



# СТИНГРЕЙ

- Что такое мобильное приложение
- Кому нужен MAST
- Как проверить мобильное приложение без доступа к исходному коду
- Интеграция в CI/CD-процессы



# Почему уязвимы мобильные приложения?

- Устаревшие или непроверенные технологии
- Ошибки в коде
- Халатность при разработке приложений
- Отсутствие контроля со стороны заказчика
- Недостаточные знания разработчиков в области ИБ и отсутствие ответственности аутсорсеров за последствия

**ЕСЛИ ВАМ КАЖЕТСЯ, ЧТО ВЫ С ЧЕМ-ТО НЕ СПРАВЛЯЕТЕСЬ, ПРОСТО ВСПОМНИТЕ, КАК ГОЛУБИ ДЕЛАЮТ СВОИ ГНЁЗДА**

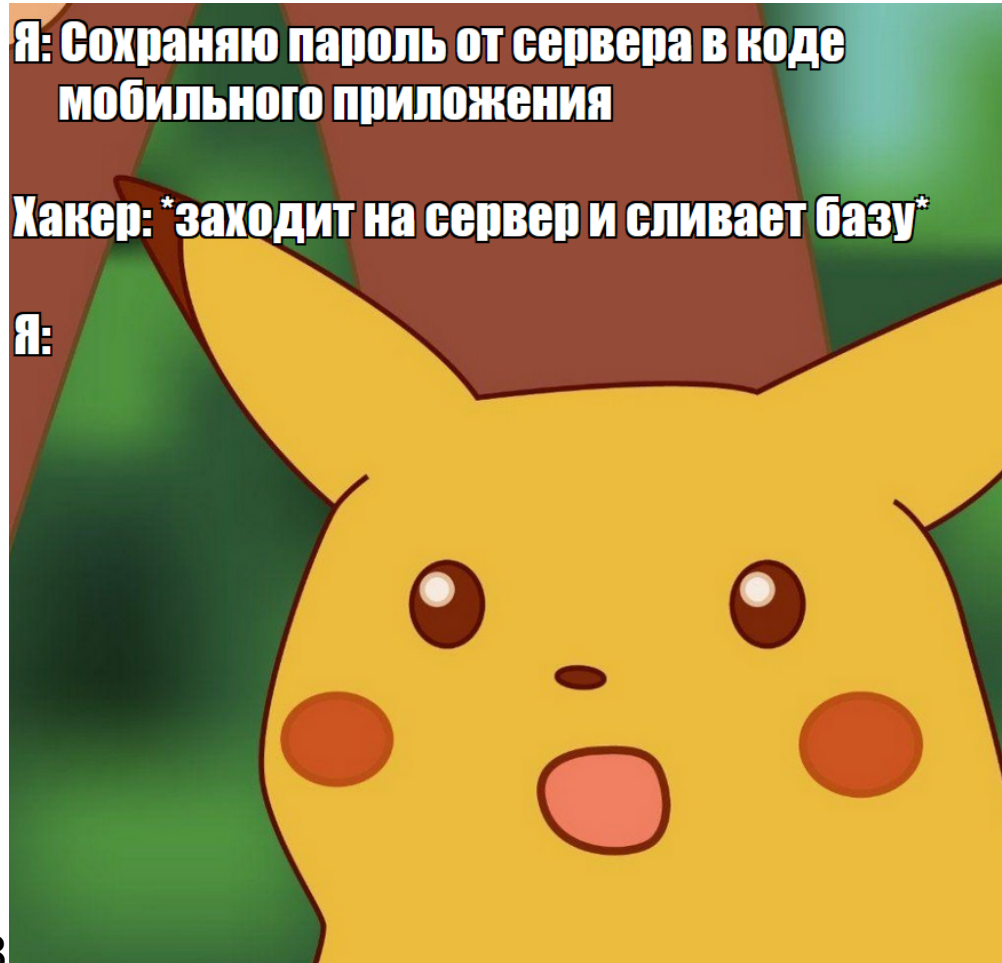


# Какие последствия?

**Я: Сохраняю пароль от сервера в коде мобильного приложения**

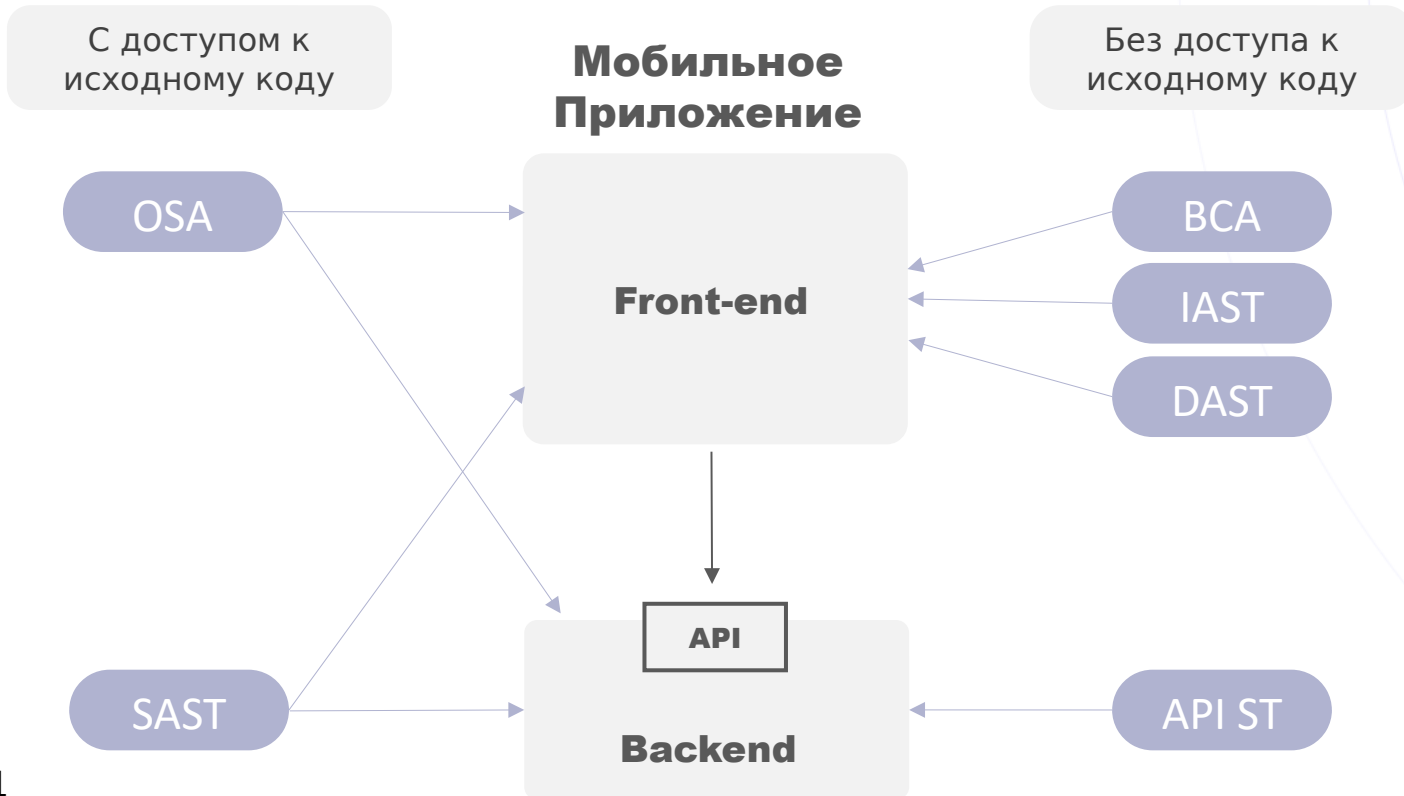
**Хакер: \*заходит на сервер и сливает базу\***

