Elcomsoft iOS Forensic Toolkit

ElcomSoft Co. Ltd.

Version 8.70

0 Введение

Настоящий документ описывает принципы работы программного продукта Elcomsoft iOS Forensic Toolkit (EIFT) для операционной системы MacOS X.

Всю информацию об ошибках, вопросы, предложения и.т.д. вы можете направить по этому адресу:

checkm8@elcomsoft.com

Содержание

0 Введение **2**

1 Содержание **5**
1.1 Физическое извлечение . . . 5
1.2 Извлечение на основе jailbreak . . . 5
1.3 Извлечение с помощью агента . . . 5
1.4 Логическое извлечение . . . 5

2 Системные требования **6**

3 Установка и запуск **6**

4 Извлечение **6**
4.1 Физическое извлечение . . . 7
4.1.1 DFU-режим . . . 7
4.1.2 Загрузка рамдиска . . . 12
4.1.3 Разблокировка пользовательских данных . . . 14
4.1.4 Извлечение данных . . . 16
4.1.5 Полное HFS извлечение . . . 16
4.2 Извлечение на основе jailbreak . . . 19
4.2.1 Утилита autolocknever . . . 19
4.2.2 Копирование связки ключей . . . 20
4.2.3 Создание Tar-архива . . . 20
4.2.4 Создание образа диска . . . 20
4.3 Извлечение с помощью агента . . . 21
4.3.1 Установка агента . . . 21
4.3.2 Копирование связки ключей . . . 22
4.3.3 Создание Tar-архива . . . 22
4.3.4 Извлечение метаданных . . . 23
4.3.5 Извлечение данных . . . 23
4.3.6 Создание образа диска . . . 23
4.3.7 Деинсталляция агента . . . 23
4.4 Логическое извлечение . . . 23
4.4.1 Отношение доверия . . . 24
4.4.2 Создание резервной копии . . . 24
4.4.3 Копирование файлов по протоколу afc . . . 25
4.4.4 Копирование системных журналов об ошибках . . . 25
4.4.5 Копирование совместно-используемых файлов . . . 26

5 Журналирование **26**

6 Дополнительные опции **26**
6.1 Общая структура EIFT_cmd . . . 26
6.2 EIFT_cmd info . . . 27
6.3 EIFT_cmd ssh . . . 27
6.4 EIFT_cmd serial . . . 28
6.5 EIFT_cmd tools . . . 28
6.6 Аппаратная реализация checkm8 . . . 28
6.6.1 Необходимое оборудование . . . 29
6.6.2 Установка прошивки . . . 29
6.6.3 Подготовка оборудования . . . 29
6.6.4 Эксплуатация . . . 30
6.6.5 Режимы работы и LED-индикатор . . . 30
6.7 Дополнительные утилиты . . . 31
6.8 Управление лицензией . . . 31

7 Анализ полученных данных **32**

8 Устранение проблем **32**
8.1 Не удается перевести устройство в DFU режим . . . 32
8.2 Разблокировка данных завершается с ошибкой . . . 32
8.3 Не удается найти SEP . . . 32

A Сокращения **33**

B Поддерживаемые устройства **33**
B.1 Вход в DFU-режим . . . 33
B.1.1 Метод 1 . . . 33
B.1.2 Метод 2 . . . 34
B.1.3 Метод 3 . . . 34
B.1.4 Метод 4 . . . 35
B.1.5 Метод 5 . . . 35
B.1.6 Метод 6 . . . 35
B.1.7 Метод 7 . . . 35
B.1.8 Method 8 (Apple Watch) . . . 35
B.2 Физическое извлечение . . . 35
B.2.1 iPhone . . . 35
B.2.2 iPod Touch . . . 36
B.2.3 iPad . . . 36
B.2.4 AppleTV . . . 36
B.2.5 Apple Watch . . . 37
B.2.6 HomePod . . . 37
B.3 Извлечение на основе Jailbreak . . . 37
B.4 Извлечение с помощью агента . . . 37
B.4.1 iPhone . . . 37
B.4.2 iPad . . . 37
B.4.3 iPod Touch . . . 37
B.5 Логическое извлечение . . . 37
B.6 Полное HFS извлечение . . . 38
B.6.1 iPhone . . . 38
B.6.2 iPod Touch . . . 38
B.6.3 iPad . . . 38
B.7 Аппаратная реализация checkm8 . . . 38
B.7.1 iPhone . . . 38
B.7.2 iPod Touch . . . 38
B.7.3 iPad . . . 38
B.7.4 Apple TV . . . 38

1 Содержание

EIFT - это программный продукт, предназначенный для извлечения данных с iOS-устройств. EIFT использует четыре различных подхода для извлечения, каждый из них будет описан ниже:

- Физическое извлечение (Раздел 1.1)
- Извлечение на основе jailbreak (Раздел 1.2)
- Извлечение с помощью агента (Раздел 1.3)
- Логическое извлечение (Раздел 1.4)

1.1 Физическое извлечение

Физическое извлечение - это наиболее предпочтительный способ извлечения данных, так как он не модифицирует информацию на устройстве. При физическом извлечении на устройство загружается специальный рамдиск (ramdisk), который никак не взаимодействует с установленной операционной системой. Разделы устройства при этом монтируются в режиме read-only, а копирование данных и записей из связки ключей (keychain) не оставляет следов в журналах системы.
Если необходимо извлечь или проверить какой-либо конкретный файл, то эксперту в этом может помочь Secure Shell (SSH)-сервер, запущенный на рамдике.
Кроме того, существует возможность вручную перемонтировать файловую систему в режиме writable и модифицировать данные, хотя подобные манипуляции окажутся лишними, если нужно просто извлечь все данные с устройсва.
Физическое извлечение считается предпочтительным способом извлечения данных, если конечно оно возможно на вашем устройстве.

1.2 Извлечение на основе jailbreak

Второй способ извлечения возможен, если на устройстве установлен jailbreak, и работает SSH-сервер. Для доступа по SSH необходимо также знать пароль пользователя root.
Второй метод обычно позволяет извлечь всю файловую систему и связку ключей (keychain). Тем не менее, объем доступных данных может различаться в зависимости от конкретного jailbreak'a. Для того, чтобы извлечь файловую систему и связку ключей, на устройство загружается специальный код, который после выполнения удаляется. Записи о доступе по SSH могут остаться в журналах системы; к сожалению, этого нельзя избежать, но и на целостность данных подобные записи особого влияния не окажут. Если физическое извлечение для вашего устройства недоступно, то следующий лучший выбор - это извлечение на основе jailbreak.

1.3 Извлечение с помощью агента

Если предыдущие два способа не подходят, то на устройство можно попробовать установить приложение-*Агент*. Для копирования файловой системы и связки ключей (keychain) *Агент* эксплуатирует известные уязвимости в ядре и повышает свои привилегии. Список поддерживаемых устройств/версий iOS находится в Разделе B.4.
Нужно помнить, что хотя EIFT и старается оставлять как можно меньше следов в системе, третий метод извлечения подразумевает установку дополнительного приложения на устройство. После извлечения *Агент* можно удалить.
Если на устройстве уже установлен jailbreak и есть доступ по SSH, то для извлечения предпочтительнее использовать предыдущий метод (на основе jailbreak). В противном случае рекомендуется прибегнуть к установке *Агента*.

1.4 Логическое извлечение

Логическое извлечение доступно для всех устройств вне зависимости от установленной версии iOS. Нет необходимости и в дополнительной установке jailbreak'a или эксплуатации уязвимостей. Если между iOS-устройством и компьютером установлены доверительные отношения (существует действующий lockdown-файл или iTunes pairing record), то логическое извлечение сработает, даже если пароль разблокировки неизвестен. Еще одно дополнительное условие для логического извлечения - устройство должно быть разблокировано хотя бы раз после включения. Четвертый метод извлечения, помимо всего прочего, позволяет создать iTunes-подобные бэкапы.

2 Системные требования

- Компьютер с установленной MacOS X
  - macOS High Sierra (и новее)
- Elcomsoft iOS Forensic Toolkit (EIFT) v8.70 и новее.
  Примечание: Данный релиз поддерживает только Mac OS X.
- Поддерживаемый iPhone или iPad (см. Раздел B). Как минимум одно из следующих требований должно выполняться для устройства:
  - Устройство уязвимо к эксплойту checkm8 и способно загрузиться в режим Device Firmware Update (DFU) (Физическое извлечение. См. Раздел B.2)
  - На устройстве установлен jailbreak (Извлечение на основе jailbreak. См. Раздел B.3)
  - Устройство функционирует (способно загрузиться) (Извлечение с помощью агента, Раздел B.4 или логическое извлечение, Раздел B.5)
- Известен пароль разблокировки.
  Примечание: Если пароль неизвестен, то часть информации с устройства все же можно извлечь, хотя ее объем будет крайне ограничен(Before First Unlock (BFU)).
- На устройстве установлена поддерживаемая версия iOS (см. Раздел B)
- Переходник lighning - USB-A (Переходник lighning - USB-C не подходит для извлечения на основе эксплойта checkm8!)
  Примечание: Некоторые устройства придется подключить через USB-хаб, в противном случае эксплойт checkm8 не сработает (относится только к компьютерам, работающим на чипе M1)

3 Установка и запуск

EIFT поставляется вместе с защитным USB-донглом. Донгл должен быть подключен в течение всего времени использования EIFT.

Скачайте Образ диска Apple (DMG)-образ с консольной версией EIFT. После ввода пароля для DMG-образа и монтирования диска, скопируйте папку *EIFT x.y* на локальный диск. Затем откройте приложение *Terminal* и с помощью команды *cd* перейдите в нужную директорию. Наберите в терминале команду "`cd` " (обратите внимание на пробел после команды), перетащите папку с EIFT прямо в окно терминала и нажмите **Enter**. Наконец, запустите EIFT, набрав команду `./EIFT_cmd`. Вся процедура установки визуализирована на Рисунке 1.
Примечание: Перед всеми *EIFT_cmd* командами, представленными в данном руководстве, необходимо добавлять символы "./"
Примечание: перед первым запуском программы необходимо также снять карантинный флаг со всех файлов, делается это только один раз. Между шагами, показанными на Рисунке 1b и Рисунке 1c, наберите в терминале следующую команду (не забудьте точку в конце команды!):

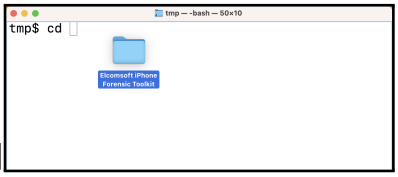
```
xattr -r -d com.apple.quarantine .
```

Вместо точки в конце команды мы можете также перетащить папку прямо в терминал.

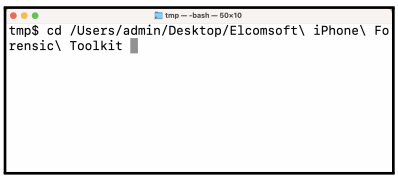
4 Извлечение

Все методы извлечения (помимо логического) позволяют осуществить следующие действия:

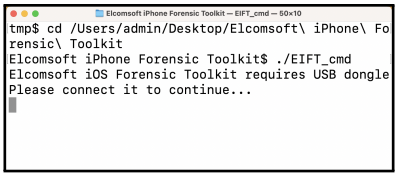
- Скопировать (и расшифровать) связку ключей (keychain)
- Скопировать файловую систему в виде Tape archive (TAR)
  - Скопировать пользовательский раздел
  - Скопировать системный раздел
- Создать образ flash-памяти



(a) Перетащите папку с EIFT в терминал



(b) Полный путь отобразится автоматически



(c) Запустите EIFT из нужной директории

Рис. 1: Установка и запуск EIFT

4.1 Физическое извлечение

При физическом извлечении эксплуатируется уязвимость в BootROM, которая позволяет загрузить специальный рамдиск и извлечь данные с устройства. Основные шаги здесь следующие:

1. Перевести устройство в DFU-режим
2. Загрузить рамдиск
3. Разблокировать устройство
4. Извлечь данные

Для старых устройств без Secure Enclave Processor (SEP), EIFT предоставляет возможность *Полного HFS извлечения* и подбора пароля. Последовательность дейсвий здесь следующая:

1. Перевести устройство в DFU-режим (Раздел 4.1.1)
2. Загрузить рамдиск (Раздел 4.1.2)
3. Выполнить *полное HFS извлечение* (Раздел 4.1.5)

Полный список устройств, для которых доступна функция *Полного HFS извлечения* и подбора пароля можно найти в Разделе B.2.

4.1.1 DFU-режим

Перевести устройство в DFU-режим необходимо вручную. Сделать это можно несколькими способами (более подробная информация доступна **здесь**).
Данная инструкция подразумевает, что устройство подключено через кабель lightning - USB-A, с кабелем lightning - USB-C она не сработает.

