Настройка двухфакторной аутентификации в домене FreeIPA на ОС Альт

Служба обеспечения совместимости <gost@basealt.ru>, Константин Белаш

Version 1.5, 14.10.2022

Оглавление

1. Общая информация

Настоящий документ можно считать практическим руководством для конечного пользователя с методическими рекомендациями по контролю за успешностью выполнения настройки.

Из этого документа станет понятно, как настроить двухфакторную аутентификацию (2ФА) в домене FreeIPA, где в качестве клиента выступает ОС «Альт Рабочая станция 10».

В процессе настройки и тестирования мы убедимся, что токены, содержащие криптографическую информацию, позволяют зарегистрироваться в домене при корректном вводе PIN-кода, без дополнительных способов аутентификации (например, ввода пароля), а также убедиться в обратном, что при вводе некорректного PIN-кода в аутентификации будет отказано. Также в аутентификации будет отказано если сертификат отозван Удостоверяющим Центром (УЦ).

Под криптографической информацией на токене здесь явно подразумеваются: сертификат пользователя, выданный УЦ, открытый и закрытый ключи пользователя.

С теоретической информацией по доменной 2ФА можно ознакомиться в документе «Двухфакторная аутентификация в доменной инфраструктуре ОС Альт. Общая информация.».

2. Базовая конфигурация стенда

Стенд состоит из сервера с контроллером домена FreeIPA и ПК клиента домена.

Контроллер домена FreeIPA (сервер) можно настроить по инструкции https://www.altlinux.org/FreeIPA/Установка_сервера_FreeIPA с интегрированным DNS и интегрированным Удостоверяющим Центром (УЦ). ПК клиента введён в домен FreeIPA согласно инструкции https://www.altlinux.org/FreeIPA/Клиент.

2.1. Обновление сервера FreeIPA и ПК клиента

Настройка выполняется на обновлённой пакетной базе и последнем ядре из репозитория:

```
$ su-
# apt-get update
# apt-get dist-upgrade
# update-kernel
# reboot
```

2.2. Создание домена FreeIPA. Кратко.

```
p10-srv ~ # hostnamectl set-hostname ipa.test3.alt
p10-srv ~ # service ahttpd stop
p10-srv ~ # a2dissite 000-default_https
Site config .conf is already disabled, or does not exist!
p10-srv ~ # a2disport https
Port config https.conf is already disabled, or does not exist!
p10-srv ~ # service httpd2 condreload
```

```
p10-srv ~ # apt-get install freeipa-server freeipa-server-dns
```

```
p10-srv ~ # ipa-server-install -U --hostname=$(hostname) -r TEST3.ALT \
    -n test3.alt -p 12345678 -a 12345678 --setup-dns --forwarder=10.33.33.1 \
    --no-reverse
```

- `TEST3.ALT` — имя разворачиваемого домена
- `--forwarder=10.33.33.1` — внешний DNS-сервер, который будет использовать контроллер домена FreeIPA

2.3. Проверка работы домена FreeIPA

2.3.1. Проверка работы службы Kerberos

```
user@ipa ~ $ kinit admin
Password for admin@TEST3.ALT:
```

```
user@ipa ~ $ klist
Ticket cache: KEYRING:persistent:0:0
Default principal: admin@TEST3.ALT

Valid starting       Expires              Service principal
03/23/2022 14:28:36  03/24/2022 14:28:30  krbtgt/TEST3.ALT@TEST3.ALT
```

2.3.2. Получение общей информации о домене FreeIPA

Перед выполнением команд "**ipa ...**" необходимо получить билет службы Kerberos.

```
user@ipa ~ $ ipa server-show ipa.test3.alt
  Server name: ipa.test3.alt
  Managed suffixes: domain, ca
  Min domain level: 1
  Max domain level: 1
  Enabled server roles: CA server, DNS server, IPA master
```

2.4. Проверка работы ПК клиента в домене FreeIPA

```
user@client ~ $ hostnamectl
   Static hostname: client.test3.alt
         Icon name: computer-vm
           Chassis: vm
      Machine ID: 0421210e5670d77b47b724f761baef2b
         Boot ID: 39f77ae33bb54569993832bd65b3e9b1
  Virtualization: kvm
Operating System: ALT Workstation 10.1 (Autolycus)
     CPE OS Name: cpe:/o:alt:workstation:10.1
          Kernel: Linux 5.10.145-std-def-alt1
    Architecture: x86-64
 Hardware Vendor: QEMU
  Hardware Model: Standard PC Q35 + ICH9, 2009
```

Проверка работы службы Kerberos:

```
user@client ~ $ kinit admin
Password for admin@TEST3.ALT:
```

```
user@client ~ $ klist
Ticket cache: KEYRING:persistent:0:0
Default principal: admin@TEST3.ALT

Valid starting       Expires              Service principal
03/23/2022 15:10:14  03/24/2022 15:10:10  krbtgt/TEST3.ALT@TEST3.ALT
```

2.5. Установка и настройка ПО для работы с токенами. На сервере и на клиенте.

Для работы с токенами необходимо установить следующие пакеты:

```
# apt-get install librtpkcs11ecp libjcpkcs11 isbc-pkcs11 \
    opensc pcsc-lite-ccid pcsc-lite pcsc-tools gnutls-utils \
    libp11 libp11-kit
```

- **librtpkcs11ecp, libjcpksc11, isbc-pkcs11** — библиотеки PKCS#11 вендоров токенов: Актив, Аладдин и ISBC, соответственно;
- **opensc, pcsc-lite-ccid, pcsc-lite, pcsc-tools, gnutls-utils** — утилиты и библиотеки, необходимые для обеспечения работы интерфейсов PC/SC(+CCID) и PKCS#11;
- **libp11** — библиотека для работы с токенами в openssl (openssl engine);
- **libp11-kit** — прокси-библиотека для работы с библиотеками вендоров токенов по протоколу PKCS#11.

Если каких-то пакетов в репозитории нет или с ними выявлены проблемы, необходимо установить их с сайта производителя:

- Актив Рутокен (librtpkcs11ecp): https://www.rutoken.ru/support/download/pkcs/
- Аладдин JaCarta (libjcPKCS11-2): https://www.aladdin-rd.ru/support/downloads/jacarta_client (теперь ещё и в составе «Единого Клиента JaCarta»)
- ISBC ESMART (libisbc_pkcs11_main): https://esmart.ru/download/

2.5.1. Настройка Polkit. На сервере и на клиенте.

По умолчанию Policy Kit позволяет работать с токенами только в активном сеансе (после входа в ОС) и только локальному пользователю. Чтобы можно было работать с токенами на этапе логина доменных пользователей, необходимо поменять все запреты "**>no<**" на разрешения "**>yes<**" в файле `/usr/share/polkit-1/actions/org.debian.pcsc-lite.policy`.

В итоге конфигурационный файл `polkit` примет следующий вид:

```
# cat /usr/share/polkit-1/actions/org.debian.pcsc-lite.policy
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE policyconfig PUBLIC
```

```
 "-//freedesktop//DTD PolicyKit Policy Configuration 1.0//EN"
 "http://www.freedesktop.org/standards/PolicyKit/1.0/policyconfig.dtd">
<policyconfig>
  <vendor>The PCSC-lite Project</vendor>
  <vendor_url>https://pcsclite.apdu.fr/</vendor_url>
<!--  <icon_name>smart-card</icon_name> -->
  <action id="org.debian.pcsc-lite.access_pcsc">
    <description>Access to the PC/SC daemon</description>
    <message>Authentication is required to access the PC/SC daemon</message>
    <defaults>
      <allow_any>yes</allow_any>
      <allow_inactive>yes</allow_inactive>
      <allow_active>yes</allow_active>
    </defaults>
  </action>
  <action id="org.debian.pcsc-lite.access_card">
    <description>Access to the smart card</description>
    <message>Authentication is required to access the smart card</message>
    <defaults>
      <allow_any>yes</allow_any>
      <allow_inactive>yes</allow_inactive>
      <allow_active>yes</allow_active>
    </defaults>
  </action>
</policyconfig>
```

2.5.2. Проверка работы PC/SC интерфейса. На сервере и на клиенте.

Работу интерфейса PC/SC обеспечивает служба **pcscd.service**, которая запускается через одноимённый сокет — **pcscd.socket**.