**Я:**



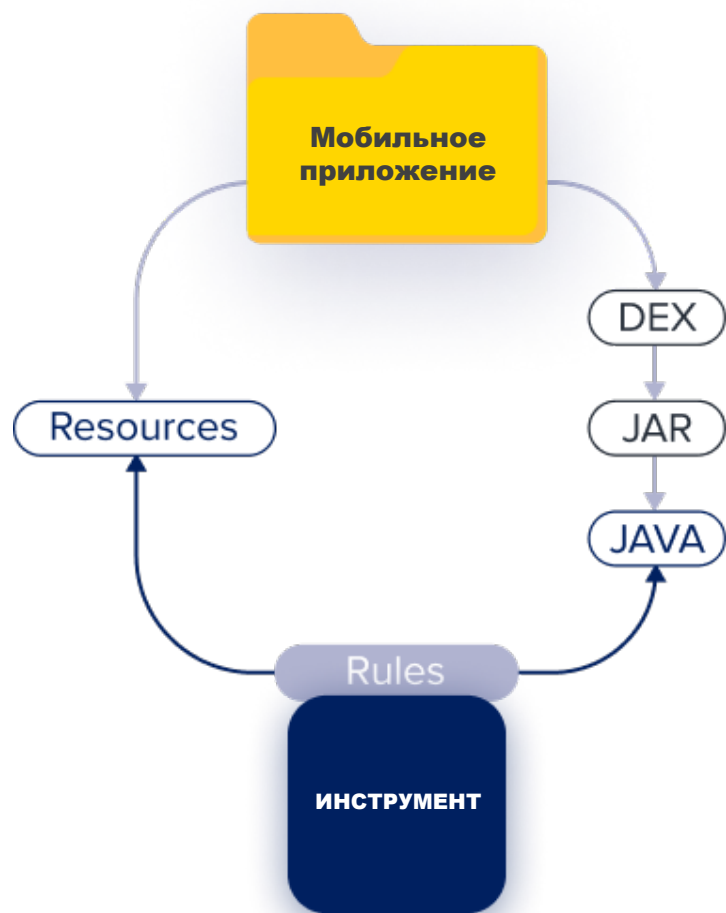
- Атака на пользователей и администраторов вашего приложения, кража данных, подмена контента и самого приложения, перенаправление на вредоносные веб-сайты, списание бонусов и денежных средств, запуск шпионского и вредоносного кода на мобильных устройствах от имени вашего приложения.
- Использование найденных в приложении ключей, сертификатов, паролей, токенов, контактов, адресов для атаки на вашу сеть, хранилища кода, облачную инфраструктуру.
- Продвинутое атаки на ваш API с дополнительной информацией, полученной из мобильного приложения.
- Репутационные потери, падение стоимости акций, всеобщее высмеивание и порицание.

# Практики **MAST** – Mobile Application Security Testing



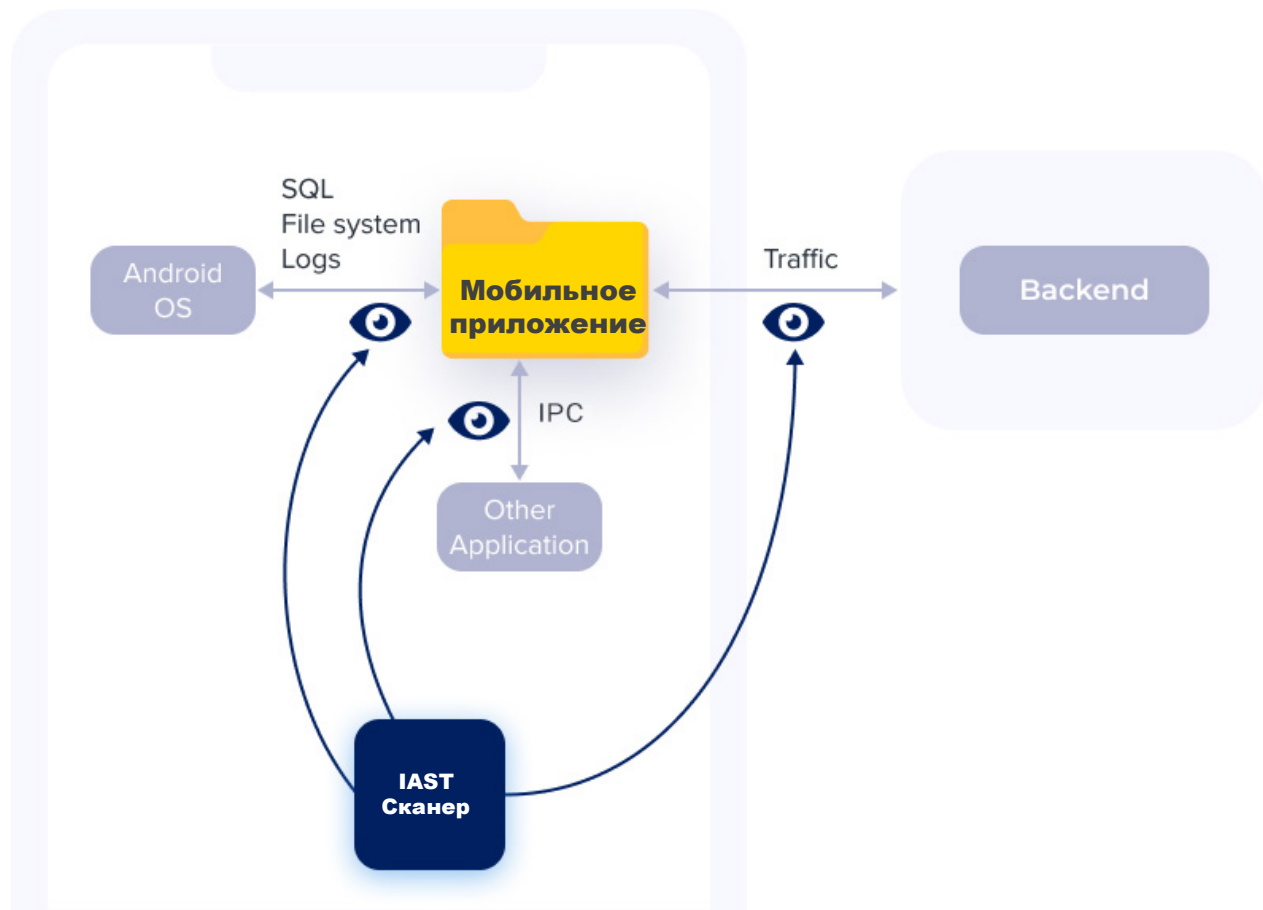
Все практики MAST можно разделить на две группы:

