# ГЛАВА 1. ИССЛЕДОВАНИЕ УГРОЗ И УЯЗВИМОСТЕЙ МОБИЛЬНОГО БАНКОВСКОГО ПРИЛОЖЕНИЯ

* 1. **Введение**

В данном исследовании приведены уязвимости клиентских мобильных банковских приложений, связанные с ошибками в коде, недостатками клиент-серверного взаимодействия, а также ошибками реализации механизмов защиты. Другие распространенные проблемы информационной безопасности (к примеру, недостатки управления обновлениями ПО) не рассматриваются. Уровень риска уязвимостей оценивался исходя из степени влияния потенциальной атаки на пользовательские данные и само приложение, а также с учетом сложности проведения атаки; выделены качественные оценки высокого, среднего и низкого уровней риска.

**Операционные системы**

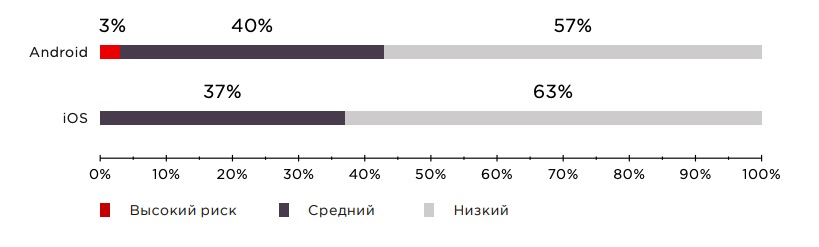
Самые распространённые операционные системы смартфонов является Android от Google, и iOS от Apple. Обе операционные системы на рынке очень востребованы, и большая часть мобильных банковских приложений разрабатываются под них.

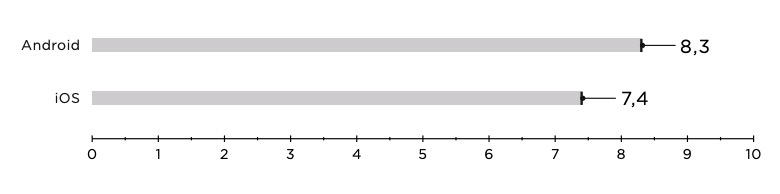
**Клиентская часть**

Клиентская часть мобильного банка — это мобильное приложение, которое устанавливается непосредственно на устройство пользователя.

Согласно исследованиям Positive Technologies:

* + - В 90% клиентских частей возможен доступ к данным пользователей.
    - 76% уязвимостей в мобильных банках могут быть проэксплуатированы без физического доступа к устройству.
* Более трети уязвимостей не требуют административных прав для эксплуатации (jailbreak или root).
  1. **Уязвимости клиентских частей приложений**

Клиентские части мобильных банковских приложений, разработанные для iOS, содержали меньше уязвимостей, чем приложения для Android. Все недостатки в мобильных банках для iOS были не выше среднего уровня риска. В то время как 29% приложений для Android содержали уязвимости высокого уровня риска.

*Рисунок 1. Доля уязвимостей различного уровня риска*

*Рисунок 2. Среднее количество уязвимостей на одно приложение*

Наиболее опасные уязвимости выявлены в Android-приложениях и связаны с небезопасной обработкой ссылок deeplink. Технология deep linking используется по-разному в версиях для iOS и для Android. Разработчикам Android-приложений предоставляется больше свободы в реализации различной функциональности. Это причина большего количества уязвимостей в сравнении с iOS-приложениями. Но это не значит, что разработчики iOSприложений не могут допустить ошибки. Безопасность

мобильного банка в первую очередь зависит от практики безопасной разработки (Security Development Lifecycle, SDL).

Во 100% клиентских частей мобильных банков содержат уязвимости в коде приложений:

* отсутствует обфускация кода;
* отсутствует защита от внедрения кода и «перепаковки»;
* в коде содержатся имена классов и методов.

Исследование показало, что банки не защищаются от угрозы анализа исходного кода, которая возникает в случае недостаточной его защиты. Для эксплуатации уязвимостей в коде злоумышленникам нужно получить к нему доступ, а для этого достаточно скачать приложение из Google Play или App Store и затем его декомпилировать.

Отсутствие обфускации кода позволяет его анализировать и находить такие важные

данные, как:

* тестовые логины и пароли;
* ключи шифрования или параметры, из которых они однозначно получаются;
* соль, используемую для хеширования или шифрования.

Эта информация может быть в дальнейшем использована злоумышленниками для получения аутентификационных данных и доступа к веб-серверам. Кроме того, хакеры могут проанализировать алгоритм работы приложения и затем воспользоваться недостатками в бизнес-логике. Также информация о том, как устроено приложение, может быть интересна конкурентам для внесения в свои продукты новой функциональности.

**Сравнительный анализ мобильного банковского   
приложения для iOS и Android**

|  |  |
| --- | --- |
| IOS | Android |
| Использование HTTP – соединение | Запуск на Root – устройстве |
| Невозможность полного завершения сессии | Пароль в файлах после завершения сессии |
| Отсутствие 2FA | Отсутствие 2FA |
| Отсутствие SSL Pinning | Раскрытие информации о текстовом окружении |
| Раскрытие координат пользователя | Раскрытие координат пользователя |
| Критичная информация в памяти приложения | Возможность утечки данных через Backup сессии приложения |
| SSL – сертификаты в файлах приложений | SSL – сертификаты в файлах приложений |
| Запуск на Jailbreak – устройстве | Пароль в памяти после завершения сессии |
| Раскрытие информации о тестовом окружении | Отсутствие предупреждения на устройстве с Root |

Таблица 1. Сравнительный анализ мобильного банковского приложения для iOS и Android

**1.3 Модель нарушителя**

В настоящем разделе определяется совокупность условий и факторов, создающих опасность нарушения характеристик безопасности возможных объектов угроз.

В данном разделе под угрозами будут пониматься атаки.

По признаку принадлежности к ИС мобильного устройства все нарушители делятся на две группы:

**Внешние** нарушители – физические лица, не имеющие физический доступ к устройству клиента или же находящийся рядом с жертвой.

