ИССЛЕДОВАНИЕ ИНФОРМАЦИОННОЙ ЗАЩИЩЕННОСТИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

***Сафин Ленар Камилевич**, аспирант факультета компьютерных технологий и информатики Санкт-Петербургского Государственного Электротехнического Университета «ЛЭТИ» имени В.И. Ульянова (Ленина), г. Санкт-Петербург*
E- mail: safin.l91@gmail.com
***Чернов Александр Владимирович**, кандидат физико-математических наук, доцент факультет вычислительной математики и кибернетики Московского Государственного Университета имени М. В. Ломоносова, г. Москва*
E- mail: blackav@gmail.com
***Александров Ярослав Алексеевич**, аспирант факультет вычислительной математики и кибернетики Московского Государственного Университета имени М. В. Ломоносова, г. Москва*
E- mail: yaroslavalexandrov@gmail.com
***Трошина Катерина Николаевна**, кандидат физико-математических наук, Генеральный директор компании SmartDec, г. Москва*
E- mail: katerina@smartdec.ru

Необходимым этапом разработки мобильных приложений как программных компонентов информационных систем, взаимодействующих с критичными ресурсами, является исследование разработанных программных решений в контексте информационной безопасности. В статье представлен подход к исследованию информационной защищенности мобильных приложений и систематизация типовых уязвимостей приложений, выработанная в процессе анализа набора мобильных приложений, к которым предъявляются повышенные требования безопасности.

***Ключевые слова**: безопасность программ, мобильные приложения, статический анализ, динамический анализ, Android, iOS, Windows Phone, декомпиляция, обратная разработка.*

A STUDY OF MOBILE APPLICATION SECURITY

***Lenar Safin**, Computer Science and Technology department, Saint Petersburg Electrotechnical University "LETI", post-graduate student Saint Petersburg*
E- mail: safin.l91@gmail.com
***Alexander Chernov**, Ph. D.(Math.), Lomonosov Moscow State University, Computational Mathematics and Cybernetics department, Moscow*
E- mail: blackav@gmail.com
***Yaroslav Alexandrov**, Lomonosov Moscow State University, Computational Mathematics and Cybernetics department, post-graduate student, Moscow*
E- mail: yaroslavalexandrov@gmail.com
***Katerina Troshina**, Ph. D.(Math.), General Director SmartDec Company, Moscow,*
E- mail: katerina@smartdec.ru

Security analysis is the one of definitely expedient milestones in the application lifecycle. The process of security analysis should be applied to every program component in the information system, which needs to access sensitive data, particularly to mobile applications. This paper presents a practical step–by–step approach to evaluation of mobile application security and systematization of applications' typical vulnerabilities, based in the results of security analysis of a number of security–critical applications.

***Keywords**: application security, mobile applications, static analysis, dynamic analysis, Android, iOS, Windows Phone, decompilation, reverse engineering.*

Введение

Неотъемлемой частью процесса разработки информационных систем, взаимодействующих с критичными ресурсами, является этап анализа информационной защищенности таких систем. Целесообразность исследования информационных систем в контексте защищенности информационной среды обуславливается рисками частичной либо полной компрометации системы и конфиденциальных данных вследствие наличия программных уязвимостей в компонентах таких систем.

Множество компонентов информационных систем, взаимодействующих с критичными ресурсами, включает программные решения для мобильных платформ. Актуальность таких решений определяется значительным развитием мобильных экосистем и, в совокупности с востребованностью аудита информационной безопасности как этапа разработки защищенных систем обработки данных, обуславливает насущность исследования мобильных приложений в контексте информационной безопасности.

Предметом исследования данной работы является анализ информационной защищенности мобильных приложений, к которым предъявляются повышенные требования информационной безопасности. В разделе 1 описывается методика исследования программ на предмет соответствия требованиям информационной безопасности. Описываются применяемые методы анализа мобильных приложений, в зависимости от доступности исходного кода приложения на языке высокого уровня. В разделе 2 систематизированы типовые уязвимости мобильных приложений и связанные с такими уязвимостями риски. Полученные результаты основаны на результатах анализа исследуемой выборки мобильных приложений, а также существующих исследований типовых уязвимостей компонентов информационных систем, в частности [1], [2], [3].

1. Методика исследования программ на предмет соответствия требованиям информационной безопасности

В процессе оценки защищенности мобильных приложений были исследованы приложения, к которым предъявляются повышенные требования безопасности. В частности, были исследованы мобильные приложения банковских организаций РФ и стран ближнего зарубежья, операторов мобильной связи, электронных платежных систем и электронных денежных систем. Общее число исследованных приложений составило 52 мобильных приложений, находящихся в открытом доступе, исполняющихся под управлением операционных систем Android, iOS и Windows Phone.