Включаем pcscd.socket:

```
# systemctl enable --now pcscd.socket
Created symlink /etc/systemd/system/sockets.target.wants/pcscd.socket →
/lib/systemd/system/pcscd.socket.
```

Убеждаемся, что токен виден в выводе утилиты `pcsc_scan`, запущенной с правами пользователя, и однозначно определяется его модель:

```
$ pcsc_scan -r
Using reader plug'n play mechanism
Scanning present readers...
0: Aktiv Rutoken ECP 00 00
1: ESMART Token GOST [ESMART Token] 01 00
2: Aladdin R.D. JaCarta 02 00
```

Если в выводе команды `pcsc_scan -r` вы не видите токенов, то для решения этого вопроса

обратитесь к документу «Методика тестирования токенов» (доступна по запросу в службу обеспечения совместимости).

2.5.3. Проверка работы библиотек вендоров PKCS#11. На сервере и на клиенте.

Библиотеки PKCS#11 являются основой для работы с токенами и будут использоваться всеми компонентами, участвующими в 2ФА. В общем виде работа с библиотеками вендоров токенов выглядит следующим образом:

```
$ pkcs11-tool --module путь_до_библиотеки_вендора --list-token-slots
```

Для каждого токена мы должны получить информацию о нём и о доступных слотах. Если в выводе вышеуказанных команд вы не видите слотов токена, то для решения этого вопроса обратитесь к документу «Методика тестирования токенов» (доступна по запросу в службу обеспечения совместимости).

2.5.3.1. ESMART Token

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/libisbc_pkcs11_main.so \
    --list-token-slots
Available slots:
Slot 0 (0x1): ESMART Token GOST [ESMART Token] 00 00
  token label        : esmart_64
  token manufacturer : ISBC
  token model        : ESMART Token
  token flags        : login required, rng, token initialized, PIN initialized
  hardware version   : 0.0
  firmware version   : 2.4
  serial num         : 206F6060C102
  pin min/max        : 4/8
```

2.5.3.2. Rutoken ECP

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/librtpkcs11ecp.so \
    --list-token-slots
Available slots:
Slot 0 (0x0): Aktiv Rutoken ECP 00 00
  token label        : r_token_ecp2
  token manufacturer : Aktiv Co.
  token model        : Rutoken ECP
  token flags        : login required, rng, SO PIN to be changed, token initialized,
PIN initialized, user PIN to be changed
  hardware version   : 54.1
  firmware version   : 23.2
  serial num         : 3c784ce9
```

```
  pin min/max        : 6/32
```

2.5.3.3. JaCarta-2 SE

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/libjcPKCS11-2.so \
    --list-token-slots
Available slots:
Slot 0 (0x1ffff): Aladdin R.D. JaCarta 00 00
  token label        :  STANDART.SE
  token manufacturer : Aladdin R.D.
  token model        : eToken GOST
  token flags        : login required, rng, token initialized, PIN initialized
  hardware version   : 1.0
  firmware version   : 1.6
  serial num         : 6082004469937675
  pin min/max        : 6/32
Slot 1 (0x2ffff): Aladdin R.D. JaCarta 00 00
  token label        : STANDART.SE
  token manufacturer : Aladdin R.D.
  token model        : JaCarta GOST 2.0
  token flags        : login required, rng, token initialized, PIN initialized, other
flags=0x800
  hardware version   : 1.0
  firmware version   : 2.55
  serial num         : 6082004469937675
  pin min/max        : 6/32
Slot 2 (0x3ffff): Aladdin R.D. JaCarta 00 00
  token label        : STANDART.SE
  token manufacturer : Aladdin R.D.
  token model        : JaCarta Laser
  token flags        : login required, rng, token initialized, PIN initialized
  hardware version   : 1.1
  firmware version   : 1.0
  serial num         : 6082004469937675
  pin min/max        : 4/10
```

2.5.3.4. p11-kit-proxy. На сервере.

Для упрощения работы с тремя разными токенами вместо трёх библиотек вендоров можно использовать одну — **p11-kit-proxy**. Эта библиотека является промежуточным звеном (прокси) между утилитами (например, **pkcs11-tool**) и библиотеками вендоров (например, **librtpkcs11ecp.so**). Таким образом, для работы с разными токенами, для которых есть **модуль pkcs11**, можно использовать одну библиотеку.

Обратите внимание, что разные библиотеки PKCS#11 по разному присваивают идентификатор слота, он указывается после индекса слота в скобках — Slot 2 (**0x3ffff**).

Чтобы использовать **p11-kit-proxy**, необходимо убедиться что необходимые модули PKCS#11 для работы с библиотеками вендоров присутствуют:

```
user@ipa ~ $ ls -l /etc/pkcs11/modules/
total 20
-rw-r--r-- 1 root root 31 Aug 31  2018 isbc.module
-rw-r--r-- 1 root root 45 Mar 24 13:41 jcpkcs11.module
-rw-r--r-- 1 root root 26 Dec  3 19:35 rutokenecp.module
-rw-r--r-- 1 root root 94 Mar 22 19:05 softhsm2.module
```

Все необходимые модули (**isbc.module**, **jcpkcs11.module**, **rutokenecp.module**), для используемых трёх токенов, присутствуют. Проверим работу библиотеки `p11-kit-proxy`:

для Rutoken ECP

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/p11-kit-proxy.so \
    --list-token-slots
Available slots:
Slot 0 (0x30): Aktiv Rutoken ECP 00 00
  token label        : r_token_ecp2
  token manufacturer : Aktiv Co.
  token model        : Rutoken ECP
  token flags        : login required, rng, SO PIN to be changed, token initialized,
PIN initialized, user PIN to be changed
  hardware version   : 54.1
  firmware version   : 23.2
  serial num         : 3c784ce9
  pin min/max        : 6/32
```

для ESMART Token

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/p11-kit-proxy.so \
    --list-token-slots
Available slots:
Slot 0 (0x10): ESMART Token GOST [ESMART Token] 00 00
  token label        : esmart_64
  token manufacturer : ISBC
  token model        : ESMART Token
  token flags        : login required, rng, token initialized, PIN initialized
  hardware version   : 0.0
  firmware version   : 2.4
  serial num         : 206F6060C102
  pin min/max        : 4/8
```

для JaCarta-2 SE

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/p11-kit-proxy.so \
```

```
    --list-token-slots
Available slots:
Slot 0 (0x11): Aladdin R.D. JaCarta 00 00
  token label        :  STANDART.SE
  token manufacturer : Aladdin R.D.
  token model        : eToken GOST
  token flags        : login required, rng, token initialized, PIN initialized
  hardware version   : 1.0
  firmware version   : 1.6
  serial num         : 6082004469937675
  pin min/max        : 6/32
Slot 1 (0x12): Aladdin R.D. JaCarta 00 00
C_GetTokenInfo() failed: rv = unknown PKCS11 error
Slot 2 (0x13): Aladdin R.D. JaCarta 00 00
  token label        : STANDART.SE
  token manufacturer : Aladdin R.D.
  token model        : JaCarta Laser
  token flags        : login required, rng, token initialized, PIN initialized
  hardware version   : 1.1
  firmware version   : 1.0
  serial num         : 6082004469937675
  pin min/max        : 4/10
```

Ошибка в `Slot 1` связана с тем, что данный слот токена `JaCarta-2 SE` работает только с ГОСТовыми протоколами шифрования и для этого слота требуется библиотека `libjckt2`. Эта библиотека присутствует в пакете `libjcpksc11`, но `p11-kit-proxy` ничего о ней не знает, так как ищет эту библиотеку в `/usr/lib64/pkcs11/`, а её там нет.

