УДК 004.054

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ТЕСТИРОВАНИЯ ПРОНИКНОВЕНИЕ УСТРОЙСТВ ПОД УПРАВЛЕНИЕМ ОПЕРАЦИОННОЙ СИСТЕМЫ IOS С ОБХОДОМ СУЩЕСТВУЮЩИХ ВСТРОЕННЫХ СИСТЕМ ЗАЩИТЫ

Симаков А.А., Дудкин А.С. (Военно-космической академии имени А.Ф.Можайского)
Научный руководитель – к.т.н. Дудкин А.С. (Военно-космической академии имени А.Ф.Можайского)

Аннотация. Основной идеей настоящего исследования является разработка устройства для тестирования на проникновение устройств под управлением операционной системы IOS с обходом существующих встроенных систем защиты. Разработанное программное средство предназначено для отладки устройств, понижения версии прошивки, получения данные загрузчика операционной системы, а также, в перспективе, извлечения информации о вызовах, SMS-текстов, сообщений из мессенджеров, информации об аккаунтах пользователя (частично), истории подключений к сетям Wi-Fi, данных о сопряжении с Bluetooth-устройствами, системных журналов WhatsApp, информации об аккаунте Viber, списка заблокированных контактов, plist-файлов. Так же рассматривается извлечение системных файлов и журналов, на основе которых можно понять, как функционировало устройство.
Основная цель разработанного программного средства – создание условий для не декларированной отладки устройств, выявление уязвимых мест операционной системы IOS, а также предотвращение их эксплуатации.
Ключевые слова: ios, операционная система, извлечение данных, обход защиты, конфиденциальная информация.

Введение

Операционная система (ОС) IOS является одной из самых популярных в мире. Доля проданных устройств на данной ОС на мировом рынке занимает более 23%. Устройства Apple считаются одними из самых безопасных. Компания приложила немало усилий чтобы повысить стойкостью своих устройств к взлому.

В основу программного средства входит описание эксплойта checkm8 зарубежного исследователя безопасности axi0mX. При использовании разработанного устройства все средства защиты операционной системы отключатся.

Главная особенность эксплойта в том, что уязвимость была обнаружена не на программном, а на аппаратном уровне устройств Apple, причем охватывает она очень большой диапазон моделей, начиная с самых старых устройств на чипе A5 вроде iPhone 4S и заканчивая вполне современным iPhone X.

Основы эксплойта

Брешь скрыта в механизме BootROM, который играет ключевую роль в процессе загрузки i-устройств. Причем исправить ее программными методами невозможно: для того, чтобы решить проблему, нужно пересмотреть аппаратную конфигурацию самого устройства.

BootROM нельзя обновить. Он помещается во внутреннюю read-only память при изготовлении устройства. BootROM - это аппаратный корень доверия цепочки загрузки. Уязвимости в нем могут позволить получить контроль над дальнейшим процессом загрузки и исполнять неподписанный код на устройстве. Под данную уязвимость попадают 70% всех устройств данной ОС. Архитектурно SecureROM, он же BootROM, представляет собой первое звено цепочки безопасной загрузки, придуманной Apple для защиты — вредоносных программ. В SecureROM вшит криптографический ключ Apple, используемый для расшифровки образов, которые задействованы на последующих этапах загрузки, а также имеется необходимый инструментарий для работы с криптоалгоритмами. Получив управление от SecureROM, загрузчик iBoot расшифровывает и запускает ядро операционной системы, после чего загружается образ самой iOS с графическим интерфейсом пользователя. Однако все эти этапы запуска i-устройств выполняются, только если инициализация SecureROM прошла успешно.

Именно поэтому все существовавшие инструменты, которые позволяли получить доступ к файловой системе, старались всячески обойти этот механизм. Ведь их первоочередная задача — загрузить измененный образ iOS, допускающий установку программ из сторонних источников, чего не должно происходить с использованием SecureROM, стоящего на страже безопасной загрузки. Именно полный контроль над процессом запуска операционной системы гарантирует невозможность проникновения на устройство всевозможных буткитов, руткитов и прочего.

Если кратко обобщить проделанную работу, можно сказать, что из-за найденной в SecureROM ошибки в механизме создания и уничтожения USB-стека происходит утечка памяти, которая может использоваться для формирования состояния heap (кучи), дающего возможность управлять выделением памяти при размещении буфера. В результате с помощью Use-after-Free можно выполнить запись в выделенную память для получения контролируемого косвенного перехода (controlled indirect branch) при выходе из DFU (режима восстановления).

По принципу своего действия разработанное устройство на основе эксплоита checkm8 реализует типичный буткит. Основная его задача состоит в том, чтобы дать устройству нормально загрузиться, но при этом скомпрометировать каждое звено в цепочке загрузки после того, как отработает BootROM.

Результаты

На сегодняшний день уже удается:

• понизить версию прошивки iPhone 3GS;

• выгрузка SecureROM на устройствах с процессорами S5L8920/S5L8922/S5L8930;

• выгрузка NOR устройств с процессорами S5L8920

• зашифровать или расшифровать шестнадцатеричные данные на подключенном устройстве в режиме Pwned DFU, используя его ключ GID или UID;

• включать verbose-загрузку iOS;

• включить отладку уязвимых устройств с помощью специального JTAG/SWD кабелей.

Заключение

Одна из основных сильных сторон Apple — это безопасность, которую они обеспечивают своими сервисами, вплоть до цепочки загрузки, где критический код, отвечающий за загрузку устройства, используется для создания безопасной цепочки доверия от одного уровня абстракции к другому. Если цепочка загрузки скомпрометирована, каждая отдельная служба, выполняемая после нее, уязвима. Поэтому у данного устройства широкие перспективы.

Не мало возможностей было достигнуто с помощью эксплойта checkm8, но останавливаться на достигнутом нельзя. В будущем планируется модифицировать устройство: перейти на платы Raspberry Pi, использовать урезанную ОС Ubuntu с настроенной на ней ipwndfu, что бы при физическом контакте с i-устройством автоматически извлекать из него информацию о вызовах, SMS-тексты, сообщения из мессенджеров, а также информация об аккаунтах пользователя (частично), история подключений к сетям Wi-Fi, данные о сопряжении с Bluetooth-устройствами, файл DataUsage.sqlite, системные журналы WhatsApp, информация об аккаунте Viber, список заблокированных контактов, plist-файлы и другие материалы. Так же рассматривается извлечение системных файлов и журналов, на основе которых можно понять, как функционировало устройство.

Симаков А.А. (автор)

Дудкин А.С. (научный руководитель)