Таблица 1. *Распределение исследуемых мобильных приложений*

Мобильная операционная система	Количество исследуемых приложений
Android	33
iOS	13
Windows Phone	6

Процесс анализа мобильного приложения включал в себя анализ программного кода приложения. Программный код мобильного приложения был исследован как с применением инструментальных средств статического анализа, так и вручную. При этом методика анализа программного кода существенно зависит от доступности исходного кода мобильного приложения или его компонентов. В рамках данной работы различались два подхода к исследованию программного кода.

Анализ программного кода с доступным исходным кодом мобильного приложения может быть осуществлен с использованием инструментальных средств статического анализа. Метод анализа программного кода с доступным исходным кодом применялся при анализе приложений для мобильных платформ Android, iOS и Windows Phone. Исследование программного кода таких мобильных приложений осуществлялось следующим образом:

- Анализ исходного кода на наличие стороннего программного кода из открытых источников. При наличии стороннего программного кода, такой код был исследован на наличие известных уязвимостей в открытых источниках информации.
- Анализ исходного кода мобильного приложения инструментальными средствами статического анализа. Результаты работы средств статического анализа программного кода были исследованы для определения ошибок первого рода.
- Анализ исходного кода приложения вручную.

Анализ программного кода с частично либо полностью недоступным исходным кодом мобильного приложения был осуществлен с использованием методов обратной разработки программного кода. Метод анализа программного кода с частично либо полностью недоступным исходным

кодом применялся при анализе приложений для мобильных платформ Android и iOS. Исследование программного кода таких мобильных приложений осуществлялось следующим образом:

- Анализ программного кода на наличие стороннего программного кода из открытых источников. В контексте исследования программных компонентов, исходный код которых недоступен, наличие стороннего программного кода определялось на основании символьной информации в двоичном образе таких компонентов. При наличии стороннего программного кода, такой код был исследован на наличие известных уязвимостей в открытых источниках информации.
- Применение методов обратной разработки к программному коду мобильного приложения. В процессе исследования программного кода применялись как инструментальные средства дизассемблирования программного кода, так и средства восстановления исходного кода с использованием методов декомпиляции.
- В случае, если были применены методы восстановления исходного кода, производился анализ такого кода инструментальными средствами статического анализа. Результаты работы средств статического анализа программного кода были исследованы для определения ошибок первого рода.
- Анализ дизассемблированного кода вручную.

Анализ работы мобильных приложений был осуществлен с использованием средств динамического анализа программ и средств отладки. В процессе анализа были использованы среды выполнения на физических мобильных устройствах и средствах программной эмуляции мобильных устройств. При этом был произведен анализ

- потоков управления и данных мобильного приложения;
- межсетевых взаимодействий, осуществляемых мобильным приложением;
- вызовов системных функций, в частности процедур чтения и записи на устройства вспомогательной (энергонезависимой) памяти и записей в системный журнал событий;
- использования сервисов, предоставляемых операционной системой, например сообщений SMS/USSD, системного журнала событий, модуля NFC и других;
- межпроцессных взаимодействий.

В некоторых случаях, например при анализе межпроцессных взаимодействий, осуществлялось тестирование поведения мобильного приложения при передаче приложению случайных данных (Fuzz Testing).

2. Результаты исследования программ на предмет соответствия требованиям информационной безопасности

В данном разделе систематизирована информация о результатах исследования мобильных приложений на предмет соответствия требованиям информационной безопасности. В разделе рассматриваются типовые уязвимости мобильных приложений, присущие мобильным приложениям каждой из исследованных мобильных платформ.

2.1. Систематизация типовых уязвимостей мобильных приложений

В данном подразделе рассматриваются типовые уязвимости мобильных приложений, обнаруженные в процессе исследования информационной защищенности мобильных приложений, к которым предъявляются повышенные требования безопасности.

2.1.1. Использование незащищенных протоколов передачи информации

В контексте сервисов, к которым предъявляются повышенные требования безопасности, необходимо использовать защищенные протоколы передачи данных для осуществления межсетевых запросов. В противном случае, злоумышленник может осуществить атаку на канал связи типа «человек посередине», что может привести к утрате конфиденциальности передаваемых данных или к другим атакам на мобильное приложение.

В результате анализа исследуемой выборки мобильных приложений были получены следующие результаты. 13 приложений осуществляли запросы к веб-серверу по незащищенному протоколу передачи данных HTTP. При этом по незащищенному соединению передавались конфиденциальные данные пользователя, в частности аутентификационные данные пользователя.

2.1.2. Небезопасная конфигурация защищенного соединения

В процессе анализа межсетевых взаимодействий мобильных приложений была произведена оценка настроек защищенного соединения, используемых веб-серверами, в частности, настроек пакетов реализации протокола SSL. В случае, если настройка пакета реализации защищенного протокола произведена некорректно, злоумышленник может осуществить атаку на защищенное

соединение, что может привести к полной утрате конфиденциальности передаваемых данных.

