УДК 004.056

Исследование технологий создания, управления и защиты информации мобильных приложений кредитно-финансовых организаций

Исследованы технологии создания, управления и защиты информации мобильных приложений кредитно-финансовых организаций (КФО). Для изучения качества взаимодействия мобильной операционной системы Android с APK-сборками (.apk) КФО, в том числе оценки технологического уровня и степени безопасности, проведено их исследование на предмет возможности получения информации о структуре построения, алгоритмах функционирования, технологиях защиты информации, незадекларированных функциях.

Ключевые слова:
потенциально вредоносное приложение, реверсный инжиниринг программного обеспечения, обфускация кода, Android.

В.В. МАЛИКОВ,
канд. техн. наук, доцент, начальник цикла технических и специальных дисциплин УО «Центр повышения квалификации руководящих работников и специалистов» Департамента охраны МВД Республики Беларусь

А.Н. КОВАЛЕНКО,
заместитель начальника кафедры специальных и инженерно-технических дисциплин – начальник службы (технических средств охраны) УО «Военная академия Республики Беларусь»

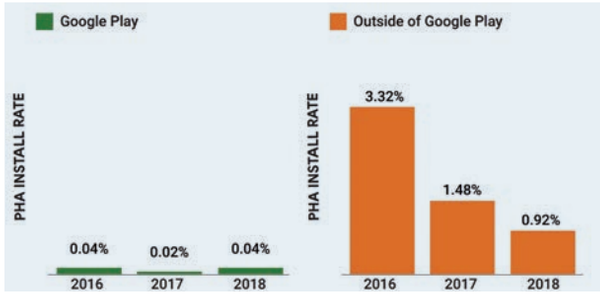
Введение. В настоящее время в мире и Республике Беларусь наиболее перспективной для КФО является деятельность, основанная на дистанционном оказании услуг на основе сетевых ресурсов, размещенных в глобальной сети интернет [1]. При этом конечные пользователи таких услуг для проведения финансово-экономических операций часто используют мобильные устройства с установленным специализированным программным обеспечением, имеющим, как правило, различные недостатки/уязвимости как в мобильных операционных системах, так и специализированных приложениях КФО. Указанные недостатки/уязвимости в программном обеспечении активно используются киберпреступниками для проведения удаленных атак на пользователей услуг КФО и совершения других противоправных действий.

Приведенные в рамках данной статьи материалы носят исключительно научно-исследовательский характер. Исследование проводилось авторами строго в научных целях, его результаты не являются и не могут признаваться руководством к совершению каких-либо противоправных действий (fair use/fair dealing). Информация, содержащаяся в статье, получена из источников, рассматриваемых авторами как надежные. При проведении исследования авторы действовали в рамках законодательства Республики Беларусь. Авторы не несут ответственности за инциденты в сфере информационной безопасности, имеющие отношение к тематике исследования.

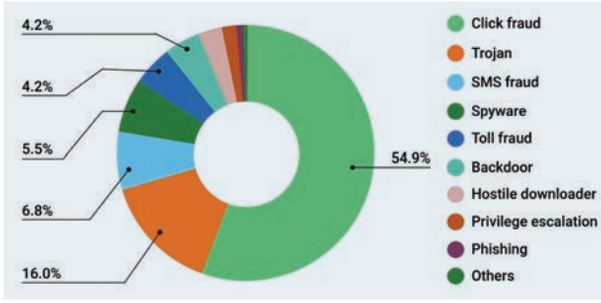
Содержательная постановка задачи. Наиболее распространенной мобильной операционной системой в мире и Республике Беларусь в настоящее время является Android [2]. Согласно отчетным данным по работе защитного алгоритма Google Play Protect, встроенного в приложение Google Play Store на всех авторизованных устройствах на Android и проверяющего устройство при установке/обновлении приложений из каталога Google Play или стороннего источника на признаки потенциально вредоносного приложения (Potentially Harmful Applications, PHA), наблюдается значительное число таких вредоносных приложений [3] (рисунок 1).