**Внутренние** нарушители – физические лица, имеющие физический доступ к устройству клиента или же находящийся рядом с жертвой.

В качестве внешнего нарушителя информационной безопасности, рассматривается нарушитель, который не имеет непосредственного доступа к техническим средствам и ресурсам системы.

Возможности внешнего и внутреннего нарушителя существенным образом зависят от действующих в ИС ограничительных факторов, из которых основным является реализация комплекса технических мер защиты.

Состав имеющихся у нарушителя средств, которые он может использовать для реализации угроз ИБ, а также возможности по их применению зависят от многих факторов, включая реализованные на устройстве методы защиты, финансовые возможности и компетенцию нарушителей. Поэтому объективно оценить состав имеющихся у нарушителя средств реализации угроз в общем случае практически невозможно.

Предполагается, что внешний нарушитель может воздействовать на защищаемую информацию во время ее передачи по каналам связи или путем загрузки на устройство клиента вредоносное или измененное приложение.

Поэтому, для создания устойчивой ИС предполагается, что вероятный нарушитель имеет все необходимые для реализации угроз средства, возможности которых не превосходят возможности аналогичных средств реализации угроз на информацию, содержащую сведения, не составляющие государственную тайну, и технические и программные средства, обрабатывающие эту информацию.

К **внешним нарушителям** могут относиться:

* Администратор безопасности (категория I);
* Администраторы конкретных подсистем или баз данных (категория II);

К **внутренним нарушителям** могут относиться:

* Лица, обладающие возможностью доступа к системе (категория V);
* Обслуживающий персонал (категория VII);

Предполагается, что наиболее совершенными средствами реализации угроз обладают лица категории III и лица категории VIII.

На лиц категорий I-II возложены задачи по администрированию программно-аппаратных средств и баз данных для интеграции и обеспечения взаимодействия различных подсистем, входящих в состав ИС мобильного приложения. Администраторы потенциально могут реализовывать угрозы ИБ, используя возможности по непосредственному доступу к защищаемой информации, обрабатываемой и хранимой в ИС а также к техническим и программным средствам, включая средства защиты, используемые в конкретных АС, в соответствии с установленными для них административными полномочиями.

Эти лица хорошо знакомы с основными алгоритмами, протоколами, реализуемыми и используемыми в конкретных подсистемах и ИС в целом, а также с применяемыми принципами и концепциями безопасности.

Предполагается, что они могли бы использовать стандартное оборудование либо для идентификации уязвимостей, либо для реализации угроз ИБ. Данное оборудование может быть, как частью штатных средств, так и может относиться к легко получаемому (например, программное обеспечение, полученное из общедоступных внешних источников).

К лицам категорий I-II ввиду их исключительной роли ИС должен применяться комплекс особых организационно-режимных мер по их подбору, принятию на работу, назначению на должность и контролю выполнения функциональных обязанностей.

Предполагается, что в число лиц категорий I-II будут включаться только доверенные лица и поэтому указанные лица исключаются из числа вероятных нарушителей.

Предполагается, что лица категорий III-VIII относятся к вероятным нарушителям.

Предположения об имеющейся у нарушителя информации об объектах реализации угроз.

В качестве основных уровней знаний нарушителей об АС можно выделить следующие:

* Общая информация – информации о назначении и общих характеристиках ИС
* Эксплуатационная информация – информация, полученная из эксплуатационной документации;
* Чувствительная информация – информация, дополняющая эксплуатационную информацию об ИС (например, сведения из проектной документации).

В частности, нарушитель может иметь:

* Данные об организации работы, структуре и используемых технических, программных и программно-технических средствах;
* Сведения об информационных ресурсах: порядок и правила создания, хранения и передачи информации, структура и свойства информационных потоков;
* Данные об уязвимостях, включая данные о недокументированных (недекларированных) возможностях технических, программных и программно-технических средств;
* Данные о реализованных в программных средствах защиты информации принципах и алгоритмах;
* Исходные тексты программного обеспечения;
* Сведения о возможных каналах реализации угроз;
* Информацию о способах реализации угроз.

Предполагается, что лица категории III и категории IV владеют только эксплуатационной информацией, что обеспечивается организационными мерами. При этом лица категории IV не владеют парольной, аутентифицирующей и ключевой информацией, используемой в автоматизированной информационной системе (АИС), к которым они не имеют санкционированного доступа.

Предполагается, что лица категории V владеют в той или иной части чувствительной и эксплуатационной информацией о системе передачи информации и общей информацией об АИС, использующих эту систему передачи информации, что обеспечивается организационными мерами. При этом лица категории V не владеют парольной и аутентифицирующей информацией, используемой в АИС.

Предполагается, что лица категории VI и лица категории VII по уровню знаний не превосходят лица категории V.

Предполагается, что лица категории VIII обладают чувствительной информацией о ИС и функционально ориентированных АС, включая информацию об уязвимостях технических и программных средств. Организационными мерами предполагается исключить доступ лиц категории VIII к техническим и программным средствам в момент обработки с использованием этих средств защищаемой информации.

Таким образом, наиболее информированными о ИС являются лица категории III и лица категории VIII.

Степень информированности нарушителя зависит от многих факторов, включая реализованные конкретные технические меры и компетенцию нарушителей. Поэтому объективно оценить объем знаний вероятного нарушителя в общем случае практически невозможно.

В связи с изложенным, с целью создания определенного запаса прочности, предполагается, что вероятные нарушители обладают всей информацией, необходимой для подготовки и реализации угроз, за исключением информации, доступ к которой со стороны нарушителя исключается системой защиты информации. К такой информации, например, относится парольная, аутентифицирующая и ключевая информация.

Предположения об имеющихся у нарушителя средствах реализации угроз:

* аппаратные компоненты средства защиты;
* доступные в свободной продаже технические средства и программное обеспечение;
* специально разработанные технические средства и программное обеспечение.

**Нарушители согласно банку данных угроз ФСТЭК России**

Дополнительно в банке данных угроз ФСТЭК России определены три типа внешних и внутренних нарушителей – с низким потенциалом, со средним потенциалом и с высоким потенциалом.