Если инструкция выполнена правильно, то экран устройства останется темным, а iTunes или Finder (в зависимости от версии macOS) обнаружит устройство в Recovery-режиме (в действительности это будет DFU-режим).

Если кнопки на устройстве сломаны, и выполнить инструкцию не представляется возможным, то воспользуйтесь **этой** статьей. В ней описано, как ввести в режим DFU, разобрав устройство и замкнув специальные контакты.

4.1.1.1 Способ 1: iPhone / iPad / iPod, у которых есть кнопка *Home*

Первый способ перехода в DFU-режим подойдет для iOS-устройств, у которых есть кнопка *Home*: iPod, iPad и iPhone (версии iPhone 6s и ниже). Полный список устройств, совместимых с первым способом, можно найти в Разделе B.1.1. Для iPhone 7 и более новых моделей рекомендуется воспользоваться Способом 2 (см. Раздел 4.1.1.2) или Способом 3 (см. Раздел 4.1.1.3).

1. Убедитесь, что устройство не подключено к компьютеру и выключено
2. Зажмите кнопку *Home*, подсоедините lightning-кабель, продолжая удерживать кнопку *Home* до тех пор, пока на экране не появится надпись “Connect to iTunes"; отпустите кнопку *Home*. После этого устройство окажется в *Recovery*-режиме
3. Зажмите кнопки *Home* и *Sleep/Power* (расположена вверху, либо сбоку справа) на 10 секунд (Экран потухнет на 7 секунде)
4. Отпустите кнопку *Sleep/Power*, но продолжайте удерживать кнопку *Home* еще 15 секунд
5. Отпустите кнопку *Home*

4.1.1.2 Способ 2: iPhone 7

Второй способ перехода в DFU-режим подойдет только для iPhone 7. Полный список устройств, совместимых со вторым способом, можно найти в Разделе B.1.2. Для iPhone 6s и более старых моделей рекомендуется воспользоваться Способом 1 (см. Раздел 4.1.1.1.) Для iPhone 8 и более новых моделей рекомендуется воспользоваться Способом 3 (см. Раздел 4.1.1.3)

1. Убедитесь, что устройство не подключено к компьютеру и выключено
2. Зажмите кнопку уменьшения громкости *Volume Down* и подключите кабель. Продолжайте удерживать кнопку уменьшения громкости до тех пор, пока на устройстве не появится надпись “Connect to iTunes"; затем отпустите кнопку *Volume Down*. Устройство теперь находится в *Recovery*-режиме.
3. Зажмите кнопки *Volume Down* и *Sleep/Power* (расположена вверху, либо сбоку справа) на 10 секунд (Экран потухнет на 7 секунде)
4. Отпустите кнопку *Sleep/Power*, но продолжайте удерживать кнопку *Volume Down* еще 15 секунд
5. Отпустите кнопку уменьшения громкости *Volume Down*

4.1.1.3 Способ 3: iPhone / iPad / iPod, у которых нет кнопки *Home*

Третий способ перехода в DFU-режим подойдет для моделей iPad и iPhone, у которых нет кнопки *Home*. Полный список устройств, совместимых с третьим способом, можно найти в Разделе B.1.3. Для iPhone 7 и более старых моделей рекомендуется воспользоваться Способом 1 (см. Раздел 4.1.1.1), либо Способом 2 (см. Раздел 4.1.1.2).

1. Убедитесь, что устройство подключено к компьютеру
2. Быстро нажмите и отпустите кнопку увеличения громкости *Volume Up*
3. Быстро нажмите и отпустите кнопку уменьшения громкости *Volume Down*
4. Зажмите и удерживайте кнопку *Power/Side* (расположена вверху или сбоку) на 10 секунд
5. Продолжайте удерживать кнопку *Power/Side* и дополнительно зажмите кнопку уменьшения громкости *Volume Down* на 5 секунд
6. Отпустите кнопку *Power/Side*, но продолжайте удерживать кнопку уменьшения громкости еще 15 секунд
7. Отпустите кнопку уменьшения громкости *Volume Down*

4.1.1.4 Способ 4: AppleTV 3

Полный список устройств, совместимых с четвертым способом, можно найти в Разделе B.1.4.

1. Убедитесь, что устройство подключено к источнику питания и включено
2. Убедитесь, что устройство подключено к компьютеру с помощью micro-USB кабеля
3. Нажмите и удерживайте кнопки *Down (1)* и *Menu (2)* до тех пор (около 6 секунд), пока LED-индикатор не начнет быстро мигать
4. Отпустите обе кнопки
5. Нажмите и удерживайте кнопки *Play (3)* и *Menu (2)* до тех пор (около 6 секунд), пока LED-индикатор не начнет быстро мигать
6. Отпустите обе кнопки

Примечание: важно именно отпустить обе кнопки. Если вы будете удерживать кнопку *Menu*, то перейти в DFU-режим не удастся.
Кнопки *Down*, *Menu* and *Play* отмечены на Рисунке 2b как *(1)*, *(2)* and *(3)* соответсвенно.

(a) Apple TV 3

(b) пульт дистанционного управления

Рис. 2: Apple TV 3 и пульт дистанционного управления

4.1.1.5 Способ 5: AppleTV HD (4th Generation)

Полный список устройств, совместимых с пятым способом, можно найти в Разделе B.1.5.

1. Убедитесь, что устройство подключено к источнику питания и включено
2. Убедитесь, что устройство подключено к компьютеру с помощью USB-C кабеля
3. Нажмите и удерживайте кнопки *Menu* и *Play* до тех пор (около 6 секунд), пока LED-индикатор не начнет быстро мигать
4. Отпустите обе кнопки

Кнопки *Menu* и *Play* отмечены на Рисунках 3b (Siri Remote 1) и 3c (Siri Remote 2).

(a) Apple TV HD (b) Siri Remote 1 (c) Siri Remote 2

Рис. 3: Apple TV HD и пульты дистанционного управления

4.1.1.6 Способ 6: AppleTV 4K (первая модель)

Полный список устройств, совместимых с шестым способом, можно найти в Разделе B.1.6. AppleTV 4K внешне очень похож на AppleTV HD (см. Рисунок 3a), за тем лишь исключением, что на корпусе AppleTV 4K отсутствует USB-порт. Вместо USB-разъема в корпусе AppleTV 4K оборудован специальный диагностический порт, скрытый внутри Ethernet-разъема. Для подключения AppleTV 4K к компьютеру и его перевода в DFU-режим необходим так называемый "DCSD кабель" и переходник с Lightning-портом. После подключения "DCSD кабеля" устройство автоматически перейдет в DFU-режим при загрузке: никаких дополнительных действий от пользователя не требуется. Скрытый диагностический порт изображен на Рисунке 4, а способ подключения AppleTV 4K к компьютеру - на Рисунке 5.

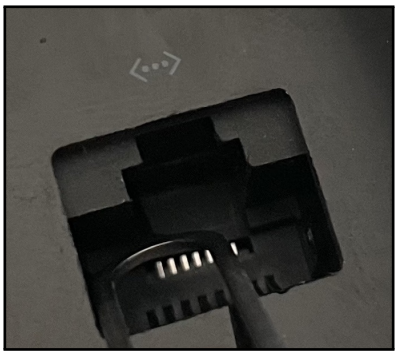


Рис. 4: Скрытый диагностический порт Apple TV 4K

4.1.1.7 Метод 7: HomePod (первая модель)

Полный список устройств, совместимых с седьмым способом, можно найти в Разделе B.1.7.
Для того чтобы перевести HomePod в DFU-режим, его необходимо сначала перевернуть (так, чтобы светодиоды Siri LED смотрели в стол) и подключить к компьютеру через USB. И только после этого подключить HomePod к питанию - HomePod автоматически перейдет в dfu - режим.
Для подключения HomePod к компьютеру необходимо снять нижнюю заглушку и подключить специальный адаптер. Пример подключения показан на Рисунке 6. Более подробную информацию можно найти в нашем блоге blog.elcomsoft.com.

4.1.1.8 Способ 8: Apple Watch

Полный список устройств, совместимых с восьмым способом, можно найти в Section B.1.8.

1. Убедитесь, что устройство включено

Рис. 5: Подключение Apple TV 4K к компьютеру

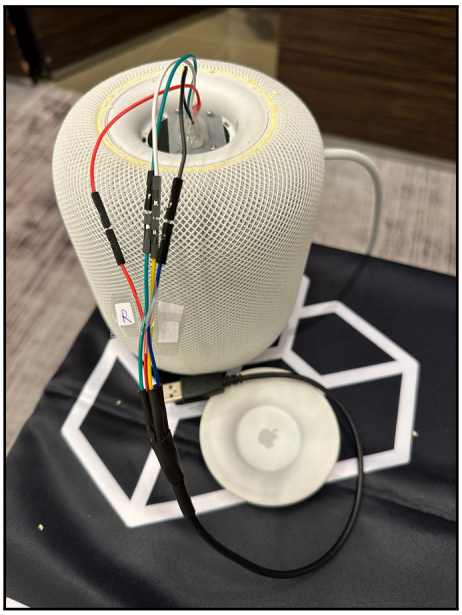


Рис. 6: Подключение HomePod через адаптер, напечатанный на 3D-принтере

2. Подключите устройство к PC с помощью адаптора (устанавливать отношение доверия при этом необязательно)
3. Нажмите и удерживайте колесико *Digital Crown* и кнопку *Side* на 10 секунд (до тех пор, пока экран не потухнет)
4. Отпустите кнопку *Side*, продолжая удерживать колесико *Digital Crown* до тех пор, пока EIFT автоматически не обнаружит устройство в DFU режиме.
5. Отпустите колесико

Примечание: Перевод очень старых устройств (например, Apple Watch Series 0) в DFU режим может занять продолжительное время и несколько попыток. Загрузка рамдиска на S0 занимает порядка 5 минут, также для этих устройств критически важна надежность соединения и кабеля. Чтобы сэкономить время, рекомендуется сначала перевести часы в Recovery режим, а затем уже в DFU.
Для перевода часов в Recovery режим можно воспользоваться следующей командой:

```
./EIFT_cmd tools recovery
```

4.1.2 Загрузка рамдиска

Загрузка рамдиска выполняется специальной командой `EIFT_cmd boot`. В большинстве случаев рекомендуется вызывать ее с параметром `-w`: тогда EIFT будет ожидать, пока устройство не окажется в DFU-режиме.

```
./EIFT_cmd boot -w
Started logging Thread!
Waiting for device in DFU mode....
```

Как только устройство окажется в DFU-режиме, EIFT попробует эксплуатировать уязвимость в BootROM. После успешной эксплуатации EIFT сохраняет копию установленного iBoot (на некоторых устройствах) и применяет патчи в памяти BootROM.

Примечание: Некоторые устройства (например iPhone 4s) несовместимы с программной реализацией checkm8. BootROM подобных устройств придется эксплуатировать другим способом, и только после этого подключать устройство (в режиме PWND-DFU) для дальнейшего извлечения. Один из способов - это аппаратная реализация checkm8 на микроконтроллере (см. Раздел 6.6). Аппаратная реализация эксплойта может послужить хорошей альтернативой программной реализации, так как она более надежна на одних платформах, а для других может оказаться единственно возможной.