Чтобы это исправить, создадим символическую ссылку в каталоге `/usr/lib64/pkcs11/` на библиотеку `libjckt2.so`:

```
ipa ~ # cd /usr/lib64/pkcs11/
ipa pkcs11 # ln -s ../libjckt2.so .
```

Убедимся, что все слоты на JaCarta-2 SE отображаются корректно:

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/p11-kit-proxy.so \
    --list-token-slots
Available slots:
Slot 0 (0x11): Aladdin R.D. JaCarta 00 00
  token label        :  STANDART.SE
  token manufacturer : Aladdin R.D.
  token model        : eToken GOST
  token flags        : login required, rng, token initialized, PIN initialized
  hardware version   : 1.0
  firmware version   : 1.6
  serial num         : 6082004469937675
  pin min/max        : 6/32
```

```
Slot 1 (0x12): Aladdin R.D. JaCarta 00 00
  token label        : STANDART.SE
  token manufacturer : Aladdin R.D.
  token model        : JaCarta GOST 2.0
  token flags        : login required, rng, token initialized, user PIN count low, PIN
initialized, other flags=0x800
  hardware version   : 1.0
  firmware version   : 2.55
  serial num         : 6082004469937675
  pin min/max        : 6/32
Slot 2 (0x13): Aladdin R.D. JaCarta 00 00
  token label        : STANDART.SE
  token manufacturer : Aladdin R.D.
  token model        : JaCarta Laser
  token flags        : login required, rng, token initialized, PIN initialized
  hardware version   : 1.1
  firmware version   : 1.0
  serial num         : 6082004469937675
  pin min/max        : 4/10
```

Для токена JaCarta-2 SE мы будем использовать **Slot 2 (0x13)(token model: JaCarta Laser)**. Только этот слот поддерживает протоколы шифрования RSA в данном токене.

Во время настройки 2ФА на клиенте для токена JaCarta-2 SE необходимо будет указать конкретный слот, на котором будет находиться сертификат. Служба sssd (служба аутентификации) будет определять этот слот по метке токена (**token label**). И метки для каждого слота должны быть уникальны. На новых токенах JaCarta метки слотов могут совпадать.

Чтобы установить метку слота его необходимо проинициализировать.

При инициализации слота все данные на нём будут удалены!

Проинициализируем **Slot 2 (0x13)** и установим метку **JC2SE-Laser**:

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/p11-kit-proxy.so \
    --init-token --slot 0x13 --label JC2SE-Laser
Please enter the new SO PIN:
Please enter the new SO PIN (again):
Token successfully initialized
```

- **SO PIN** — PIN-код администратора
- **0x13** — идентификатор слота

После инициализации слота необходимо установить PIN-код пользователя:

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/p11-kit-proxy.so \
```

```
    --init-pin --slot 0x13 --login
Logging in to "JC2SE-Laser".
Please enter SO PIN:
Please enter the new PIN:
Please enter the new PIN again:
User PIN successfully initialized
```

Теперь слоты на JaCarta-2 SE будут выглядеть следующим образом:

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/p11-kit-proxy.so \
    --list-token-slots
Available slots:
Slot 0 (0x11): Aladdin R.D. JaCarta 00 00
  token label        :  STANDART.SE
  token manufacturer : Aladdin R.D.
  token model        : eToken GOST
  token flags        : login required, rng, token initialized, PIN initialized
  hardware version   : 1.0
  firmware version   : 1.6
  serial num         : 6082004469937675
  pin min/max        : 6/32
Slot 1 (0x12): Aladdin R.D. JaCarta 00 00
  token label        : STANDART.SE
  token manufacturer : Aladdin R.D.
  token model        : JaCarta GOST 2.0
  token flags        : login required, rng, token initialized, user PIN count low, PIN
initialized, user PIN locked, other flags=0x800
  hardware version   : 1.0
  firmware version   : 2.55
  serial num         : 6082004469937675
  pin min/max        : 6/32
Slot 2 (0x13): Aladdin R.D. JaCarta 00 00
  token label        : JC2SE-Laser
  token manufacturer : Aladdin R.D.
  token model        : JaCarta Laser
  token flags        : login required, token initialized, PIN initialized
  hardware version   : 1.0
  firmware version   : 1.0
  serial num         : 6082004469937675
  pin min/max        : 4/10
```

3. Настройка на сервере FreeIPA

Создадим трёх пользователей для трёх токенов. Перед работой с утилитой `ipa` необходимо получить билет Kerberos администратора домена:

```
user@ipa ~ $ kinit admin
Password for admin@TEST3.ALT:
```

для Rutoken ECP

```
user@ipa ~ $ ipa user-add rt --first=Active --last=Rutoken
---------------
Added user "rt"
---------------
  User login: rt
  First name: Active
  Last name: Rutoken
  Full name: Active Rutoken
  Display name: Active Rutoken
  Initials: AR
  Home directory: /home/rt
  GECOS: Active Rutoken
  Login shell: /bin/bash
  Principal name: rt@TEST3.ALT
  Principal alias: rt@TEST3.ALT
  Email address: rt@test3.alt
  UID: 1450000001
  GID: 1450000001
  Password: False
  Member of groups: ipausers
  Kerberos keys available: False
```

для JaCarta-2 SE

```
user@ipa ~ $ ipa user-add jc --first=Aladdin --last=Jacarta
---------------
Added user "jc"
---------------
  User login: jc
  First name: Aladdin
  Last name: Jacarta
  Full name: Aladdin Jacarta
  Display name: Aladdin Jacarta
  Initials: AJ
  Home directory: /home/jc
  GECOS: Aladdin Jacarta
  Login shell: /bin/bash
```

```
  Principal name: jc@TEST3.ALT
  Principal alias: jc@TEST3.ALT
  Email address: jc@test3.alt
  UID: 1450000003
  GID: 1450000003
  Password: False
  Member of groups: ipausers
  Kerberos keys available: False
```

для ESMART Token

```
user@ipa ~ $ ipa user-add es --first=ISBC --last=ESMART
---------------
Added user "es"
---------------
  User login: es
  First name: ISBC
  Last name: ESMART
  Full name: ISBC ESMART
  Display name: ISBC ESMART
  Initials: IE
  Home directory: /home/es
  GECOS: ISBC ESMART
  Login shell: /bin/bash
  Principal name: es@TEST3.ALT
  Principal alias: es@TEST3.ALT
  Email address: es@test3.alt
  UID: 1450000004
  GID: 1450000004
  Password: False
  Member of groups: ipausers
  Kerberos keys available: False
```

Обратите внимание, что для созданных пользователей не установлен пароль. Устанавливать его смысла нет, так как аутентификация будет производиться по сертификату на токене.

3.1. УЦ и Key Distribution Center (KDC)

Так как установка сервера FreeIPA производилась с интегрированным УЦ, то на сервере уже присутствует его сертификат, равно как и сертификат KDC. В процессе PKINIT задействована следующая цепочка сертификатов: **сертификат УЦ — сертификат KDC — сертификат клиента**. На сервере используется сертификат УЦ и сертификат KDC. На клиенте используется сертификат УЦ и сертификат клиента (на токене).

Сертификат УЦ на сервере находится в базе NSS. Его нужно экспортировать, чтобы перенести на ПК клиента. Для этого необходимо знать его серийный номер в базе NSS.

Определим серийный номер сертификата УЦ:

```
user@ipa ~ $ ipa cert-find | grep -i \
    'Subject: CN=Certificate Authority,O=TEST3.ALT' -A4
  Subject: CN=Certificate Authority,O=TEST3.ALT
  Issuer: CN=Certificate Authority,O=TEST3.ALT
  Not Before: Tue Mar 22 19:06:38 2022 UTC
  Not After: Sat Mar 22 19:06:38 2042 UTC
  Serial number: 1
```

