БЕЗОПАСНОСТЬ ЦИФРОВЫХ ТЕХНОЛОГИЙ. – 2023. – № 4 (111). – 47–63

МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 004.056.5 DOI: 10.17212/2782-2230-2023-4-47-63

ФОРМИРОВАНИЕ ТИПОВЫХ ТРЕБОВАНИЙ БЕЗОПАСНОСТИ ПРИ РАЗРАБОТКЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ*

А.В. НОСЕНКО[1], Т.М. ПЕСТУНОВА[2]

[1] *630090, г. Новосибирск, ул. Пирогова, 2, Новосибирский государственный университет, магистрант направления «Информатика и вычислительная техника» ФИТ. E-mail: a.nosenko1@g.nsu.ru*

[2] *630090, г. Новосибирск, ул. Пирогова, 2, Новосибирский государственный университет, кандидат технических наук, доцент, кафедра компьютерных систем ФИТ. E-mail: t.pestunova@g.nsu.ru*

В статье рассмотрена проблема обеспечения защищенности мобильных приложений, разработка которых часто осуществляется без привлечения специалистов по ИТ-безопасности и в отсутствие доступа к специализированной инфраструктуре программной инженерии. Проанализированы возможные причины недостаточной защищенности мобильного программного обеспечения (МО) и применяемые на практике методы безопасной разработки. На основе лучших практик и рекомендаций разработан базовый перечень требований безопасности к функциям нативного мобильного приложения, инфраструктуре и методам его разработки, реализация которых доступна одиночным разработчикам, не имеющим возможности экспертной поддержки процесса разработки мобильного ПО.

Ключевые слова: мобильное приложение, защита информации, информационная безопасность, безопасное программное обеспечение, требования безопасности, жизненный цикл разработки, тестирование защищенности, угрозы безопасности

ВВЕДЕНИЕ

Мобильные приложения прочно вошли в жизнь организаций и индивидуальных пользователей, деятельность которых подвержена множеству информационных угроз. По статистике, мобильными устройствами пользуется 68 %

* Статья получена 09 ноября 2023 г.

населения мира [1, 12]. Мобильные приложения являются одним из наиболее распространенных объектов атак [3, 7, 17, 18]. Кроме прочего, это можно объяснить и тем, что мобильные устройства стали неотъемлемой частью рабочей среды и коммуникаций, а значит, содержат значительное количество критичной для бизнеса информации.

По оценке экспертов, более 80 % мобильных приложений российских разработчиков содержат уязвимости высокого или критичного уровня [2]. Последние исследования показывают, что наиболее распространены уязвимости небезопасного хранения данных [2, 21]. К примеру, приложения хранят конфиденциальную информацию в исходном коде или в открытом виде. Эти уязвимости являются критическими, поскольку позволяют злоумышленникам получить доступ к конфиденциальной информации пользователей и компаний. Также распространены уязвимости, позволяющие реализовать угрозы целостности приложений. Они могут позволить злоумышленнику изменить логику работы приложения путем модификации кода или внедрения в него вредоносных фрагментов. Например, можно перенаправлять пользователей на фишинговые сайты или перехватывать данные, передаваемые между приложением и сервером. Кроме того, актуальны уязвимости, связанные с сетевой безопасностью, такие как отсутствие шифрования или небезопасная конфигурация сетевого взаимодействия.

Многие из распространенных недостатков мобильных приложений, позволяющих реализовывать эффективные атаки на них, входят в список самых распространенных уязвимостей, по версии OWASP, с 2016 года и по настоящее время [6].

1. СЛЕДОВАНИЕ ЛУЧШИМ ПРАКТИКАМ КАК СПОСОБ ПОВЫШЕНИЯ ЗАЩИЩЕННОСТИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

По оценкам исследователей, при разработке мобильных приложений уделяется недостаточно внимания применению методов безопасного программирования [4, 5, 19]. В целом, аналитики сходятся во мнении, что многие разработчики приложений не имеют достаточной подготовки в практике безопасного программирования. Как правило, повышение квалификации в вопросах ИБ происходит за счет обучения на собственном опыте и требует проявления энтузиазма. Подчеркивается, что нет универсального подхода к мотивации разработчика уделять должное внимание

вопросам безопасности. Главными приоритетами в большинстве случаев являются написание чистого и масштабируемого кода, соблюдение принципов проектирования, в то время как безопасность считается скорее необязательным критерием. Более того, существенно усложняет процесс безопасной разработки отсутствие доступных методов обеспечения безопасности, не требующих привлечения специалистов по ИБ. Таким образом, к основным причинам низкой защищенности мобильных приложений, в частности, можно отнести незаинтересованность разработчиков и отсутствие возможности обращения к специалистам ИБ.