Для изучения качества взаимодействия мобильной операционной системы Android с APK-сборками (.apk) КФО [4], в том числе оценки технологического уровня и степени безопасности, проведем их исследование на предмет возможности получения информации о структуре построения, алгоритмам функционирования, технологиям защиты информации, незадекларированным функциям. Информация, необходимая для оценки возможности взаимодействия, технологического уровня и степени безопасности ранее не была доступна из других источников.

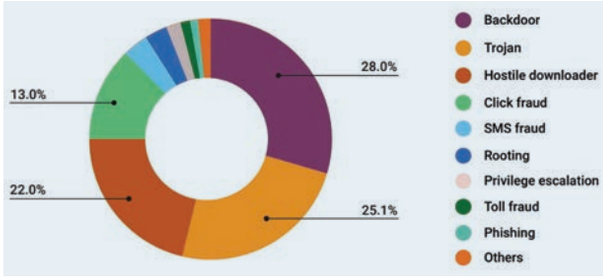
Результаты и обсуждение. Типовой архивный исполняемый файл-приложение APK (.apk) для Android имеет стандартизированную структуру разработки проекта и фиксированный процесс компиляции/сборки приложения [5] (рисунок 2).



а) количество PHA от общего числа приложений, загруженных из/вне Google Play (за 2016–2018 годы)



б) структура PHA, загруженных из Google Play за 2018 год



в) структура PHA, загруженных вне Google Play за 2018 год

Рисунок 1 – Статистические данные по информационной безопасности Android [3]

В качестве базовой защиты архивного исполняемого файла-приложения APK (.apk) для Android используется:

уникальный идентификатор приложения (имя пакета приложения), подписанный цифровым сертификатом разработчика;

каталог META-INF, содержащий файлы: MANIFEST.MF / CERT.MF (список всех файлов пакета и их контрольных сумм), CERT.RSA (открытый ключ разработчика и созданную с помощью закрытого ключа цифровую подпись файла CERT.MF).

Для дополнительной усиленной защиты исполняемых файлов-приложений может использоваться:

обфускация кода (obfuscate) - преобразование исполняемого кода программы к виду, сохраняющему ее функциональность [6] (рисунок 3), но затрудняющему проведение реверсного инжиниринга программного обеспечения (software reverse engineering, SRE) [7];

дополнительная упаковка защищаемых файлов – переименование, перемещение, шифрование основных файлов приложения с передачей функции управления программе упаковщику [8].

Для изучения качества взаимодействия мобильной операционной системы Android с APK-сборками (.apk) КФО, в том числе оценки технологического уровня и степени безопасности проведем исследование одной APK-сборки (.apk) на предмет возможности реализации SRE с использованием программного обеспечения «Ghidra» (программа «Technology Transfer Program», NSA) [9].

Для сохранения конфиденциальности и соблюдения правовых норм: название КФО заменено порядковым номером, идентификатор приложения изменен, листинг (дизассемблер/декомпилятор) исполняемого кода (classes.dex) и алгоритмы функционирования не приводятся (для иллюстраций фрагментов применено маскирование /перестановка/ данных), копия APK-сборки (.apk) получена легальным путем, согласисие владельца получено (fair use / fair dealing) [10].

Основной целью для реверсного инжиниринга APK-сборки (.apk) при проведении исследования будет являться подтверждение возможности:

определения общей структуры APK-сборки (.apk);

выделения информации о APK-сборке (файл AndroidManifest.xml);

получения листинга (дизассемблер/декомпилятор) исполняемого кода (файл classes.dex);

построения функциональных графов (декомпилятор) базовых функций (файл classes.dex).