Нарушители с низким потенциалом имеют возможность получить информацию об уязвимостях отдельных компонент информационной системы, опубликованную в общедоступных источниках. Также такие нарушители имеют возможность получить информацию о методах и средствах реализации угроз безопасности информации (компьютерных атак), опубликованных в общедоступных источниках, и (или) самостоятельно осуществляют создание методов и средств реализации атак и реализацию атак на информационную систему.

Нарушители со средним потенциалом обладают всеми возможностями нарушителей с низким потенциалом. Имеют осведомленность о мерах защиты информации, применяемых в информационной системе данного типа. Имеют возможность получить информацию об уязвимостях отдельных компонент информационной системы путем проведения, с использованием имеющихся в свободном доступе программных средств, анализа кода прикладного программного обеспечения и отдельных программных компонент общесистемного программного обеспечения. Имеют доступ к сведениям о структурно-функциональных характеристиках и особенностях функционирования информационной системы.

Нарушители с высоким потенциалом обладают всеми возможностями нарушителей с низким и средним потенциалами. Имеют возможность осуществлять несанкционированный доступ из выделенных (ведомственных, корпоративных) сетей связи, к которым возможен физический доступ (незащищенных организационными мерами). Имеют возможность получить доступ к программному обеспечению чипсетов (микропрограммам), системному и прикладному программному обеспечению, телекоммуникационному оборудованию и другим программно-техническим средствам информационной системы для преднамеренного внесения в них уязвимостей или программных закладок. Имеют хорошую осведомленность о мерах защиты информации, применяемых в информационной системе, об алгоритмах, аппаратных и программных средствах, используемых в информационной системе. Имеют возможность получить информацию об уязвимостях путем проведения специальных исследований (в том числе с привлечением специализированных научных организаций) и применения специально разработанных средств для анализа программного обеспечения. Имеют возможность создания методов и средств реализации угроз безопасности информации с привлечением специализированных научных организаций и реализации угроз с применением специально разработанных средств, в том числе обеспечивающих скрытное проникновение в информационную систему и воздействие на нее. Имеют возможность создания и применения специальных технических средств для добывания информации (воздействия на информацию или технические средства), распространяющейся в виде физических полей или явлений.