1. Когда доступен исходный код.
2. Когда нет доступа к исходному коду, а есть только готовое приложение.



## BCA – Bytecode Analysis

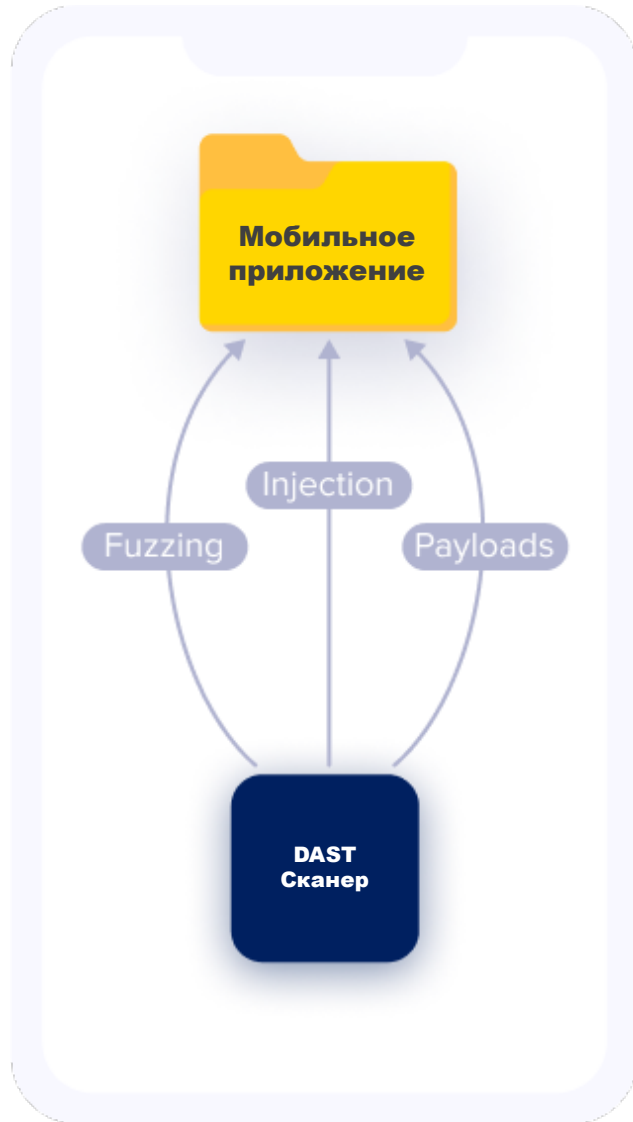
BCA полезен для проверки окончательной версии приложения, чтобы убедиться, что она собрана корректно и, как минимум, включает в себя только файлы и конфигурацию, необходимые для работы, в сборке нет никаких лишних файлов и данных.