Экспортируем сертификат УЦ в файл **ca.pem** для последующего переноса на ПК клиента:

```
user@ipa ~ $ ipa cert-show 1 --certificate-out=ca.pem
  Issuing CA: ipa
  Certificate:
MIIEhTCCAu2gAwIBAgIBATANBgkqhkiG9w0BAQsFADA0MRIwEAYDVQQKDAlURVNUMy5BTFQxHjAcBgNVBAMMFU
NlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0yMjAzMjIxNjA2MzhaFw00MjAzMjIxNjA2MzhaMDQxEjAQBgNVBAoM
CVRFU1QzLkFMVDEeMBwGA1UEAwwVQ2VydGlmaWNhdGUgQXV0aG9yaXR5MIIBojANBgkqhkiG9w0BAQEFAAOCAY
8AMIIBigKCAYEAuK1ItgX18UgpSR2jeiSNxJB4TvF3dthUfdSIJgnP5dlXxcOaTs5W/C1SoEcyn3GuuSRuZpbx
hSE0ng2PCGD+jlIsYhdF+fzS72N2XTz3Srki8uoNBFnY1PFt0smDOSOKkHq50W8ymOrLJlqExGoNwpbRoKuwdB
/DxuMYOkycCls6MwlcJKy1T+YzO6zptvueg2l2NGn4oLH1l/mUpTE2kgJe311dn30wqVt2nCoywMut/3d8Ew/7
OuTUriaUykWsKPl6z3zKP2Z5ZJbJA+bjVncvYw5b0LSHyVt7JkaGfBhqH5QdjHyRIahgAk2ZpCNzmN9usLp+gk
PiK2xgYrvjyEACmwDNNFEX/3QkhoVOjZjN7Lh5P0okpXizz7HyGEyDBQjBwn1P2iNvW6IUSbdvv1eGdYrh4IY1
2NlZml0MmI7iaAsQpFgNUFG6yMMbhR/oFte3rSNwBVd2KqgDm9YByZ46AS5g1rcDbN+YOagQFrhVVqdt7yQt+U
SXxiwUD7hTAgMBAAGjgaEwgZ4wHwYDVR0jBBgwFoAU0WzZal5XQlVEDShruukh21AO3n0wDwYDVR0TAQH/BAUw
AwEB/zAOBgNVHQ8BAf8EBAMCAcYwHQYDVR0OBBYEFNFs2WpeV0JVRA0oa7rpIdtQDt59MDsGCCsGAQUFBwEBBC
8wLTArBggrBgEFBQcwAYYfaHR0cDovL2lwYS1jYS50ZXN0My5hbHQvY2Evb2NzcDANBgkqhkiG9w0BAQsFAAOC
AYEAoocqsyLTgCQIrIjv5Q/9s0fZ210wA7xyV2KiXsVS3jS61CleBsqwjV6imyVZto6mcknBt7AdhYUYOv2Fjg
uctWTSFeVVGekkzfmHGyz0lCdCBbX146ZUVygimG88ZXqEHNmbYlZTr1RUJC/AYOnhNZlkqIIi1nR53HTfpuF1
t3tq+XcWpDr1Jq9/mA7hhQl/k+ZO5D0CM/Fx8bisW+vtYYvYb06fRIN+Eeggo58R4ENO7rK7kqJ1X8wQ+nbXSo
/FqslmCW/vth9MFghcvqCoFlw46JPWFPrGle5OydmNqmJIrr1AWPsKPs196hseLj2E0S+hwE5hhLxxgIdMr4Rl
dPjhWBCZ3go5pxuAMYNf0GQMQfpY0AfAtQRT1JDtr86YA51uzbPrZd+U6g/VDPyd6qGs9pF1nMYe6HBxlF7xvD
UDe8whJyiQo/HhyuM8E805/7Mv3U2LLE19u+CwA6VWa/HGvEc9a33zX9tcd0BhPsfjoWyMt97T6kRgxWvVDKz3
  Subject: CN=Certificate Authority,O=TEST3.ALT
  Issuer: CN=Certificate Authority,O=TEST3.ALT
  Not Before: Tue Mar 22 16:06:38 2022 UTC
  Not After: Sat Mar 22 16:06:38 2042 UTC
  Serial number: 1
  Serial number (hex): 0x1
  Revoked: False
```

Сертификат KDC можно просмотреть следующей командой:

```
ipa ~ # ipa-getcert list
...
Request ID '20220322160917':
        status: MONITORING
        stuck: no
```

```
        key pair storage: type=FILE,location='/var/lib/kerberos/krb5kdc/kdc.key'
        certificate: type=FILE,location='/var/lib/kerberos/krb5kdc/kdc.crt'
        CA: IPA
        issuer: CN=Certificate Authority,O=TEST3.ALT
        subject: CN=ipa.test3.alt,O=TEST3.ALT
        expires: 2024-03-22 19:09:17 MSK
        principal name: krbtgt/TEST3.ALT@TEST3.ALT
        key usage: digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment
        eku: id-kp-serverAuth,id-pkinit-KPKdc
        pre-save command:
        post-save command: /usr/libexec/ipa/certmonger/renew_kdc_cert
        track: yes
        auto-renew: yes
```

Обратите внимание на EKU (Extended Key Usage). Здесь указано, что данный сертификат может использоваться для аутентификации на сервере (**id-kp-serverAuth**, <http://oid-info.com/get/1.3.6.1.5.5.7.3.1>), а также в KDC на контроллере домена (**id-pkinit-KPKdc**, <http://oid-info.com/get/1.3.6.1.5.2.3.5>).

Убедимся, что Kerberos PKINIT поддерживается на сервере изначально и готов к работе:

```
user@ipa ~ $ ipa pkinit-status
----------------
1 server matched
----------------
  Server name: ipa.test3.alt
  PKINIT status: enabled
----------------------------
Number of entries returned 1
----------------------------
```

3.2. Выпускаем сертификаты для пользователей

Перед выпуском сертификатов необходимо сгенерировать ключевую пару на токене. Перед генерирацией необходимо определить механизмы шифрования поддерживаемые конкретным токеном. Просмотреть механизмы можно при помощи утилиты `pkcs11-tool`.

Пример для JaCarta-2 SE. В этом токене три слота. Нас интересует третий слот, в котором поддерживается протокол RSA.

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/p11-kit-proxy.so \
    --list-mechanisms --token-label JC2SE-Laser
Supported mechanisms:
  RSA-PKCS-KEY-PAIR-GEN, keySize={1024,2048}, hw, generate_key_pair
  RSA-PKCS, keySize={1024,2048}, hw, encrypt, decrypt, sign, verify, wrap, unwrap
  SHA1-RSA-PKCS, keySize={1024,2048}, sign, verify
  RSA-PKCS-OAEP, keySize={1024,2048}, hw, encrypt, decrypt, wrap, unwrap
  SHA256-RSA-PKCS, keySize={1024,2048}, sign, verify
```

```
  SHA384-RSA-PKCS, keySize={1024,2048}, sign, verify
  SHA512-RSA-PKCS, keySize={1024,2048}, sign, verify
  DES2-KEY-GEN, keySize={128,128}, generate
  DES3-KEY-GEN, keySize={192,192}, generate
  DES3-ECB, keySize={24,24}, encrypt, decrypt
  DES3-CBC, keySize={24,24}, encrypt, decrypt
  DES3-MAC, keySize={24,24}, sign, verify
  DES3-MAC-GENERAL, keySize={24,24}, sign, verify
  MD5, digest
  MD5-HMAC, sign, verify
  SHA-1, digest
  SHA-1-HMAC, sign, verify
  SHA256, digest
  SHA256-HMAC, sign, verify
  SHA224, digest
  SHA384, digest
  SHA384-HMAC, sign, verify
  SHA512, digest
  SHA512-HMAC, sign, verify
  TLS-PRE-MASTER-KEY-GEN, hw, generate
  TLS-MASTER-KEY-DERIVE, hw, derive
  TLS-KEY-AND-MAC-DERIVE, hw, derive
  TLS-MASTER-KEY-DERIVE-DH, hw, derive
  mechtype-0x378, hw, derive
  mechtype-0x500, hw, sign
  AES-KEY-GEN, keySize={16,32}, generate
  AES-ECB, keySize={16,32}, encrypt, decrypt
  AES-CBC, keySize={16,32}, encrypt, decrypt
  AES-MAC, keySize={16,32}, sign, verify
  AES-MAC-GENERAL, keySize={16,32}, sign, verify
```

Слот JC2SE-Laser на JaCarta-2 SE поддерживает генерацию ключевой пары длиной 1024 и 2048 бит (RSA-PKCS-KEY-PAIR-GEN).

3.2.1. Генерируем ключевую пару на токенах

Подключаем токен по одному и генерируем ключевую пару.

для Rutoken ECP

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/p11-kit-proxy.so \
    --pin 12345678 --login --keypairgen --key-type rsa:2048 --id 7001 \
    --label rt_2fa_ipa
Using slot 32 with a present token (0x30)
Key pair generated:
Private Key Object; RSA
  label:      rt_2fa_ipa
  ID:         7001
  Usage:      decrypt, sign, unwrap
  Access:     sensitive, always sensitive, never extractable, local
```

```
Public Key Object; RSA 2048 bits
  label:      rt_2fa_ipa
  ID:         7001
  Usage:      encrypt, verify, wrap
  Access:     local
```