**1.4 Модель угроз**

| **№ п/п** | **Идентификатор** | **Описание угрозы** | **Способ реализации угрозы** | **Источники угрозы** | **Объекты воздействия** | **Нарушаемые свойства безопасности информации** |
| --- | --- | --- | --- | --- | --- | --- |
|  | УБИ.003 | Угроза использования слабостей криптографических алгоритмов и уязвимостей в программном обеспечении их реализации. Угроза заключается в возможности выявления слабых мест в криптографических алгоритмах или уязвимостей в реализующем их программном обеспечении.  Данная угроза обусловлена слабостями криптографических алгоритмов, а также ошибками в программном коде криптографических средств, их сопряжении с системой или параметрах их настройки. | Реализация угрозы возможна в случае наличия у нарушителя сведений об применяемых в системе средствах шифрования, реализованных в них алгоритмах шифрования и параметрах их настройки | Внутренний нарушитель со средним потенциалом  Внешний нарушитель со средним потенциалом | Метаданные, системное программное обеспечение | К, Ц |
|  | УБИ.006 | Угроза внедрения кода или данных. Угроза заключается в возможности внедрения нарушителем в дискредитируемую информационную систему или IoT-устройство вредоносного кода, который может быть в дальнейшем запущен «вручную» пользователями, автоматически при выполнении определённого условия (наступления определённой даты, входа пользователя в систему и т.п.) или с использованием аутентификационных данных, заданных «по умолчанию», а также в возможности несанкционированного внедрения нарушителем некоторых собственных данных для обработки в дискредитируемую информационную систему, фактически осуществив незаконное использование чужих вычислительных ресурсов, и блокирования работы устройства при выполнении определенных команд.  Данная угроза обусловлена:  наличием уязвимостей программного обеспечения;  слабостями мер антивирусной защиты и разграничения доступа;  наличием открытого Telnet-порта на IoT-устройстве (только для IoT-устройств). | Реализация данной угрозы возможна:  в случае работы дискредитируемого пользователя с файлами, поступающими из недоверенных источников;  при наличии у него привилегий установки программного обеспечения;  в случае неизмененных владельцем учетных данных IoT-устройства (заводских пароля и логина) | Внешний нарушитель с низким потенциалом | Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение | К, Ц, Д |
|  | УБИ.036 | Угроза исследования механизмов работы программы. Угроза заключается в возможности проведения нарушителем обратного инжиниринга кода программы и дальнейшего исследования его структуры, функционала и состава в интересах определения алгоритма работы программы и поиска в ней уязвимостей.  Данная угроза обусловлена слабостями механизма защиты кода программы от исследования. | Реализация данной угрозы возможна в случаях:  наличия у нарушителя доступа к исходным файлам программы;  наличия у нарушителя доступа к дистрибутиву программы и отсутствия механизма защиты кода программы от исследования | Внутренний нарушитель со средним потенциалом  Внешний нарушитель со средним потенциалом | Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение, микропрограммное обеспечение | К, Д. |
|  | УБИ.100 | Угроза обхода некорректно настроенных механизмов аутентификации. Угроза заключается в возможности получения нарушителем привилегий в системе без прохождения процедуры аутентификации за счёт выполнения действий, нарушающих условия корректной работы средств аутентификации (например, ввод данных неподдерживаемого формата).  Данная угроза обусловлена в случае некорректных значений параметров конфигурации средств аутентификации и/или отсутствием контроля входных данных. | Реализация данной угрозы возможна при условии наличия ошибок в заданных значениях параметров настройки механизмов аутентификации | Внутренний нарушитель с низким потенциалом  Внешний нарушитель с низким потенциалом | Системное программное обеспечение, сетевое программное обеспечение | К, Ц, Д |
|  | УБИ.116 | Угроза перехвата данных, передаваемых по вычислительной сети. Угроза заключается в возможности осуществления нарушителем несанкционированного доступа к сетевому трафику дискредитируемой вычислительной сети в пассивном («прослушивание» трафика) или активном (подмена пакетов, изменение их содержимого) режиме для сбора и анализа сведений (например, аутентификационной информации), которые могут быть использованы в дальнейшем для реализации других угроз, оставаясь при реализации данной угрозы невидимым (скрытным) получателем перехватываемых данных. Кроме того, нарушитель может проводить исследования других типов потоков данных, например, радиосигналов.  Данная угроза обусловлена слабостями механизмов сетевого взаимодействия, предоставляющими сторонним пользователям открытые данные о дискредитируемой системе, а также ошибками конфигурации сетевого программного обеспечения. | Реализация данной угрозы возможна в следующих условиях:  наличие у нарушителя доступа к дискредитируемой вычислительной сети;  неспособность технологий, с помощью которых реализована передача данных, предотвратить возможность осуществления скрытного прослушивания потока данных | Внешний нарушитель с низким потенциалом | Сетевой узел, сетевой трафик | К |
|  | УБИ.127 | Угроза подмены действия пользователя путём обмана. Угроза заключается в возможности нарушителя выполнения неправомерных действий в системе от имени другого пользователя с помощью методов социальной инженерии (обмана пользователя, навязывание ложных убеждений) или технических методов (использование прозрачных кнопок, подмена надписей на элементах управления и др.)  Данная угроза обусловлена слабостями интерфейса взаимодействия с пользователем или ошибками пользователя. | Реализация данной угрозы возможна при условии наличия у дискредитируемого пользователя прав на проведение нужных от него нарушителю операций | Внешний нарушитель со средним потенциалом | Прикладное программное обеспечение, сетевое программное обеспечение | К, Ц, Д |
|  | УБИ.169 | Угроза наличия механизмов разработчика. Угроза заключается в возможности перехвата управления программой за счёт использования отладочных механизмов (специальных программных функций или аппаратных элементов, помогающих проводить тестирование и отладку средств во время их разработки).  Данная угроза обусловлена недостаточностью мер по контролю за ошибками в ходе разработки средств защиты информации. | Реализация данной угрозы возможна при условии, что в программе не удалены отладочные механизмы | Внешний нарушитель со средним потенциалом | Программное обеспечение, техническое средство | К, Ц, Д |
|  | УБИ.175: | Угроза «фишинга». Угроза заключается в возможности неправомерного ознакомления нарушителем с защищаемой информацией (в т.ч. идентификации/аутентификации) пользователя путём убеждения его с помощью методов социальной инженерии (в т.ч. посылкой целевых писем (т.н. spear-phishing attack), с помощью звонков с вопросом об открытии вложения письма, имитацией рекламных предложений (fake offers) или различных приложений (fake apps)) зайти на поддельный сайт (выглядящий одинаково с оригинальным), на котором от дискредитируемого пользователя требуется ввести защищаемую информацию или открыть заражённое вложение в письме.  Данная угроза обусловлена недостаточностью знаний пользователей о методах и средствах «фишинга». | Реализация данной угрозы возможна при условии наличия у нарушителя:  сведений о конкретных сайтах, посещаемых пользователем, на которых требуется ввод защищаемой информации;  средств создания и запуска поддельного сайта;  сведений о контактах пользователя с доверенной организацией (номер телефона, адрес электронной почты и др.).  Для убеждения пользователя раскрыть информацию ограниченного доступа (или открыть вложение в письмо) наиболее часто используются поддельные письма от администрации какой-либо организации, с которой взаимодействует пользователь (например, банк) | Внешний нарушитель с низким потенциалом | Рабочая станция, сетевое программное обеспечение, сетевой трафик | К |
|  | УБИ.186 | Угроза внедрения вредоносного кода через рекламу, сервисы и контент. Угроза заключается в возможности внедрения нарушителем в информационную систему вредоносного кода посредством рекламы, сервисов и (или) контента (т.е. убеждения пользователя системы активировать ссылку, код и др.) при посещении пользователем системы сайтов в сети Интернет или установкой программ с функцией показа рекламы.  Данная угроза обусловлена слабостями механизмов фильтрации сетевого трафика и антивирусного контроля на уровне организации. | Реализация данной угрозы возможна при условии посещения пользователями системы с рабочих мест сайтов в сети Интернет | Внутренний нарушитель с низким потенциалом | Сетевое программное обеспечение | Ц, Д |
|  | УБИ.188 | Угроза подмены программного обеспечения. Угроза заключается в возможности осуществления нарушителем внедрения в систему вредоносного программного обеспечения за счёт загрузки и установки вредоносного программного обеспечения, скрытого под видом легитимного свободно распространяемого программного обеспечения.  Данная угроза обусловлена наличием у пользователя прав для установки программного обеспечения из сети Интернет. | Реализация данной угрозы возможна при скачивании программного обеспечения в сети Интернет | Внешний нарушитель со средним потенциалом | Прикладное программное обеспечение, сетевое программное обеспечение, системное программное обеспечение | К, Ц, Д |
|  | УБИ.189 | Угроза маскирования действий вредоносного кода. Угроза заключается в возможности сокрытия в системе действий вредоносного кода за счет применения специальных механизмов маскирования кода (архивирование, изменение формата данных и др.), которые препятствуют его дальнейшему анализу.  Данная угроза обусловлена наличием способов маскирования программного кода, не учтенных сигнатурными базами средств защиты информации, а также механизмов операционной системы, позволяющих осуществить поиск модулей средств защиты информации. | Реализация данной угрозы возможна при условии использования в системе устаревших версий средств защиты информации | Внешний нарушитель со средним потенциалом | Системное программное обеспечение, сетевое программное обеспечение | Ц, Д |
|  | УБИ.194 | Угроза несанкционированного использования привилегированных функций мобильного устройства. Угроза заключается в возможности снятия нарушителем предустановленных производителем ограничений на конфигурирование привилегированных функций мобильного устройства. Данная угроза обусловлена наличием уязвимостей в операционных системах мобильного устройства, позволяющих получить доступ к настройкам привилегированных функций. | Реализация данной угрозы возможна при получении нарушителем доступа к мобильному устройству | Внешний нарушитель с высоким потенциалом | Мобильное устройство | К, Ц, Д |
|  | УБИ.196 | Угроза контроля вредоносной программой списка приложений, запущенных на мобильном устройстве. Угроза заключается в возможности использования вредоносной программы для контроля списка приложений, запущенных на мобильном устройстве.  Данная угроза обусловлена недостаточностью мер по антивирусной защите, что позволяет выполнить неконтролируемый запуск вредоносных программ (отсутствие контроля разрешенного программного обеспечения). Реализация данной угрозы возможна при условии, что вредоносная программа внедрена на мобильном устройстве и непреднамеренно запущена самим пользователем |  | Внешний нарушитель с высоким потенциалом | Мобильное устройство | Ц, Д |
|  | УБИ.202 | Угроза несанкционированной установки приложений на мобильные устройства. Угроза заключается в возможности установки приложений на виртуальные машины мобильных устройств, работающих под управлением операционной системы Android, несанкционированно запущенных внедренной вредоносной программой. Вредоносная программа запускает виртуальную машину на мобильном устройстве, размещает (устанавливает) в этой виртуальной машине неограниченное количество приложений.  Данная угроза обусловлена недостаточностью мер по контролю за запуском прикладного программного обеспечения, что позволяет выполнить неконтролируемый запуск вредоносного прикладного программного обеспечения по факту совершения пользователем различных действий в системе (например, при попытке закрытия пользователем нежелательной рекламы). | Реализация данной угрозы возможна при условии наличия на мобильном устройстве вредоносной программы, способной запустить виртуальную машину и установить в эту виртуальную машину приложение | Внешний нарушитель со средним потенциалом | Мобильные устройства | К |
|  | УБИ.213 | Угроза обхода многофакторной аутентификации. Угроза заключается в возможности обхода многофакторной аутентификации путем внедрения вредоносного кода в дискредитируемую систему и компоненты, участвующие в процедуре многофакторной аутентификации.  Данная угроза обусловлена:  наличием уязвимостей программного обеспечения;  слабостями мер антивирусной защиты и разграничения доступа. | Реализация данной угрозы возможна:  в случае работы дискредитируемого пользователя с файлами, поступающими из недоверенных источников;  при наличии у него привилегий установки программного обеспечения | Внешний нарушитель с высоким потенциалом | Системное программное обеспечение, микропрограммное обеспечение, учетные данные пользователя | К, Ц, Д |