## IAST - Interactive Application Security Testing

Практика IAST построена на основе наблюдения за поведением приложения, как оно взаимодействует с операционной системой, приложениями, вашим API.

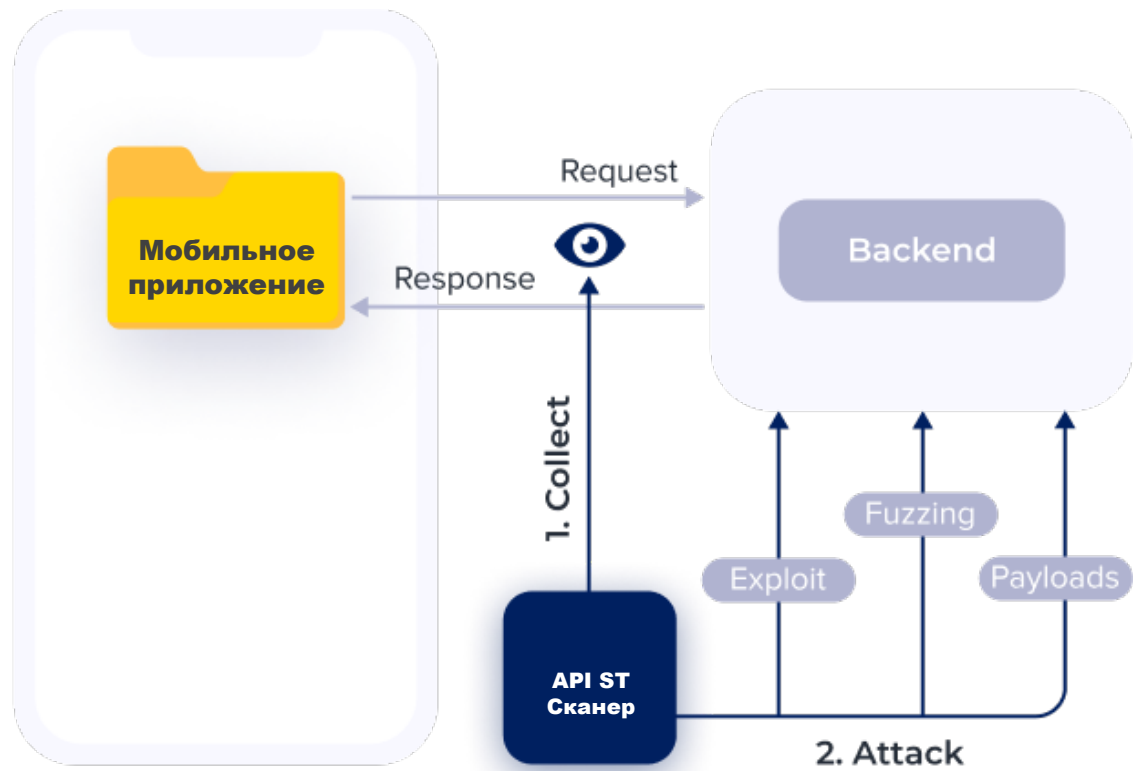
Благодаря этому, можно идентифицировать и определить всю конфиденциальную информацию, с которой работает приложение, и понять, как она обрабатывается и хранится.



## **DAST** - Dynamic Application Security Testing

Практика DAST нацелена на поиск уязвимостей, которые могут быть реализованы без root / jailbreak доступа и основаны на специфике используемых в приложении способов взаимодействия со сторонними приложениями.

Другими словами – это эмуляция злоумышленника, который представляет собой приложение, установленное рядом с вашим.



## API ST - API Security Testing

Тестирование безопасности API (API ST) применяется для тестирования серверной части мобильных приложений (API).

Можно считать эту практику частью DAST для серверной части, но чтобы не путаться в аббревиатурах, в безопасности мобильных приложений применяется понятие API Security Testing.



# Важность динамического анализа

При работе с запущенным приложением можно анализировать его меняющееся состояние, реакции на поступающие данные и накапливающиеся ошибки логики обработки пользовательских действий.

## Подтверждение уязвимостей

При помощи динамического анализа можно подтвердить уязвимости, выявленные другими практиками и определить, какие из них на самом деле эксплуатируемые.

## Обнаружение уязвимостей

При динамическом анализе выявляются уязвимости, которые невозможно определить другими практиками.

