**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ**

**для разработчиков по предупреждению   
и противодействию нарушениям  
 информационной безопасности**

Москва

2020

# Введение

## 1. Цели и задачи Методических рекомендаций

Методические рекомендации по разработке и принятию мер по предупреждению и противодействию нарушениям информационной безопасности (далее – Методические рекомендации) разработаны во исполнение отдельных положений Федерального закона № 149   
«Об информации, информационных технологиях и о защите информации».

Целью Методических рекомендаций является формирование единого подхода к обеспечению работы по профилактике и противодействию нарушениям информационной безопасности в организациях независимо от их форм собственности, организационно-правовых форм, отраслевой принадлежности и иных обстоятельств.

Задачами Методических рекомендаций являются:

* информирование организаций о нормативно-правовом обеспечении работы по обеспечении информационной безопасности и ответственности за совершение правонарушений;
* определение основных принципов противодействия нарушениям в организациях;
* методическое обеспечение разработки и реализации мер, направленных на профилактику и противодействие нарушениям ИБ в организации.

## 2. Круг субъектов, для которых разработаны Методические рекомендации

Настоящие Методические рекомендации разработаны для использования

широким кругом лиц:

* Администраторов бекенда
* Бекенд-разработчиков
* Мобильных разработчиков

# Рекомендации

## Поддерживать высокий уровень осведомленности команды разработчиков в вопросах информационной безопасности .

## Уделять особое внимание вопросам безопасности на этапе проектирования архитектуры.

## Проводить аудит исходного кода.

## Проводить анализ защищенности приложения.

## Поддерживать последние версии SDK.

## Контролировать распространение приложения в сети Интернет.

## Своевременно закрывать уязвимости и выпускать обновления.

## Отслеживать попадающие в финальные версии продукта файлы и данные.

## Повысить уровень интеграции банковских сервисов с операторами сотовой связи.

## Используйте методы запутывания кода, усложняющие злоумышленникам его чтение и анализ. Примером запутывания может служить процедура удаления символов, проводимая на этапе сборки приложения. Она заключается в замене исходных имен классов и методов случайными или однобуквенными именами. Можно использовать специализированные программные средства, например, ProGuard для Android или Sirius Obfuscator и SwiftShield для iOS.

## При реализации deep linking появляется еще одна точка входа в приложение, которой может воспользоваться злоумышленник. Необходимо учитывать, что все параметры, передаваемые с помощью механизма deep linking, поступают из ненадежного источника и должны проходить проверку и фильтрацию перед передачей в соответствующие методы исходного кода.

## На мобильном устройстве нужно хранить только необходимый объем данных. Требуемые данные должны запрашиваться с сервера только во время работы с приложением и после завершения работы должны быть удалены. Шифруйте конфиденциальную информацию, хранящуюся на устройстве, но при этом обеспечьте безопасное управление ключами шифрования. Для защиты данных на снимках состояния экрана используйте специальное фоновое изображение, которое будет перекрывать экран мобильного банка, содержащий важную информацию.

## Интегрируйте в жизненный цикл разработки принципы SDL, в которые входит анализ защищенности кода еще в процессе его написания.