Иногда во время выполнения эксплойта EIFT может попросить пользователя переподключить кабель. В этом случае рекомендуется переподключить USB-кабель именно со стороны компьютера.
С некоторыми устройствами эксплуатация будет всегда завершаться с ошибкой, если их не подсоединить через USB-хаб. Поэтому, если вы испытываете трудности при эксплуатации устройства, поробуйте воспользоваться USB-хабом.
Далее, пользователю необходимо предоставить файл прошивки конкретной версии iOS. В зависимости от устройства это будет либо последняя доступная версия iOS (например, *iOS 10.3.4* для *iPhone 5*), либо версия, которая установлена на устройстве в данный момент (*iPhone 5s* и новее).
EIFT анализирует сохраненную копию iBoot и показывает версию установленной iOS. Иногда у разных минорных версий iOS могут быть одинаковые версий iBoot, поэтому EIFT предложит несколько вариантов. В такой ситуации постарайтесь предугадать установленную версию iOS, используя всю имеющуюся информацию. Ошибочный выбор минорной версии обычно не влияет на успешность извлечения.
Впоследствии EIFT определит точную версию установленной iOS. Если загруженная версия отличается от установленной, EIFT выдаст предупреждение. Пользователь затем сам решает, перезагрузиться с правильной версией iOS или продолжить с уже выбранной. Попытка загрузить несовместимую версию SEP как правило приводит к аварийной остановке системы. Если это произошло, то просто начните сначала, выбрав для загрузки правильную версию. Процесс выбора прошивки показан на Листинге 1.

```
Attepting to load required iOS version for device...
This device (0x00008000) doesn't have a hardcoded required iOS version. Proceed with parsing
↪  iBoot dump!
Parsing iBoot version...
Got iBoot: iBoot-7429.42.2

Either one of this iOS versions is installed:
15.1 (19B74): https://updates.cdn-apple.com/2021FallFCS/fullrestores/071-63917/
↪  94C75F99-97F0-4B67-8559-A83CA667CF99/iPhone_5.5_15.1_19B74_Restore.ipsw

If you don't know exactly which version is installed, just choose the first one.
At a later stage Toolkit will check the exact build number and will inform you
about versions mismatch

Please drag and drop an IPSW to this screen!
Got:
```

Листинг 1: EIFT firmware selection dialogue

Скачайте прошивку по предложенной ссылке, перетащите ее в окно терминала и нажмите **Enter**. EIFT продолжит загрузку рамдиска. После загрузки экран устройства должен выглядеть следующим образом (см. Рисунок 7)

Рис. 7: iPhone 6s Plus загруженный в режиме рамдиска

Примечание: при загрузке iOS 16.1 и выше с операционной системы Linux, на экране компьютера может появится окно, как показано на Рисунке 8. Для продолжения работы EIFT в таком случае вам необходимо указать корректные учетные данные. При загрузке с MacOS X авторизация hdiutil больше не требуется.

Рис. 8: Окно авторизации hdiutil

После загрузки рамдиска доступны следующие команды (см. Листинг 2)

```
Commands:             | Parameters:           | Description:

--Ramdisk commands:
mount         | [-u]                  | Mount relevant filesystems
loadnfcd      | [-u]                  | Mount relevant filesystems, start ncfd (Sometimes
↪  needed for unlock)
sepboot       | [-u]                  | Mount relevant filesystems, boot SEP
unlockdata    | [-u -s]               | Mount relevant filesystems, boot SEP and unlock Data
↪  partition

--Available after mount (non-SEP devices only):
dumpkeys      | -o -p [-u -a -np]     | Dump device keys required for decrypting HFS dump
passcode      | [-u -a]               | Bruteforce device passcode

--Available after unlockdata:
keychain      | -o [-u]               | Decrypt keychain
tar           | -o [-u]               | Dump a tar image of the device (Default: Data only)
                [--system]            ( Dump system partition        )
                [--data]              ( Dump data partition          )

--Always available:
diskdump      | -o [-u]               | Dump a raw disk image of the device (Default:
↪  HFS=Data/APFS=everything)
                [--system]            ( Dump system partition        )
                [--data]              ( Dump data partition          )
fsck_hfs      | [-u] <system or data> | Run HFS filesystem check
                --system              ( Run: fsck_hfs /dev/disk0s1s1 )
                --data                ( Run: fsck_hfs /dev/disk0s1s2 )
```

Листинг 2: Доступные команды в режиме рамдиска (список команд можно получить также, набрав в терминале `EIFT_cmd ramdisk -h`)

Если устройство поддерживает *Полное HFS извлечение*, переходите к Разделу 4.1.5.
Полный список устройств, для которых доступна функция *Полного HFS извлечения* и подбора пароля можно найти в Разделе B.2.

Если *Полное HFS извлечение* устройством не поддерживается, то последовательность действий следующая:

1. Смонтировать файловую систему (`EIFT_cmd ramdisk mount`)
2. Загрузить SEP (`EIFT_cmd ramdisk sepboot`)
3. Разблокировать пользовательские данные (`EIFT_cmd ramdisk unlockdata`)

Первые два шага необязательны, они выполняются автоматически при запуске команды `unlockdata`. Тем не менее, они могут оказаться полезными для выяснения причины ошибки, если таковая произошла при разблокировке пользовательских данных. Если вы столкнулись с проблемой на одном из этих шагов, пожалуйста, обратитесь в тех-поддержку.

4.1.3 Разблокировка пользовательских данных

Перед извлечением файловой системы и связки ключей необходимо произвести разблокировку устройства. Для разблокировки выполните следующую команду:

```
EIFT_cmd ramdisk unlockdata
```

EIFT попросит вас также ввести пароль от устройства; после ввода нажмите **ENTER**.
ВНИМАНИЕ: Количество попыток ввода пароля ограничено, аналогично тому, как это происходит на экране устройства! Для предотвращения полной блокировки количество попыток ограничено 7 попытками.

```
unlockdata version: 0.68-4ae02facef2348019917bde0412036258e9d3dd9-RELEASE
[+] InitUserClient
[+] KeyBagGetSystem
Keybag wasn't inited, initing now...
[+] KeyBagGetSystem
[*] Got system keybag id=1
[!] Device is locked
[-] Failed unlock attempts: 1
Please input device passcode: 0000
[+] unlockdevice with passcode '0000'
[*] Data volume is unlocked!
```

Листинг 3: Использование утилиты `unlockdata` при физическом извлечении

Важно: iPhone 8 / iPhone X с установленной iOS 14 & iOS 15 не могут быть разблокированы, если установлен пароль. Перед физическим извлечением необходимо убрать пароль.

Важно: iPhone 8 / iPhone X с установленной iOS 16 не получится разблокировать, если пароль был установлен хотя бы раз после восстановления устройства (restore). Снятие пароля в этом случае не поможет.

4.1.3.1 BFU-разблокировка

Если пароль неизвестен, можно провести так называемую BFU-разблокировку и получить ограниченное число файлов. Количество доступных файлов будет отличаться в зависимости от версии установленной iOS; но в общем случае: чем новее версия, тем меньше файлов доступно. Более подробная информация о BFU-извлечении доступна в нашем блоге blog.elcomsoft.com.
Перед acbfu-извлечением необходимо так же вызвать утилиту `unlockdata`, чтобы подготовить файловую систему.
Есть два способа. Первый: напрямую вызвать `unlockdata` с аргументом `'--no-unlock'`:

```
EIFT_cmd ramdisk unlockdata --argument='--no-unlock'
```

Второй способ: просто не вводить пароль, когда утилита об этом попросит: (см. Листинг 4).

```
unlockdata version: 0.68-4ae02facef2348019917bde0412036258e9d3dd9-RELEASE
[+] InitUserClient
[+] KeyBagGetSystem
Keybag wasn't inited, initing now...
[+] KeyBagGetSystem
[*] Got system keybag id=1
[!] Device is locked
[-] Failed unlock attempts: 1
Please input device passcode:
[!] Got an empty passcode. Canceling device unlock
[!] Data volume is locked!
```

Листинг 4: BFU-разблокировка при физическом извлечении

4.1.3.2 Seashat-разблокировка

Иногда разблокировка пользовательских данных завершается с ошибкой. В подобных случаях в логах можно найти следующие строки:

```
Device needs 0xe007c015-unlock, retrying...
```

Если вы столкнулись с такой проблемой, то необходима *Seashat-разблокировка*.
В первую очередь, загрузите демон *nfcd* командой

```
EIFT_cmd ramdisk loadnfcd
```

Затем, вызовите утилиту `unlockdata` с параметром `'-s'` (с загруженным общим кешем):

```
EIFT_cmd ramdisk unlockdata -s
```

4.1.4 Извлечение данных

После разблокировки пользовательских данных доступны следующие варианты:

1. Создание tar-архива (Раздел 4.1.4.1)
2. Создание образа диска (Раздел 4.1.4.2)
3. Копирование связки ключей (Раздел 4.1.4.3)

4.1.4.1 Создание Tar-архива

Перед созданием tar-архива файловой системы устройство должно быть разблокировано (см. Раздел 4.1.3). Следующая команда создаст TAR-файл с именем *data.tar*, который будет содержать все файлы пользовательского раздела:

```
EIFT_cmd ramdisk tar -o data.tar
```

Если вам нужен архив системного раздела, то воспользуйтесь такой командой:

```
EIFT_cmd ramdisk tar --system -o system.tar
```

4.1.4.2 Создание образа диска

Вы также можете создать полный образ диска для дальнейшего исследования. Самый простой способ - вызвать следующую команду:

```
EIFT_cmd ramdisk diskdump -o dump.dmg
```

При вызове *diskdump* без каких-либо параметров произойдет одно из следующего:

1. Если формат файловой системы **HFS**, то создастся образ пользовательского раздела/**dev/disk0s1s2** (Data)
2. Если формат файловой систмеы **APFS**, то создастся образ всего диска /**dev/disk0s1** (Full storage)

Для устройств с файловой системой **HFS** возможно передать дополнительный параметр `--system`. В таком случае *diskdump* создаст образ системного раздела.

```
EIFT_cmd ramdisk diskdump --system -o dump.dmg
```

Примечание: Хотя пользовательский раздел и зашифрован, EIFT способен расшифровать его на устройствах, совместимых с *Полным HFS извлечением* (см. Раздел B.6). Остальные разделы не зашифрованы, и их можно анализировать без какой-либо дополнительной обработки.

4.1.4.3 Копирование связки ключей

Перед копированием связки ключей устройство должно быть разблокировано (см. Раздел 4.1.3)!
Следующая команда скопирует связку ключей в файл *keychain.xml*:

```
EIFT_cmd ramdisk keychain -o keychain.xml
```

4.1.5 Полное HFS извлечение

Полное извлечение - это наилучший способ извлечения данных с устройства. Он идеально подходит для программно-технической экспертизы, так как не вносит никаких изменений в данные устройства и имеет несколько других преимуществ по сравнению с другими методами.
Полный список устройств, которые поддерживают *Полное HFS извлечение* можно найти в Разделе B.6.
Процедура полного HFS извлечения состоит из трех этапов: создание полного образа диска (Раздел 4.1.5.1), получение полного набора ключей шифрования (Раздел 4.1.5.2) и расшифровка образа диска (Раздел 4.1.5.3).

4.1.5.1 Создание полного образа диска

В этот разделе описывается, как создать полный образ пользовательского раздела, извлечь BFU-ключи и при необходимости создать образ системного раздела.

Создание образа пользовательского раздела Следующая команда создаст побитовый образ пользовательского раздела:

```
EIFT_cmd ramdisk diskdump -o data.dmg
```

Примечание: если устройство завершило работу некорректно, либо файловая система повреждена, возможна такая ошибка:

```
[Error] [!] Data partition is in an unclean state, please run fsck first to fix potential
    inconsistencies!
    Alternatively pass --unclean, to ignore this and proceed with dumping anyways!
```

В таком случае к команде можно добавить флаг `--unclean`, чтобы проигнорировать ошибку и создать образ в любом случае.

```
  EIFT_cmd ramdisk diskdump --unclean -o data.dmg
```

ВНИМАНИЕ: Если файловая система действительно повреждена, то вам возможно придется рабираться с этим в последующих шагах и вносить исправления в образе вручную. В таком случае рекомендуется создать копию образа и исправлять копию, а не оригинальный образ.
Во время процедуры создания образа данные на устройстве никак не модифицируются!
Опционально можно создать образ системного раздела. Системные разделы немодифицированных устройств не содержат ничего интересного, а вот jailbreak и вредоносное программное обеспечение могут вносить значимые изменения в разделе, которые рекомендуется проанализировать.

```
EIFT_cmd ramdisk diskdump --system -o system.dmg
```

Примечание: если устройство завершило работу некорректно, либо файловая система повреждена, возможна ошибка. В таком случае добавьте к команде флаг `--unclean`, чтобы все равно создать образ.

Получение BFU ключей Чтобы вычислить BFU ключи выполните следующую команду:

```
EIFT_cmd ramdisk dumpkeys -n -o keys_bfu.plist
```

О том, как получить полный набор ключей, расказано в следующем разделе.

4.1.5.2 Получение полного набора ключей шифрования

Для доступа к пользовательским данным необходим полный набор ключей шифрования, без него будет расшифровано только ограниченное число данных (BFU). В следующем параграфе описано, как получить полный набор ключей с целевого устройства.
ВАЖНО: Следующие действия необходимо выполнять на том же устройстве, с которого был снят образ диска. Другое устройство не подойдет.
Примечание: Получение набора ключей можно проводить даже если на утройстве произошли какие-то изменения с момента снятия образа диска. Эти изменения никак не повлияют на целостность образа, сделанного ранее.