для ESMART Token

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/p11-kit-proxy.so \
    --pin 12345678 --login --keypairgen --key-type rsa:2048 --id 7002 \
    --label es_2fa_ipa
Using slot 0 with a present token (0x10)
Key pair generated:
Private Key Object; RSA
  label:      es_2fa_ipa
  ID:         7002
  Usage:      decrypt, sign, unwrap
  Access:     sensitive, always sensitive, never extractable, local
Public Key Object; RSA 2048 bits
  label:      es_2fa_ipa
  ID:         7002
  Usage:      encrypt, verify, wrap
  Access:     local
```

для JaCarta-2 SE

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/p11-kit-proxy.so \
    --pin 11111111 --login --keypairgen --key-type rsa:2048 --id 7003 \
    --label jc_2fa_ipa --token-label JC2SE-Laser
Key pair generated:
Private Key Object; RSA
  label:      jc_2fa_ipa
  ID:         7003
  Usage:      decrypt, sign, unwrap
warning: PKCS11 function C_GetAttributeValue(ALWAYS_AUTHENTICATE) failed: rv =
CKR_ATTRIBUTE_TYPE_INVALID (0x12)

  Access:     sensitive, always sensitive, never extractable, local
Public Key Object; RSA 2048 bits
  label:      jc_2fa_ipa
  ID:         7003
  Usage:      encrypt, verify, wrap
  Access:     local
```

Утилита `pkcs11-tool` при работе с токенами JaCarta иногда выдаёт предупреждение `"warning: PKCS11 function …"`. Данное предупреждение ошибкой не является и на работоспособность не влияет.

3.2.2. Генерируем запрос на сертификат

Теперь, имея закрытые и открытые ключи, создадим запросы на сертификаты.

Создадим файл расширений (extensions) для выпуска сертификатов пользователей:

```
user@ipa ~ $ cat user_cert.inf
[ req ]
prompt = no
encrypt_key = no

distinguished_name = dn
req_extensions = exts

[ dn ]
commonName = ${ENV::NAME}

[ exts ]
subjectAltName=email:${ENV::NAME}@test3.alt
```

`${ENV::NAME}` — переменная, в которую будем передавать имя пользователя сертификата.

Пример данного файла расширений — https://www.freeipa.org/page/Howto/Client_Certificate_Authentication_with_LDAP

3.2.2.1. openssl engine

Для выпуска сертификата пользователя `openssl` должен «попросить» токен подписать запрос на выпуск сертификата при помощи приватного ключа, который находится на токене, и доступа к которому ни у кого нет, кроме самого токена. Но `openssl` не умеет напрямую обращаться к токенам по протоколу **PKCS#11**. Чтобы `openssl` смог передать токену «просьбу» о подписании запроса на сертификат, необходимо использовать так называемый «**engine**».

Engine — это некий механизм, своего рода прокси, который позволяет `openssl` переложить криптофункции на «чужие плечи». В данном случае на стороннюю библиотеку **PKCS#11**. А уже она, в свою очередь, умея общаться с токеном, сможет донести «просьбу» о подписании запроса на сертификат.

Если в системе установлены пакеты `libp11` и `libp11-kit` (ранее были установлены), то `openssl engine` работает без дополнительной конфигурации.

Проверим доступность `openssl engine`:

```
user@ipa ~ $ openssl engine pkcs11 -t
(pkcs11) pkcs11 engine
     [ available ]
```

Проверим что для `engine pkcs11` доступен протокол шифрования RSA:

```
user@ipa ~ $ openssl engine pkcs11 -c
(pkcs11) pkcs11 engine
 [RSA, rsaEncryption, id-ecPublicKey]
```

Подключаем токены по одному и создаём запросы на сертификаты:

для пользователя **rt** (токен Rutoken ECP)

```
user@ipa ~ $ export NAME="rt"; openssl req -new -out rt.csr \
    -keyform engine -engine pkcs11 -key "pkcs11:object=rt_2fa_ipa" \
    -passin pass:12345678 -config user_cert.inf
```

- `export NAME="rt"` — передаём в файл расширений для выпуска сертификатов имя пользователя
- `rt.csr` — файл запроса на выдачу сертификата
- `rt_2fa_ipa` — метка, по которой openssl определяет какими ключами на токене подписывать запрос
- `-passin pass:12345678` — PIN-код токена

для пользователя **es** (токен ESMART Token)

```
user@ipa ~ $ export NAME="es"; openssl req -new -out es.csr \
    -keyform engine -engine pkcs11 -key "pkcs11:object=es_2fa_ipa" \
    -passin pass:12345678 -config user_cert.inf
engine "pkcs11" set.
```

для пользователя **jc** (токен JaCarta-2 SE)

```
user@ipa ~ $ export NAME="jc"; openssl req -new -out jc.csr \
    -keyform engine -engine pkcs11 \
    -key "pkcs11:model=JaCarta%20Laser;object=jc_2fa_ipa" -passin pass:11111111 \
    -config user_cert.inf
engine "pkcs11" set.
```

Здесь, для `openssl engine`, определяем с каким именно слотом нужно работать при помощи `pkcs11:<URL>`, в котором указываем модель слота — `model=JaCarta%20Laser` (%20 — кодирование пробела). Также для определения слота можно как и раньше использовать его метку. Тогда `pkcs11:<URL>` будет выглядеть так:

```
"pkcs11:token=JC2SE-Laser;object=jc_2fa_ipa"
```

3.2.3. Выпускаем сертификаты

Выпущенные сертификаты сохраняются в профиле соответствующего пользователя, в базе LDAP на контроллере домена.

Получаем билет Kerberos, если он не был получен ранее, для выпуска сертификата на сервере:

```
user@ipa ~ $ kinit admin
Password for admin@TEST3.ALT:
user@ipa ~ $ klist
Ticket cache: KEYRING:persistent:500:500
Default principal: admin@TEST3.ALT

Valid starting       Expires              Service principal
03/30/2022 15:32:47  03/31/2022 15:32:43  krbtgt/TEST3.ALT@TEST3.ALT
```

Выпускаем сертификат для пользователя **rt** (токен Rutoken ECP):

```
user@ipa ~ $ ipa cert-request rt.csr --principal rt
  Issuing CA: ipa
  Certificate:
MIIEkTCCAvmgAwIBAgIBCzANBgkqhkiG9w0BAQsFADA0MRIwEAYDVQQKDAlURVNUMy5BTFQxHjAcBgNVBAMMFU
NlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0yMjAzMzAxMjQ4NTVaFw0yNDAzMzAxMjQ4NTVaMCExEjAQBgNVBAoM
CVRFU1QzLkFMVDELMAkGA1UEAwwCcnQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDixyi3OLL16b
8dxldRWIPLWVDtWS5yBtyiPJw7akrO0Vuhz7+Ei280Pnubsjv/q1h2OKZ1YbHJhuQayLk+JPBTzn4AEoARHUfW
U3XIZtSF1Fu/udTRi1ANIeoKyuSL2AQiz217dsU9Zfne15bWTudJCHQJ2CTLDtYV/mhapSwoxNsrBAKLGgv4xZ
XPI9AQ1aA/ZgP1FFedHzkzh89VDuMQlpHROXDS5jFDhsC9AMZcslettypr3/hcbQoJJmOZE28NtAKb/15EqIFk
t+qalz4jsNBYsW7EZ9bJQV1rhotQLj8EdE1U3UwDwYFTDRaIDS4slL3le0/hk96xVO28lNeTAgMBAAGjggE/MI
IBOzAfBgNVHSMEGDAWgBTRbNlqXldCVUQNKGu66SHbUA7efTA7BggrBgEFBQcBAQQvMC0wKwYIKwYBBQUHMAGG
H2h0dHA6Ly9pcGEtY2EudGVzdDMuYWx0L2NhL29jc3AwDgYDVR0PAQH/BAQDAgTwMB0GA1UdJQQWMBQGCCsGAQ
UFBwMBBggrBgEFBQcDAjB0BgNVHR8EbTBrMGmgMaAvhi1odHRwOi8vaXBhLWNhLnRlc3QzLmFsdC9pcGEvY3Js
L01hc3RlckNSTC5iaW6iNKQyMDAxDjAMBgNVBAoMBWlwYWNhMR4wHAYDVQQDDBVDZXJ0aWZpY2F0ZSBBdXRob3
JpdHkwHQYDVR0OBBYEFCFX+7558EUtkCOET2oPWR17nTy3MBcGA1UdEQQQMA6BDHJ0QHRlc3QzLmFsdDANBgkq
hkiG9w0BAQsFAAOCAYEAXSzbY7CrpoI1Tv6Ejmsvan545DU+2FXICE9vUFlP9601yUU4HNRQobPc8huSBkqZUk
dAtERHBUZ/JrwY777+G/J8S1au6SiDK9LwuRwABTV7jJ3hhWB7qxTrWhu1ErdoCToN1kRsek5W/ske+LWM/Ubx
9q2vdW/eUjK69GtwbLAWgWZFA1U7PfpDrhk1RPJ7RPPU+o0qToKhz/02JZRsZOZE1/KR5hpGcgOWeE0cG2aAID
xGu4wN2KlHqY4bcefIiLLnwXyZ1Z3jhwYa4A/0qlhqUN7wn3lve/+MctFzh7HA3ULt+egUz8q8Ub1BfN33i1nb
dFtxbK6LllJaszxtILIMpu62XeZt33qMpm4ZvuiEJRTF3ewz+lhWYJwLoZLgxHLhtjbht2/Xebd4i2g31jsMus
J556Yq0chpFEw293q+FXjpbHJGiCtgUEsVU77ufKD7+gmqjMGZTkPyvlhZDaXZpKkZeAtXcH39yOs0w6haTBMS
FA+HszrseRkSVV2u
  Subject: CN=rt,O=TEST3.ALT
  Subject email address: rt@test3.alt
  Issuer: CN=Certificate Authority,O=TEST3.ALT
  Not Before: Wed Mar 30 12:48:55 2022 UTC
  Not After: Sat Mar 30 12:48:55 2024 UTC
  Serial number: 11
  Serial number (hex): 0xB
```

