ООО Фирма «Инфокрипт»

Vitamin-BIO
Руководство администратора

11485466.72.21.12.192 90 01

2026

Содержание

1 Введение 4

1.1 Область применения 4

1.2 Краткое описание возможностей системы 4

2 Назначение и условия применения 5

2.1 Назначение ПО Vitamin-BIO 5

2.2 Условия применения ПО Vitamin-BIO 5

3 Установка программного обеспечения Vitamin-BIO 6

3.1 Подготовка к работе 6

3.2 Настройка 6

3.2.1 Глобальный конфигурационный файл 7

3.2.2 Пользовательский конфигурационный файл 7

4 Удаление ПО Vitamin-BIO 8

Обозначения и сокращения

В настоящем документе используются следующие обозначения и сокращения:

Обозначение	**Описание**
АРМ	Автоматизированное рабочее место
ОС	Операционная система
ПО	Программное обеспечение

1 Введение

1.1 Область применения

Настоящий документ содержит описание основных возможностей и процесса установки программного обеспечения Vitamin-BIO.

Документ предназначен для системного администратора.

1.2 Краткое описание возможностей системы

ПО Vitamin-BIO обеспечивает аутентификацию на основе биометрической информации в браузере, установленном на АРМ пользователя под управлением ОС, основанной на Debian GNU/Linux.

ПО Vitamin-BIO предназначено для работы в автоматическом режиме. Вмешательство пользователя не требуется.

2 Назначение и условия применения

2.1 Назначение ПО Vitamin-BIO

ПО Vitamin-BIO предназначен для аутентификации в браузере семейства Chromium (например, Chrome или SberBrowser) на основе биометрической информации во время работы библиотеки NSS, заменяя штатную аутентификацию в браузере.

2.2 Условия применения ПО Vitamin-BIO

ПО Vitamin-BIO устанавливается на компьютеры, удовлетворяющие следующим аппаратным и программным требованиям:

- На АРМ должен быть установлен модуль TPM 2.
- АРМ должно работать под управлением операционной системы, основанной на Debian GNU/Linux (например, SberOS).
- На АРМ должна быть установлена библиотека биометрической аутентификации, удовлетворяющая требованиям, описанным в Приложении 1.

3 Установка программного обеспечения Vitamin-BIO

Для того чтобы установить клиентскую часть ПО Vitamin-BIO на АРМ, необходимо выполнить команду:

```
sudo dpkg -i vitaminbio_*.deb,
```

где * — номер версии.

В состав программного комплекса «Vitamin-BIO» входят следующие компоненты:

- динамическая библиотека libgetbiopin.so для получения ПИН-кода от сторонней библиотеки биометрической аутентификации;
- динамическая библиотека libic_p11_proxy.so, управляющая работой с браузером, поддерживающая аутентификацию в браузере.
- приложение ic_p11_proxy_ctrl, для получения диагностической информации о работе программного комплекса «Vitamin-BIO».

3.1 Подготовка к работе

Для подготовки к работе необходимо выполнить следующие действия:

1. На АРМ проинициализировать модуль TPM 2 с помощью утилиты tpm2_ptool.
2. Создать токен с для Веб-аутентификации.
3. Зарегистрировать библиотеку ic_p11_proxy в браузере.

При установке ПО Vitamin-BIO формируется и записывается в базу данных список токенов, подходящих для аутентификации. Проверка состояния токенов выполняется при загрузке библиотеки-прокси libic_p11_proxy.so. Если состояние токенов изменилось, библиотека не начинает работу.

Проверка состояния токенов может быть выполнена пользователем с помощью приложения ic_p11_proxy_ctrl.

3.2 Настройка

Для настройки ПО Vitamin-BIO необходимо с помощью приложения ic_p11_proxy_ctrl выполнить следующие действия:

1. Создать базу данных токенов и пользовательский конфигурационный файл, выполнив команду

```
ic_p11_proxy_ctrl -t
```

2. Получить пронумерованный список подходящих для аутентификации токенов, выполнив команду

```
ic_p11_proxy_ctrl -e
```

3. Выбрать в полученном списке токен для БИО-аутентификации и выполнить команду

```
ic_p11_proxy_ctrl -l -n NUMBER
```

указав в качестве NUMBER номер выбранного токена.