В качестве параметров оценки применения дополнительной защиты APK-сборки (.apk) будем использовать:

изменение стандартной структуры APK-сборки (.apk);

наличие значительного числа ошибок, неопределенных типов данных/меток в листинге (дизассемблер/декомпилятор) исполняемого кода (файл classes.dex);

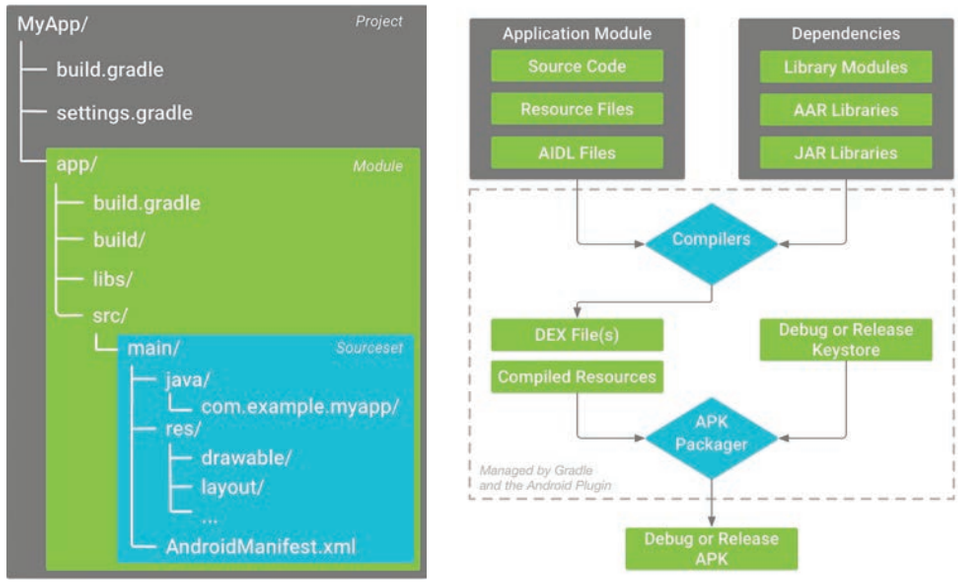
высокий уровень энтропии в листинге (дизассемблер) исполняемого кода (файл classes.dex) /общий диапазон значений энтропии: 0.0÷8.0; из них низкий уровень: 0.0÷4.0, средний уровень: 4.1÷6.0, высокий уровень: 6.1÷8.0/ [9].

В рамках данной статьи инструмент ProGuard из комплекта Android Studio [5] будем относить к базовой защите.

Общий интерфейс работы SRE-программы «Ghidra» показан на рисунке 4.

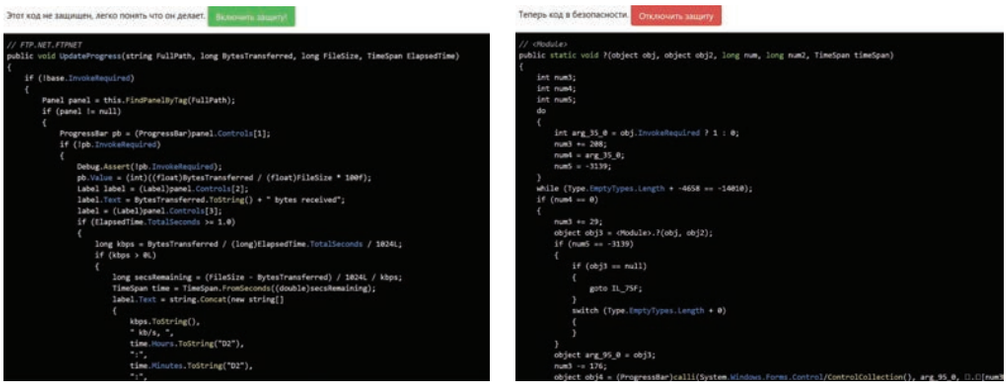
Основные результаты проведенного исследования APK-сборки (.apk) приведены в таблицах 1 и 2.