**1.5 Методы оценки безопасности клиентского приложения**

**Статический анализ**

* Анализ исходного кода (если доступен)
* Reverse Engineering
* Декомпиляция
* Дизассемблирование
* Анализ информации в архиве, поставляемом из магазина приложений
* Анализ метаданных

**Динамический анализ**

* Отладка запущенного приложения
* Анализ сетевого трафика
* Анализ взаимодействия с файловой системой
* Анализ памяти приложения
* Анализ работы приложения с механизмами безопасности, предусмотренными операционной системой
* Фаззинг
* URI схем
* Параметров приложения

**1.6 Результаты исследования безопасности iOS приложений**

**Согласно исследованиям Digital Security**

**Длина SMS 2FA-кода:**

* 4/16 – 4 символа
* 2/16 – 5 символов
* 6/16 – 6 символов

**Анализ API:**

* 1/16 – Использование HTTP-соединения
* 3/16 – Отсутствие 2FA
* 7/16 – Раскрытие координат пользователя
* 10/16 – SSL-сертификаты в файлах приложений

**Используемые SDK:**

* 50% - SDK 8.0
* 50% - SDK 9.0
* 2/16 – Использование App Transport Security (ATS)

**Поведение после завершения сессии:**

* 2/16 – Сессию нельзя полностью завершить
* 6/16 – Остаются данные клиента в Keychain
* 7/16 – Остаются данные клиента в памяти
* 10/16 – Остаются данные клиента в файлах
* 11/16 – Остаются данные клиента в cookie
* 13/16 – Раскрытие информации о тестовом окружении

**Раскрытие критичной информации:**

* 9/16 – Критичная информация в памяти приложения
* 10/16 – Использование NS User Defaults
* 10/16 – Использование Keychain
* 13/16 – Некорректная работа со Snapshot
* Механизмы защиты:
* 11/16 – Запуск на Jailbreak устройстве
* 6/16 – Отсутствие SSL Pinning

**Метаданные приложения**

**Используемые SDK**

На основании этой информации можно сделать выводы как о частоте обновлений, так и о безопасности приложений. Так, в новых версиях SDK улучшены механизмы защиты и внедрены новые API для упрощения взаимодействия между различными компонентами системы, а также организована поддержка новых устройств.

**Использование App Transport Security (ATS)**

ATS – это набор гибких правил ограничения сетевого взаимодействия приложения, одной из ключевых особенностей которого является обязательное внедрение защищенного взаимодействия клиента с сервером. Некоторые разработчики вовсе отказываются от таких механизмов защиты, что снижает уровень сетевой безопасности приложения. Мы рекомендуем не пренебрегать этим.

**Анализ API**

**SSL-сертификаты в файлах приложений**

Часто разработчики встраивают сертификаты в код приложения, в отдельных случаях они используются для SSL pinning’а, а в некоторых – не применяются вовсе. Во многих сертификатах указаны адреса, к которым они относятся. Среди адресов указываются внутренние хосты, адреса платежных шлюзов, URL и API-серверы.

При разработке приложения сторонней компанией использовались сертификаты, содержащие информацию о хостах других компаний, адреса сервисов и иную информацию. Необходимо тщательно исследовать финальное приложение на наличие подобной информации.

**Использование HTTP-соединения**

Использование незащищенного HTTP-соединения дает возможность злоумышленнику при проведении успешной атаки «человек посередине» получать трафик в незашифрованном виде, видоизменять и просматривать его. Рекомендуется внедрять механизмы шифрования трафика на основании протокола TLS.

**Раскрытие координат пользователя**

Приложение может собирать и передавать по сети, данные геопозиции пользователя (широта и долгота) с точностью вплоть до одного метра. Необходимо предупреждать пользователя о собираемых данных и об их использовании.