Извлечение файла systembag.kb Для извлечения системной сумки с ключами (systembag.kb) вам понадобится образ диска (`data.dmg`) и соответсвующие bfu ключи (`keys_bfu.plist`).
Примечание: на этом этапе можно также использовать и полный набор ключей.
Следующая команда извлечет системную сумку с ключами:

```
EIFT_cmd hfstool -i data.dmg -p /keybags/systembag.kb -e -o systembag.kb -k keys_bfu.plist
↪  --no-passcode
```

Команда создаст новый файл под именем `systembag.kb`. Проверьте, что файл расшифрован корректно: он должен начинаться с **bplist**. На UNIX системах это можно проверить следующей командой:

```
head -c 6 systembag.kb | hexdump -C
```

Ожидаемый вывод:

```
00000000  62 70 6c 69 73 74  |bplist|
00000006
```

Как вариант, полученный файл можно попытаться открыть в любом plist-редакторе. Если файл открывается, то расшифровка прошла успешно.

Подбор пароля Если пароль неизвестен, на этом этапе его можно попробовать подобрать.
Для подбора пароля вам понадобится системная сумка с ключами (`systembag.kb`) и BFU ключи (`keys_bfu.plist`).
Следующая команда запустит подбор пароля с параметрами по умолчанию:

```
EIFT_cmd ramdisk passcode -b systembag.kb -k keys_bfu.plist
```

Возможно также передать утилите passcode дополнительные аргументы, с помощью параметра `--argument`
Например, см. Листинг 5:

- `EIFT_cmd ramdisk passcode --argument='-y'` (Check 4-digit PINs only)
- `EIFT_cmd ramdisk passcode --argument='-z'` (Check 6-digit PINs only)

Листинг 5: Варианты перебора пароля

Для атаки перебором (bruteforce) утилите `passcode` можно передать длину пароля и набор символов для перебора, пример показан на Листинге 6:

- `EIFT_cmd ramdisk passcode --argument='-l 5 -c abc123'` (Перебор всех 5-значных паролей, составленных из комбинации символов 'abc123')

Листинг 6: Атака перебором

Утилита `passcode` поддерживает также атаку по словарю. Для этого просто передайте путь к файлу со словарем через параметр `-i`, как это показано на Листинге 7:

- `EIFT_cmd ramdisk passcode -i /path/to/dictionary`

Листинг 7: Атака по словарю

Полный список аргументов утилиты passcode можно увидеть, передав ей параметр `'-h'`
(`EIFT_cmd ramdisk passcode --argument='-h'`).

Получение полного набора ключей шифрования Для получения полного набора ключей шифрования потребуется системная сумка с ключами (`systembag.kb`), BFU ключи (`keys_bfu.plist`) и пароль разблокировки. Если пароль неизвестен, вы можете попробовать подобрать его, как описано на предыдущем шаге.
Следующая команда получит полный набор ключей шифрования:

```
EIFT_cmd ramdisk dumpkeys -k keys_bfu.plist -b systembag.kb -o keys.plist -p <PASSCODE>
```

Замените `<PASSCODE>` на пароль разблокировки. Например, есои пароль `0000`, то команда будет выглядеть следующим образом:

```
EIFT_cmd ramdisk dumpkeys -k keys_bfu.plist -b systembag.kb -o keys.plist -p 0000
```

Если пароля на устройстве нет, параметр `-p` можно опустить.
В результате создастся новый файл `keys.plist`, содержащий все ключи, необходимые для расшифровки всех файлов.

4.1.5.3 Расшифровка образа диска

Для расшифровки образа диска (`data.dmg`) потребуется полный набор ключей шифрования (`keys.plist`).
О том, как получить ключи с того же устройства, рассказано в предыдущем разделе.

Расшифровка образа диска Для расшифровки образа диска запустите следующую команду:

```
EIFT_cmd hfstool -i data.dmg -o data_dec.dmg -k keys.plist -j 16 -d
```

Примечание: параметр `-j` устанавливает количество потоков, ипользуемых для расшифровки. На современных компьютерах рекомендуется использовать 16 потоков. Количество потоков можно увеличить или уменьшить в зависимости от вычислительной мощности машины.
По завершении расшифровки утилита выдаст отчет об ошибках (если они есть) и статистику по расшифрованным файлам (см. Листинг 8).

```
------------------------------------ Report ------------------------------------


----------------------------------- Summary ------------------------------------
[.] Took  0 minutes and  0 seconds to unwrap 6107 filekeys
[.] Took  0 minutes and  0 seconds to copy dmg
[-] Took  0 minutes and  2 seconds to decrypt files
[*] Total files 6133
[*] Decryption succeded on 6107 files
[*] Not decrypted 0 files
[*] Not encrypted files 26
```

Листинг 8: Пример отчета утилиты `hfstool`

Примечание: полученный расшифрованный образ можно смонтировать и исследовать с помощью встроенной в MacOS утилиты `hfstool --mount`. На Windows предварительно нужно установить пакет `WinFsp` (включен в дистрибутив EIFT).

Извлечение связки ключей Следующая команда извлечет связку ключей из образа диска:

```
EIFT_cmd tools keychain -i data.dmg -k keys.plist -o keychain.xml
```

Примечание: не имеет значения зашифрованный или расшифрованный образ подается на вход команде. Если образ зашифрован, необходимые файлы расшифруются на лету.
В результате создастся новый файл с именем `keychain.xml`.

4.2 Извлечение на основе jailbreak

В основе второго метода лежит возможность подключиться к устройству через SSH. Для этого необходимо знать пароль пользователя root (если он отличается от стандартного). Подменю `EIFT_cmd jb` предоставляет следующие опции:

- предотвращение блокировки экрана
- копирование связки ключей
- копирование системного/пользовательского раздела в виде tar-архива
- создание образа диска

Важно знать, что в течение всего времени извлечения данных, устройство должно оставаться разблокированным. В противном случае, некоторые файлы или записи связки ключей окажутся недоступными.
Чтобы предотвратить автоблокировку (включена по умолчанию), перейдите в меню *Settings* → *Display & Brightness* → *Auto Lock* и выберите значение *Never*.
Отключение автоблокировки проиллюстрировано на Рисунке 9.

4.2.1 Утилита autolocknever

Иногда отключить автоблокировку напрямую из настроек невозможно (например, когда на устройство установлен профиль управления).
В таких случаях поможет утилита `autolocknever`, которая изменит необходимые конфигурационные файлы напрямую:

```
EIFT_cmd jb autolocknever
```

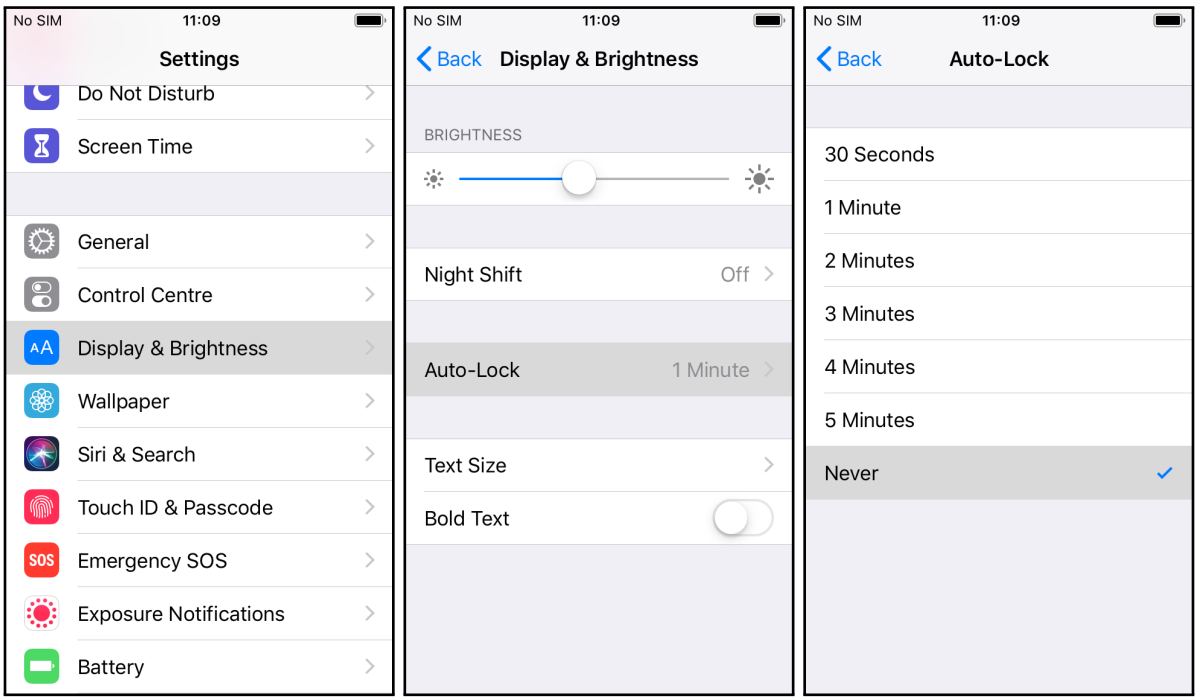


Рис. 9: Отключение автоблокировки в Настройках

4.2.2 Копирование связки ключей

Перед копированием связки ключей устройство должно быть разблокировано. Проследите также, что экран устройства остается разблокированным во время всего процесса копирования. Этого можно добиться, например, запустив какое-либо приложение.
Следующая команда сохранит связку ключей в файле *keychain.xml*:

```
EIFT_cmd jb keychain -o keychain.xml
```

4.2.3 Создание Tar-архива

Перед созданием tar-архива устройство должно быть разблокировано. Проследите также, что экран устройства остается разблокированным во время всего процесса создания архива. Этого можно добиться, например, запустив какое-либо приложение.
Следующая команда создаст tar-архив *data.tar*, содержащий все файлы пользовательского раздела:

```
EIFT_cmd jb tar -o data.tar
```

Если вам нужен tar-архив системного раздела (*system.tar*), то запустите такую команду:

```
EIFT_cmd jb tar --system -o system.tar
```

4.2.4 Создание образа диска

Вы также можете создать полный образ диска для дальнейшего исследования. Самый простой способ - вызвать следующую команду:

```
EIFT_cmd jb diskdump -o dump.dmg
```

При вызове *diskdump* без каких-либо параметров произойдет одно из следующего:

1. Если формат файловой системы **HFS**, то создастся образ пользовательского раздела/**dev/disk0s1s2** (Data)

2. Если формат файловой систмеы **APFS**, то создастся образ всего диска /**dev/disk0s1** (Full storage)

Для устройств с файловой системой **HFS** возможно передать дополнительный параметр `--system`. В таком случае *diskdump* создаст образ системного раздела.

```
EIFT_cmd jb diskdump --system -o dump.dmg
```

Для файловой системы **APFS** доступно только создание образа всего диска.
Вы можете также вручную передать параметры `--system` и `--data` и создать образ раздела **/dev/disk0s1**.

```
EIFT_cmd jb diskdump --system --data -o dump.dmg
```

4.3 Извлечение с помощью агента

В третьем методе извлечения на устройство устанавливается специальное приложение - *Агент*. После установки агент пытается проэксплуатировать уязвимость в ядре, повысить свои привилегии и (в случае удачи) сохранить копию файловой системы и расшифрованную связку ключей на компьютере эксперта. После извлечения приложение-*Агент* можно деинсталлировать с устройства.
Поддерживаемые устройства перечислены в Разделе B.4.
Порядок действий следующий:

1. Установить *Агента* на устройство
2. Извлечь данные (tar-архив файловой системы, связку ключей, образ диска, данные и метаданные)
3. Деинсталлировать *Агента*

Для установки агента вам понадобится действующий аккаунт Apple. Mac-версия EIFT работает как с обычными аккаунтами, так и с аккаунтами разработчика. При установке *Агента* вам потребуется ввести пароль от Apple - аккаунта (обычный, не одноразовый) и пройти двухфакторную аутентификацию с помощью кода, присланного на доверенное устройство.
Важно: к аккаунту разработчика можно привязать до 100 устройств каждого типа (например, 100 iPhone и 100 iPad), к обычному же аккаунту только три. Тем не менее, после достижения предела можно вручную отвязать лишние устройства.

