Локальная аутентификация в Linux по ГОСТ 2012

включая Astra Linux

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:

```
 $ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 005: ID 0a89:0030 Aktiv Rutoken ECP
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

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

1 Устанавливаем необходимые пакеты

```
sudo apt-get install libccid pcscd opensc openssl libpam-pkcs11
```

2 Добавляем PAM модуль с поддержкой ГОСТ-2012

Загружаем библиотеку через браузер

https://download.rutoken.ru/Rutoken/PAM/1.0.0/x86_64/librtpam.so.1.0.0

или через Терминал

```
 $ wget --no-check-certificate https://download.rutoken.ru/Rutoken/PAM/1.0.0/x86_64/librtpam.so.1.0.0
```

Копируем в системную папку

```
$ sudo cp librtpam.so.1.0.0 /usr/lib/x86_64-linux-gnu/
$ sudo chmod 644 /usr/lib/x86_64-linux-gnu/librtpam.so.1.0.0
```

3 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

```
$ sudo cp librtpkcs11ecp.so /usr/lib/x86_64-linux-gnu/
$ sudo chmod 644 /usr/lib/x86_64-linux-gnu/librtpkcs11ecp.so
```

4 Проверяем что Рутокен ЭЦП работает в системе

В Терминале

```
$ pkcs11-tool --module /usr/lib/x86_64-linux-gnu/librtpkcs11ecp.so -T
```

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

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

5 Считываем сертификат

Проверяем что на устройстве есть сертификат

Откройте Терминал

```
$ pkcs11-tool --module /usr/lib/x86_64-linux-gnu/librtpkcs11ecp.so -O
```

Если после строчки

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

- **выводится информация** о ключах и сертификатах, то необходимо считать сертификат и сохранить на диск. Для этого выполните следующую команду, где вместо {id} нужно подставьте ID-сертификата, который вы увидели в выводе предыдущей команды:

```
$ pkcs11-tool --module /usr/lib/x86_64-linux-gnu/librtpkcs11ecp.so -r -y cert --id {id} --output-file cert.crt
```

В случае, если файл cert.crt создан переходим к пункту 6

- **нет ничего**, значит устройство пустое. Обратитесь к администратору или создайте ключи и сертификат самостоятельно следуя следующему шагу

5.1 Создаем самоподписанный сертификат

Собирайте ветку pkcs11-tool с поддержкой ГОСТ-2012, или используйте релиз OpenSC 0.20.0 или новее

Откройте Терминал

Генерируем ключевую пару с параметрами:

`--key-type`: `GOSTR3410-2012-512:` (ГОСТ-2012 512 бит с парамсетом A), `GOSTR3410-2012-256:A` (ГОСТ-2012 256 бит с парамсетом A)

`--id`: идентификатор объекта (CKA_ID) в виде двузначных номеров символов в hex из таблицы ASCII. Используйте только ASCII-коды печатных символов, т.к. id нужно будет передать OpenSSL в виде строки. Например ASCII-кодам “3132” соответствует строка "12".

Для удобства, можно воспользоваться онлайн-сервисом конвертации строки в ASCII-коды.

```
$ ./pkcs11-tool --module /usr/lib/librtpkcs11ecp.so --keypairgen --key-type GOSTR3410-2012-512:A -l --id 3132
```

создаем самоподписанный сертификат. Для этого сначала надо установить и настроить для работы с Рутокен OpenSSL 1.1 или новее через специальный модуль `rtengine` по инструкции: Установка и настройка OpenSSL для работы с rtengine 0.7.x

Чтобы использовать этот `id` ключевой пары, созданной через pkcs11-tool, в OpenSSL – надо использовать hex-символы из таблицы ASCII, соответствующие этим кодам.

Для удобства, можно воспользоваться онлайн-сервисом конвертации ACSII-кодов в строку.

Например: для '`--id 3132`' в OpenSSL надо указывать "`pkcs11:id=12`".

```
$ openssl req -utf8 -x509 -keyform engine -key "pkcs11:id=12" -engine rtengine -out cert.cer
```

загружаем сертификат на устройство

```
$ pkcs11-tool --module /usr/x86_64-linux-gnu/librtpkcs11ecp.so -l -y cert -w cert.crt --id 3132
```

6 Регистрируем сертификат в системе

Откройте Терминал

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
```

7 Настраиваем аутентификацию

Откройте Терминал

```
$ sudo nano /usr/share/pam-configs/rtpam-gost
```

создайте в файл с содержимым

```
Name: Rutoken PAM GOST
Default: yes
Priority: 800
Auth-Type: Primary
Auth: sufficient /usr/lib/x86_64-linux-gnu/librtpam.so.1.0.0 /usr/lib/x86_64-linux-gnu/librtpkcs11ecp.so
```

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

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

в появившемся окне ставим галку в "Rutoken PAM GOST" и нажимаем OK

8 Проверка

Откройте Терминал

```
$ sudo login
```

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

9 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`

Для различных дистрибутивов Линукс, команда которая вызывает блокировку учетной записи будет отличаться.

Пример файла конфигурации представлен ниже:

```
     pkcs11_eventmgr
     {
         #
         daemon = true;

         #
         debug = false;

         #
         polling_time = 1;

         #  -
         # - 0
         expire_time = 0;

         #  pkcs11
         pkcs11_module = /usr/lib/x86_64-linux-gnu/librtpkcs11ecp.so;

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

             action = "/bin/false";
         }

         #
         event card_remove {
             on_error = ignore;

             #

             #  GNOME
             action = "dbus-send --type=method_call --dest=org.gnome.ScreenSaver /org/gnome/ScreenSaver org.
     gnome.ScreenSaver.Lock";

            #  XFCE
            # action = "xflock4";

            #  Astra Linux (FLY)
            # action = "fly-wmfunc FLYWM_LOCK";
    }

    #
    event expire_time {
        #     (  )
        on_error = ignore;

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

После этого добавьте приложение pkcs11_eventmgr в автозагрузку и перезагрузитесь.