Выпускаем сертификат для пользователя **es** (токен ESMART Token):

```
user@ipa ~ $ ipa cert-request es.csr --principal es
  Issuing CA: ipa
  Certificate:
MIIEkDCCAvigAwIBAgIBDDANBgkqhkiG9w0BAQsFADA0MRIwEAYDVQQKDAlURVNUMy5BTFQxHjAcBgNVBAMMFU
NlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0yMjAzMzAxMjU1NTNaFw0yNDAzMzAxMjU1NTNaMCExEjAQBgNVBAoM
CVRFU1QzLkFMVDELMAkGA1UEAwwCZXMwggEhMA0GCSqGSIb3DQEBAQUAA4IBDgAwggEJAoIBAHB7rEXrujSWW3
OmUWqHXDxbFKNFU2j76pRQmK+5UbnnopoWfY6qGdb3EiYXnf9bBn1J06QU0FuDS8vZhNqYeb6Oc3zMKWMHtK+0
EJXx/9BA+X7Yu8TsvKEAPKqHE8s44BsF/SYCyQ3Psoo550hDEnoy8b/gl2r8bs6YnFXns5sYDvL2O8frNtrN36
jZIyqo7cE+OU+AXYyId+iXEnWEhUzoNeRAhoA6BV/gnnjsBdE7xkR2WoM6tlFROP5cXbL3rS/pwWR35uQlevgr
rijnQS2a0ZJQ4pITm2wseefe0e0fyZwP+ot0xtENIRK2Ks08J70X21lDaIIQEPtdNzpYpo8CAwEAAaOCAT8wgg
E7MB8GA1UdIwQYMBaAFNFs2WpeV0JVRA0oa7rpIdtQDt59MDsGCCsGAQUFBwEBBC8wLTArBggrBgEFBQcwAYYf
aHR0cDovL2lwYS1jYS50ZXN0My5hbHQvY2Evb2NzcDAOBgNVHQ8BAf8EBAMCBPAwHQYDVR0lBBYwFAYIKwYBBQ
UHAwEGCCsGAQUFBwMCMHQGA1UdHwRtMGswaaAxoC+GLWh0dHA6Ly9pcGEtY2EudGVzdDMuYWx0L2lwYS9jcmwv
TWFzdGVyQ1JMLmJpbqI0pDIwMDEOMAwGA1UECgwFaXBhY2ExHjAcBgNVBAMMFUNlcnRpZmljYXRlIEF1dGhvcm
l0eTAdBgNVHQ4EFgQU/vz0WK5daVHKe2KGpl6scCRkGXswFwYDVR0RBBAwDoEMZXNAdGVzdDMuYWx0MA0GCSqG
SIb3DQEBCwUAA4IBgQA9tFqv/4F5AFJafw9rYYba1L5czU6Z3dwlSe65nGfbMKZXckkR3a7zta0UoR1FHJS16m
q477ycS/RUKNt2BczPsT2B2lN3A32gII46TTsplK8pRPFnDTX5iCDBY3XIrgIFFtGCaNn9Cdj55aKHj9oVgary
+V8mxehoPZF3Y0X9Y1EE7ji1fNsDrJneX/RYNUJZVKmkPeM2tTccthK5k+ftsykEMFKadIzA9QbDnw6YQ7RxHT
Ik5sUP004qJZFMVDzWiUhd1ruvoSUz/sgLNXfl1avokPIumojZuP5ZPy15Gm3Q2vXsEGm0qROHVBWUTVBhps5R
gdvMF77nGppPTwo82MFQCinVS/oKFt8geQ1EMx8TTHLMxHDF5FiSg6QXl4fPyfI1wMWgssIkNBP6Y4nwygERJt
tW7cHs2cBufyjFmm1xfh2I6nmFJUIXyIZU+llKLIlmT2GIK+hzBXwb82zvv++lDsP6hmNX6DcF7AUGwxEab/R8
zLRCBxGSVHnBsjg=
  Subject: CN=es,O=TEST3.ALT
  Subject email address: es@test3.alt
  Issuer: CN=Certificate Authority,O=TEST3.ALT
  Not Before: Wed Mar 30 12:55:53 2022 UTC
  Not After: Sat Mar 30 12:55:53 2024 UTC
  Serial number: 12
  Serial number (hex): 0xC
```

Выпускаем сертификат для пользователя **jc** (токен JaCarta-2 SE):

```
user@ipa ~ $ ipa cert-request jc.csr --principal jc
  Issuing CA: ipa
  Certificate:
MIIEkTCCAvmgAwIBAgIBETANBgkqhkiG9w0BAQsFADA0MRIwEAYDVQQKDAlURVNUMy5BTFQxHjAcBgNVBAMMFU
NlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0yMjA0MTMxNjE5MjBaFw0yNDA0MTMxNjE5MjBaMCExEjAQBgNVBAoM
CVRFU1QzLkFMVDELMAkGA1UEAwwCamMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCilMBH0MxHRt
gRkDr17Vqtk3e028xHEhwzSflyYNo4L0hKqa3NYduzw9JdWQ6DqvMaypeCpVEfMy6mIMQQqGJd8sqxsBJZEh+Q
6yuKZuZpLjg1fm7NNi74WJMjaSoh+C8n4U42LmGvMPrYQiX5f0K+mj5qi3zhAh1GbtIEoipZKPppp0v9NpJ5EP
S1X2jSfWJVJR2WLG4HyZGfJVOmYBayLjmR6433AhQbT+8yPirenjkvs7+EH1rICnMxACRS4GFWpjkc+EZPV+s+
4Xee2xhZwtw4i3imz4l/jJO2lfHnUXddLnpc4pWhvkuR4TY6IyCzam1lCCvWgBa6dKpHFDFdAgMBAAGjggE/MI
IBOzAfBgNVHSMEGDAWgBTRbNlqXldCVUQNKGu66SHbUA7efTA7BggrBgEFBQcBAQQvMC0wKwYIKwYBBQUHMAGG
H2h0dHA6Ly9pcGEtY2EudGVzdDMuYWx0L2NhL29jc3AwDgYDVR0PAQH/BAQDAgTwMB0GA1UdJQQWMBQGCCsGAQ
UFBwMBBggrBgEFBQcDAjB0BgNVHR8EbTBrMGmgMaAvhi1odHRwOi8vaXBhLWNhLnRlc3QzLmFsdC9pcGEvY3Js
L01hc3RlckNSTC5iaW6iNKQyMDAxDjAMBgNVBAoMBWlwYWNhMR4wHAYDVQQDDBVDZXJ0aWZpY2F0ZSBBdXRob3
JpdHkwHQYDVR0OBBYEFOKE4VGJ5KepoGfhDsC1eIMIK8M+MBcGA1UdEQQQMA6BDGpjQHRlc3QzLmFsdDANBgkq
```

```
hkiG9w0BAQsFAAOCAYEAZ2BHv4jNC2A1mjQ+XQWGpgvcwOaKqgAqrgN8gWVFG1SEdQ0yUVpAzLJUWdofp4quvi
pkmia6bH/7paniEK6GqPEgljRw9HQ+rOHeXbPuxODDpsiT1ItWTk3l9biE8EOxewJM10jgNpkdOZ4IrWJ/LcSl
97fMHfaxdZvJaNqGVP4AW0LrA3PWfD3S0PMOjvSJLhEO2TIJSdjfrkqEfQQntEvGZfNMTeaXT5JLNEwHcoYWNx
HtF4SXtXUwugYLsn45CC7hyTrX7Cv3hE37Q1XeOTzCH8sVaYsneJX9uKsQ5O1zDvpVsHoSvUulKWeLIBuLRsOS
ECmut1vqi+Lixfi4eHg3ocghqdmS9pNdNRTFafoW4CK1PGD4cyXsBYJ40mVwj0GjXMGCmo0f3EF8f97tqlROre
UH3y3W/bD1jiYM2pMEE/8wngA+LXW1dP/vdrTJ2OujqsLphiN74HKFoooME0CkXbAVMz3zw80AAcmEbypcGc+V
UD9vHx78L8ZE6BB0
  Subject: CN=jc,O=TEST3.ALT
  Subject email address: jc@test3.alt
  Issuer: CN=Certificate Authority,O=TEST3.ALT
  Not Before: Wed Apr 13 16:19:20 2022 UTC
  Not After: Sat Apr 13 16:19:20 2024 UTC
  Serial number: 17
  Serial number (hex): 0x11
```