К потенциальным уязвимостям защищенных соединений следует отнести:

- Использование устаревших протоколов. В частности, протокол SSLv2 является устаревшим и обладает рядом известных уязвимостей [4].
- Использование уязвимых протоколов обмена криптографическими ключами, слабых алгоритмов проверки целостности сообщений, уязвимых алгоритмов шифрования или алгоритмов шифрования с малой эффективной стойкостью ключа шифрования.
- Использование небезопасного протокола повторного рукопожатия (Insecure renegotiation) [5].
- Использование алгоритмов сжатия [6].
- Известные уязвимости пакетов реализации протокола SSL, например [7], [8] и [9].

В результате анализа исследуемой выборки мобильных приложений были получены следующие результаты. 24 приложения взаимодействовали с веб-сервером, использующим небезопасные настройки защищенного соединения. В частности, в 10 случаях было установлено, что веб-сервер использует пакет реализации OpenSSL [10], уязвимый к атакам типа «человек посередине» [9], что может привести к утрате конфиденциальности передаваемых данных.

2.1.3. Небезопасная аутентификация веб-сервера

Интерфейсы классов, используемые мобильными приложениями для соединения с веб-сервером, предоставляют методы, позволяющие изменить процесс проверки сертификата X.509 веб-сервера во время инициализации защищенного соединения. В частности, в 14 случаях мобильные приложения частично либо полностью отключали механизм аутентификации веб-сервера. В случае, если мобильное приложение не осуществляет проверку сертификата X.509 веб-сервера, злоумышленник может осуществить атаку типа «человек посередине», что может привести к полной утрате конфиденциальности передаваемых данных.

В процессе исследования мобильных приложений на предмет соответствия требованиям информационной безопасности было отмечено, что значительная часть приложений использует политику проверки сертификата X.509 по умолчанию. Количество таких приложений составило 35 мобильных приложений. В таком случае при проверке сертификата X.509 мобильным приложением используется набор доверенных (корневых) сертификатов, установленных в системе. Данный набор сертификатов X.509 может быть изменен как пользователем мобильного устройства, так и злоумышленником, при наличии несанкционированного привилегированного доступа к устройству. В частности, мобильное приложение может быть подвержено атакам типа «человек посередине».

Для предотвращения атак такого типа, мобильное приложение должно использовать технику привязки сертификата X.509 веб-сервера (Certificate Pinning). Количество приложений, осуществляющих привязку сертификата составило 3 мобильных приложения. В реализации техники привязки сертификата X.509 одного из таких приложений была допущена программная ошибка, что привело к ослаблению правил проверки сертификата X.509 по умолчанию, вследствие чего мобильное приложение было уязвимо к атакам типа «человек посередине».

2.1.4. Использование небезопасных криптографических методов

В случае, если мобильным приложением применяются криптографические методы, такие как шифрование или криптографические функции свертки, мобильными приложениями могут применяться методы, обладающие известными уязвимостями. К таким методам следует отнести:

- Слабые и устаревшие алгоритмы симметричного шифрования, такие как DES, Triple DES с двумя ключами шифрования [11], [12].
- Алгоритмы шифрования с малой эффективной стойкостью ключа шифрования.
- Алгоритмы формирования ключа шифрования на основании парольной фразы с малым количеством итераций.
- Слабые и уязвимые режимы алгоритмов шифрования, например ECB или OFB [11].
- Слабые и устаревшие криптографические функции свертки, такие как MD4, MD5 [13], [14] и SHA-1 [12], [15].

В случае, если мобильным приложением используются уязвимые криптографические методы или при компрометации ключей шифрования (включая синхропосылку, при наличии), злоумышленник может восстановить исходные данные, что может привести к компрометации конфиденциальных данных пользователя.

Одной из распространенных ошибок реализации механизмов шифрования является использование режимов шифрования по умолчанию. Так, например, в реализации алгоритма симметрично-

го блочного шифрования AES в пакете расширений стандартной библиотеки языка Java в классе javax.crypto.Cipher по умолчанию используется режим шифрования ECB и метод выравнивания PKCS #5 [16].

В процессе анализа исследуемой выборки мобильных приложений было обнаружено, что 23 мобильных приложений используют небезопасные криптографические методы шифрования и/или функции свертки. Кроме того, в 11 исследуемых приложениях использовались методы шифрования с постоянным ключом шифрования или доступным при анализе данных приложения. В одном из исследуемых мобильных приложений использовался нестандартный протокол обмена криптографическими ключами, уязвимый к атакам типа «человек посередине» и «повторное воспроизведение запроса».

2.1.5. Небезопасное хранение конфиденциальных данных

Мобильные операционные системы предоставляют системные интерфейсы для взаимодействия мобильных приложений с файловой системой. В случае если политики безопасности мобильной операционной системы не были подвержены изменениям, мобильные приложения имеют разрешения на чтение и запись данных в домашней директории приложения [17], [18], [19] и, в отдельных случаях, к внешнему устройству вспомогательной памяти.