4.3.1 Установка агента

Перед установкой агента необходимо установить отношение доверия между устройством и компьютером (см. Раздел 4.4.1)
Для установки агента вызовите следующую команду (см. Листинг 9).
Вам потребуется ввести e-mail и пароль от appleID. Если ваш аккаунт привязан к организации, то EIFT попросит уточнить, какой именно аккаунт использовать: индивидуальный или корпоративный. В примере из листинга аккаунт пользователя *Mr. Forensic Expert* привязяан также к организации *Elcomsoft s.r.o.*.
Если у вас есть доступ к платному аккаунту разработчика, то для установки *Агента* предпочтительнее использовать его, так как при этом не потребуется интернет-соединения. В примере из листинга был выбран корпоративный аккаунт.
Примечание: Для проверки сертификата разаработчика с бесплатным аккаунтом больше не требуется интернет-соединение. Добавлена поддержка полностью офлайновой установки для всех учётных записей разработчика и независимо от их возраста.
Примечание: Если для аккаунта включена двухфакторная аутентификация, то вам также может потребоваться ввести одноразовый код, полученный либо через SMS, либо через доверенное устройство.

```
./EIFT_cmd agent install
Started logging Thread!
[*] Looking for device...
[*] Installing Agent to device <udid>
Provision IPA...
Getting authinfo...
Please input AppleID email: <type email here>
Please input AppleID password: <type password here>
Authentication success!
Can proceed without two step verification.
        [0] ABCDEFGHIJ - Mr. Forensic Expert - active - Individual
        [1] KLMNOPQRST - Elcomsoft s.r.o. - active - Company/Organization

Getting dev teams...
Please select id of the developer team you want to use: <select number here>
Setting selected devteam id KLMNOPQRST ...
Signing IPA...
Installing IPA...
Install: CreatingStagingDirectory (5%)
Install: ExtractingPackage (15%)
Install: InspectingPackage (20%)
Install: TakingInstallLock (20%)
Install: PreflightingApplication (30%)
Install: InstallingEmbeddedProfile (30%)
Install: VerifyingApplication (40%)
Install: CreatingContainer (50%)
Install: InstallingApplication (60%)
Install: PostflightingApplication (70%)
Install: SandboxingApplication (80%)
Install: GeneratingApplicationMap (90%)
Process has been completed.
Done
```

Листинг 9: EIFT Процесс установки *Агента* в EIFT

4.3.2 Копирование связки ключей

Перед копированием связки ключей убедитесь, что приложение *Агента* открыто и работает. Держите приложение открытым в течение всего процесса извлечения, в том числе и между различными операциями (копированием связки ключей, созданием tar-архива, образа диска, извлечения данных и метаданных). Следующая команда сохранит связку ключей под именем *keychain_ UDID_ timestamp.xml* в папке, путь к которой указан в параметре `-o`:

```
EIFT_cmd agent keychain -o /path/to/folder/
```

Примечание: метка времени (timestamp) в имени файла указывается в формате GMT.

4.3.3 Создание Tar-архива

Перед созданием tar-архива убедитесь, что приложение *Агента* открыто и работает. Держите приложение открытым в течение всего процесса извлечения, в том числе и между различными операциями (копированием связки ключей, созданием tar-архива, образа диска, извлечения данных и метаданных). Следующие команды создадут tar-архив *image_ UDUD_ timestamp.tar*, содержащий все файлы пользовательского раздела в папке, путь к которой указан в параметре `-o`:.

```
EIFT_cmd agent tar -o /path/to/folder/
```

или

```
  EIFT_cmd agent tar --user -o /path/to/folder/
```

Для архивирования только системного раздела воспользуйтесь командой:

```
EIFT_cmd agent tar --system -o /path/to/folder/
```

Для архивирования обоих разделов воспользуйтесь немного другой командой:

```
EIFT_cmd agent tar --full -o /path/to/folder/
```

Примечание: метка времени (timestamp) в имени файла указывается в формате GMT.

4.3.4 Извлечение метаданных

Перед извлечением метаданных убедитесь, что приложение *Агента* открыто и работает. Держите приложение открытым в течение всего процесса извлечения, в том числе и между различными операциями (копированием связки ключей, созданием tar-архива, образа диска, извлечения данных и метаданных).
Для извлечения метаданных файла или папки необходимо указать через параметр `-p` путь файла/папки для извлечения, а через параметр `-o` путь к папке, в которую извлекутся метаданные. Например, следующая команда извлечет метаданные папки */private/var/mobile/Library/Health* и сохранит их в файле *metadata_ UDID_ timestamp.plist* в папке, путь к которой указан в параметре `-o`:

```
EIFT_cmd agent metadata -p /private/var/mobile/Library/Health -o /path/to/folder
```

Параметр `-p` является необязательным, по умолчанию извлекаются метаданные папки */private/var*.
Примечание: метка времени (timestamp) в имени файла указывается в формате GMT.

4.3.5 Извлечение данных

Перед извлечением данных убедитесь, что приложение *Агента* открыто и работает. Держите приложение открытым в течение всего процесса извлечения, в том числе и между различными операциями (копированием связки ключей, созданием tar-архива, образа диска, извлечения данных и метаданных).
Для извлечения данных файла или папки необходимо указать через параметр `-p` путь файла/папки для извлечения, а через параметр `-o` путь к папке, в которую извлекутся метаданные. Например, следующая команда извлечет данные папки */private/var/mobile/Library/Health* и сохранит их в TAR-файле *data_ UDID_ timestamp.tar* в папке, путь к которой указан в параметре `-o`:

```
EIFT_cmd agent extract -p /private/var/mobile/Library/Health -o /path/to/folder
```

Следующая команда извлечет данные файла */private/var/mobile/Media/DCIM/1.JPG* и сохранит их как бинарный файл *data_ UDID_ timestamp.bin* в папке, путь к которой указан в параметре `-o`:

```
EIFT_cmd agent extract -p /private/var/mobile/Media/DCIM/1.JPG -o /path/to/folder
```

Параметр `-p` является необязательным, по умолчанию извлекаются данные папки */private/var*.
Примечание: метка времени (timestamp) в имени файла указывается в формате GMT.

4.3.6 Создание образа диска

К сожалению, эта опция еще недоступна, но мы постараемся добавить ее в следующих версиях EIFT.

4.3.7 Деинсталляция агента

Для деинсталляции *Агента* после извлечения либо удалите значок приложения с экрана стандартным способом, либо выполните следующую команду:

```
EIFT_cmd agent uninstall
```

4.4 Логическое извлечение

Логическое извлечение доступно для всех устройств вне зависимости от установленной на них версии iOS. Тем не менее, количество информации, получаемой при таком методе извлечения, значительно уступает другим методам.
Логическое извлечение позволяет сделать следующее:

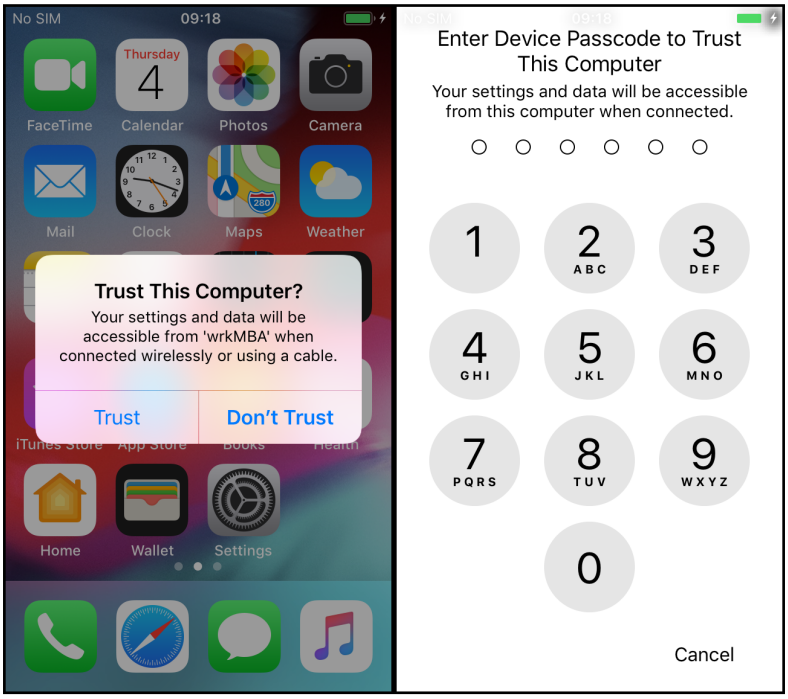
- создать iTunes-подобный бэкап
- создать копии файлов, доступных по протоколу Apple File Conduit (AFC) (например, фотографии)
- скопировать системные логи об ошибках
- скопировать совместно-используемые файлы приложений

4.4.1 Отношение доверия

Прежде чем совершать любое действие в логическом извлечении, между устройством и компьютером необходимо установить отношения доверия (см. Раздел 4.4.1) Можно также использовать уже существующий файл-запись о доверии (pairing record), в таких случаях просто добавляйте параметр с именем файла к каждой команде из последующих разделов (например, `-r record.plist`).
Для установки доверия выполните следующую команду:

```
EIFT_cmd normal pair
```

После этого на устройстве появится диалог, спрашивающий, хотите ли вы доверять компьютеру (Рисунок 10a). Нажмите `'Trust'`, и на следующем экране введите пароль от устройства (Рисунок 10b).



(a) Диалог доверия (b) Экран ввода пароля

Рис. 10: Установка отношения доверия между iOS-устройством и компьютером

4.4.2 Создание резервной копии

Перед созданием резервной копии необходимо убедиться, что между устройством и компьютером установлены отношения доверия (либо передать файл-запись о доверии через параметр). Более подробная информация доступна в Разделе 4.4.1.
Для извлечения максимально возможной информации резервная копия должна быть зашифрована. В незашифрованную резервную копию попадает не вся информация.

4.4.2.1 Проверка пароля на резервную копию

В Листинге 10 показано, как проверить, установлен пароль на резервную копию или нет. Кроме того, в листинге вы также видите, как передать файл-запись о доверии (если отношения о доверии не установлены).

```
./EIFT_cmd normal backuppwcheck -r record.plist
Started logging Thread!
Got device:
Mode: [normal]
BuildVersion: 16H50
DeviceName: iPhone
HardwareModel: N53AP
Paired: YES
PasswordProtected: NO
ProductName: iPhone OS
ProductType: iPhone6,2
ProductVersion: 12.5.4
SerialNumber: <serial number>
udid: <udid>

Loading custom record from=record.plist
Checking backup password...
Backup password is DISABLED
Done
```

Листинг 10: Проверка пароля резервной копии

4.4.2.2 Установка пароля на резервную копию

Следующая команда установит пароль *123* на резервную копию:

```
EIFT_cmd normal backuppwset -p "123"
```

4.4.2.3 Создание резервной копии

Следующая команда создаст резервную копию в текущей директории:

```
EIFT_cmd normal backup -o ./
```

Примечание: если пароль на резервную копию не установлен, то в нее попадут не все данные (см. Раздел 4.4.2.2).

4.4.2.4 Снятие пароля с резервной копии

Следующая команда снимет пароль с резервной копии, при этом понадобится предоставить текущий пароль:

```
EIFT_cmd normal backuppwunset -p "123"
```

4.4.3 Копирование файлов по протоколу afc

Перед копированием файлов по протоколу AFC необходимо убедиться, что между устройством и компьютером установлены отношения доверия (либо передать файл-запись о доверии через параметр). Более подробную информацию можно найти в Разделе 4.4.1.
Следующая команда скопирует все доступные по протоколу AFC данные и сохранит их в файле *afcdump.tar*:

```
EIFT_cmd normal dumpafc -o afcdump.tar
```

4.4.4 Копирование системных журналов об ошибках

Перед копированием системных журналов об ошибках необходимо убедиться, что между устройством и компьютером установлены отношения доверия (либо передать файл-запись о доверии через параметр). Более подробную информацию можно найти в Разделе 4.4.1.
Следующая команда сохранит все системные журналы об ошибках в файле *crashlogs.tar*:

```
EIFT_cmd normal dumpcrash -o crashlogs.tar
```

4.4.5 Копирование совместно-используемых файлов

Перед копированием совместно-используемых файлов необходимо убедиться, что между устройством и компьютером установлены отношения доверия (либо передать файл-запись о доверии через параметр). Более подробную информацию можно найти в Разделе 4.4.1.
Следующая команда сохранит все совместно-используемые файлы приложений в файле *container.tar*:

```
EIFT_cmd normal dumpshared -o container.tar
```

5 Журналирование

По умолчанию EIFT журналирует все проводимые операции, за исключением `ssh`-операций (см. Раздел 6.3).
Журналы сохраняются в папке **/Users/<username>/Elcomsoft/EIFT/logs**, и имеют следующий формат: **EIFT_YYYY_MM_DD-hh_mm_ss.txt**.
Журналы могут оказаться полезными как для самого пользователя (информация о предыдущих извлечениях), так и для тех-поддержки (информация, необходимая для исправления ошибок).
Журналы не шифруются, то есть пользователь может посмотреть, какая именно информация находится в журнале и при необходимости отредактировать ее (например, конфиденциальные данные).
EIFT никогда не отсылает журналы автоматически!
Если пользователь решит отправит журнал в отдел тех-поддержки (для устранения ошибки), то он должен сделать это вручную!