**Отсутствие 2FA**

Отдельные финансовые организации полностью полагаются на пару логин-пароль и не требуют иных факторов подтверждения личности пользователя. Скомпрометировав данные учетной записи однажды, злоумышленник всегда сможет получать доступ к банковскому счету пользователя. Сегодня неприемлемо отказываться от второго фактора аутентификации.

**Длина SMS 2FA-кода**

Банки используют различную длину SMS-кода для авторизации в своей системе. Сегодня оптимальная длина кода второго фактора — 6 и более символов.

**Хранение критичной информации**

**Использование Keychain**

Keychain в iOS — это специальное зашифрованное хранилище, предназначенное для хранения критичной информации приложений.

Неиспользование Keychain не означает, что приложение хранит свои данные в открытом виде. Оно может вовсе не хранить никакой информации на клиентском устройстве. На устройствах с Jailbreak данные из Keychain могут быть получены злоумышленником. Если локальный файл используется как хранилище критичной информации, он может быть прочитан удалённо. Лучшее решение — не хранить никакую критичную информацию на устройстве. К сожалению, это не всегда возможно.

**Использование NSUserDefaults**

NSUserDefaults — это массив, отдельный для каждого приложения. Обычно он используется разработчиками для хранения неключевых настроек приложения, например, информации о языке приложения или дате последнего входа, что может пригодиться для статистики. Некоторые разработчики помещают в этот массив данные об учетной записи пользователя, карточную или иную критичную информацию. Для ее хранения рекомендуется использовать Keychain.

**Некорректная работа со Snapshot**

При сворачивании приложения в iOS автоматически создается снимок экрана, который устройство сохраняет до следующего открытия программы. Он отображается в разделе мультизадачности устройства, а также при разворачивании приложения. Скриншот экрана может содержать критичную информацию: переписку с банком, баланс счета и другие данные. Их необходимо маскировать.

**Критичная информация в памяти приложения**

Некорректная работа с памятью приложения в некоторых случаях приводит к попаданию критичных данных (логин, пароль, карточные данные и так далее) в память устройства в открытом виде. Сразу после использования необходимо освобождать память переменных, содержащих критичную информацию. На устройствах с Jailbreak это позволяет получить личные данные из снимка памяти. В отдельных случаях другие приложения могут получить доступ к памяти программы из-за ошибок системы.

**Остаются данные клиента в Keychain**

После завершения активной сессии за счет полного выхода из приложения, часть данных, принадлежащих прежнему пользователю, остается в Keychain устройства. Рекомендуется очищать всю клиентскую информацию из Keychain после завершения сессии.

**Остаются данные клиента в памяти**

После завершения активной сессии посредством полного выхода из приложения, часть данных, принадлежащих прежнему пользователю, остается в Cache устройства. Рекомендуется очищать всю клиентскую информацию из Cache файлов после завершения сессии.

**Остаются данные клиента в файлах**

После завершения активной сессии посредством полного выхода из приложения, часть данных, принадлежащих прежнему пользователю, остаётся в Cache устройства. Рекомендуется очищать всю клиентскую информацию из Cache файлов после завершения сессии.

**Остаются данные клиента в Cookie**

После завершения активной сессии посредством полного выхода из приложения, часть аутентификационных данных (cookie), принадлежащих прежнему пользователю, остаётся в файлах устройства. Рекомендуется очищать всю клиентскую информацию из Cookie после завершения сессии.

**Раскрытие информации о тестовом окружении**

В результате анализа в составе пакетов приложений было найдено большое количество отладочной, тестовой и иной информации, определяющей окружение инфраструктуры разработчиков приложений, самой компании или компаний, относящихся к банку. Помимо этих данных, некоторые приложения содержали файлы с информацией для внутреннего использования, приватные ключи или информацию о реализации нового функционала.

Перед каждой публикацией новой версии в App Store необходимо тщательно исследовать финальное приложение на наличие подобной информации.

**Механизмы защиты**

* Запуск на Jailbreak устройстве
* Отсутствие SSL Pinning

В связи со спецификой модели выполнения кода, на Swift и Objective-C легко восстановить информацию о назначении используемых классов и их методах даже без исходного кода, что значительно упрощает процесс восстановления работы программы и поиска уязвимостей в ней. Устройство с установленным Jailbreak в значительной степени подвержено различным угрозам безопасности. Это связано с тем, что большинство механизмов безопасности при Jailbreak отключается. Как правило, приложения, определяющие наличие Jailbreak, либо вообще отказываются работать на таком устройстве, либо предоставляют ограниченный функционал. Но даже при использовании всех вышеперечисленных техник приложение все равно можно исследовать и запускать на устройстве с Jailbreak. Они лишь усложняют и замедляют процесс исследования приложения, требуя более высокой квалификации исследователя.

**1.7 Результаты исследования безопасности Android приложений**

**Согласно исследованиям Digital Security**

**Длина SMS 2FA-кода:**

* 4/16 – 4 символа
* 2/16 – 5 символов
* 6/16 – 6 символов

**Анализ API:**

* 3/16 – Отсутствие 2FA
* 4/16 – Раскрытие координат пользователя
* 5/16 – Возможность утечки данных через Backup
* 6/16 – Наличие URI схемы
* 16/16 – Сбор метаданных

**Механизмы защиты:**

* 12/16 – Отсутствие предупреждения на устройстве с Root
* 1/16 – Отсутствие SSL Pinning

**Данные разработки:**

* 10/16 – SSL-сертификаты в приложении
* 4/16 – Раскрытие информации о тестовом окружении

**Хранение критичной информации:**

* 1/16 – Пароль в файлах после завершения сессии
* 1/16 – Запись отладочной информации в журнал устройства (logcat)
* 5/16 – Данные клиента в памяти после завершения сессии
* 6/16 – Пароль в память после завершения сессии

**Анализ API**

**Возможность утечки данных через Backup:**

На устройствах под управлением ОС Android приложения могут быть настроены через файл AndroidManifest.xml для включения их файлов в резервную копию системы. Если опция создания резервного дубликата включена, запасная копия может быть загружена в Google Cloud при наличии прикреплённого к устройству Google-аккаунта. При этом, если приложение не указывает, какие конкретно файлы должны попадать в копию, дублируется директория приложения целиком. Это особенно важно в случае, если приложение хранит в своей директории критичные или аутентификационные данные пользователя. В использовании резервных копий нет ничего плохого, однако включение этой функции требует дополнительного контроля содержимого файлов приложения.