Бизнес, заинтересованный в обеспечении защищенности своих продуктов, выделяет средства на безопасность. По последним отчетам, такие затраты составляют в среднем 30 % от ИТ-бюджета [11]. Обратимся к опыту организаций [8, 22]. Существует множество методов, позволяющих повысить защищенность приложений, среди них обучение разработчиков, проведение «кодревью» и тестирования на проникновение, использование автоматического статического, динамического и интерактивного анализа приложений, выстраивание процесса своевременного обновления зависимостей, моделирование и оценка угроз и в целом построение жизненного цикла безопасной разработки (Security Development Lifecycle). Для выполнения этих задач необходимо привлечение специалистов ИБ.

У индивидуальных разработчиков, в частности, часто отсутствует возможность обращения к специалистам по информационной безопасности. Кроме того, они находятся в условиях ограниченных ресурсов, что делает недоступными некоторые методы обеспечения безопасности. В ходе создания ПО разработчик обычно опирается на собственные знания о безопасности, поэтому их необходимо постоянно актуализировать. Разумно использовать стандарты безопасной разработки, но они сложны для понимания и реализации. Существуют также рекомендации в форме лучших практик. Они могут относиться к некоторой конкретной технологии или фреймворку, носят рекомендательный характер, а при создании защищенного приложения могут быть взяты за основу для определения базовых требований. Их структура должна учитывать три аспекта: требования к процессу разработки, требования к инфраструктуре и функциональные требования безопасности (рисунок).

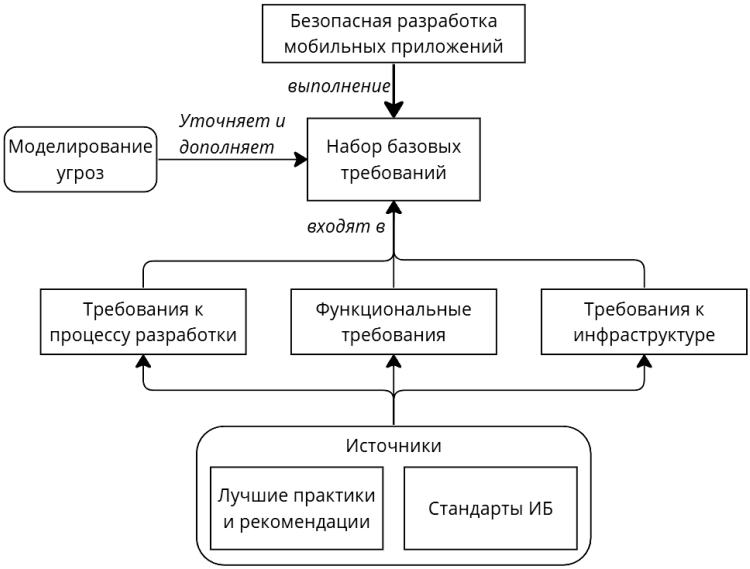

Схема безопасной разработки

Diagram of the system requirements creation process

Для выявления дополнительных требований, направленных на предотвращение актуальных угроз, необходимо проводить периодическое моделирование угроз. Сформированный набор мер может быть использован при разработке методов оценки приложений на предмет соответствия требованиям информационной безопасности.

2. ИСТОЧНИКИ ТРЕБОВАНИЙ

В роли основного источника выступили рекомендации по безопасной разработке от OWASP [13]. Они представляют собой краткое справочное руководство по основным вопросам безопасности приложений, включая практические советы и методы повышения защищенности, а также объяснение рекомендуемых к применению технологий и обоснование требований стандарта MASVS (Mobile Application Security Verification Standard). Стандарт верификации требований к безопасности приложений MASVS является широко используемым многими организациями и специалистами по безопасности во всем мире [9]. Стандарт содержит набор критериев безопасности мобильных

приложений, разделенных по восьми категориям. OWASP рекомендует использовать MASVS как для оценки безопасности продукта, так и в качестве руководства по безопасной разработке. Стандарт поддерживается Google (разработчиком ОС Android), NIST (Национальным институтом стандартов и технологий США) и другими правительственными и образовательными учреждениями. Несомненным преимуществом MASVS является его взаимосвязанность с другими методологиями и стандартами. Во-первых, это CWE – общий перечень недостатков безопасности программного обеспечения. Соответствие между требованиями стандарта и этим перечнем помогает получить представление о потенциальной уязвимости, а также позволяет связывать MASVS с другими стандартами, которые поддерживают CWE, в частности, NIST SP 800-163 [16], Common Criteria for Information Technology Security Evaluation [23] и др. Также стандарт имеет ссылки на MSTG (Mobile Security Testing Guide) – руководство по тестированию защищенности мобильных приложений [15]. Таким образом, все требования стандарта, за исключением требований раздела «Архитектура», имеют соответствующий сценарий проверки выполнимости. Более того, на основе стандарта создан чек-лист MAS (Mobile Application Security), который удобен для применения в процессе разработки и также имеет ссылки на сценарии тестирования. MASVS можно использовать и в качестве инструмента, помогающего обеспечить соответствие мобильных приложений требованиям, изложенным в отраслевых стандартах, поскольку его структура и критерии во многом им соответствуют. В частности, к стандартам, требованиям которых соответствуют рекомендации MASVS, относятся PCI DSS, NIST SP 800-53.