По результатам исследования (таблицы 1 и 2) можно сделать следующий вывод: APK-сборка (.apk) типовой КФО может быть подвергнута реверсному инжинирингу с получением общей/



а) структура разработки проекта в Android Studio б) процесс компиляции/сборки APK (.apk)

Рисунок 2 – Разработка архивных исполняемых файлов-приложений APK (.apk) [5]



а) исходный код приложения (.NET) б) измененный код приложения (.NET)

Рисунок 3 – Пример обфускации кода файла-приложения (.NET) [6]

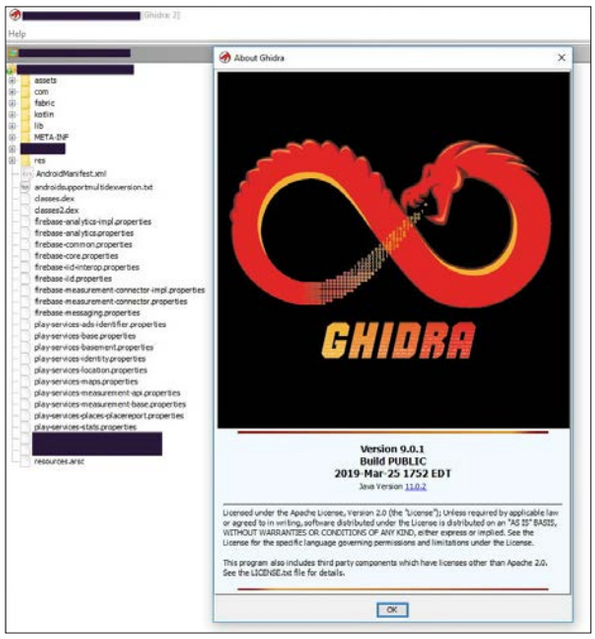
детальной информации о структуре APK-сборки (файл AndroidManifest.xml), листинга (дизассемблер/декомпилятор) исполняемого кода (файл classes.dex), функциональных графов (декомпилятор) базовых функций (файл classes.dex).

В качестве защиты APK-сборки (.apk) типовой КФО использовался базовой алгоритм (включая обфускацию кода от ProGuard из комплекта Android Studio) [5], признаков дополнительной защиты (усиленная обфускация кода / дополнительная упаковка защищаемых файлов) не выявлено.

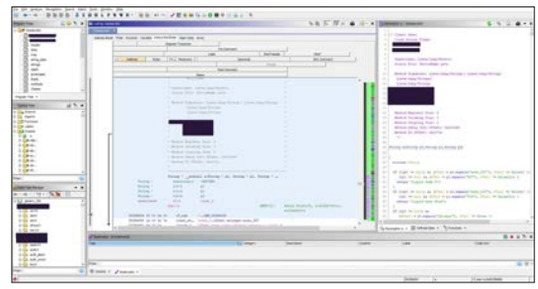
Для реализации дополнительной защиты пользователей КФО при авторизации в специализированном Android-приложении, как правило, используется метод двухфакторной аутентификации (2FA / Two-Factor authentication/). Наиболее распространенным способом реализации 2FA являются технологии OTP (one time password) [11], реализующие алгоритм получения одноразового пароля на основе SMS или TOTP [12].

Android-приложения, имеющие признаки PHA/malware, ранее скачанные пользователем из/вне Google Play Store [4] и инсталлированные на мобильное устройство, могут использоваться киберпреступниками для перехвата одноразовых паролей. При этом приложения PHA/malware могут быть подписаны легитимными цифровыми сертификатами удостоверяющих центров [13]. Указанные приложения PHA/malware могут генерировать фальшивые push-уведомления (API Notifications/Push) [14], использовать возможности сервиса для людей с ограниченными возможностями (Accessibility Service) [15] и другие технологии, реализующие атаку «человек посередине» (MITM /Man in the middle/) [16].