**Сбор метаданных:**

Зачастую приложения имеют доступ к информации об устройстве: модель и производитель, название Wi-Fi сети, к которой подключен пользователь, а также наименованиям ближайших сетей, IMSI устройства, IMEI абонента и так далее. Нередко приложения передают эти данные на свои серверы или сервисам по сбору метаданных (одни из самых распространённых – AppMetrica и Crashlytics). Банковские приложения могут использовать эту информацию для запоминания устройства, с которого заходит пользователь, чтобы в дальнейшем при обнаружении подозрительной активности запросить дополнительные проверки и/или предупредить его. Необходимо информировать пользователей о собираемых и используемых данных.

**Раскрытие координат пользователя:**

Иногда кроме стандартного набора метаданных программа может также устанавливать точное местоположение устройства и передавать его на свои серверы либо сервисам по сбору метаданных. Пользователь должен знать о том, что информация о его местоположении попадает на серверы компании или сторонних сервисов.

**Отсутствие 2FA:**

Ряд финансовых организаций полностью полагается на пару логин-пароль и не требует иных факторов подтверждения пользователя. Скомпрометировав данные учетной записи один раз, злоумышленник всегда сможет получать доступ к банковскому счету пользователя. При аутентификации должен использоваться второй фактор для повышения безопасности пользователей приложения.

**Длина SMS кода 2FA:**

Разные банки используют различную длину SMS-кода для авторизации. Сегодня оптимальная длина кода второго фактора – 6 и более символов.

**Хранение критичной информации**

**Пароль в памяти после завершения сессии:**

После принудительного завершения активной сессии в памяти процесса остаются аутентификационные данные пользователя приложения. Следует убирать критичную информацию из памяти после работы.

**Пароль в файлах после завершения сессии:**

После принудительного завершения активной сессии в файлах процесса остаются аутентификационные данные пользователя приложения. Если в приложении поставлена настройка резервного копирования файлов, его аутентификационные данные окажутся в любой резервной копии устройства. Следует убирать критичные данные из памяти после работы с устройством.

**Данные клиента в памяти после завершения сессии:**

После принудительного завершения активной сессии в памяти процесса остаются личные данные пользователя приложения. Следует убирать критичные данные из памяти после работы с устройством.

**Запись отладочной информации в журнал устройства:**

Разработчики могут пользоваться функцией записи в журнал устройства для отладки своего приложения. Однако они могут оставить эту функцию при переходе из тестовой версии в окончательную. Эти записи иногда содержат критичную информацию или аутентификационные данные пользователя. Стоит отметить, что доступ к ним могут получить только пользователи, имеющие доступ по ADB к устройству, или злоумышленник с правами пользователя Root (как правило, это невозможно в современных версиях Android, если сам пользователь не повысил свои привилегии). Отладочная информация не должна попадать в журнал устройства.

**SSL-сертификаты в приложении:**

Банковские клиенты могут содержать в архиве приложения сертификаты или хранилище ключей с сертификатами. Некоторые из них используются для обеспечения SSL Pinning, а другие применялись в тестовой среде и не были удалены. Кроме имени домена они также могут содержать в себе информацию о компании-разработчике и имена конкретных разработчиков программы. Необходимо контролировать информацию и файлы, которые попадают в окончательную версию приложения.

**Раскрытие информации о тестовом окружении:**

При разработке банковских приложений может использоваться тестовая среда для анализа и отладки. При переходе к финальной версии продукта и выпуске его в магазин приложений операционной системы в теле программы могут остаться файлы или элементы разработки: IP-адреса или домены локальной сети, информация о среде разработки, ники или имена программистов, информация о компании-разработчике и так далее. Необходимо контролировать информацию и файлы, которые попадают в финальную версию приложения.

**Механизмы защиты**

**Отсутствие предупреждения на устройстве с Root:**

Многие приложения определяют наличие повышенных привилегий на устройстве. Однако реагируют на этот фактор лишь некоторые из них. Самая распространенная реакция — это предупреждение пользователя в начале работы о том, что информация приложения может быть скомпрометирована и устройство с Root-доступом потенциально небезопасно. В этом случае клиент осведомлён о том, что на его устройстве есть Root-доступ (если он был получен без его ведома), а также о том, что это опасное окружение для приложений с банковской информацией. Клиент должен быть предупреждён о том, что он работает в небезопасном окружении.

**Отсутствие SSL Pinning:**

Один из самых надёжных способов обеспечения безопасности передачи информации между клиентом и сервером – это дополнительная проверка приложения на использование конкретного сертификата безопасности при передаче данных. Учитывая простоту имплементации, при нынешнем уровне развития разработки мобильных приложений наличие SSL Pinning в программе является одним из первых способов защиты от атак «человек посередине».

**Методология анализа взаимодействия операторов сотовой связи с банками**

Анализ взаимодействия операторов сотовой связи с кредитно-финансовыми организациями может быть осуществлен несколькими способами. После применения любого из этих способов производится проверка работоспособности мобильного и онлайн ДБО, в том числе на возможность вывода денег с помощью сторонних сервисов. Во всех случаях банковский счёт пользователя привязан к номеру мобильного оператора, который будет использован для проверок.

**Примеры взаимодействия:**

1. Пользователь заказывает перевыпуск SIM-карты у мобильного оператора. В этом случае старая SIM-карта блокируется, новая становится активной, а через 24 часа после перевыпуска на новой карте активируются SMS-сервисы;
2. Пользователь банковского счёта инициирует процедуру переноса номера абонента к другому поставщику услуг связи (MNP). Перенос занимает несколько дней.

**Подмена Caller ID с помощью SIP-операторов:**

**Были осуществлены проверки:**

* поведения банков при подмене Caller ID абонента;
* запроса дополнительной информации для аутентификации в системе;
* уровня раскрываемых данных с минимальным количеством информации о клиенте.