6 Дополнительные опции

Основной функционал EIFT был подробно описан в предыдущих разделах. Этот раздел описывает общую структуры консольной версии EIFT.
Если ваша цель заключается *только* в извлечении данных, то вы можете пропустить этот раздел и перейти напрямую к Разделу 4.1 для физического извлечения, Разделу 4.2 для извлечения на основе jailbreak, к Разделу 4.3 для извлечения с помощью агента или к Разделу 4.4 для логического извлечения.

6.1 Общая структура EIFT_cmd

Запуск команды `EIFT_cmd` без параметров выдаст вам список доступных подменю (см. Рисунок 11).

```
Usage: EIFT_cmd <tool> [args]
Collection of EIFT commandline tools

Tools:
        agent     - Interact with EIFT agent (Agent device acquisition)
        boot      - Boot to EIFT ramdisk (Physical aquisition)
        info      - Display info about connected device
        ssh       - Connecte to a device via SSH
        scp       - Copy file from/to a device via SSH
        jb        - Interact with device in jailreak mode (Jailbroken device acquisition)
        normal    - Interact with device in normal mode (Logical device aquisition)
        ramdisk   - Interact with EIFT ramdisk (Physical device aquisition)
        serial    - Interact with serial console (requires special cable)
        tools     - Collection of EIFT tools
```

Рис. 11: Запуск команды `EIFT_cmd` для получения списка доступных подменю

Для каждого подменю можно также вызвать экран справки, передав параметр `-h` или `--help`.
Например, следующая команда выдаст справку для подменю **agent**:

```
EIFT_cmd agent -h
```

Подменю **agent**, **jb**, **normal** и **ramdisk** выполняют функционал извлечения с помощью агента, извлечения на основе jailbreak, логического и физического извлечения соответсвенно. См. Раздел 4.3, Раздел 4.2, Раздел 4.4 и Раздел 4.1. Подменю **boot** используется для загрузки рамдиска на начальном этапе физического извлечения (см. Раздел 4.1.2).

6.2 EIFT_cmd info

После запуска команды `EIFT_cmd info`, EIFT начнет непрерывно опрашивать порты компьютера и выводить базовую информацию о подключенных iOS-устройствах. Отменить данную операцию можно нажатием клавиш **CTRL-C**.
Пример показан на Рисунке 12.

```
Started logging Thread!
[iDevice normal attached] <udid>
Press CTRL-C to exit
----------------------------
Got device:
Mode: [normal]
BuildVersion: 19B74
DeviceName: iPhone
HardwareModel: N66AP
Paired: YES
PasswordProtected: NO
ProductName: iPhone OS
ProductType: iPhone8,2
ProductVersion: 15.1
SerialNumber: <serial number>
udid: <udid>

Press CTRL-C to exit
```

Рис. 12: Результат запуска команды `EIFT_cmd info` для получения информации о подключенных iOS-устройствах

Чтобы получить более подробную информацию, передайте команде параметр `-a` или or `--all`.
Вывод команды можно сохранить в отдельном файле, передав параметр `-s` или `--save`, например:

```
EIFT_cmd info -a -s           #Сохранить в текущей папке
EIFT_cmd info -a --save=/tmp  #Сохранить в папке'/tmp'
```

В результате в указанной папке создастся несколько файлов: `udid_<UDID>_info.txt`, `udid_<UDID>_info.plist`, `udid_<UDID>_apps.plist`.

6.3 EIFT_cmd ssh

Если на устройстве работает SSH-сервер (например после загрузки рамдиска или после установки jailbreak), то команда `EIFT_cmd ssh` позволит подключится к устройству. Номер порта (если он отличается от стандартного), можно передать параметром `-p` или `--port`.
В режиме рамдиска аутентификация пользователя производится автоматически; в других случаях (например на устройствах с jailbreak) необходимо вручную ввести пароль пользователя root.
Примечание: По умолчанию пароль пользователя root - *alpine*.
Возможно также копировать файлы с устройства и на устройства с помощью протокола Secure Copy (SCP). Синтаксис команды следующий: `EIFT_cmd` *mode source destination*, где *mode* принимает значение либо **scpd** для передачи файлов с компьютера на iOS-устройство, либо **scph** для передачи с устройства на компьютер.
Примеры показаны на Рисунке 13.

```
EIFT_cmd scph /var/mobile/Media/DCIM/100APPLE/IMG_0001.MOV .
```

(a) Копирование файлов с устройства на компьютер

```
EIFT_cmd scpd ./tar /usr/bin
```

(b) Копирование файла с компьютера на устройство

Рис. 13: Примеры копирования файлов с/на iOS-устройство

Примечание: следующей командой можно выключить устройство

```
EIFT_cmd ssh halt
```

6.4 EIFT_cmd serial

iOS-устройства также поддерживают вывод информации в Universal Asynchronous Receiver-Transmitter (UART)-интерфейс, часто называемый еще *серийным* интерфейсом. Для взаимодействия по этому интерфейсу необходим специальный серийный кабель.
Более подробную информацию о серийных кабелях можно найти в **этой** статье.
Для обычного извлечения серийный кабель необязателен, тем не менее с его помощью можно получить дополнительную полезную информацию для отладки ошибок. Для получения подробных инструкции по пользованию серийным кабелем обратитесь в тех-поддержку.
Обычно для перенаправления вывода на серийный интерфейс достаточно такой команды:

```
EIFT_cmd serial
```

Тем не менее, если автоопределение интерфейса не сработало, либо серийных интерфейсов несколько, то необходимо вручную указать нужный интефейс. Например, если серийный кабель подключен к интерфейсу **/dev/tty.usbserial-A403B2AD**, то команда будет выглядеть следующим образом:

```
EIFT_cmd serial /dev/tty.usbserial-A403B2AD
```

6.5 EIFT_cmd tools

Подменю `EIFT_cmd tools` содержит набор дополнительных утилит (см. Рисунок 14):

```
Started logging Thread!
[Error] Missing command
Usage: EIFT_cmd tools <command> [parameters]
Collection of EIFT tools

  -h, --help                Displays this helpscreen
  -e, --ecid <ECID>        Target device ECID
  -i, --input <path>        Input argument
  -o, --output <path>        Output argument
  -k, --keys <path>        Path to device keys.plist
  -w, --wait                Wait for device

Commands:             | Required parameters   | Description
        keychain      | -i -o -k              | Decrypt keychain
        autobootTrue  | [-e] [-w]             | Kick out of recovery loop
        autobootFalse | [-e] [-w]             | Make device not boot past recovery
        grabkeys      | -o         <ipsw>     | Get keys for ipsw and store them in directory specified
```

Рис. 14: Список утилит в подменю `EIFT_cmd tools`

Утилита **keychain** позволяет расшифровать связку ключей в том случае, если файл *keychain-2.db* скопирован с устройства вручную. Для расшифровки также необходим файл с ключами шифрования *keys.plist* (см. Раздел 4.1.5.2).
Замечание: После checkm8 эксплойта устройство по умолчанию всегда будет грузится в Recovery-режиме. Следующая команда выведет устройство из этого режима:

```
EIFT_cmd tools autobootTrue
```

Замечание: ручная расшифровка связки ключей официально не поддерживается!

6.6 Аппаратная реализация checkm8

В некоторых случаях (например, в случае с iPhone 4S) программная реализация эксплойта checkm8 не сработает. Для успешной работы на подобных устройствах эксплойту необходим больший контроль над USB-стеком, чтобы посылать на устройство особым образом сформированные USB-пакеты. Более того,

аппаратная реализация checkm8 менее подвержена ошибкам, чем программная версия. Полный список устройств, которые поддерживают аппаратную реализацию, можно найти в Разделе B.7.
Далее описывается аппаратная реализация эксплойта checkm8 на основе микроконтроллера **Raspberry Pi Pico**[1].

6.6.1 Необходимое оборудование

Для аппаратной эксплуатации checkm8 потребуется следующее оборудование:

- **Raspberry Pi Pico**[1]
- micro-USB On-The-Go (OTG)-адаптер
- стандартный micro-USB кабель
- Прошивка **picom8.uf2** для Raspberry Pi Pico от ElcomSoft
- источник питания 5V и провода
- (необязательно) паяльное оборудование

6.6.2 Установка прошивки

Перед первым использованием необходимо установить прошивку **picom8.uf2** на Raspberry Pi Pico. Установку достаточно произвести только один раз, так как прошивка остается в памяти микроконтроллера (даже после перезагрузки) до тех пор, пока пользователь вручную не сбросит память.
Ниже описана процедура установки прошивки на микроконтроллер.
Ниже описана процедура установки прошивки на микроконтроллер. Сначала подсоедините Raspberry Pi Pico к компьютеру в режиме *загрузчика*. Для этого, удерживая кнопку **BOOTSEL** зажатой, подключите микроконтроллер к компьютеру с помощью micro-USB кабеля. На Рисунке 15 изображено местоположение кнопки **BOOTSEL** и micro-USB кабель.

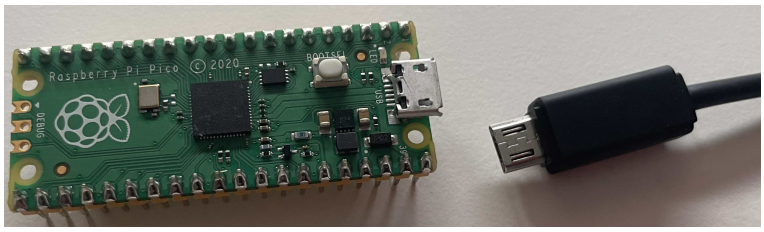


Рис. 15: Raspberry Pi Pico и micro-USB кабель

Если все сделано правильно, компьютер обнаружит устройство как съемный носитель **RPI-RP2**. Для установки прошивки просто перетащите файл **picom8.uf2** в корневую папку **RPI-RP2** (см. Рисунок 16). После этого устройство отсоединится и перезагрузится уже с новой прошивкой. При удачной загрузке LED-индикатор будет медленно мигать.

6.6.3 Подготовка оборудования

Рекомендуется подсоединять iOS-устройство к Raspberry Pi Pico через micro-USB OTG-адаптер. Дополнительно к Raspberry Pi Pico и iOS-устройству необходимо подключить источник питания 5V.
Замечание: Хотя для загрузки Raspberry Pi Pico достаточно и меньшего напряжения (3.3V), iOS-устройсву для перехода в DFU-режим этого недостаточно, поэтому убедитесь, что напряжение источника питания минимум 5V.

Для подключения к источнику питания 5V рекомендуется также воспользоваться старым USB-кабелем. Обрежьте кабель и подсоедините контакт №40 (PIN 40) Raspberry Pi Pico к питанию 5V, а контакт №38 (PIN 38) - к заземлению. Провода можно либо напрямую припаять к печатной плате микроконтроллера, либо соединить с помощью пружинных зажимов. Второй вариант проиллюстрирован на Рисунке 17a.
Затем подключите OTG-адаптер к micro-USB порту Raspberry Pi Pico, и микроконтроллер готов к эксплуатации checkm8 (см. Рисунок 17b).