Тем не менее, в том случае если к данным приложения не применяются алгоритмы шифрования, доступ к данным мобильных приложений может быть получен злоумышленником при наличии у него доступа к файловой системе устройства. Например, доступ к файловой системе может быть получен вредоносными приложениями на устройствах с измененными политиками безопасности операционной системы, в результате атаки типа «обход файлового пути», а также при наличии у злоумышленника физического доступа к устройству. В случае, если мобильное устройство хранит конфиденциальные данные пользователя в домашней директории приложения или на внешнем устройстве вспомогательной памяти, возможна компрометация таких данных.

В результате анализа исследуемой выборки мобильных приложений были получены следующие результаты. 23 приложения хранили конфиденциальные данные пользователя в домашней директории приложения. При этом такими приложениями сохранялись персональные и аутентификационные данные пользователя, а также данные банковских карт. Мобильных приложений, хранящих конфиденциальные данные пользователя на внешних устройствах вспомогательной памяти, обнаружено не было.

Кроме того, злоумышленник может использовать возможность изменить данные приложения, существенно влияющих на поток управления мобильного приложения, в том числе приводящих к ослаблению правил безопасности приложения. Например, в 9 мобильных приложения такая возможность могла быть использована злоумышленником для подмены адреса веб-сервиса. В 7 случаях такая возможность могла быть использована злоумышленником для обхода экрана аутентификации пользователя при наличии физического доступа к мобильному устройству.

2.1.6. Непреднамеренная компрометация данных

Компоненты мобильного приложения могут осуществлять действия, которые могут привести к компрометации конфиденциальной информации, прозрачно для разработчика приложения. К таким действиям следует отнести, в частности:

- хранение базы данных подсистемы кэширования запросов и ответов веб-сервера, а также баз данных HTTP Cookie и Web Storage компонентой веб-обозревателя в домашней директории приложения;
- хранение базы данных подсистемы кэширования вводимой пользователем информации;
- хранение конфиденциальной информации в системном буфере обмена данными.

Так, в результате анализа исследуемой выборки мобильных приложений было обнаружено 18 приложений, сохраняющих конфиденциальные данные в базах данных компоненты веб-обозревателя. При наличии у злоумышленника доступа к файловой системе пользовательского устройства, возможна компрометация конфиденциальных данных пользователя.

2.1.7. Вывод конфиденциальной информации в системный журнал событий

Мобильные операционные системы предоставляют системные интерфейсы взаимодействия с системным журналом событий мобильного устройства. В процессе разработки мобильных приложений такие интерфейсы могут быть использованы, например, для вывода отладочной информации и при обработке исключительных ситуаций.

В результате анализа исследуемой выборки мобильных приложений были получены следующие результаты. 13 приложений выводили конфиденциальную информацию пользователя

в системный журнал событий. При этом в 11 случаях, приложениями выводились данные, полученные в процессе межсетевых взаимодействий с веб-сервисами, такие как HTTP-запросы и ответы веб-сервера. Такие данные, в частности, включали конфиденциальную информацию пользователя, включая аутентификационные данные пользователя и данные банковский карт.

2.1.8. Небезопасное использование возможностей межпроцессного взаимодействия

Мобильные приложения получают возможность взаимодействовать в контексте исполнения под управлением операционной системы через интерфейсы межпроцессного взаимодействия. В данном пункте рассматриваются такие возможности мобильных операционных систем, а также типовые атаки на приложения с использованием механизмов межпроцессного взаимодействия.

Мобильная платформа Android предоставляет возможность процессам отправлять и получать объекты типа android.content.Intent [20], используемые в контексте взаимодействия компонентов приложений. В случае, если адресат таких сообщений указывается неявно или используется механизм широковещательной рассылки сообщений, данные таких сообщений могут быть скомпрометированы. Кроме того, вредоносными приложениями могут использоваться механизмы делегирования управления процесса, такие как неявные вызовы компонентов приложений или, например, объекты типа android.app.PendingIntent [21], для перехвата потока управления приложения или атак типа «Phishing Attack».

Мобильные платформы Android, iOS и Windows Phone предоставляют возможность взаимодействия компонентов приложений через обработчики пользовательских протоколов URI. Злоумышленник может использовать механизм взаимодействия компонентов через обработчики пользовательских протоколов для осуществления атак с целью мошенничества.

Мобильные приложения для операционной системы Android могут предоставлять интерфейсы для доступа к данным приложения через компоненты приложения типа Content Provider. В случае, если такие компоненты не защищены правами доступа, данные приложения могут быть скомпрометированы.

В результате анализа исследуемой выборки мобильных приложений были получены следующие результаты:

- 3 приложения передавали конфиденциальные данные в сообщениях, включая аутентификационные данные пользователя, которые могли быть скомпрометированы вредоносными приложениями;
- в 7 приложениях были обнаружены уязвимости, позволяющие злоумышленнику осуществить атаку типа «Phishing Attack»;
- 2 приложения не защищали правами доступа компоненты типа Content Provider, содержащие конфиденциальные данные пользователя, включая аутентификационные данные пользователя.