**Необходимыми компонентами для осуществления подобных проверок являются:**

* контракт с SIP оператором, позволяющим осуществлять подмену Caller ID;
* устройство, позволяющее осуществлять звонки через VOIP.

**1.2 Разработка требований к мобильному приложению и выбор средства разработки**

Разработка приложений для Android является наиболее предпочтительным вариантом для приложения Исламской ипотеки. Здесь у разработчика имеется в распоряжении: Android Studio, Ganymotion, библиотеку SDK, а также JDK (Java Development Kit). Большинство разработчиков предпочитают использовать [службы разработки приложений для Android](https://devtechnosys.com/android-app-development.php) из-за многочисленных преимуществ, которые он может предложить. Он очень прост в использовании и поставляется с библиотекой фрагментов кода для ускорения процесса разработки.

**Java IDE**

Для разработки мобильного банковского приложения Ак Барс Ипотека использовалась среда разработки (IDE) Java SE 14. Java – это универсальный язык программирования, основанный на классах, объектно-ориентированный и предназначенный для того, чтобы иметь как можно меньше зависимостей реализации. Он предназначен для того, чтобы позволить разработчикам приложений писать один раз, запускать в любом месте (WORA), что означает, что скомпилированный код Java может работать на всех платформах, поддерживающих Java, без необходимости перекомпиляции.

Приложения Java обычно компилируются в байт-код, который может работать на любой виртуальной машине Java (JVM) независимо от базовой архитектуры компьютера. Синтаксис Java аналогичен C и C++, но он имеет меньше низкоуровневых средств, чем любой из них. По состоянию на 2019 год Java была одним из самых популярных языков программирования, используемых в соответствии с GitHub, особенно для клиент-серверных веб-приложений, с сообщением о 9 миллионах разработчиков.

К основным особенностям Java относится:

* Можно работать на разных операционных системах (Windows, Mac OS X, Linux, Solaris);
* Программа позволяет разрабатывать приложения на разных языках, такие как Eclipse, Java, Python, PHP, Cobol, C и С++, Perl;
* В Java все является объектом. Java может быть легко расширена, так как она основана на объектной модели.
* В отличие от многих других языков программирования, включая C и c++, когда Java компилируется, она не компилируется в конкретную машину платформы, а в независимый от платформы байтовый код. Этот байтовый код распространяется по сети и интерпретируется виртуальной машиной (JVM) на любой платформе, на которой он выполняется.
* Java создана для того, чтобы быть легкой в освоении. Если вы понимаете основную концепцию ООP Java, то это будет легко освоить.
* С помощью безопасной функции Java он позволяет разрабатывать системы, свободные от вирусов и несанкционированного доступа. Методы аутентификации основаны на шифровании с открытым ключом.
* Компилятор Java генерирует архитектурно-нейтральный формат объектного файла, который делает скомпилированный код исполняемым на многих процессорах, с наличием Java runtime system.
* Будучи нейтральным к архитектуре и не имея зависимых от реализации аспектов спецификации, Java переносится. Компилятор в Java написан на языке ANSI C с чистой границей переносимости, которая является подмножеством POSIX.
* Java прилагает усилия для устранения ситуаций, подверженных ошибкам, уделяя основное внимание проверке ошибок во время компиляции и проверке времени выполнения.
* С помощью многопоточной функции Java можно писать программы, которые могут выполнять множество задач одновременно. Эта конструктивная особенность позволяет разработчикам создавать интерактивные приложения, которые могут работать плавно.
* Байт-код Java переводится на лету в собственные машинные инструкции и нигде не хранится. Процесс разработки является более быстрым и аналитическим, поскольку связывание является постепенным и легким процессом.
* С помощью компиляторов Just-In-Time Java обеспечивает высокую производительность.
* Java предназначена для распределенной среды интернета.
* Java считается более динамичной, чем C или C++, поскольку она предназначена для адаптации к меняющейся среде. Программы Java могут нести большой объем информации во время выполнения, которая может быть использована для проверки и разрешения доступа к объектам во время выполнения.

**SDK**

SDK состоит из инструментов, которые включают исполняемую программу для разработки корпоративного программного обеспечения, такую ​​как эмуляторы, отладчики, документация, профилировщики и многое другое.

**Редакторы и IDE**

Можно написать приложение в широко используемом текстовом редакторе или командной строке, но предпочтительно вместо этого выбрать «Интегрированную среду разработки». Вы можете легко интегрировать инструмент со всеми доступными инструментами SDK, поскольку это позволит вам использовать ресурсы без каких-либо проблем.

Помимо Android Studio, также можно использовать Eclipse, который поставляется с множеством плагинов, которые можно использовать для расширения кода на любом другом языке программирования.

Другой популярный вариант - IntelliJ IDEA, который поставляется в платной версии и может быть настроен в соответствии с вашими личными требованиями и процессом разработки.

**Библиотеки**

Библиотеки имеют большую полезность, так как содержат фрагменты ранее написанного кода, которые помогают упростить работу кодера.

Это позволило бы сэкономить много времени и обеспечить разработку приложения с помощью доступных бесплатных решений. Одним из наиболее популярных вариантов является использование GSON, который будет использоваться для сериализации объектов Java для обеспечения надлежащего взаимодействия с API.

Другой популярный доступный вариант - это Retrofit и Event Bus, которые можно использовать для обеспечения связи с несколькими элементами приложения.

Чтобы настроить параметр, вы можете организовать процесс с помощью внешних библиотек. Разработчики также могут использовать инструмент под названием Gradle для сборки приложения.

**Плагины**

Основной аспект использования библиотек заключается в том, что их можно использовать для автоматизации задач. Кроме того, плагины могут быть созданы путем расширения программного инструмента.

В этом случае это будет IDE. Если вы хотите использовать комплексные решения для улучшения своего кода, то вы можете использовать популярные инструменты, предназначенные для разработки под Android.

Приложения «Ак Барс» разрабатываются с использованием компонентов, добавленных в Android API 14 (Application programming interface), поэтому данные приложения поддерживают Android версии 4.0 и выше.