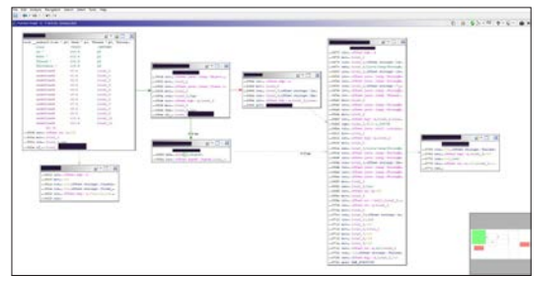
Например, malware на основе Android.Banker [17], инсталлированное с root-правами на мобильное устройство, имеет возможность перехвата одноразового пароля из SMS, которое сгенерировано для авторизации в типовом Android-приложении



а) структура APK (.apk) и информация о SRE-программе Ghidra



б) листинг (дизассемблер/декомпилятор) файла classes.dex из APK (.apk)



в) функциональный граф базовой функции файла classes.dex из APK (.apk)

Рисунок 4 – Пример результатов работы SRE-программы Ghidra /идентификатор приложения изменен, данные маскированы / [9]

КФО [4]. Эффективно противодействовать PHA/malware в данных случаях могут специальные защитные приложения [18].

Заключение. На основании проведенных исследований можно сделать следующие выводы:

1. В настоящее время использование профессиональных программ для реверсного инжиниринга [9] архивных исполняемых файлов-приложений APK (.apk) для Android позволяет получить достаточно полную информацию о их структуре, алгоритмах функционирования, а также листинг (дизассемблер/декомпилятор) исполняемого кода (таблица 1, рисунок 4).

2. В качестве основной, как правило, используется базовая защита архивного исполняемого файла-приложения APK (.apk), основанная на применении цифрового сертификата КФО/разработчика и встроенных механизмах компиляции/сборки Android Studio и др. (включая обфускацию кода от ProGuard и др.) (таблица 2).

3. Владельцам КФО необходимо уделять больше внимания использованию новых технологий/инструментов для создания APK-сборок (.apk), а также достаточному уровню обеспечения защиты информации на основе цифровых сертификатов удостоверяющих центров [13].

4. Пользователям услуг КФО необходимо осуществлять загрузку программного обеспечения только из проверенных источников [3], а также использовать на мобильных устройствах специальные защитные приложения [18].

Таблица 1 – Общие результаты проведенного исследования APK-сборки (.apk)

№ КФО	Возможность определения общей структуры APK-сборки (.apk)	Возможность выделения информации о APK-сборке (AndroidManifest.xml)	№ файла типа classes.dex	Возможность получения листинга (дизассемблер/декомпилятор) исполняемого кода (classes.dex)	Значение хэш-функции (MD5) листинга (дизассемблер/декомпилятор) исполняемого кода (classes.dex) /показана часть значения/	Возможность построения функциональных графов (декомпилятор) базовых функций (classes.dex)
1	+	+	1	+/+	da24b0b3...d06617e3	+
			2	+/+	a4b80f8d...5390bcaf	+

Таблица 2 – Результаты оценки качества проведенного исследования APK-сбороки (.apk)

№ КФО	№ файла типа classes.dex	Ошибки (дизассемблер) исполняемого кода, шт.	Уровень энтропии (дизассемблер) исполняемого кода	Наличие неопределенных типов данных/меток (функций) (дизассемблер/декомпилятор)	Максимальный размер базовой функции (декомпилятор), байт	Дополнительная защита APK-сборки	
						Признаки усиленной обфускации кода	Признаки дополнительной упаковки файлов
1	1	-	средний	-/-	10458	-	-
	2	-	средний	-/-	5930	-	-

