Локальная аутентификация в Astra Linux Смоленск и Рутокен ЭЦП

0 Проверка модели устройства

1. Подключите USB-токен к компьютеру.
2. Для определения названия модели USB-токена откройте **Терминал** и введите команду:

```
$ lsusb
```

В результате в окне Терминала отобразится название модели USB-токена:

```
[dmitrieva@localhost ~]$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 004: ID 0a89:0030 Aktiv Rutoken ECP
Bus 002 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 002 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

Убедитесь, что используете: **Aktiv Rutoken ECP**

1 Доустанавливаем необходимые пакеты с диска

Пуск - Настройки - Менеджер пакетов

через Быстрый фильтр или через поиск находим и отмечаем к установке следующие пакеты:

- libccid
- pcscd
- libpam-p11
- libpam-pkcs11
- libp11-2
- libengine-pkcs11-openssl
- opensc

В Astra Linux SE 1.6 pkcs11 libengine-pkcs11-openssl версии 1.0.2 не совместим с библиотекой librtpkcs11ecp.so. Для корректного функционирования, следует скачать и установить подписанный пакет libengine-pkcs11-openssl1.1 версии 0.4.4-4 для Смоленска 1.6:

libengine-pkcs11-openssl1.1_0.4.4-4_amd64.deb

2 Добавляем библиотеку librtpkcs11ecp.so

Загружаем библиотеку через браузер.

Для 64-битной системы используйте ссылку:

https://download.rutoken.ru/Rutoken/PKCS11Lib/Current/Linux/x64/librtpkcs11ecp.so

Для 32-битной системы используйте ссылку:

https://download.rutoken.ru/Rutoken/PKCS11Lib/Current/Linux/x32/librtpkcs11ecp.so

или через консоль

Пуск - Утилиты - Терминал Fly

Для 64-битной системы используйте:

```
$ wget --no-check-certificate https://download.rutoken.ru/Rutoken/PKCS11Lib/Current/Linux/x64/librtpkcs11ecp.so
```

Для 32-битной системы используйте:

```
$ wget --no-check-certificate https://download.rutoken.ru/Rutoken/PKCS11Lib/Current/Linux/x32/librtpkcs11ecp.so
```

Копируем в системную папку.

Для 32- и 64-битной системы используйте:

```
$ sudo cp librtpkcs11ecp.so /usr/lib
$ sudo chmod 644 /usr/lib/librtpkcs11ecp.so
```

3 Проверяем что Рутокен ЭЦП работает в системе

Пуск - утилиты - Терминал Fly

```
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so -T
```

В случае если увидите вот такую строку, значит все хорошо.

```
Rutoken ECP <no label>
```

4 Считываем сертификат

Проверяем что на устройстве есть сертификат

Пуск - утилиты - Терминал Fly

```
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so -O
```

Если после строчки

```
Using slot 0 with a present token (0x0)
```

нет ничего, значит устройство пустое. Обратитесь к администратору или создайте ключи и сертификат самостоятельно следуя пункту 4.1

Если после строчки

```
Using slot 0 with a present token (0x0)
```

выводится информация о ключах и сертификатах то необходимо считать сертификат

```
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so -r -y cert  --id {id} > cert.crt
```

вместо {id} нужно подставить ID который вы увидите в выводе команды

```
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so -O
```

В случае, если файл cert.crt создан переходим к пункту 5

4.1 Создаем самоподписанный сертификат

Пуск - утилиты - Терминал Fly

генерируем ключевую пару

```
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so --keypairgen --key-type rsa:2048 -l --id 45
```

создаем самоподписанный сертификат

```
$ openssl
OpenSSL> engine dynamic -pre SO_PATH:/usr/lib/x86_64-linux-gnu/engines-1.1/pkcs11.so -pre ID:pkcs11 -pre
LIST_ADD:1  -pre LOAD -pre MODULE_PATH:/usr/lib/librtpkcs11ecp.so
OpenSSL> req -engine pkcs11 -new -key 0:45 -keyform engine -x509 -out cert.crt -outform DER
```

загружаем сертификат на устройство

```
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so -l -y cert -w cert.crt --id 45
```

5 Регистрируем сертификат в системе

Пуск - утилиты - Терминал Fly

Конвертируем сертификат в текстовый формат

```
OpenSSL> x509 -in cert.crt -out cert.pem -inform DER -outform PEM
```

Добавляем сертификат в список доверенных сертификатов

```
$ mkdir ~/.eid
$ chmod 0755 ~/.eid
$ cat cert.pem >> ~/.eid/authorized_certificates
$ chmod 0644 ~/.eid/authorized_certificates
```

6 Настраиваем аутентификацию

Пуск - утилиты - Терминал Fly

```
$ sudo nano /usr/share/pam-configs/p11
```

записываем в файл следующую информацию

```
Name: Pam_p11
Default: yes
Priority: 800
Auth-Type: Primary
Auth: sufficient pam_p11_opensc.so /usr/lib/librtpkcs11ecp.so
```

сохраняем файл, нажимаем Alt + X, а затем Y
после этого выполняем

```
$ sudo pam-auth-update
```

в появившемся окне ставим галку в Pam_p11 и нажимаем OK

7 Проверка

Пуск - утилиты - Терминал Fly

```
$ sudo login
```

введите имя пользователя и в случае если система потребует PIN-код от устройства значит все настроено правильно

8 Блокировка компьютера при извлечении токена

В состав пакета libpam-pkcs11 входит утилита pkcs11_eventmgr, которая позволяет выполнять различные действия при возникновении событий PKCS#11.

Для настройки pkcs11_eventmgr служит файл конфигурации - /etc/pam_pkcs11/pkcs11_eventmgr.conf

Пример файла конфигурации представлен ниже:

```
pkcs11_eventmgr
{
        #
    daemon = true;

        #
        debug = false;

        #
        polling_time = 1;

        #  -
        # - 0
        expire_time = 0;

        #  pkcs11
        pkcs11_module = /usr/lib/librtpkcs11ecp.so;

        #
        #  :
        event card_insert {
                #     (  )
                on_error = ignore ;

                action = "/bin/false";
        }

        #
        event card_remove {
                on_error = ignore;

                #
                action = "fly-wmfunc FLYWM_LOCK";
        }

        #
        event expire_time {
                #     (  )
                on_error = ignore;

                action = "/bin/false";
        }
}
```

После этого добавьте приложение pkcs11_eventmgr в автозагрузку и перезагрузитесь.