Аутентификация в РЕД ОС 7.3 при помощи ГОСТ ключей на Рутокен ЭЦП

Настройка системы

Перед началом работы, установите следующие пакеты:

```
sudo yum update

sudo yum install ccid opensc pam_pkcs11 gdm-plugin-smartcard p11-kit openssl
```

Проверьте, что у вас установлен openssl версии 1.1 и выше

Скачайте pam модуль и положите его по адресу /usr/lib64

Для этого из папки с pam модулем выполните команду:

```
sudo cp librtpam.so.1.0.0 /usr/lib64/
```

Установите права доступа:

```
sudo chmod 644 /usr/lib64/librtpam.so.1.0.0
```

Загрузите модуль librtpkcs11ecp.so и установите:

```
sudo rpm -i librtpkcs11ecp_X.X.X.X-X_x86_64.rpm
```

Проверяем, что все настроили правильно:

```
pkcs11-tool --module /usr/lib64/librtpkcs11ecp.so -T
```

Далее потребуется скачать сертификат с токена, если его нету, то генерируем его согласно следующему пункту

Генерация сертификата и запись его на Рутокен

Генерируем ключевую пару с параметрами:

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

`--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/lib64/librtpkcs11ecp.so --keypairgen --key-type GOSTR3410-2012-512:A -l --id 3132
```

Скачайте и разархивируйте комплект разработчика

```
unzip rutoken-sdk-latest.zip
```

В sdk находится необходимый нам rtengine. Поместите его в engines-1.1

```
sudo cp /home/user//sdk/openssl/bin/1.1/rtengine-1.1/linux_glibc-x86_64/lib/librtengine.so /usr/lib64/engines-1.1/librtengine.so

sudo mv /usr/lib64/engines-1.1/librtengine.so /usr/lib64/engines-1.1/rtengine.so
```

Зайдите в файл конфигурации openssl.cnf

```
sudo nano /etc/ssl/openssl.cnf
```

В начале файла исправьте следующее:

```
openssl_conf = openssl_def
```

И в конце файла добавьте следующее:

```
[ openssl_def ]

engines = engine_section


[ engine_section ]

rtengine = gost_section


[ gost_section ]

engine_id = rtengine

dynamic_path = /usr/lib64/engines-1.1/rtengine.so

pkcs11_path = /usr/lib64/librtpkcs11ecp.so

default_algorithms = CIPHERS, DIGEST, PKEY, RAND
```

Отредактируйте файл /etc/crypto-policies/back-ends/openssl.config

```
@SECLEVEL=1:aGOST:aGOST01:kGOST:GOST94:GOST89MAC:kEECDH:kRSA:kEDH:kPSK:kDHEPSK:kECDHEPSK:-aDSS:-3DES:!DES:!RC4:!RC2:!IDEA:-SEED:!eNULL:!aNULL:!MD5:-SHA384:-CAMELLIA:-ARIA:-AESCCM8
```

Далее необходимо выполнить следующую команду

```
export OPENSSL_CONF=/etc/ssl/openssl.cnf
```

Создаём самоподписанный сертификат

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`".

Теперь создадим самоподписанный сертификат для наших ключей на токене. Для ключей ГОСТ 512 необходимо выполнить следующую команду

```
openssl req -utf8 -x509 -keyform engine -key "pkcs11:id=12" -engine rtengine -out cert.cer -md_gost12_512
```

Загружаем его на токен

```
pkcs11-tool --module /usr/lib64/librtpkcs11ecp.so -l -y cert -w cert.cer --id 3132
```

Регистрация сертификата в системе

Скачиваем сертификат с токена (если вы пользовались вышеописанной инструкцией для получения сертификата, то ID = 3132)

```
pkcs11-tool --module /usr/lib64/librtpkcs11ecp.so -r -y cert --id 3132 --output-file cert.crt
```

Конвертируем его в PEM 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
```

Настройка аутентификации

Открываем файл /etc/pam.d/system-auth

```
sudo nano /etc/pam.d/system-auth
```

И записываем в самом начале следующую строчку:

```
auth sufficient /usr/lib64/librtpam.so.1.0.0 /usr/lib64/librtpkcs11ecp.so
```