Еще один источник требований – веб-сайт для разработчиков ОС Android [24], на котором предоставлена информация о возможностях платформы, в том числе о функциях безопасности. Документация содержит разделы о безопасности мобильных приложений, включая лучшие практики безопасной разработки с описанием правильного применения функций безопасности Android, рекомендациями по защите передаваемых по сети данных, применению криптографии, хранению конфиденциальной информации и др.

Также источниками требований послужили рекомендации по безопасной разработке от компаний Digital Security [20] и «Стингрей Технолоджиз» [14], специализирующиеся на анализе защищенности мобильных приложений. Digital Security предлагает чек-лист, содержащий требования к использованию нативного API системы, к хранению данных и к архитектуре. Лучшие практики от «Стингрей Технолоджиз» охватывают такие вопросы, как менеджмент ключей и сертификатов, хранение и передача критичной информации, логирование, конфигурация.

3. ОСНОВНЫЕ ТРЕБОВАНИЯ БЕЗОПАСНОСТИ К РАЗРАБОТКЕ И ФУНКЦИЯМ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

На основе перечисленных источников разработан набор базовых требований (табл. 1) к процессу разработки (П) и функциональные требования безопасности (Ф). Требования к инфраструктуре можно рассматривать как требования к инфраструктуре разработчика или к среде функционирования приложения. В приведенных источниках рекомендаций по безопасности для мобильных приложений этот класс требований не рассматривается.

Таблица 1

Table 1

Перечень требований

List of requirements

Ф1. Требования к хранению критичной информации	
Ф1.1. Хранение данных в оперативной памяти	
Требования	
Ф1.1.1	**Необходимо** хранить критичные данные в примитивных типах данных, таких как байтовый или символьный массивы. **Не рекомендуется** использовать неизменяемые и непримитивные типы
Ф1.1.2	Если данные хранятся в непримитивных объектах, то **необходимо** выполнять явный вызов сборщика мусора и корректное удаление ссылок. Ссылки должны быть перезаписаны перед удалением вне метода finalize
Ф1.1.3	Переменные, содержащие конфиденциальную информацию, после использования **необходимо** перезаписать значениями других переменных или случайными значениями. Чтобы обойти оптимизации компилятора, **рекомендуется** сохранить переменные после перезаписи во временный файл, например /dev/null, однако это может негативно сказаться на производительности
Ф1.1.4	**Рекомендуется** обрабатывать критичную информацию минимальным числом компонентов
Ф1.2. Хранение данных на устройстве	
Ф1.2.1	**Рекомендуется** запретить резервное копирование, для этого в манифесте нужно установить значение false параметра allowBackup
Ф1.2.2	Для хранения критичных данных в открытом виде **необходимо** использовать зашифрованные хранилища: Encrypted SQLite Database (SQLCipher), Encrypted Realm Database, EncryptedSharedPreferences, EncryptedFile
Ф1.2.3	Если используется зашифрованное хранилище, то ключ шифрования не должен находиться в коде или храниться в открытом виде. **Рекомендуется** получать ключ из пароля или PIN

Продолжение табл. 1

Continuation of the Tab. 1

Ф1.2.4	Для хранения критичных данных в зашифрованном виде **рекомендуется** использовать хранилища: Shared Preferences, SQLite Database, Realm Database, Internal Storage
Ф1.2.5	Если используется Shared Preferences или Internal Storage, то **необходимо** установить режим MODE_PRIVATE
Ф1.3. Хранение криптографических материалов	
Ф1.3.1	Криптографические материалы **необходимо** хранить в специализированном хранилище (Keychain/Keystore)
Ф1.3.2	Если поддерживаются, то **необходимо** использовать специализированное хранилище с аппаратной поддержкой, иначе **необходимо** использовать программную реализацию
Ф1.3.3	Если используется Keystore API, то **необходимо** использовать AndroidKeystore-реализацию
Ф1.3.4	При использовании AndroidKeystore **необходимо** установить флаг unlockedDeviceRequied в *true* для предотвращения расшифровки ключей приложения при заблокированном устройстве.
Ф1.3.5	При использовании аппаратного хранилища **необходимо** установить флаг setIsStrongBoxBacked в *true* для защиты ключей модулем StrongBox Keymaster
Ф1.3.6	Если используется программная реализация Keystore, то **необходимо** отказаться от явной передачи имени провайдера в KeyStore.getInstance
Ф1.3.7	Если используется программная реализация Keystore, то для разблокировки и проверки ее целостности **необходимо** использовать ключ, который должен быть получен из PIN или пароля пользователя
Ф1.3.8	**Необходимо** использовать биометрическую аутентификацию для управления доступом к элементам Keystore. Биометрическая аутентификация пользователя должна выполняться при каждом использовании ключа
Ф1.3.9	**Необходимо** обеспечить инвалидацию хранимых ключей при регистрации новой биометрической информации
Ф1.4. Валидация данных из общедоступных хранилищ	
Ф1.4.1	**Необходимо** валидировать данные, извлекаемые из общедоступных хранилищ, таких как Shared Preferences. Валидацию **необходимо** производить в момент чтения
Ф1.4.2	**Необходимо** проверять целостность хранимых данных
Ф1.5. Отображение данных в пользовательском интерфейсе	
Ф1.5.1	**Необходимо** отключить кэш-клавиатуры, автозаполнение и проверку правописания для полей ввода критичных данных. Для этого нужно установить флаг *android:inputType textNoSuggestions*
Ф1.5.2	**Необходимо** маскировать поля с критичными данными, чтобы они не были видны через пользовательский интерфейс