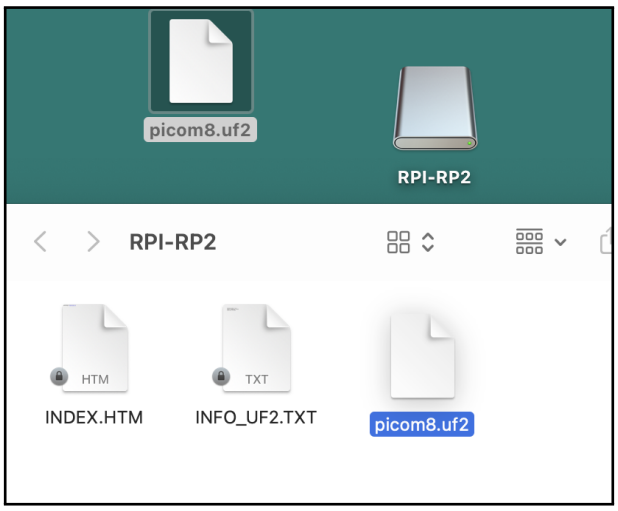
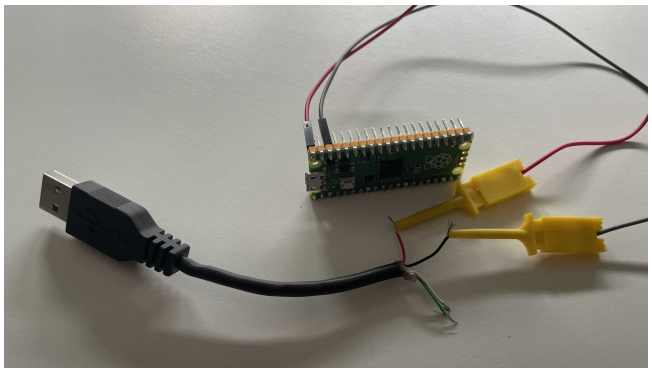
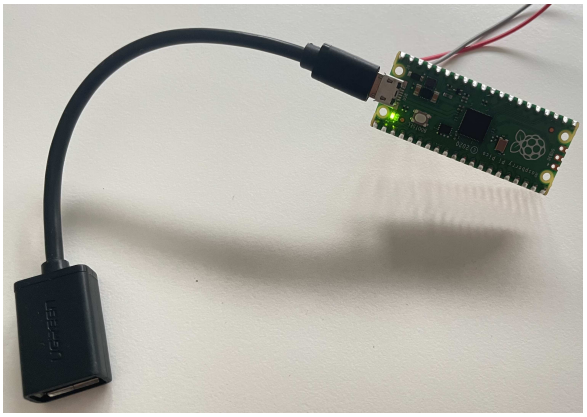


Рис. 16: Установка прошивки **picom8.uf2** на Raspberry Pi Pico



(a) Запитывание через USB с помощью пружинных зажимов



(b) Подключение OTG-адаптера к Pico

Рис. 17: Запитывание Raspberry Pi Pico через USB и подключение OTG-адаптера

6.6.4 Эксплуатация

После установки прошивки (см. Раздел 6.6.2) и запитывания микроконтроллера (см. Раздел 6.6.2) Raspberry Pi Pico готов к эксплуатации iOS-устройств с помощью checkm8. Сначала включите Raspberry Pi Pico. Если вы следовали указаниям предыдущего раздела и для запитывания использовали старый USB-кабель, то сейчас просто подключите этот кабель к источнику питания.
После включения LED-индикатор должен сигнализировать о том, что микроконтроллер находится в режиме **ожидания**. Более подробную информацию о режимах можно найти в Разделе 6.6.5.
Затем подсоедините iOS-устройство к USB-разъему OTG-адаптера. Если к этому моменту утройство еще не находится в DFU режиме, то следует перевести его в этот режим сейчас.
Подробные инструкции о переходе в DFU режим можно найти в Разделе 4.1.1.
Как только Raspberry Pi Pico обнаружит iOS-устройство в DFU режиме, микроконтроллер сразу начнет выполнять checkm8, и LED-индикатор просигнализирует о переходе в режим **эксплуатации**. Эксплуатация занимает примерно 10-15 секунд.
Затем микроконтроллер перейдет в режим **устройство не опознано**. Это ожидаемый этап в работе эксплойта, режим изменится, после того как USB-стек iOS-устройства будет снова проинициализирован. Теперь переподключите iOS-устройство. Если все сделано правильно, то LED-индикатор будет сигнализировать о режиме **завершено**. iOS-устройство теперь можно подключить к компьютеру и приступить к извлечению с помощью EIFT.

6.6.5 Режимы работы и LED-индикатор

Микроконтроллер содержит встроенный LED-индикатор. Прошивка **picom8** использует этот индикатор для обозначения различных режимов работы. Более подробная информация о режимах содержится в

Таблице 1.
Когда к микроконтроллеру не подключено никакое устройство, LED всегда обозначает режим **ожидания**. В противном случае (если LED сигнализирует о другом режиме, либо LED не мигает совсем) попробуйте перезагрузить микроконтроллер. Если проблема сохраняется, то обратитесь в службу поддержки для получения дальнейшей помощи.
Незадолго после подключения поддерживаемого iOS-устройства микроконтроллер перейдет в режим **эксплуатации**. Эксплуатация обычно занимает 10-15 секунд, в течение которых устройство может несколько раз автоматически переподключиться (микроконтроллер периодически будет переходить в режим **ожидания**). В конце эксплуатации LED-индикактор должен сигнализировать о режиме **устройство неопознано**.
Режим **устройство неопознано** еще не говорит об ошибке, а является стандартным и ожидаемым шагом в процессе эксплуатации. Из-за оссобенностей работы эксплойта далее вам потребуется вручную переподключить iOS-устройство, после чего оно будет обнаружено в режиме PWND-DFU.
Микроконтроллер может перейти в режим **устройство не опознано** по двум причинам. Первая: подключено неподдерживаемое устройство. В этом случае, если отключить устройство, LED-индикатор опять перейдет в режим **ожидания**. Вторая причина: подключено поддерживаемое устройство и эксплойт завершился. В таком случае просто переподключите устройство, чтобы понять, успешно ли прошла эксплуатация.
Если эксплойт завершился удачно, то после переподключения iOS-устройства к микроконтроллеру LED-индикатор будет сигнализировать о режиме **завершено**.
Затем просто подключите устройство к компьютеру и приступите к извлечению данных с помощью EIFT.
Если на каком-либо этапе микроконтроллер перешел в режим **ошибки**, попробуйте отсоединить и перезагрузить iOS-устройство.
Если после отключения iOS-устройства микроконтроллер не перешел обрано в режим **ожидания**, то это свидетельствет об ошибке уже в самой прошивке **picom8**. Попробуйте перезагрузить микроконтроллер и обратитесь в службу поддержки, если проблема не устранится.
Если же LED-индикатор вообще не горит или последовательность миганий не похожа ни на одну из описанных в Таблице1, то, вероятно, произошло одно из следующего: программа на прошивке зависла, аварийно завершилась, не загрузилась, либо прошивка была неправильно установлена. Попробуйте перезагрузить микроконтроллер и/или переустановить прошивку согласно Разделу 6.6.2; если проблема не исчезла, то обратитесь в службу поддержки для получения дальнейшей помощи.

Режим	**LED-индикатор**
Ожидание	медленное мигание
Эксплуатация	быстрое мигание
Завершено	1 быстрое мигание, длинная пауза
Устройство не опознано	2 быстрых мигания, длинная пауза
Ошибка	4 быстрых мигания, длинная пауза
Фатальная ошибка	индикатор не мигает

Таблица 1: Индикация режимов микроконтроллера

6.7 Дополнительные утилиты

EIFT поставляется вместе с дополнительными утилитами, которые находятся в папке `tools`. В стандартных случаях, нет необходимости использовать эти утилиты вручную, EIFT делает все автоматически.
Продвинутые пользователи могут прибегнуть к помощи этих утилит для выполнения расширенного функционала. Служба поддержки также может попросить пользователя выполнить определенные действия с утилитами для отладки.

6.8 Управление лицензией

В папке `tools` содержится также утилита `elmc`, предназначенная для управления лицензией.
Следующая команда выведет основную информацию о лицензии и срок ее действия:

```
elmc --info
```

С помощью следующей командв можно обновить лицензию и донгл:

```
elmc --update IOFT-...
```

Примечание: Для выполнения этих операций необходим подключенный донгл и интернет-соединение.

7 Анализ полученных данных

Для анализа tar-архивов, полученных при извлечении, воспользуйтесь программой Elcomsoft Phone Viewer, либо любой другой программой для программно-технической экспертизы, которая умеет анализировать tar-файлы.
Для анализа расшифрованного образа диска (DMG) его сначала нужно подмонтировать к системе. В операционной системе macOS для монтирования достаточно дважды кликнуть на DMG-файл. В операционной системе Windows для монтрования понадобится сторонняя утилита.
Для анализа связки ключей воспользуйтесь программой Elcomsoft Phone Breaker.

8 Устранение проблем

8.1 Не удается перевести устройство в DFU режим

Убедитесь, что устройство подкючено к компьютеру с помощью кабеля lightning - USB-A. Если компьютер имеет только USB-C порты, то воспользуйтесь переходником USB-A - USB-C.
Не подключайте устройство через кабель lightning - USB-C! Вам **никогда** не удастся перевести устройство в DFU-режим, если вы подключите его через кабель lightning - USB-C.
Устройство также **никогда** не перейдет в DFU-режим, если оно не подключено к компьютеру.
Если кнопки на устройстве сломаны, и перевести устройство в режим DFU стандартными способами невозможно, то воспользуйтесь **этой** статьей. В ней описано, как ввести в режим DFU, разобрав устройство и замкнув специальные контакты.

8.2 Разблокировка данных завершается с ошибкой

Иногда разблокировка пользовательских данных может завершиться с ошибкой. Если в журнале EIFT вы нашли следующие строчки, то необходима так называемая *Seashat-разблокировка*:

```
Device needs 0xe007c015-unlock, retrying...
```

Seashat-разблокировка более подробно описана в Разделе 4.1.3.2.

8.3 Не удается найти SEP

Иногда при загрузке SEP может возникнуть такая ошибка:

```
Looking for SEP in PREBOOT volume
[Error] Failed to find SEP on PREBOOT Volume, but it's not on System Volume either!
```

Похожая проблема может возникнуть при поиске ядра:

```
Mounting required filesystems
[Error] Failed to find kernel on PREBOOT Volume, but it's not on System Volume either!
```

По умолчанию EIFT монтирует *реальную* корневую файловую систему. Это необходимо для того, чтобы обнаружить модификации, вызванные jailbreak или вредоносными программами.
Однако, в некоторых случаях на немодифицированных системах ядро или SEP может отсутствовать на реальной корневой файловой системе.
Такое может произойти, к примеру, при предстоящем Over-The-Air (OTA) - обновлении.
Обычно (в iOS 12 и новее), отсутствие этих файлов не критично, так как вместо реальной файловой системы монтируется снимок APFS.
Для разблокировки устройства и доступа к пользовательским данным EIFT требуется загрузить корректно подписанную SEP.
В случае если SEP отсутствует на реальной корневой фаловой системе, можно вместо этого смонтировать снимок. Сразу после загрузки рамдиска выполните следующую команду:

```
EIFT_cmd ramdisk mount --snapshot
```

Примечание: Команда не сработает, если реальная корневая файловая система уже была ранее смонтирована какой-либо командой EIFT (например, `unlockdata`). В таком случае устройство необходимо перезагрузить и загрузить на него рамдиск заного.
После монтирования снимка файловой системы попробуйте снова загрузить SEP, выполнив следующую команду:

```
EIFT_cmd ramdisk sepboot
```

В случае успеха можно продолжить разблокировку и извлечение пользовательских данных.

A Сокращения

AFC Apple File Conduit

AFU After First Unlock (после разблокировки: устройство в данный момент разблокировано)

partial-AFU частичная разблокировка: устройство в данный момент заблокировано, но было разблокировано хотя бы раз после перезагрузки

BFU Before First Unlock(до первой разблокировки)

DFU Device Firmware Update

DMG Образ диска Apple

EIFT Elcomsoft iOS Forensic Toolkit

OTG On-The-Go

OTA Over-The-Air

PCB Печатная плата

SCP Secure Copy

SEP Secure Enclave Processor

SSH Secure Shell

TAR Tape archive

UART Universal Asynchronous Receiver-Transmitter

B Поддерживаемые устройства

B.1 Вход в DFU-режим

B.1.1 Метод 1

Следующие устройства можно перевести в DFU-режим первым способом, как это описано в Разделе 4.1.1.1.

B.1.1.1 iPhone

- iPhone (Original) (iPhone1,1): A1203
- iPhone 3G (iPhone1,2): A1241
- iPhone 3Gs (iPhone2,1): A1303, A1324, A1325
- iPhone 4 (iPhone3,1/iPhone3,2/iPhone3,3): A1332, A1349
- iPhone 4s (iPhone4,1): A1387, A1431
- iPhone 5 (iPhone5,1): A1428, A1429, A1428
- iPhone 5c (iPhone5,4): A1507, A1526, A1529, A1516
- iPhone 5S (iPhone6,1/iPhone6,2): A1453, A1533, A1457, A1518, A1528, A1530

- iPhone 6 (iPhone7,2): A1549, A1586, A1589
- iPhone 6 Plus (iPhone7,1): A1522, A1524, A1593
- iPhone 6s (iPhone8,1): A1633, A1688, A1691, A1700
- iPhone 6s Plus (iPhone8,2): A1634, A1687, A1690, A1699
- iPhone SE (iPhone8,4): A1662, A1723, A1724

B.1.1.2 iPod Touch

- iPod touch 1g (iPod1,1): A1213
- iPod touch 2g (iPod2,1): A1288, A1319
- iPod touch 3g (iPod3,1): A1318
- iPod touch 4g (iPod4,1): A1367
- iPod touch 5g (iPod5,1): A1421, A1509
- iPod touch 6g (iPod7,1): A1574
- iPod touch 7g (iPod9,1): A2178

B.1.1.3 iPad

- iPad 1 (iPad1,1): A1219, A1337
- iPad 2 (iPad2,1/iPad2,2/iPad2,3/iPad2,4): A1395, A1396, A1397
- iPad 3 (iPad3,1/iPad3,2/iPad3,3): A1403, A1416, A1430
- iPad 4 (iPad3,4/iPad3,5/iPad3,6): A1458, A1459, A1460
- iPad 5 (iPad6,11/iPad6,12): A1822, A1823
- iPad 6 (iPad7,5/iPad7,6): A1893, A1954
- iPad mini 1 (iPad2,6/iPad2,7/iPad2,8): A1432, A1454, A1455
- iPad mini 2 (iPad4,4/iPad4,5/iPad4,6): A1489, A1490, A1491
- iPad mini 3 (iPad4,7/iPad4,8/iPad4,9): A1599, A1600, A1601
- iPad mini 4 (iPad5,1/iPad5,2): A1538, A1550
- iPad Air 1 (iPad4,1/iPad4,2/iPad4,3): A1474, A1475, A1476
- iPad Air 2 (iPad5,3/iPad5,4): A1566, A1567
- iPad Pro 1 (iPad6,3/iPad6,4/iPad6,7/iPad6,8): A1584, A1652, A1673, A1674
- iPad Pro 2 (iPad7,1/iPad7,2/iPad7,3/iPad7,4): A1670, A1671, A1701, A1709

B.1.2 Метод 2

Следующие устройства можно перевести в DFU-режим вторым способом, как это описано в Разделе 4.1.1.2.

