Р. Р. Глушень

ИССЛЕДОВАНИЕ ЧАСТЫХ ОШИБОК ПРИ ХРАНЕНИИ ДАННЫХ В МОБИЛЬНЫХ ПРИЛОЖЕНИЯХ

Рассматриваются существующие проблемы в области хранения данных в долговременной памяти при разработке мобильных приложениях.

Введение

В современных реалиях, мобильные устройства занимают важную роль в жизни человека. Однако очень редко разработчики мобильных приложений обеспечивают должный уровень защиты данных пользователя. Одной из причин этого явления служит отсутствие стандартизированного подхода к решению проблемы.

I. Хранение данных в открытом виде

Разработчики зачастую склонны сохранять информацию в защищенные системные хранилища и даже в файловую систему без дополнительной защиты, поскольку системные механизмы хорошо сопротивляются взлому. Однако уровень их стойкости падает до минимума в случае, если устройство рутованное. Если злоумышленник физически достигает мобильного устройства, злоумышленник подключает мобильное устройство к компьютеру с помощью свободно доступного программного обеспечения. Эти инструменты позволяют злоумышленнику видеть все сторонние каталоги приложений, которые часто содержат хранимую личную информацию или другие конфиденциальные информационные активы.

Уязвимость также касается хранения секретных данных внутри кода (в статических константных строках, в ресурсах приложения и т.п.). Яркие примеры: хранение соли для пароля (password salt) в константе или макросе, которая применяется по всему коду для шифрования паролей; хранение приватного ключа для асимметричных алгоритмов; хранение паролей и логинов для серверных узлов или баз данных. Легко вскрывается третьей стороной при наличии базовых навыков декомпиляции.

Защита: Пользовательские данные не должны использоваться в приложении без дополнительного шифрования. Как только надобность в "открытой"; информации отпала — она немедленно должна быть либо зашифрована, либо уничтожена. Любые данные перед выходом за пределы приложения должны быть зашифрованы. Локальные хранилища платформы не являются областью приложения, они тоже должны получать на вход только зашифрованные данные.

II. Применение алгоритмов с хранением приватного ключа

Уязвимость актуальна в случае, когда приватный ключ вынужденно сохраняется в коде или ресурсах мобильного приложения. Легко вскрывается с помощью реверс-инжиниринга.

Защита: В мобильной разработке желательно применять только современные симметричные алгоритмы с генерируемым случайным одноразовым ключом, обладающие высокой стойкостью с взлому методом грубой силы, либо выводить асимметричный приватный ключ за пределы приложения, либо персонализировать этот ключ (приватным ключом может выступать пользовательский код входа, сохраненный в зашифрованном виде или не сохраняемые вообще).

III. Выводы

Часто приложения позиционируют себя как защищенные, но на деле таковыми не являются, так как содержат внутри себя средства для расшифровки персональной информации.

Хорошим примером реализации безопасного хранения данных является запрашивание у пользователя PIN-кода, затем использование PIN-кода как связанных данных(associated data) с использованием AEAD-режима блочного шифрования. Таким образом, возможность или невозможность расшифровки пользовательских данных будут служить индикатором правильности PIN-кода. При этом PIN-код вообще не сохраняется на устройстве, но операции шифрования основаны на PIN-коде.

1. Drake, J. J. Android Hacker's Handbook / J. Drake, P. Fora, Z. Lanier, C. Mulliner, S. Ridley, G. Wicherski // John Wiley & Sons, Inc., Indianapolis, Indiana. – 2014. – 545 с.
2. Elenkov, N. Android Security Internals / N. Elenkov // No Starch Press, Inc., San Francisco, CA. – 2015. – 433 с.

Глушень Руслан Русланович, магистрант кафедры информационных технологий автоматизированных систем БГУИР, ruslan.hlushen@gmail.com.

Научный руководитель: Матвеенко Владимир Владимирович, кандидат технических наук, доцент кафедры вычислительных методов и программирования, vladzimir66@bsuir.by.