Продолжение табл. 1

Continuation of the Tab. 1

Ф1.5.3	**Необходимо** скрывать критичные данные в фоновом режиме
Ф1.5.4	**Необходимо** запретить скриншоты на экранах с критичными данными. Для этого нужно установить флаг *FLAG_SECURE*. Флаг указывает системе, что содержимое экрана не должно попасть в скриншот
Ф2. Требования к аутентификации	
Ф2.1. Общие требования	
Ф2.1.1	Если приложение предоставляет пользователям доступ к удаленным сервисам, **необходимо** проводить аутентификацию на бэкенде
Ф2.1.2	**Необходимо** реализовать безопасный второй фактор аутентификации, такой как OTP на основе времени (Time-based One-time Password Algorithm), push-уведомление или иной, который также необходимо использовать при регистрации и восстановлении аккаунта
Ф2.1.3	Проверку аутентификации **необходимо** проводить до обработки диплинков и push-сообщений
Ф2.1.4	Если используется access-токен, то при включенном входе по PIN-коду или биометрии **необходимо** хранить его только в памяти. Токен не должен храниться на файловой системе устройства
Ф2.1.5	Если используется access-токен, то его **необходимо** подписывать безопасным криптоалгоритмом на бэкенде
Ф2.1.6	Если используются сессии, то идентификатор сессии **необходимо** генерировать случайно на бэкенде с помощью безопасного генератора
Ф2.1.7	Если используются сессии, то **необходимо** удалять на бэкенде существующую сессию при выходе пользователя из системы
Ф2.1.8	Если пользователь аутентифицируется по паролю, то **необходимо** настроить парольную политику. **Рекомендуется** установить минимальную длину пароля в 12 символов и разрешить любые печатные символы Unicode
Ф2.1.9	Если пользователь аутентифицируется по паролю, то **необходимо** предоставить возможность менять пароль. При смене пароля **необходимо** ввести старый и новый пароли. При смене пароля **необходимо** сбрасывать существующие сессии и токены
Ф2.1.10	**Необходимо** реализовать на бэкенде защиту от перебора авторизационных данных
Ф2.1.11	**Необходимо** предоставить пользователю возможность просматривать список устройств и сессий и возможность блокировать определенные устройства, а также инвалидировать определенную сессию
Ф2.2. Требования к аутентификации по PIN-коду	
Ф2.2.1	Необходимо избегать хранения PIN-кода на устройстве или серверной части
Ф2.2.2	Для генерации ключа шифрования из PIN необходимо использовать криптостойкие алгоритмы формирования ключа, такие как Argon2 или PBKDF2

Продолжение табл. 1

Continuation of the Tab. 1

Ф2.2.3	При реализации шифрования **необходимо** соблюдать требования по выбору алгоритма шифрования, длины ключа, криптографических параметров и материалов, описанных в разделе *3.3 Шифрование*
Ф2.3. Требования к биометрической аутентификации	
Ф2.3.1	**Необходимо** основывать биометрическую аутентификацию на разблокировке доступа к записям в Keychain/Keystore. Нельзя основывать ее на определенном событии, таком как вызов api
Ф2.3.2	При изменении настроек биометрии вход в приложение по ним должен блокироваться, для этого **необходимо** проверять наличие изменений в хранилище после входа
Ф3. Требования к криптографии	
Ф3.1. Генераторы случайных чисел	
Ф3.1.1	Генерацию криптографически сильных случайных чисел **необходимо** производить встроенными средствами языка программирования
Ф3.1.2	Для инициализации генератора **необходимо** использовать значение с достаточной энтропией
Ф3.1.3	Для генерации криптографически сильных случайных чисел **необходимо** использовать java.security.SecureRandom. **Рекомендуется** использовать метод getInstanceStrong()
Ф3.1.4	**Необходимо** выбирать стойкий алгоритм генерации. По умолчанию используется стойкий NativePRNGBlocking
Ф3.2. Управление ключами	
Ф3.2.1	**Необходимо** импортировать ключи только из доверенных мест
Ф3.2.2	Генерацию симметричного ключа из пароля **рекомендуется** производить с помощью алгоритма PBKDF2
Ф3.2.3	Для генерации пары публичного и приватного ключа **рекомендуется** использовать KeyPairGenerator с KeyGenParameterSpec
Ф3.2.4	Для генерации симметричного ключа **рекомендуется** использовать KeyGenerator с KeyGenParameterSpec, или криптографически стойкий генератор случайных чисел
Ф3.2.5	**Необходимо** безопасно хранить ключи согласно требованиям раздела по хранению криптографических материалов
Ф3.2.6	При передаче ключей между устройствами или между клиентской и серверной частями приложения **необходимо** обеспечить шифрование
Ф3.3. Шифрование	
Ф3.3.1	Криптографические операции **необходимо** осуществлять встроенными средствами Android SDK
Ф3.3.2	**Необходимо** использовать распространенные алгоритмы с доказанной стойкостью, такие как AES или RSA