## Работа без исходного кода

Динамический анализ работает без необходимости доступа к исходному коду и анализирует поведение работающего приложения на устройстве.

# Как это делает Стингрей?



# Автоматизация тестирования

## ЗАПИСЬ

Система записывает все действия пользователя и отклик приложения на эти действия, и на основе записи формирует сценарий проверки.

Либо специалист, проверяющий приложение, предоставляет готовый сценарий в формате Arrium

## ВОСПРОИЗВЕДЕНИЕ

Система воспроизводит записанные автотесты, анализирует, привели ли действия к ожидаемым результатам и, при необходимости, отправляет тесты на адаптацию.

## АДАПТАЦИЯ

С помощью методов машинного обучения и интеграции с операционной системой Стингрей производит адаптацию автотеста под изменения элементов интерфейса без перезаписи теста.

# Автоматические проверки

## Дамп архива приложения

Расшифровка приложения, дампы запущенного приложения из памяти.

## Анализ поведения

Activity/Intent для Android. Отслеживание сообщений и взаимодействия с соседними приложениями и сервисами.

## Анализ сетевой активности

Перехват HTTP/HTTPS/WebSocket, сбор информации о конечных точках, анализ передаваемых данных.

## Анализ систем защиты

Проверка на изменение поведения приложения в зависимости от того, запущено оно на эмуляторе или нет

## Анализ файлов, баз данных, системного журнала и дампа памяти приложения

Сбор баз данных, которые используются в приложении (включая зашифрованные базы данных), анализ запросов и ответов.

Анализ файлов, которые использует приложение во время своей работы.

Анализ изменений памяти приложения во время работы.

Анализ записей системного журнала.

## Анализ сборки (SAST)

Декомпиляция исходного кода приложения, проверка на обфускацию, анализ качества конфигурации и сборки.

## Поиск чувствительной информации

Поиск ключей, имен пользователей и паролей, сертификатов, токенов, введенных данных.

+ рекурсивный поиск найденной и производной информации по всем источникам данных.

## Поиск уязвимостей

Обнаружение уязвимостей, связанных с небезопасным хранением и передачей данных, небезопасной аутентификацией, слабой криптостойкостью.

Анализ поведения приложения на различные входные данные: пользовательский ввод, deep links.

# Отчеты о найденных дефектах

## Дефекты

Название	Критичность	Инструмент	Статус	Состояние
STG-127109 Хранение sensitive-информации в исходном коде приложения	Критический			
STG-127108 Небезопасная конфигурация App Transport Security	Высокий			
STG-127096 Чувствительная информация в исполняемом файле	Средний			
STG-127097 Чувствительная информация в исполняемом файле	Средний			
STG-127098 Чувствительная информация в исполняемом файле	Средний			
STG-127099 Чувствительная информация в исполняемом файле	Средний			

**Хранение sensitive-информации в исходном коде приложения**  
Приложение хранит чувствительную информацию в исходном коде приложения.

[Скачать отчёт](#) [Рекомендации по устранению](#)

Состояние: **Новый** | Критичность: **Критический** | Статус: **Не обработан** | [Сохранить](#)

Место возникновения 1 | **Хранение sensitive-информации в исходном коде приложения**

Чувствительная информация  
keychain-access-groups

Критичность: **КРИТИЧНЫЙ**  
Способ обнаружения: **DAST, FILES**

**Описание**  
Приложение хранит чувствительную информацию в исходном коде приложения. Очень часто ошибочно считается, что данные, которые зашиты в исходном коде приложений защищены и недоступны после компиляции и обфускации. Однако, в декомпилированном приложении все строковые ресурсы остаются в неизменном виде.

**Рекомендации**  
Несмотря на то, что восстановить исходный код в iOS из пакета приложения представляет собой трудоемкую задачу, статические данные (строки, константы, числа) хранятся в открытом виде и легко считываются из исполняемого файла

Если необходимо хранить конфиденциальную информацию, исходный код не самое лучшее место для этого. Оптимальным вариантом является получение такой информации с сервера и, при необходимости её хранения на устройстве, использование шифрования. Для обеспечения конфиденциальности данных iOS оснащена множеством криптографических функций и методов, с помощью которых приложения iOS могут безопасно осуществлять шифрование и дешифрование (для обеспечения конфиденциальности), а также аутентификацию сообщений (MAC) и цифровые подписи (для проверки целостности).