Суммарно, в результате анализа исследуемой выборки мобильных приложений было обнаружено 10 уязвимостей приложений в контексте межпроцессных взаимодействий.

2.1.9. Небезопасная обработка входных данных

Мобильные приложения, не проверяющие или некорректно обрабатывающие входные данные могут быть подвержены непредусмотренному изменению поведения приложения, например компрометации конфиденциальных данных или изменению потока управления программы, а также атакам типа «отказ в обслуживании». К входным данным, используемым мобильными приложениями, следует отнести данные, полученные в процессе межсетевого взаимодействия, в процессе межпроцессного взаимодействия, при взаимодействии приложения с файловой системой мобильного устройства или пользовательскими данными.

Некорректная обработка входных данных приложением может привести к непредусмотренному изменению поведения приложения. В зависимости от контекста, в котором используются такие входные данные, техники, используемые для атаки на приложение, и связанные с такой атакой риски могут значительно отличаться. К типовым классам атак следует отнести такие атаки как «внедрение SQL запросов», «обход файлового пути», ошибки переполнения буфера.

В результате анализа исследуемой выборки мобильных приложений были получены следующие результаты. 18 приложений некорректно обрабатывали входные данные и были подвержены атакам, приводящим к непредусмотренному поведению приложения. При этом в 6 случаях недостаточная проверка входных данных могла быть использована злоумышленником для атак типа «отказ в обслуживании».

2.1.10. Небезопасное использование возможностей сети GSM

Мобильные приложения могут использовать возможности сетей GSM для осуществления операций в контексте сервиса. В частности, в контек-

сте сервиса могут использоваться возможности протоколов обмена сообщениями SMS/USSD, реализованных в стандарте мобильной сотовой связи GSM. К потенциально небезопасным операциям, осуществляемым по протоколам обмена сообщениями в сетях GSM, следует отнести взаимодействие с критичными ресурсами и передачу конфиденциальных данных.

Типовые атаки на сервисы, использующие протоколы SMS/USSD сетей GSM для осуществления потенциально небезопасных операций, включают в себя:

- атаки на SIM-карты [22], [23];
- атаки на алгоритмы шифрования данных в сетях GSM [24], [25], [26] и [27], включая атаки на алгоритм A5/3 [28], [29];
- атаки на протокол аутентификации [30].

Злоумышленник, реализующий одну из атак из описанных атак, может перехватывать и/или произвольным образом видоизменять данные, передаваемые в сообщениях SMS/USSD.

Кроме того, мобильные платформы могут предоставлять системные интерфейсы взаимодействия с подсистемой обмена сообщениями SMS/USSD. Например, мобильные приложения, исполняющиеся под управлением операционной системы Android, могут запрашивать разрешения на чтение, получение и отправку сообщений SMS. Кроме того, доступ к подсистеме обмена сообщениями SMS может быть получен приложением при наличии несанкционированного привилегированного доступа к мобильному устройству.

В результате анализа исследуемой выборки мобильных приложений были получены следующие результаты. В 16 приложениях использовались возможности протоколов SMS/USSD для взаимодействия с критичными ресурсами или при передаче конфиденциальных данных. В частности, в исследуемой выборке мобильных приложений, 12 приложений использовали сообщения SMS передачи конфиденциальных данных, таких как аутентификационные данные пользователя и данные банковских карт. В 7 приложениях функциональность протоколов обмена сообщениями SMS/USSD использовалась для осуществления банковских операций.

2.1.11. Отсутствие проверок наличия несанкционированного привилегированного доступа к мобильному устройству

Мобильные операционные системы могут быть подвергнуты изменениям политик безопасности по умолчанию. В случае, если мобильные приложения, исполняющиеся под управлением таких систем, взаимодействуют с критичными ресурсами и конфиденциальными данными, вредоносные приложения, при наличии несанкционированного привилегированного доступа к мобильному устройству, могут получить доступ к таким ресурсам. Например, вредоносное приложение, исполняющееся с привилегиями операционной системы, может осуществлять перехват системных вызовов или методов мобильного приложения, получать доступ к адресному пространству процесса мобильного приложения или доступ к другим данным приложения.

В контексте информационной защищенности мобильных приложений, к которым предъявляются повышенные требования безопасности, представляется целесообразным использовать алгоритмы обнаружения несанкционированного привилегированного доступа к мобильному устройству для ограничения возможностей взаимодействия приложения с критичными ресурсами и конфиденциальными данными пользователя. При осуществлении проверки целостности модели безопасности исходной мобильной платформы необходимо опираться на эвристические критерии целостности модели безопасности, по возможности уникальные для мобильного приложения, что позволит свести к минимуму риски, связанные с атаками на мобильное приложение при наличии несанкционированного привилегированного доступа к мобильному устройству.