Продолжение табл. 1

Continuation of the Tab. 1

Ф3.3.3	**Необходимо** выбирать достаточную длину ключа и алгоритм шифрования. Для выбора длины ключа и алгоритма шифрования **рекомендуется** пользоваться стандартом NIST SP 800-57. Нужно ориентироваться на время хранения данных
Ф3.3.4	В качестве симметричного алгоритма шифрования **рекомендуется** использовать AES с длиной ключа 256 бит и GCM-режимом шифрования
Ф3.3.5	В качестве асимметричного алгоритма шифрования **рекомендуется** использовать RSA с длиной ключа 3072 бит
Ф3.3.6	Для генерации случайных значений **необходимо** использовать криптографически стойкие генераторы случайных чисел
Ф3.3.7	**Необходимо** обеспечить уникальность вектора инициализации
Ф3.3.8	**Необходимо** использовать безопасный режим блочного шифрования. Рекомендуется CBC
Ф3.3.9	**Необходимо** использовать PKCS7 алгоритм дополнения
Ф3.4. Контроль целостности и HMAC	
Ф3.4.1	Генерацию HMAC **необходимо** осуществлять исключительно встроенными средствами языка
Ф3.4.2	**Необходимо** использовать безопасные алгоритмы хеширования: SHA-256, SHA-384, SHA-512, Blake2, SHA-3
Ф4. Требования к передаче данных	
Ф4.1	**Необходимо** использовать SSL Pinning или Certificate Transparency (CT). Для поддержки CT рекомендуется использовать библиотеки *Certificate Transparency for Android* и *Conscrypt – A Java Security Provider*
Ф4.2	В WebView **необходимо** использовать SSL Pinning
Ф4.3	Любую передачу данных **необходимо** проводить по защищенному соединению с использованием TLS
Ф4.4	**Необходимо** использовать TLS v1.2 или v1.3
Ф4.5	Если используются TLS v1.0 или v1.1, то **необходимо** отключить поддержку небезопасных шифронаборов. Для определения рекомендованных шифронаборов рекомендуется обратиться к таблице IANA
Ф5. Требования к ведению журнала	
Ф5.1	**Необходимо** избегать попадания критичной информации в журнал
Ф5.2	При отсутствии необходимости вести журнал в производственной версии приложения, **необходимо** удалить все операторы логирования
Ф6. Требования к использованию WebView	
Ф6.1. Общие требования	
Ф6.1.1	**Необходимо** производить сетевое взаимодействие с серверами только по защищенному каналу
Ф6.1.2	**Необходимо** отключить поддержку Javascript, если она не требуется

Окончание табл. 1

End of the Tab. 1

Ф6.2. Требования к ограничению ресурсов	
Ф6.2.1	**Необходимо** настроить белый список разрешенных ресурсов
Ф6.2.2	**Необходимо** настроить белый список разрешенных протоколов. **Рекомендуется** разрешить только https. **Необходимо** запретить поддержку потенциально опасных URL-схем: file, tel, app-id
Ф6.2.3	**Необходимо** создать контрольные суммы для локальных HTML- и Javascript-файлов, загружаемых в WebView, и проверять их во время запуска
Ф6.2.4	**Рекомендуется** использовать SafeBrowsing API библиотеку SafetyNet с возможностью настройки URL-схем
Ф6.2.5	**Рекомендуется** использовать VirusTotal API
Ф6.2.6	**Необходимо** запретить доступ к content provider. Для этого установить setAllowContentAccess в *false*
Ф6.2.7	**Необходимо** запретить доступ к файловой системе. Для этого установить setAllowFileAccess в *false*
Ф6.2.8	**Необходимо** запретить доступ из Javascript, работающего в контексте file:// схемы URL, к содержимому других ресурсов с file:// схемой URL. Для этого флаг setAllowFileAccessFromFileURLs должен быть установлен в *false*
Ф6.2.9	**Необходимо** запретить доступ из Javascript, работающего в контексте file:// схемы URL, к содержимому других ресурсов с любого origin. Для этого флаг setAllowUniversalAccessFromFileURLs должен быть установлен в *false*
Ф6.3. Требования к загрузке локальных ресурсов	
Ф6.3.1	**Необходимо** минифицировать локальные Javascript файлы
Ф6.3.2	**Необходимо** размещать локальные HTML- и JS-файлы в каталоге приложения
Ф6.3.3	**Необходимо** использовать WebViewAssetLoader для доступа к локальным HTML и JS файлам по http://
Ф6.3.4	**Необходимо** запретить возможность изменения имени файла, пути, а также содержимого файла со стороны пользователя
Ф6.4. Требования к загрузке внешних ресурсов	
Ф6.4.1	**Необходимо** проводить сетевое взаимодействие только по https
Ф6.4.2	**Необходимо** запретить возможность изменения URL пользователем
Ф6.4.3	**Необходимо** очищать кэш, хранилище и загруженные ресурсы WebView перед уничтожением WebView
П1. Требования к производственной версии приложения	
П1.1	**Необходимо** отключить режим отладки в производственной версии
П1.2	**Необходимо** подписывать приложение валидным сертификатом и реализовать проверку подписи
П1.3	**Необходимо** удалить файлы с информацией для внутреннего использования, приватные ключи и другую раскрывающую логику информацию
П1.4	**Необходимо** запрашивать только минимальное количество разрешений для правильной работы приложения