Просмотрим профиль пользователя и убедимся что в нём есть сертификат (на примере **rt**):

```
user@ipa ~ $ ipa user-show rt
  User login: rt
  First name: Active
  Last name: Rutoken
  Home directory: /home/rt
  Login shell: /bin/bash
  Principal name: rt@TEST3.ALT
  Principal alias: rt@TEST3.ALT
  Email address: rt@test3.alt
  UID: 1450000001
  GID: 1450000001
  Certificate:
MIIEkTCCAvmgAwIBAgIBCzANBgkqhkiG9w0BAQsFADA0MRIwEAYDVQQKDAlURVNUMy5BTFQxHjAcBgNVBAMMFU
NlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0yMjAzMzAxMjQ4NTVaFw0yNDAzMzAxMjQ4NTVaMCExEjAQBgNVBAoM
CVRFU1QzLkFMVDELMAkGA1UEAwwCcnQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDixyi3OLL16b
8dxldRWIPLWVDtWS5yBtyiPJw7akrO0Vuhz7+Ei280Pnubsjv/q1h2OKZ1YbHJhuQayLk+JPBTzn4AEoARHUfW
U3XIZtSF1Fu/udTRi1ANIeoKyuSL2AQiz217dsU9Zfne15bWTudJCHQJ2CTLDtYV/mhapSwoxNsrBAKLGgv4xZ
XPI9AQ1aA/ZgP1FFedHzkzh89VDuMQlpHROXDS5jFDhsC9AMZcslettypr3/hcbQoJJmOZE28NtAKb/15EqIFk
t+qalz4jsNBYsW7EZ9bJQV1rhotQLj8EdE1U3UwDwYFTDRaIDS4slL3le0/hk96xVO28lNeTAgMBAAGjggE/MI
IBOzAfBgNVHSMEGDAWgBTRbNlqXldCVUQNKGu66SHbUA7efTA7BggrBgEFBQcBAQQvMC0wKwYIKwYBBQUHMAGG
H2h0dHA6Ly9pcGEtY2EudGVzdDMuYWx0L2NhL29jc3AwDgYDVR0PAQH/BAQDAgTwMB0GA1UdJQQWMBQGCCsGAQ
UFBwMBBggrBgEFBQcDAjB0BgNVHR8EbTBrMGmgMaAvhi1odHRwOi8vaXBhLWNhLnRlc3QzLmFsdC9pcGEvY3Js
L01hc3RlckNSTC5iaW6iNKQyMDAxDjAMBgNVBAoMBWlwYWNhMR4wHAYDVQQDDBVDZXJ0aWZpY2F0ZSBBdXRob3
JpdHkwHQYDVR0OBBYEFCFX+7558EUtkCOET2oPWR17nTy3MBcGA1UdEQQQMA6BDHJ0QHRlc3QzLmFsdDANBgkq
hkiG9w0BAQsFAAOCAYEAXSzbY7CrpoI1Tv6Ejmsvan545DU+2FXICE9vUFlP9601yUU4HNRQobPc8huSBkqZUk
dAtERHBUZ/JrwY777+G/J8S1au6SiDK9LwuRwABTV7jJ3hhWB7qxTrWhu1ErdoCToN1kRsek5W/ske+LWM/Ubx
9q2vdW/eUjK69GtwbLAWgWZFA1U7PfpDrhk1RPJ7RPPU+o0qToKhz/02JZRsZOZE1/KR5hpGcgOWeE0cG2aAID
xGu4wN2KlHqY4bcefIiLLnwXyZ1Z3jhwYa4A/0qlhqUN7wn3lve/+MctFzh7HA3ULt+egUz8q8Ub1BfN33i1nb
dFtxbK6LllJaszxtILIMpu62XeZt33qMpm4ZvuiEJRTF3ewz+lhWYJwLoZLgxHLhtjbht2/Xebd4i2g31jsMus
J556Yq0chpFEw293q+FXjpbHJGiCtgUEsVU77ufKD7+gmqjMGZTkPyvlhZDaXZpKkZeAtXcH39yOs0w6haTBMS
FA+HszrseRkSVV2u
  Account disabled: False
  Password: False
```

```
  Member of groups: ipausers
  Kerberos keys available: False
```

3.2.4. Записываем сертификаты на токены

Перед записью сертификатов на токены их сначала необходимо экспортировать из профиля пользователя. Чтобы экспортировать сертификаты, нужно знать серийный номер сертификата. Серийный номер сертификата указывается при его выпуске. Также узнать его можно при помощи следующей команды:

```
user@ipa ~ $ ipa cert-find --user=rt
---------------------
1 certificate matched
---------------------
  Issuing CA: ipa
  Subject: CN=rt,O=TEST3.ALT
  Issuer: CN=Certificate Authority,O=TEST3.ALT
  Not Before: Wed Mar 30 15:48:55 2022 UTC
  Not After: Sat Mar 30 15:48:55 2024 UTC
  Serial number: 11
  Serial number (hex): 0xB
  Status: VALID
  Revoked: False
----------------------------
Number of entries returned 1
----------------------------
```