В процессе анализа исследуемой выборки мобильных приложений было обнаружено 5 приложений, осуществляющих проверку наличия несанкционированного привилегированного доступа к мобильному устройству.

2.1.12. Недостаточная защита пакета приложения и его компонентов

К потенциальным уязвимостям мобильного приложения, связанных с недостаточной защитой пакета приложения и его компонентов, следует отнести:

- недостаточное покрытие программного кода маскирующими преобразованиями;
- отсутствие проверки целостности пакета приложения и его компонентов;
- отсутствие механизма обнаружения отладчика программного кода;
- наличие отладочной и символьной информации о программном коде в пакете приложения.

В процессе анализа исследуемой выборки мобильных приложений было обнаружено 8 приложений, к программному коду которых были приме-

нены маскирующие преобразования. При этом, для запутывания программного кода не были применены методы шифрования к строковым литералам, а изменения, внесенные в поток управления и поток данных были отмечены как несущественные.

Мобильных приложений, полностью удовлетворяющих указанным требованиям по обеспечению информационной защищенности пакета приложения и его компонентов, в исследуемой выборке обнаружено не было.

2.1.13. Небезопасная конфигурация и использование потенциально уязвимых методов в контексте обработки управляемых ресурсов

Средства разработки программных компонентов для мобильных платформ включают механизмы защиты, увеличивающих сложность эксплуатации уязвимостей мобильных приложений, связанных с ошибками обработки управляемых ресурсов. В данном пункте рассматриваются такие механизмы защиты и их применения в программных компонентах.

К основным механизмом защиты мобильных приложений следует отнести:

- Трансляция исходного кода программных компонентов в двоичный код, не зависящий от расположения модуля в адресном пространстве. Сегменты двоичного образа такой программы могут быть отображены на произвольные смещения адресного простанства процесса программы (Address Space Layout Randomization, ASLR).
- Использование средств защиты программы от ошибок переполнения буфера. Трансляторы исходного кода могут добавлять дополнительную область в стековый фрейм подпрограммы, значение которой устанавливается равным предустановленному значению при вызове подпрограммы и проверяется при выходе из нее. Реализация такой техники существенно зависит от используемого транслятора исходного кода. Частный случай реализации техники в семействе компиляторов GCC описан в [31].
- Использование механизмов автоматического управления памятью динамических структур данных, а также механизмов контроля жизни таких структур (RAII, подсчет ссылок на структуру).

В процессе анализа исследуемой выборки мобильных приложений было обнаружено 8 приложений, в которых не были использованы указанные механизмы защиты, увеличивающих сложность эксплуатации уязвимостей мобильных приложений, связанных с ошибками обработки управляемых ресурсов.

2.2. Результаты исследования типовых уязвимостей мобильных приложений

В данном подразделе приводятся результаты исследования типовых уязвимостей мобильных приложений, обнаруженных в контексте исследования набора мобильных приложений, к которым предъявляются повышенные требования информационной безопасности.

В таблице 2 рассматривается соответствие между рассматриваемыми в работе классами типовых уязвимостей мобильных приложений и количеством мобильных приложений, которым присущи такие уязвимости.

***Таблица 2**. Распределение классов уязвимостей мобильных приложений*

Класс уязвимости	Количество уязвимых приложений
Использование незащищенных протоколов передачи информации	13
Небезопасная конфигурация защищенного соединения	24
Небезопасная аутентификация веб-сервера	15
Использование небезопасных криптографических методов	23
Небезопасное хранение конфиденциальных данных	23
Непреднамеренная компрометация данных	18
Вывод конфиденциальной информации в системный журнал событий	13
Небезопасное использование возможностей межпроцессного взаимодействия	10
Небезопасная обработка входных данных	18
Небезопасное использование возможностей сети GSM	16
Отсутствие проверок наличия несанкционированного привилегированного доступа к мобильному устройству	47
Недостаточная защита пакета приложения и его компонентов	52
Небезопасная конфигурация и использование потенциально уязвимых методов в контексте обработки управляемыми ресурсов	8

Следует отметить, что значительная часть указанных уязвимостей была обнаружена в процессе

статического анализа программного кода мобильных приложений. В частности, методами статического анализа были обнаружены программные уязвимости мобильных приложений, связанные с небезопасной аутентификацией веб-сервера, использованием небезопасных криптографических методов, небезопасным хранением конфиденциальных данных, выводом конфиденциальной информации в системный журнал событий и другие. В контексте разработки информационных систем, к которым предъявляются повышенные требования безопасности, такие уязвимости следует рассматривать как уязвимости критичной степени важности. Прикладным фактором, обусловившим целесообразность статического анализа программного кода в процессе исследования информационной защищенности, является снижение затрат ресурсов на такое исследование при возросшем качестве результатов.