Таким образом, функциональные требования включают требования к хранению критичных данных, требования к аутентификации, требования к криптографии, требования к передаче данных по сети, требования к ведению журнала, требования к использованию WebView, требования к сборке. Они согласуются с документацией Android и лучшими практиками от Digital Security и «Стингрей Технолоджиз» (табл. 2).

Таблица 2

Table 2

Соответствие требований и источников

Matching requirements and sources

Раздел	MASVS	ОС Android для разработчиков	Digital Security	«Стингрей Технолоджиз»
Хранение критичной информации	+	+	+	+
Аутентификация	+	+	+	–
Криптография	+	+	+	+
Передача данных	+	+	+	+
Ведение журнала	+	+	+	–
Использование WebView	+	+	+	+
Производственная версия приложения	+	+	+	–

ЗАКЛЮЧЕНИЕ

В результате исследований определены требования безопасности к разработке мобильных приложений. Они могут быть соотнесены с типовыми угрозами безопасности. В целом, данные требования соответствуют стандарту MASVS на уровне L1, за исключением ряда требований (к архитектуре, дизайну, модели угроз и стойкости к некоторым трудоемким атакам), являющихся сложными для реализации разработчиками, не имеющими возможности экспертной поддержки специалистами по безопасности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Digital 2023: Global Overview Report. – URL: https://datareportal.com/reports/digital-2023-global-overview-report (accessed: 05.12.2023).

2. Оценка защищенности мобильных приложений российских разработчиков. Исследование «Стингрей Технолоджиз». – URL: https://stingray-mobile.ru/wp-content/uploads/2022/12/stingray-annual-report-2022.pdf (дата обращения: 05.12.2023).

3. *Шишкова Т.* Развитие информационных угроз в первом квартале 2022 года. Мобильная статистика. – URL: https://securelist.ru/it-threat-evolution-in-q1-2022-mobile-statistics/105235/ (дата обращения: 05.12.2023).

4. *Weir C., Hermann B., Fahl S.* From needs to actions to secure apps? The effect of requirements and developer practices on app security // Proceedings of the 29th USENIX Security Symposium. – USENIX Association, 2020. – P. 289–305. – URL: https://www.usenix.org/system/files/sec20-weir.pdf (accessed: 05.12.2023).

5. *Weir C., Rashid A., Noble J.* How to improve the security skills of mobile app developers? Comparing and contrasting expert views // Proceedings of the 2016 ACM Workshop on Security Information Workers. – USENIX Association, 2016. – URL: https://eprints.lancs.ac.uk/id/eprint/80016/1/SOUPS2016_SIW_AppDev_CW7June16_submitted.pdf (accessed: 05.12.2023).

6. OWASP. Mobile Top 10 2023: Updates. – URL: https://owasp.org/www-project-mobile-top-10/ (accessed: 05.12.2023).

7. *Townsend K.* Как смартфоны стали одной из главных целей кибератак. – URL: https://blog.avast.com/ru/smartphones-and-increasing-mobile-threats-avast (дата обращения: 05.12.2023).

8. *Михайлова А.* Мобильные угрозы и методы борьбы с ними. – URL: https://www.securitylab.ru/analytics/501302.php (дата обращения: 05.12.2023).

9. OWASP. Mobile Application Security. – URL: https://mas.owasp.org/MASVS/ (accessed: 05.12.2023).

10. Mobile Security Primer. – URL: https://books.nowsecure.com/secure-mobile-development/en/primer/mobile-security.html (accessed: 05.12.2023).

11. Калькулятор бюджета на информационную безопасность «Лаборатории Касперского». – URL: https://calculator.kaspersky.com/ru (accessed: 05.12.2023).