Если нужного токена в списке нет, необходимо проверить, выполняются ли требования к токену, который используется для аутентификации:

- Имена токенов уникальны.
- Имя токена должно заканчиваться на суффикс, указанный в параметре suff в глобальном конфигурационном файле (см. подраздел 3.2.1).
- На токене должна быть только одна ключевая пара (закрытый ключ, открытый ключ, сертификат). Все объекты ключевой пары имеют одно и тоже значение идентификатора (атрибут CKA_ID).

ПО для своей работы использует два конфигурационных файла: глобальный и пользовательский.

3.2.1 Глобальный конфигурационный файл

Глобальный конфигурационный файл /etc/infocrypt/ic_p11_proxy/ic_p11_proxy.ini содержит общие настройки, которые разделяются между всеми пользователями:

- путь к целевой библиотеке PKCS11;
- путь к библиотеке libgetbiopin.so;
- путь к ресурсам для библиотеки libgetbiopin.so;
- путь к библиотеке БИО-аутентификации;
- переменные окружения для библиотеки libtpm2_pkcs11.so;
- имя файла журнала;
- имя файла, в котором хранятся сведения о токенах;
- суффиксы для имен токенов в базе данных токенов. Если отсутствует, подходят все токены.

3.2.2 Пользовательский конфигурационный файл

Пользовательский конфигурационный файл ~/.ic_p11_proxy/ic_p11_proxy.ini содержит имя токена, который используется для аутентификации (параметр label).

4 Удаление ПО Vitamin-BIO

Для того чтобы удалить ПО Vitamin-BIO на АРМ под управлением операционной системы SberOS, необходимо выполнить команду

```
sudo dpkg -r vitaminbio_*,
```

где * — номер версии.

Приложение 1

Сторонняя библиотека биометрической аутентификации должна поддерживать следующие форматы и значения:

```
#ifndef SBPASSLESS_BIO_SUGGEST_PIN_H
#define SBPASSLESS_BIO_SUGGEST_PIN_H

#ifdef __cplusplus
extern "C" {
#endif

// флаги для init / iter / restart

// по умолчанию: 0 = аутентификация по геометрии лица, позволяет включить камеру
#define CSP_DISABLE_PRESTART_CAMERA (1) // не позволяет включить камеру
#define CSP_USE_FINGER (2) // аутентификация по отпечатку пальца, должен быть очищен для аутентификации по геометрии лица

// возвращаемые значения для init / iter / restart
   // init or restart
#define CSP_OK (0)
#define CSP_ERROR_FAILED_TO_INIT_BIO (1)
#define CSP_ERROR_FAILED_TO_INIT_CAMERA (2)
#define CSP_ERROR_FAILED_TO_LOAD_BIO_PIN (3)
#define CSP_ERROR_TIMEOUT (4)
#define CSP_ERROR_UNEXPECTED_PIPELINE_ERROR (5)
#define CSP_ERROR_ITER_CALLED_BEFORE_INIT (6)
#define CSP_ERROR_RESTART_CALLED_BEFORE_TIMEOUT (7)
#define CSP_ERROR_FAILED_TO_INIT_FINGER (8) // аутентификация по отпечатку пальца, не позволяет включить сенсор
#define CSP_ERROR_REG_MISMATCH (9) // аутентификация по геометрии лица или отпечатку пальца, зарегистрированные данные не соответствуют предъявленным
#define CSP_ERROR_TIMEOUT_BAD_LIVENESS (10) // отрицательный результат проверки живого присутствия
```

#define CSP_ERROR_BUFFER_INVALID (11) // входной буфер, например, для структуры API, равен нулю

#define CSP_ERROR_BUFFER_TOO_SMALL (12) входной буфер, например, для структуры API, слишком мал

#define CSP_ERROR_HOME_PATH_INVALID (13) // неверный путь к домашнему каталогу пользователя

// iter

#define CSP_BIO_STATUS_INITIALIZING (99) // аутентификация по геометрии лица или отпечатку пальца

#define CSP_BIO_STATUS_OK (100) // аутентификация по геометрии лица или отпечатку пальца (палец приложен к сенсору, получить отпечаток)