Чтобы выбрать подходящий в заданных условиях метод шифрования и тип ключа, можно воспользоваться следующей схемой:

Чувствительные данные  
1 [   
2 "UAVZNE8PJA.\*"   
3 ]

Путь  
/Payload/DVIA-v2.app/en

Тип контента  
JSON

Найдено правилом  
Ключи

Название правила  
Ключи

Строка поиска  
(?:appsflyer|dev)?key

Найденные дефекты складываются в удобный список карточек с обозначением уровня критичности, полной информацией о деталях, а также ссылками на собственную базу данных инструкций по устранению.

Каждое сканирование формирует собственный список найденных дефектов, чтобы вы могли сравнить результаты между собой.

Список можно выгрузить в виде PDF-отчета для предоставления аудиторам.

# Проверка на соответствие требованиям

Найденные дефекты распределяются по пунктам стандартов, чтобы можно было легко проверить, каким стандартам и почему не соответствует ваше приложение:

- MASVS
- OWASP Mobile Top 10
- PCI DSS 4.0
- PCI Software Security Framework
- ОУД4
- ГОСТ-57580

## Дефекты

### Хранение sensitive-информации в общедоступном файле

Приложение хранит чувствительную информацию в общедоступном файле внутри директории приложения.

### Хранение приватного ключа/сертификата не защищенного паролем в директории/ресурсах приложения

Приложение хранит приватный ключ/сертификат не защищенный паролем в директории/ресурсах приложения. Такой подход к хранению ключей и сертификатов может существенно упростить подмену ключевой информации злоумышленником и нарушение целостности и логики работы приложения.

### Вывод sensitive-информации в системный лог

Приложение выводит чувствительную информацию с помощью методов класса Log или System.out/err.

### Хранение sensitive-информации в общедоступном файле

Приложение хранит чувствительную информацию в общедоступном файле вне директории приложения.

### Хранение ранее найденной чувствительной информации

Приложение хранит чувствительную информацию.

### Хранение чувствительной информации в общедоступной незащищенной базе данных

Приложение хранит чувствительную информацию в общедоступной незащищенной базе данных.

### Хранение значений Cookies в стандартной базе WebView

Приложение хранит значения cookie в стандартной базе Cookies.db в открытом виде. Такой подход к хранению информации может привести к утечке сессионных идентификаторов и повлечь за собой неправомерный доступ к данным пользователя.

### Хранение чувствительной информации в общедоступной защищенной базе данных

Приложение хранит чувствительную информацию в общедоступной защищенной базе данных.

### Хранение sensitive-информации в исходном коде приложения

Приложение хранит чувствительную информацию в исходном коде приложения.

### Хранение sensitive-информации в кэше клавиатуры

Sensitive-информация попадает в кэш клавиатуры устройства и может быть доступна в подсказках автодополнения при вводе текста.

ОУД4

OWASP MASVS

OWASP Mobile Top 10

PCI DSS

ГОСТ 57580



# Сканирование на эмуляторах и живых устройствах

Приложения запускаются на ферме из эмуляторов и специально подготовленных устройствах на базе iOS и Android.

Вместе с отчетом и собранными данными предоставляется запись с экрана устройства для анализа поведения UI и отработки всех этапов сценария автоматизированного тестирования.

Установка платформы и проведение сканирований возможны как в облаке Стингрей, так и в сети заказчика.

# Интеграции

Стингрей обеспечивает интеграцию со многими инструментами DevOps: CI / CD, дефект-трекерами, системами дистрибуции и другими инструментами



а также возможность проверки публикуемых приложений по расписанию.



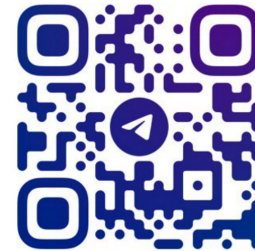
## Что делать дальше:

**Зайдите на сайт продукта  
и познакомьтесь с  
детальями**



**<https://stingray-mobile.ru/>**

**Свяжитесь с нами, чтобы  
запланировать демо/пилот**



**@MRCRRA**

Или пишите на почту:  
**[dm@afi-d.ru](mailto:dm@afi-d.ru)**

Или звоните:  
7 495 223 35 33  
8 800 550 52 23