12. State of Mobile 2023. – URL: https://www.data.ai/en/go/state-of-mobile-2023/ (accessed: 05.12.2023).

13. OWASP. MASVS Cheat Sheet. – URL: https://cheatsheetseries.owasp.org/IndexMASVS.html (accessed: 05.12.2023).

14. Рекомендации по безопасной разработке приложений. – URL: https://saas.stingray-mobile.ru/knowledgebase/2022.12/rg/ (дата обращения: 05.12.2023).

15. OWASP Mobile Application Security Testing Guide (MASTG). – URL: https://mas.owasp.org/MASTG/ (accessed: 05.12.2023).

16. Vetting the security of mobile applications: NIST SP 800-163 Rev. 1 / M. Ogata, J. Franklin, J. Voas, V. Sritapan, S. Quirolgico. – URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-163r1.pdf (accessed: 05.12.2023).

17. Security risks. – URL: https://developer.android.com/topic/security/risks (accessed: 05.12.2023).

18. 2022 Mobile Security Index Report. – URL: https://www.verizon.com/business/resources/reports/mobile-security-index/ (accessed: 05.12.2023).

19. Mobile application security: a systematic literature mapping / F.G. Rocha, I.M.L. do Nascimento, O.S.F. Campos, R. Santos, G.R. Colaborador // 16th CONTECSI-International Conference on Information Systems and Technology Management. – São Paulo, 2019. – DOI: 10.5748/16CONTECSI/SEC-6100.

20. Digital Security. Чек-лист по безопасной разработке мобильных приложений. – URL: https://dsec.ru/useful-materials/chek-list-po-bezopasnoj-razrabotke/ (дата обращения: 05.12.2023).

21. Кибербезопасность в 2022–2023. Тренды и прогнозы. – URL: https://www.ptsecurity.com/ru-ru/research/analytics/ogo-kakaya-ib/ (дата обращения: 05.12.2023).

22. *Никитин А.* Зачем и как решать проблемы безопасности мобильных приложений? // Информационная безопасность. – 2022. – № 3. – С. 57. – URL: https://www.itsec.ru/articles/zachem-i-kak-reshat-problemy-bezopasnosti-mobilnyh-prilozhenij (дата обращения: 05.12.2023).

23. Common Criteria for Information Technology Security Evaluation. Pt. 1. Introduction and general model: v. 3.1, rev. 5. CCMB-2017-04-001. – Common Criteria, 2017. – 106 p. – URL: https://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf (accessed: 05.12.2023).

Носенко Алексей Владиславович, магистрант направления «Информатика и вычислительная техника» факультета информационных технологий Новосибирского государственного университета. E-mail: a.nosenko1@g.nsu.ru

Пестунова Тамара Михайловна, кандидат технических наук, доцент кафедры компьютерных систем факультета информационных технологий Новосибирского государственного университета. E-mail: t.pestunova@g.nsu.ru

DOI: 10.17212/2782-2230-2023-4-47-63

Formulating typical security requirements for mobile application development*

A.V. Nosenko[1], T.M. Pestunova[2]

[1] *Novosibirsk State University, 2 Pirogov Street, Novosibirsk, 630090, Russian Federation, master's student in information technologies. E-mail: a.nosenko1@g.nsu.ru*
[2] *Novosibirsk State University, 2 Pirogov Street, Novosibirsk, 630090, Russian Federation, Associate Professor, Department of Computer Systems. E-mail: t.pestunova@g.nsu.ru*

The article examines the problem of ensuring the security of mobile applications, the development of which is often carried out without the involvement of IT security specialists and in the absence of access to a specialized software engineering infrastructure. The possible causes of insufficient security of mobile software (MS) and the methods of safe development used in practice are analyzed. Based on best practices and recommendations, a basic list of security requirements for the functions of a native mobile application, infrastructure and methods of its development has been developed, the implementation of which is available to single developers who do not have the ability to expert support the process of developing mobile software.

Keywords: mobile application, information protection, information security, safe software, security requirements, development life cycle, security testing, security threats

REFERENCE

1. Digital 2023: Global Overview Report. Available at: https://datareportal.com/reports/digital-2023-global-overview-report (accessed 05.12.2023).

2. Assessment of the security of mobile applications of Russian developers. "Stingray Technologies" study. (In Russian). Available at: https://stingray-mobile.ru/wp-content/uploads/2022/12/stingray-annual-report-2022.pdf (accessed 05.12.2023).

3. Shishkova T. *Razvitie informatsionnykh ugroz v pervom kvartale 2022 goda. Mobil'naya statistika* [Development of information threats in the first quarter of 2022. Mobile Statistics]. Available at: https://securelist.ru/it-threat-evolution-in-q1-2022-mobile-statistics/105235/ (accessed 05.12.2023).

4. Weir C., Hermann B., Fahl S. From needs to actions to secure apps? The effect of requirements and developer practices on app security. *Proceedings of the 29th USENIX Security Symposium*. USENIX Association, 2020, pp. 289–305. Available at: https://www.usenix.org/system/files/sec20-weir.pdf (accessed 05.12.2023).