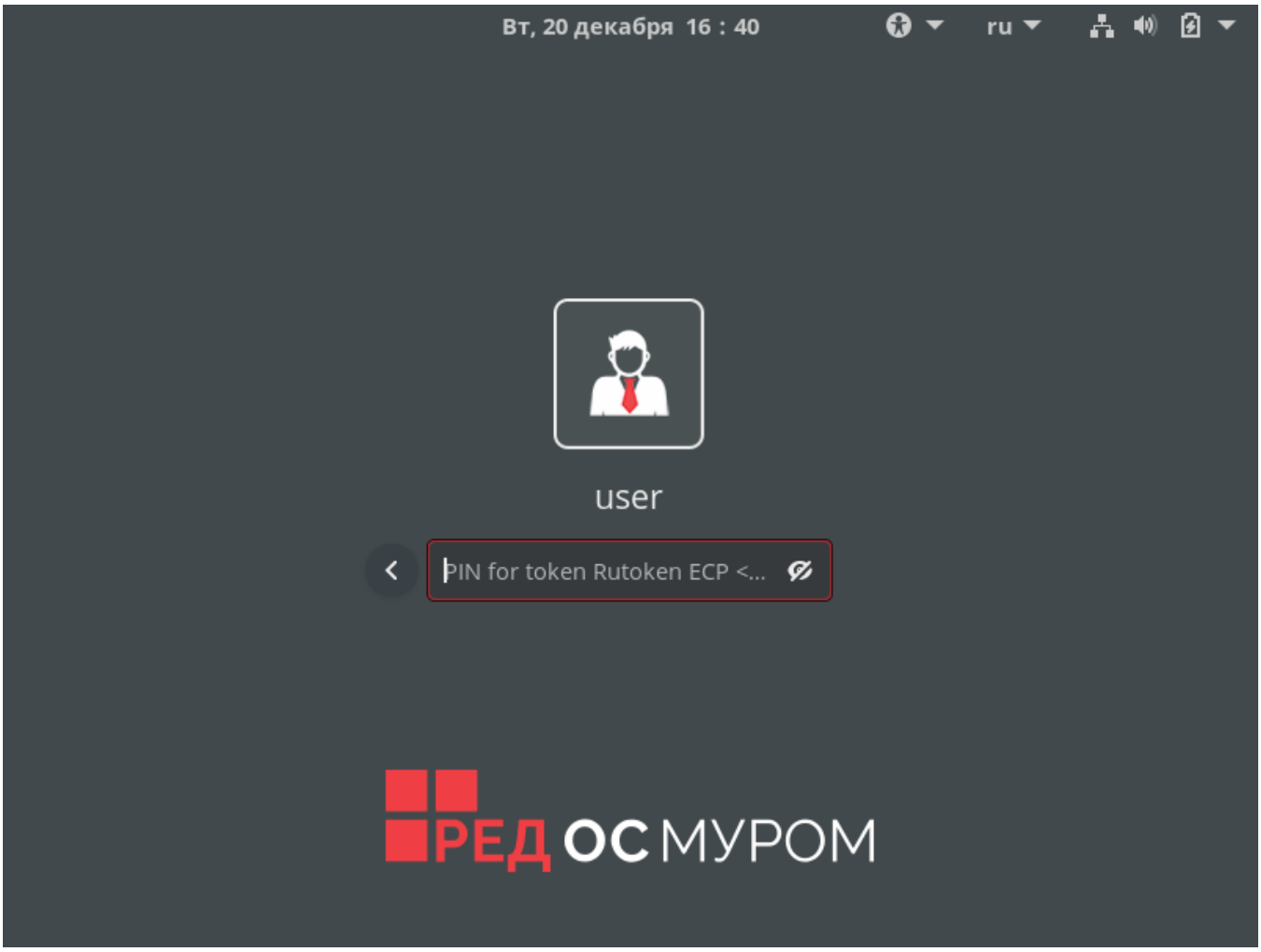
Аналогично делаем с **/etc/pam.d/password-auth.**

Пробуем пройти аутентификацию

```
su <username>
```

Если все прошло успешно, то появится просьба ввести PIN токена.

В окне экрана приветствия аналогично:



Настройка автоблокировки

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

Для того, чтобы аутентификация корректно работала на лок скрине. В настройках pkcs11_eventmgr нужно указать название сервиса, использующегося при аутентификации через лок скрин, чтобы сделать его доверенным. У каждой графической оболочки свое название данного сервиса. Узнать название вашей графической оболочки можно с помощью команды:

Название графической оболочки

```
echo $XDG_CURRENT_DESKTOP
```

Вот список соответствий названий графических оболочек и сервиса, используемого лок скрином. Данный список не является полным.

MATE → mate-screensaver
X-Cinnamon → cinnamon-screensaver
fly → <Отсутствует>
KDE → kde
GNOME → xdg-screensaver

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/lib64/librtpkcs11ecp.so;

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

        action = "/bin/false";
    }

    #
    event card_remove {
        on_error = ignore;

        #
        action = "mate-screensaver-command --lock";
    }

    #
    event expire_time {
        #     (  )
        on_error = ignore;

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

После этого добавьте приложение pkcs11_eventmgr в автозагрузку и перезагрузите компьютер.

Для этого создайте папку ~/.config/autostart. И в данной директории создайте файл ~/.config/autostart/smartcard-screensaver.desktop

```
sudo mkdir ~/.config/autostart

sudo nano ~/.config/autostart/smartcard-screensaver.desktop
```

Содержание файла smartcard-screensaver.desktop должно быть следующим:

```
[Desktop Entry]
Type=Application
Name=Smart Card Screensaver
Comment=Application to lock screen on smart card removal.
Exec=/usr/bin/pkcs11_eventmgr daemon
```