- iPhone 7 (iPhone9,1/iPhone9,3): A1660, A1778, A1779, A1780, A1853, A1866
- iPhone 7 Plus (iPhone9,2/iPhone9,4): A1661, A1784, A1785, A1786

B.1.3 Метод 3

Следующие устройства можно перевести в DFU-режим третьим способом, как это описано в Разделе 4.1.1.3.

- iPhone 8 (iPhone10,1/iPhone10,4): A1863, A1905, A1906, A1907
- iPhone 8 Plus (iPhone10,2/iPhone10,5): A1864, A1897, A1898, A1899
- iPhone X (iPhone10,3/iPhone10,6): A1865, A1901, A1902, A1903

B.1.4 Метод 4

Следующие устройства можно перевести в DFU-режим четвертым способом, как это описано в Разделе 4.1.1.4.

- AppleTV 3 (AppleTV3,1/AppleTV3,2): A1427, A1469

B.1.5 Метод 5

Следующие устройства можно перевести в DFU-режим пятым способом, как это описано в Разделе 4.1.1.5.

- AppleTV HD (AppleTV5,3): A1625

B.1.6 Метод 6

Следующие устройства можно перевести в DFU-режим шестым способом, как это описано в Разделе 4.1.1.6.

- AppleTV 4K (AppleTV6,2): A1842

B.1.7 Метод 7

Следующие устройства можно перевести в DFU-режим седьмым способом, как это описано в Разделе 4.1.1.7.

- HomePod 1st gen (AudioAccessory1,1): A1639

B.1.8 Method 8 (Apple Watch)

Следующие устройства можно перевести в DFU-режим восьмым способом, как это описано в Разделе 4.1.1.8.

- Apple Watch S0 (Watch1,1, Watch1,2): A1553, A1554
- Apple Watch S1 (Watch2,6, Watch2,7): A1802, A1803
- Apple Watch S2 (Watch2,3, Watch2,4): A1757, A1758, A1816, A1817
- Apple Watch S3 (Watch3,1, Watch3,2, Watch3,3, Watch3,4): A1858, A1859, A1860, A1861, A1889, A1890, A1891, A189

B.2 Физическое извлечение

Физическое извлечение на данный момент доступно для следующих устройств под управлением iOS 8.0 - iOS 16.7.11:

B.2.1 iPhone

- iPhone 3Gs (iPhone2,1): A1303, A1324, A1325
- iPhone 4 (iPhone3,1/iPhone3,2/iPhone3,3): A1332, A1349
- iPhone 4s (iPhone4,1): A1387, A1431
- iPhone 5 (iPhone5,1): A1428, A1429, A1428
- iPhone 5c (iPhone5,4): A1507, A1526, A1529, A1516
- iPhone 5S (iPhone6,1/iPhone6,2): A1453, A1533, A1457, A1518, A1528, A1530
- iPhone 6 (iPhone7,2): A1549, A1586, A1589
- iPhone 6 Plus (iPhone7,1): A1522, A1524, A1593
- iPhone 6s (iPhone8,1): A1633, A1688, A1691, A1700
- iPhone 6s Plus (iPhone8,2): A1634, A1687, A1690, A1699
- iPhone SE (iPhone8,4): A1662, A1723, A1724

- iPhone 7 (iPhone9,1/iPhone9,3): A1660, A1778, A1779, A1780, A1853, A1866
- iPhone 7 Plus (iPhone9,2/iPhone9,4): A1661, A1784, A1785, A1786
- iPhone 8 (iPhone10,1/iPhone10,4): A1863, A1905, A1906, A1907
- iPhone 8 Plus (iPhone10,2/iPhone10,5): A1864, A1897, A1898, A1899
- iPhone X (iPhone10,3/iPhone10,6): A1865, A1901, A1902, A1903

Примечание: физическое извлечение iPhone 8, iPhone 8 Plus, iPhone X под управлением iOS 14 и новее доступно только если на устройстве не установлен пароль.
Примечание: физическое извлечение iPhone 8, iPhone 8 Plus, iPhone X под управлением iOS 16 и новее доступно только если на устройстве **никогда** не был установлен пароль с момента активации.

B.2.2 iPod Touch

- iPod touch 3g (iPod3,1): A1318
- iPod touch 4g (iPod4,1): A1367
- iPod touch 5g (iPod5,1): A1421, A1509
- iPod touch 6g (iPod7,1): A1574
- iPod touch 7g (iPod9,1): A2178

B.2.3 iPad

- iPad 1 (iPad1,1): A1219, A1337
- iPad 2 (iPad2,1/iPad2,2/iPad2,3/iPad2,4): A1395, A1396, A1397
- iPad 3 (iPad3,1/iPad3,2/iPad3,3): A1403, A1416, A1430
- iPad 4 (iPad3,4/iPad3,5/iPad3,6): A1458, A1459, A1460
- iPad 5 (iPad6,11/iPad6,12): A1822, A1823
- iPad 6 (iPad7,5/iPad7,6): A1893, A1954
- iPad mini 1 (iPad2,6/iPad2,7/iPad2,8): A1432, A1454, A1455
- iPad mini 2 (iPad4,4/iPad4,5/iPad4,6): A1489, A1490, A1491
- iPad mini 3 (iPad4,7/iPad4,8/iPad4,9): A1599, A1600, A1601
- iPad mini 4 (iPad5,1/iPad5,2): A1538, A1550
- iPad Air 1 (iPad4,1/iPad4,2/iPad4,3): A1474, A1475, A1476
- iPad Air 2 (iPad5,3/iPad5,4): A1566, A1567
- iPad Pro 1 (iPad6,3/iPad6,4/iPad6,7/iPad6,8): A1584, A1652, A1673, A1674
- iPad Pro 2 (iPad7,1/iPad7,2/iPad7,3/iPad7,4): A1670, A1671, A1701, A1709

B.2.4 AppleTV

- AppleTV 2 (AppleTV2,1): A1378 (not tested)
- AppleTV 3 (AppleTV3,2): A1469
- AppleTV 4 (AppleTV5,3): A1625
- AppleTV 4K (AppleTV6,2): A1842

B.2.5 Apple Watch

- Apple Watch S0 (Watch1,1, Watch1,2): A1553, A1554
- Apple Watch S1 (Watch2,6, Watch2,7): A1802, A1803
- Apple Watch S2 (Watch2,3, Watch2,4): A1757, A1758, A1816, A1817
- Apple Watch S3 (Watch3,1, Watch3,2, Watch3,3, Watch3,4): A1858, A1859, A1860, A1861, A1889, A1890, A1891, A189

B.2.6 HomePod

- HomePod 1st gen (AudioAccessory1,1): A1639

B.3 Извлечение на основе Jailbreak

Извлечение доступно для всех устрйоств с jailbreak, работающих под управлением iOS версии с 8 по 10.

B.4 Извлечение с помощью агента

Извлечение с помощью агента доступно для следующих устройств и версий iOS:

B.4.1 iPhone

- iPhone 5s, iPhone 6, iPhone 6s Plus: **iOS 12.0 - 12.5.7**
- iPhone 6s, iPhone 6S Plus, iPhone SE (1st gen), iPhone 7, iPhone 7 Plus: **iOS 12.0 - 14.3**
- iPhone 8, iPhone 8 Plus, iPhone X: **iOS 12.0 - 16.6.1**
- iPhone Xr, Xs, Xs Max: **iOS 12.0 - 16.6.1**
- iPhone 11, 11 Pro, 11 Pro Max, iPhone SE (2020): **iOS 13.0 - 16.6.1**
- iPhone 12, iPhone 12 Mini, iPhone 12 Pro, iPhone 12 Pro Max: **iOS 13.0 - 16.6.1**
- iPhone 13, iPhone 13 Mini, iPhone 13 Pro, iPhone 13 Pro Max: **iOS 15.0 - 16.6.1**

B.4.2 iPad

- iPad Mini 2 and 3, iPad Air (1st gen): **iOS 12.0 - 12.5.7**
- iPad 5th-7th gen, iPad Pro 1st and 2nd gen: **iOS 12.0 - 14.3**
- iPad Mini 5, iPad Air 3rd gen, iPad Pro 3rd and 4th gen, iPad 8th gen: **iOS 12.0 - 16.6.1**
- iPad 9th gen: **iOS 13.0 - 16.6.1**
- iPad Air 4th gen: **iOS 13.0 - 16.6.1**
- iPad Pro 5: **iOS 14.5 - 16.6.1**
- iPad Mini 6: **iOS 15.0 - 16.6.1**

B.4.3 iPod Touch

- iPod Touch 7th gen: **iOS 12.0 - 14.3**

B.5 Логическое извлечение

Логическое извлечение доступно для всех устройств вне зависимости от установленной версии iOS. Дополнительная поддержка для Apple Watch:

- Apple Watch S0, S1, S2, S3, S4, S5, S6, SE (требуется сторонний адаптер)
- Apple Watch S7 - S10, SE2, Ultra, Ultra 2 (требуется сторонний беспроводной адаптер)

B.6 Полное HFS извлечение

Полное HFS извлечение и побор пароля доступны для следующих устройств под управлением любой версии iOS (см. Раздел 4.1.5)

B.6.1 iPhone

- iPhone 3G (iPhone1,2): A1241
- iPhone 3Gs (iPhone2,1): A1303, A1324, A1325
- iPhone 4 (iPhone3,1/iPhone3,2/iPhone3,3): A1332, A1349
- iPhone 4s (iPhone4,1): A1387, A1431
- iPhone 5 (iPhone5,1): A1428, A1429, A1428
- iPhone 5c (iPhone5,4): A1507, A1526, A1529, A1516

B.6.2 iPod Touch

- iPod touch 2g (iPod2,1): A1288, A1319
- iPod touch 3g (iPod3,1): A1318
- iPod touch 4g (iPod4,1): A1367
- iPod touch 5g (iPod5,1): A1421, A1509

B.6.3 iPad

- iPad 1 (iPad1,1): A1219, A1337
- iPad 2 (iPad2,1/iPad2,2/iPad2,3/iPad2,4): A1395, A1396, A1397
- iPad 3 (iPad3,1/iPad3,2/iPad3,3): A1403, A1416, A1430
- iPad 4 (iPad3,4/iPad3,5/iPad3,6): A1458, A1459, A1460
- iPad mini 1 (iPad2,6/iPad2,7/iPad2,8): A1432, A1454, A1455

B.7 Аппаратная реализация checkm8

Эксплуатация на основе аппаратной реализации checkm8 (см. Раздел 6.6) доступна для следующих устройств:

B.7.1 iPhone

- iPhone 4s (iPhone4,1): A1387, A1431

B.7.2 iPod Touch

- iPod touch 5th gen (iPod5,1): A1421, A1509

B.7.3 iPad

- iPad 2 (iPad2,1/iPad2,2/iPad2,3/iPad2,4): A1395, A1396, A1397
- iPad 3G (iPad3,1): A1416
- iPad 3 (iPad3,2/iPad3,3): A1403, A1430
- iPad mini (iPad2,5/iPad2,6/iPad2,7): A1432, A1454, A1455

B.7.4 Apple TV

- AppleTV 3 (AppleTV3,1): A1427

Список литературы

[1] "Raspberry Pi Pico (RP2040)." https://www.raspberrypi.com/products/raspberry-pi-pico/. Accessed: 2022-03-31.