* *Received 09 November 2023.*

5. Weir C., Rashid A., Noble J. How to improve the security skills of mobile app developers? Comparing and contrasting expert views. *Proceedings of the 2016 ACM Workshop on Security Information Workers*. USENIX Association, 2016. Available at: https://eprints.lancs.ac.uk/id/eprint/80016/1/SOUPS2016_SIW_AppDev_CW7June16_submitted.pdf (accessed 05.12.2023).

6. OWASP. Mobile Top 10 2023: Updates. Available at: https://owasp.org/www-project-mobile-top-10/ (accessed 05.12.2023).

7. Townsend K. *Kak smartfony stali odnoi iz glavnykh tselei kiberatak* [How smartphones have become one of the main goals of cyber-attacks]. Available at: https://blog.avast.com/ru/smartphones-and-increasing-mobile-threats-avast (accessed 05.12.2023).

8. Mikhailova A. *Mobil'nye ugrozy i metody bor'by s nimi* [Mobile threats and methods of combating them]. Available at: https://www.securitylab.ru/analytics/501302.php (accessed 05.12.2023).

9. OWASP. Mobile Application Security. Available at: https://mas.owasp.org/MASVS/ (accessed 05.12.2023).

10. Mobile Security Primer - Available at: https://books.nowsecure.com/secure-mobile-development/en/primer/mobile-security.html/ (accessed date: 10.11.2023).

11. Kaspersky IT Security Calculator. Available at: https://calculator.kaspersky.com/ru (accessed 05.12.2023).

12. State of Mobile 2023. Available at: https://www.data.ai/en/go/state-of-mobile-2023/ (accessed 05.12.2023).

13. OWASP. MASVS Cheat Sheet. Available at: https://cheatsheet-series.owasp.org/IndexMASVS.html (accessed 05.12.2023).

14. Rekomendatsii po bezopasnoi razrabotke prilozhenii [Guidelines for Secure Application Development]. Available at: https://saas.stingray-mobile.ru/knowledgebase/2022.12/rg/ (accessed 05.12.2023).

15. OWASP Mobile Application Security Testing Guide (MASTG). Available at: https://mas.owasp.org/MASTG/ (accessed 05.12.2023).

16. Ogata M., Franklin J., Voas J., Sritapan V., Quirolgico S. *Vetting the security of mobile applications.* NIST SP 800-163 Rev. 1. Available at: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-163r1.pdf (accessed 05.12.2023).

17. Security risks. Available at: https://developer.android.com/topic/security/risks (accessed 05.12.2023).

18. 2022 Mobile Security Index Report. Available at: https://www.verizon.com/business/resources/reports/mobile-security-index/ (accessed 05.12.2023).

19. Rocha F.G., Nascimento I.M.L. do, Campos O.S.F., Santos R., Colaborador G.R. Mobile application security: a systematic literature mapping. *16th CONTECSI-International Conference on Information Systems and Technology Management*, São Paulo, 2019. DOI: 10.5748/16CONTECSI/SEC-6100.

20. Digital Security. *Chek-list po bezopasnoi razrabotke mobil'nykh prilozhenii* [Checklist for the secure development of mobile applications]. Available at: https://dsec.ru/useful-materials/chek-list-po-bezopasnoj-razrabotke/ (accessed 05.12.2023).

21. Kiberbezopasnost' v 2022–2023. Trendy i prognozy [Cybersecurity in 2022–2023. Trends and forecasts]. Available at: https://www.ptsecurity.com/ru-ru/research/analytics/ogo-kakaya-ib/ (accessed 05.12.2023).

22. Nikitin A. Zachem i kak reshat' problemy bezopasnosti mobil'nykh prilozhenii? [Why and how to solve security problems of mobile applications?]. *Informatsionnaya bezopasnost' = Information Security*, 2022, no. 3, p. 57. Available at: https://www.itsec.ru/articles/zachem-i-kak-reshat-problemy-bezopasnosti-mobilnyh-prilozhenij (accessed 05.12.2023).

23. Common Criteria for Information Technology Security Evaluation. Pt. 1. Introduction and general model: v. 3.1, rev. 5. CCMB-2017-04-001. Common Criteria, 2017. 106 p. Available at: https://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf (accessed 05.12.2023).

Для цитирования:

Носенко А.В., Пестунова Т.М. Формирование типовых требований безопасности при разработке мобильного приложения // Безопасность цифровых технологий. – 2023. – № 4 (111). – С. 47–63. – DOI: 10.17212/2782-2230-2023-4-47-63.

For citation:

Nosenko A.V., Pestunova T.M. Formirovanie tipovykh trebovanii bezopasnosti pri razrabotke mobil'nogo prilozheniya [Formulating typical security requirements for mobile application development]. *Bezopasnost' tsifrovykh tekhnologii = Digital Technology Security*, 2023, no. 4 (111), pp. 47–63. DOI: 10.17212/2782-2230-2023-4-47-63.