Заключение

В работе рассмотрена методика анализа приложений для мобильных платформ на предмет соответствия требованиям информационной безопасности. Опубликованы и систематизированы результаты исследования набора мобильных приложений, к которым предъявляются повышенные требования информационной защищенности. Результаты исследования описывают классы типовых уязвимостей мобильных приложений, а также потенциальные риски, связанные с такими уязвимостями.

Литература:

1. OWASP Mobile Security Project – OWASP [Электронный ресурс]. URL: https://www.owasp.org/index.php/OWASP_Mobile_Security_Project (дата обращения: 29.07.2015).
2. Wichers D. OWASP Top-10 2013 //OWASP Foundation, February. – 2013.
3. The Web Application Security Consortium / Thread Classification [Электронный ресурс]. URL: http://projects.webappsec.org/w/page/13246978/Threat%20Classification (дата обращения: 29.07.2015).
4. Turner S., Polk T. Prohibiting secure sockets layer (SSL) version 2.0. – 2011.
5. CVE-2009-3555 [Электронный ресурс]. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3555 (дата обращения: 29.07.2015).
6. CVE-2012-4929 [Электронный ресурс]. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-4929 (дата обращения: 29.07.2015).
7. CVE-2010-3972 [Электронный ресурс]. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-3972 (дата обращения: 29.07.2015).
8. CVE-2014-0160 [Электронный ресурс]. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160 (дата обращения: 29.07.2015).
9. CVE-2014-0224 [Электронный ресурс]. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0224 (дата обращения: 29.07.2015).
10. OpenSSL: The Open Source toolkit for SSL/TLS [Электронный ресурс]. URL: https://www.openssl.org/ (дата обращения: 29.07.2015).
11. Schneier B. Applied cryptography: protocols, algorithms, and source code in C. – john wiley & sons, 2007.
12. Barker E., Roginsky A. NIST Special Publication 800-131A Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths. – 2011.
13. Wang X., Yu H. How to break MD5 and other hash functions //Advances in Cryptology–EUROCRYPT 2005. – Springer Berlin Heidelberg, 2005. – С. 19-35.
14. Turner S., Chen L. Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms. – 2011.
15. Wang X., Yin Y. L., Yu H. Finding collisions in the full SHA-1 //Advances in Cryptology–CRYPTO 2005. – Springer Berlin Heidelberg, 2005. – С. 17-36.
16. Java Cryptography Architecture (JCA) Reference Guide [Электронный ресурс]. URL: http://docs.oracle.com/javase/8/docs/technotes/guides/security/crypto/CryptoSpec.html (дата обращения: 29.07.2015).
17. App Sandbox Design Guide: About App Sandbox [Электронный ресурс]. URL: https://developer.apple.com/library/mac/documentation/Security/Conceptual/AppSandboxDesignGuide/AboutAppSandbox/AboutAppSandbox.html (дата обращения: 29.07.2015).
18. Blazakis D. The apple sandbox //Arlington, VA, January. – 2011.
19. Android Security Overview | Android Developers [Электронный ресурс]. URL: https://source.android.com/devices/tech/security/ (дата обращения: 29.07.2015).
20. Intent | Android Developers [Электронный ресурс]. URL: https://developer.android.com/reference/android/content/Intent.html (дата обращения: 29.07.2015).
21. PendingIntent | Android Developers [Электронный ресурс]. URL: https://developer.android.com/reference/android/app/PendingIntent.html (дата обращения: 29.07.2015).
22. Nohl K. Rooting SIM cards //Black Hat. – 2013.
23. Wagner D., Goldberg I., Briceno M. GSM cloning //Web page about COMP-128 version. – 1998. – Т. 1.
24. Barkan E., Biham E., Keller N. Instant ciphertext-only cryptanalysis of GSM encrypted communication //Journal of Cryptology. – 2008. – Т. 21. – №. 3. – С. 392-429.
25. Golić J. D. Cryptanalysis of alleged A5 stream cipher //Advances in Cryptology—EUROCRYPT'97. – Springer Berlin Heidelberg, 1997. – С. 239-255.

26. Ekdahl P., Johansson T. Another attack on A5/1 //Information Theory, IEEE Transactions on. – 2003. – Т. 49. – №. 1. – С. 284-289.
27. Biryukov A., Shamir A., Wagner D. Real Time Cryptanalysis of A5/1 on a PC //Fast Software Encryption. – Springer Berlin Heidelberg, 2001. – С. 1-18.
28. Dunkelman O., Keller N., Shamir A. A Practical-Time Attack on the A5/3 Cryptosystem Used in Third Generation GSM Telephony // IACR Cryptology ePrint Archive. – 2010. – Т. 2010. – С. 13.
29. Paget C., Nohl K. GSM: SRSLY //26th Chaos Communication Congress. – 2009.
30. Meyer U., Wetzel S. On the impact of GSM encryption and man-in-the-middle attacks on the security of interoperating GSM/UMTS networks //Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on. – IEEE, 2004. – Т. 4. – С. 2876-2883.
31. Wagle P., Cowan C. Stackguard: Simple stack smash protection for gcc //Proceedings of the GCC Developers Summit. – 2003. – С. 243-255.