// iter, face

#define CSP_BIO_STATUS_NO_FACE (101)

#define CSP_BIO_STATUS_MANY_FACES (102)

#define CSP_BIO_STATUS_BAD_FACE_SCORE (103)

#define CSP_BIO_STATUS_FACE_TOO_SMALL (104)

#define CSP_BIO_STATUS_FACE_OUTSIDE (105)

#define CSP_BIO_STATUS_BAD_HEAD_ANGLE (106)

#define CSP_BIO_STATUS_BAD_LIGHTNING (107)

#define CSP_BIO_STATUS_BLURRED (108)

#define CSP_BIO_STATUS_OVERLAPPED (109)

#define CSP_BIO_STATUS_GENERIC_FAIL (110)

// iter, finger

#define CSP_BIO_STATUS_FINGER_INSERT (111) // аутентификация по отпечатку пальца, палец не приложен к сенсору, приложить палец к сенсору

#define CSP_BIO_STATUS_FINGER_KEEP_REMOVED (112) // аутентификация по отпечатку пальца, палец не приложен к сенсору, пока не прикладывать палец

#define CSP_BIO_STATUS_FINGER_REMOVE (113) // аутентификация по отпечатку пальца, палец приложен к сенсору слишком долго, убрать палец

#define CSP_BIO_STATUS_FINGER_MOVE (114) // аутентификация по отпечатку пальца, палец слишком долго находится на сенсоре в одном положении, подвигать пальцем (или убрать его и приложить по-другому)

// флаги для проверки оборудования

#define CSP_CHECK_HW_FACE (1 << 0) // проверить оборудование для аутентификации по геометрии лица

#define CSP_CHECK_HW_FINGER (1 << 1) // проверить оборудование для аутентификации по отпечатку пальца

// возвращаемые значения для проверки оборудования

#define CSP_CHECK_HW_FACE_DEVICE (1 << 0) // лицо, камера доступна

#define CSP_CHECK_HW_FACE_LIB_BIO (1 << 1) // лицо, BIO SDK доступен

#define CSP_CHECK_HW_FINGER_DEVICE (1 << 2) // палец, устройство доступно

#define CSP_CHECK_HW_FINGER_LIB_FPRINT2 (1 << 3) // палец, библиотека LIBFPRINT-2 доступна

#define CSP_CHECK_HW_FINGER_LIB_DIA (1 << 4) // палец, библиотека DIALIB доступна

#define CSP_CHECK_HW_FINGER_LIB_DIA_LICENSE (1 << 5) // палец, лицензия на DIALIB доступна

#define CSP_CHECK_HW_FINGER_LIB_DIA_SERIAL (1 << 6) // палец, серийный номер продукта для проверки лицензии на DIALIB может быть прочитан

#define CSP_CHECK_HW_FACE_CONFIG (1 << 7) // лицо, аутентификации по геометрии лица разрешена в файле config

#define CSP_CHECK_HW_FINGER_CONFIG (1 << 8) // палец, аутентификация по отпечатку пальца разрешена в файле config

// API, любая точка входа может иметь значение null

```
struct bio_api {
    int size; // не может быть больше размера передаваемого буфера
    int reserved; // отступ, в будущем может использоваться в качестве номера версии или значения переменной flags
// 1.9.0
    int (*init)(const char*, const char*, int flags); // первые два аргумента не используются
    int (*iter)(unsigned int pin_buf_size, char* pin, int flags);
    int (*restart)(unsigned int pin_buf_size, char* pin, int flags);
    void (*free)();
    int (*check_hw)(int flags);
```

```
    int (*set_home_path)(const char* home); // не может иметь значение null, по умолчанию путь к домашнему каталогу пользователя берется из $HOME
};

int crypto_suggest_get_api(bio_api* api, int size); // get available APIs, may fail if buffer pointer is invalid or buffer is too small

#ifdef __cplusplus
}
#endif

#endif  // SBPASSLESS_BIO_SUGGEST_PIN_H
```

Лист регистрации изменений

№№ п/п	Дата	Описание изменения, основание для внесения изменения	Автор
1			
2			
3			