ЛИТЕРАТУРА

1. Стратегия развития цифрового банкинга в Республике Беларусь на 2016 – 2020 годы // nbrb.by [Электрон. ресурс]. – 2000-2019. – Режим доступа: http://www.nbrb.by/Legislation/documents/DigitalBankingStrategy2016.pdf. – Дата доступа: 15.04.2019.
2. Operating System Market Share Worldwide // gs.statcounter.com [Электрон. ресурс]. – 1999-2019. – Режим доступа: http://gs.statcounter.com/os-market-share. – Дата доступа: 16.04.2019.
3. Android Security & Privacy 2018 Year In Review // source.android.com [Электрон. ресурс]. – 2019. – Режим доступа: https://source.android.com/security/reports/ Google_Android_Security_ 2018_Report_ Final.pdf. – Дата доступа: 19.04.2019.
4. Google Play // play.google.com [Электрон. ресурс]. – 2019. – Режим доступа: https://play.google.com/store/apps?hl=ru. – Дата доступа: 20.04.2019.
5. Android Studio// developer.android.com [Электрон. ресурс]. – 2019. – Режим доступа: https://developer.android.com/studio/build. – Дата доступа: 21.04.2019.
6. Appfuscator // appfuscator.net [Электрон. ресурс]. – 2014. – Режим доступа: http://appfuscator.net/#intro. – Дата доступа: 24.04.2019.
7. Practical Reverse Engineering / Gazet Alexandre, Dang Bruce, Josse Sebastien, Bachaalany Elias. – New York: Wiley, John Wiley and Sons, Ltd., 2014 – 384 с.
8. Подозрительные упаковщики и шифровальщики// kaspersky.ru [Электрон. ресурс]. – 2019. – Режим доступа: https://www.kaspersky.ru/resource-center/threats/suspicious-packers. – Дата доступа: 29.04.2019.
9. Ghidra // ghidra-sre.org [Электрон. ресурс]. – 2019. – Режим доступа: https://ghidra-sre.org/. – Дата доступа: 02.05.2019.
10. Coders' Rights Project Reverse Engineering FAQ // eff.org [Электрон. ресурс]. – 2019. – Режим доступа: https://www.eff.org/ru/issues/coders/reverse-engineering-faq. – Дата доступа: 03.05.2019.
11. One Time Password (OTP) // gemalto.com [Электрон. ресурс]. – 2006-2019. – Режим доступа: https://www.gemalto.com/companyinfo/digital-security/techno/otp. – Дата доступа: 10.05.2019.
12. How Time-based One-Time Passwords work and why you should use them in your app // freecodecamp.org [Электрон. ресурс]. – 2019. – Режим доступа: https://www.freecodecamp.org /news/how-time-based-one-time-passwords-work-and-why-you-should-use-them-in-your-app-fdd2b9ed43c3/. – Дата доступа: 11.05.2019.
13. Abusing Code Signing for Profit // medium.com [Электрон. ресурс]. – 2019. – Режим доступа: https://medium.com/@chroniclesec/abusing-code-signing-for-profit-ef80a37b50f4/. – Дата доступа: 22.05.2019.
14. Android Users Being Spammed Using Fake Missed Call Alerts // bleepingcomputer.com [Электрон. ресурс]. – 2003-2019. – Режим доступа: https://www.bleepingcomputer.com/ news/security/android-users-being-spammed-using-fake-missed-call-alerts/. – Дата доступа: 23.05.2019.
15. Новый Android-троян Gustuff опустошает счета и выводит криптовалюту // anti-malware.ru [Электрон. ресурс]. – 2005-2019. – Режим доступа: https://www.anti-malware.ru/news/2019-03-28-1447/29277. – Дата доступа: 28.05.2019.
16. Recommendations on mitigation of man-in-the-middle phishing attacks (evilginx2/Modlishka) // cert.pl [Электрон. ресурс]. – 2019. – Режим доступа: https://www.cert.pl/en/news/single/recommendations-on-mitigation-of-man-in-the-middle-phishing -attacks-evilginx2-modlishka/. – Дата доступа: 29.05.2019.
17. Android.Banker // drweb.ru [Электрон. ресурс]. – 2003-2019. – Режим доступа: https://vms.drweb.ru/search/?q=Android.Banker&lng=ru/. – Дата доступа: 30.05.2019.
18. The best antivirus software for Android // av-test.org [Электрон. ресурс]. – 2019. – Режим доступа: https://www.av-test.org/en/antivirus/mobile-devices/. – Дата доступа: 30.05.2019.

The technologies of creation, management and information protection of mobile applications of credit and financial organizations (CFO) are investigated. To study the quality of the CFO's Android mobile operating system interaction with APK assemblies (.apk), including the assessment of the technological level and security degree, it has been conducted the research of the possibility of obtaining information about the structure, functioning algorithms, information protection technologies, undeclared functions.

Получено 31.05.2019.