References:

1. OWASP Mobile Security Project – OWASP [Electronic resource]. URL: https://www.owasp.org/index.php/OWASP_Mobile_Security_Project (date accessed: 07/29/2015).
2. Wichers D. OWASP Top-10 2013 //OWASP Foundation, February. – 2013.
3. The Web Application Security Consortium / Thread Classification [Electronic resource]. URL: http://projects.webappsec.org/w/page/13246978/Threat%20Classification (date accessed: 07/29/2015).
4. Turner S., Polk T. Prohibiting secure sockets layer (SSL) version 2.0. – 2011.
5. CVE-2009-3555 [Electronic resource]. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3555 (date accessed: 07/29/2015).
6. CVE-2012-4929 [Electronic resource]. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-4929 (date accessed: 07/29/2015).
7. CVE-2010-3972 [Electronic resource]. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-3972 (date accessed: 07/29/2015).
8. CVE-2014-0160 [Electronic resource]. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160 (date accessed: 07/29/2015).
9. CVE-2014-0224 [Electronic resource]. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0224 (date accessed: 07/29/2015).
10. OpenSSL: The Open Source toolkit for SSL/TLS [Electronic resource]. URL: https://www.openssl.org/ (date accessed: 07/29/2015).
11. Schneier B. Applied cryptography: protocols, algorithms, and source code in C. – john wiley & sons, 2007.
12. Barker E., Roginsky A. NIST Special Publication 800-131A Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths. – 2011.
13. Wang X., Yu H. How to break MD5 and other hash functions //Advances in Cryptology–EUROCRYPT 2005. – Springer Berlin Heidelberg, 2005. – С. 19-35.
14. Turner S., Chen L. Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms. – 2011.
15. Wang X., Yin Y. L., Yu H. Finding collisions in the full SHA-1 //Advances in Cryptology–CRYPTO 2005. – Springer Berlin Heidelberg, 2005. – С. 17-36.
16. Java Cryptography Architecture (JCA) Reference Guide [Electronic resource]. URL: http://docs.oracle.com/javase/8/docs/technotes/guides/security/crypto/CryptoSpec.html (date accessed: 07/29/2015).
17. App Sandbox Design Guide: About App Sandbox [Electronic resource]. URL: https://developer.apple.com/library/mac/documentation/Security/Conceptual/AppSandboxDesignGuide/AboutAppSandbox/AboutAppSandbox.html (date accessed: 07/29/2015).
18. Blazakis D. The apple sandbox //Arlington, VA, January. – 2011.
19. Android Security Overview | Android Developers [Electronic resource]. URL: https://source.android.com/devices/tech/security/ (date accessed: 07/29/2015).
20. Intent | Android Developers [Electronic resource]. URL: https://developer.android.com/reference/android/content/Intent.html (date accessed: 07/29/2015).
21. PendingIntent | Android Developers [Electronic resource]. URL: https://developer.android.com/reference/android/app/PendingIntent.html (date accessed: 07/29/2015).
22. Nohl K. Rooting SIM cards //Black Hat. – 2013.
23. Wagner D., Goldberg I., Briceno M. GSM cloning //Web page about COMP-128 version. – 1998. – Т. 1.
24. Barkan E., Biham E., Keller N. Instant ciphertext-only cryptanalysis of GSM encrypted communication //Journal of Cryptology. – 2008. – Т. 21. – №. 3. – С. 392-429.
25. Golić J. D. Cryptanalysis of alleged A5 stream cipher //Advances in Cryptology—EUROCRYPT'97. – Springer Berlin Heidelberg, 1997. – С. 239-255.
26. Ekdahl P., Johansson T. Another attack on A5/1 //Information Theory, IEEE Transactions on. – 2003. – Т. 49. – №. 1. – С. 284-289.
27. Biryukov A., Shamir A., Wagner D. Real Time Cryptanalysis of A5/1 on a PC //Fast Software Encryption. – Springer Berlin Heidelberg, 2001. – С. 1-18.
28. Dunkelman O., Keller N., Shamir A. A Practical-Time Attack on the A5/3 Cryptosystem Used in Third Generation GSM Telephony // IACR Cryptology ePrint Archive. – 2010. – Т. 2010. – С. 13.
29. Paget C., Nohl K. GSM: SRSLY //26th Chaos Communication Congress. – 2009.
30. Meyer U., Wetzel S. On the impact of GSM encryption and man-in-the-middle attacks on the security of interoperating GSM/UMTS networks //Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on. – IEEE, 2004. – Т. 4. – С. 2876-2883.
31. Wagle P., Cowan C. Stackguard: Simple stack smash protection for gcc //Proceedings of the GCC Developers Summit. – 2003. – С. 243-255.