Экспортируем сертификат пользователя **rt**:

```
user@ipa ~ $ ipa cert-show 11 --certificate-out=rt.pem
  Issuing CA: ipa
  Certificate:
MIIEkTCCAvmgAwIBAgIBCzANBgkqhkiG9w0BAQsFADA0MRIwEAYDVQQKDAlURVNUMy5BTFQxHjAcBgNVBAMMFU
NlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0yMjAzMzAxMjQ4NTVaFw0yNDAzMzAxMjQ4NTVaMCExEjAQBgNVBAoM
CVRFU1QzLkFMVDELMAkGA1UEAwwCcnQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDixyi3OLL16b
8dxldRWIPLWVDtWS5yBtyiPJw7akrO0Vuhz7+Ei280Pnubsjv/q1h2OKZ1YbHJhuQayLk+JPBTzn4AEoARHUfW
U3XIZtSF1Fu/udTRi1ANIeoKyuSL2AQiz217dsU9Zfne15bWTudJCHQJ2CTLDtYV/mhapSwoxNsrBAKLGgv4xZ
XPI9AQ1aA/ZgP1FFedHzkzh89VDuMQlpHROXDS5jFDhsC9AMZcslettypr3/hcbQoJJmOZE28NtAKb/15EqIFk
t+qalz4jsNBYsW7EZ9bJQV1rhotQLj8EdE1U3UwDwYFTDRaIDS4slL3le0/hk96xVO28lNeTAgMBAAGjggE/MI
IBOzAfBgNVHSMEGDAWgBTRbNlqXldCVUQNKGu66SHbUA7efTA7BggrBgEFBQcBAQQvMC0wKwYIKwYBBQUHMAGG
H2h0dHA6Ly9pcGEtY2EudGVzdDMuYWx0L2NhL29jc3AwDgYDVR0PAQH/BAQDAgTwMB0GA1UdJQQWMBQGCCsGAQ
UFBwMBBggrBgEFBQcDAjB0BgNVHR8EbTBrMGmgMaAvhi1odHRwOi8vaXBhLWNhLnRlc3QzLmFsdC9pcGEvY3Js
L01hc3RlckNSTC5iaW6iNKQyMDAxDjAMBgNVBAoMBWlwYWNhMR4wHAYDVQQDDBVDZXJ0aWZpY2F0ZSBBdXRob3
JpdHkwHQYDVR0OBBYEFCFX+7558EUtkCOET2oPWR17nTy3MBcGA1UdEQQQMA6BDHJ0QHRlc3QzLmFsdDANBgkq
hkiG9w0BAQsFAAOCAYEAXSzbY7CrpoI1Tv6Ejmsvan545DU+2FXICE9vUFlP9601yUU4HNRQobPc8huSBkqZUk
dAtERHBUZ/JrwY777+G/J8S1au6SiDK9LwuRwABTV7jJ3hhWB7qxTrWhu1ErdoCToN1kRsek5W/ske+LWM/Ubx
9q2vdW/eUjK69GtwbLAWgWZFA1U7PfpDrhk1RPJ7RPPU+o0qToKhz/02JZRsZOZE1/KR5hpGcgOWeE0cG2aAID
xGu4wN2KlHqY4bcefIiLLnwXyZ1Z3jhwYa4A/0qlhqUN7wn3lve/+MctFzh7HA3ULt+egUz8q8Ub1BfN33i1nb
```

```
dFtxbK6LllJaszxtILIMpu62XeZt33qMpm4ZvuiEJRTF3ewz+lhWYJwLoZLgxHLhtjbht2/Xebd4i2g31jsMus
J556Yq0chpFEw293q+FXjpbHJGiCtgUEsVU77ufKD7+gmqjMGZTkPyvlhZDaXZpKkZeAtXcH39yOs0w6haTBMS
FA+HszrseRkSVV2u
  Subject: CN=rt,O=TEST3.ALT
  Subject email address: rt@test3.alt
  Issuer: CN=Certificate Authority,O=TEST3.ALT
  Not Before: Wed Mar 30 12:48:55 2022 UTC
  Not After: Sat Mar 30 12:48:55 2024 UTC
  Serial number: 11
  Serial number (hex): 0xB
  Revoked: False
  Owner user: rt
```

Экспортируем сертификат пользователя **es**:

```
user@ipa ~ $ ipa cert-show 12 --certificate-out=es.pem
  Issuing CA: ipa
  Certificate:
MIIEkDCCAvigAwIBAgIBDDANBgkqhkiG9w0BAQsFADA0MRIwEAYDVQQKDAlURVNUMy5BTFQxHjAcBgNVBAMMFU
NlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0yMjAzMzAxMjU1NTNaFw0yNDAzMzAxMjU1NTNaMCExEjAQBgNVBAoM
CVRFU1QzLkFMVDELMAkGA1UEAwwCZXMwggEhMA0GCSqGSIb3DQEBAQUAA4IBDgAwggEJAoIBAHB7rEXrujSWW3
OmUWqHXDxbFKNFU2j76pRQmK+5UbnnopoWfY6qGdb3EiYXnf9bBn1J06QU0FuDS8vZhNqYeb6Oc3zMKWMHtK+0
EJXx/9BA+X7Yu8TsvKEAPKqHE8s44BsF/SYCyQ3Psoo550hDEnoy8b/gl2r8bs6YnFXns5sYDvL2O8frNtrN36
jZIyqo7cE+OU+AXYyId+iXEnWEhUzoNeRAhoA6BV/gnnjsBdE7xkR2WoM6tlFROP5cXbL3rS/pwWR35uQlevgr
rijnQS2a0ZJQ4pITm2wseefe0e0fyZwP+ot0xtENIRK2Ks08J70X21lDaIIQEPtdNzpYpo8CAwEAAaOCAT8wgg
E7MB8GA1UdIwQYMBaAFNFs2WpeV0JVRA0oa7rpIdtQDt59MDsGCCsGAQUFBwEBBC8wLTArBggrBgEFBQcwAYYf
aHR0cDovL2lwYS1jYS50ZXN0My5hbHQvY2Evb2NzcDAOBgNVHQ8BAf8EBAMCBPAwHQYDVR0lBBYwFAYIKwYBBQ
UHAwEGCCsGAQUFBwMCMHQGA1UdHwRtMGswaaAxoC+GLWh0dHA6Ly9pcGEtY2EudGVzdDMuYWx0L2lwYS9jcmwv
TWFzdGVyQ1JMLmJpbqI0pDIwMDEOMAwGA1UECgwFaXBhY2ExHjAcBgNVBAMMFUNlcnRpZmljYXRlIEF1dGhvcm
l0eTAdBgNVHQ4EFgQU/vz0WK5daVHKe2KGpl6scCRkGXswFwYDVR0RBBAwDoEMZXNAdGVzdDMuYWx0MA0GCSqG
SIb3DQEBCwUAA4IBgQA9tFqv/4F5AFJafw9rYYba1L5czU6Z3dwlSe65nGfbMKZXckkR3a7zta0UoR1FHJS16m
q477ycS/RUKNt2BczPsT2B2lN3A32gII46TTsplK8pRPFnDTX5iCDBY3XIrgIFFtGCaNn9Cdj55aKHj9oVgary
+V8mxehoPZF3Y0X9Y1EE7ji1fNsDrJneX/RYNUJZVKmkPeM2tTccthK5k+ftsykEMFKadIzA9QbDnw6YQ7RxHT
Ik5sUP004qJZFMVDzWiUhd1ruvoSUz/sgLNXfl1avokPIumojZuP5ZPy15Gm3Q2vXsEGm0qROHVBWUTVBhps5R
gdvMF77nGppPTwo82MFQCinVS/oKFt8geQ1EMx8TTHLMxHDF5FiSg6QXl4fPyfI1wMWgssIkNBP6Y4nwygERJt
tW7cHs2cBufyjFmm1xfh2I6nmFJUIXyIZU+llKLIlmT2GIK+hzBXwb82zvv++lDsP6hmNX6DcF7AUGwxEab/R8
zLRCBxGSVHnBsjg=
  Subject: CN=es,O=TEST3.ALT
  Subject email address: es@test3.alt
  Issuer: CN=Certificate Authority,O=TEST3.ALT
  Not Before: Wed Mar 30 12:55:53 2022 UTC
  Not After: Sat Mar 30 12:55:53 2024 UTC
  Serial number: 12
  Serial number (hex): 0xC
  Revoked: False
  Owner user: es
```

Экспортируем сертификат пользователя **jc**:

```
user@ipa ~ $ ipa cert-show 17 --certificate-out=jc.pem
  Issuing CA: ipa
  Certificate:
MIIEkTCCAvmgAwIBAgIBETANBgkqhkiG9w0BAQsFADA0MRIwEAYDVQQKDAlURVNUMy5BTFQxHjAcBgNVBAMMFU
NlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0yMjA0MTMxNjE5MjBaFw0yNDA0MTMxNjE5MjBaMCExEjAQBgNVBAoM
CVRFU1QzLkFMVDELMAkGA1UEAwwCamMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCilMBH0MxHRt
gRkDr17Vqtk3e028xHEhwzSflyYNo4L0hKqa3NYduzw9JdWQ6DqvMaypeCpVEfMy6mIMQQqGJd8sqxsBJZEh+Q
6yuKZuZpLjg1fm7NNi74WJMjaSoh+C8n4U42LmGvMPrYQiX5f0K+mj5qi3zhAh1GbtIEoipZKPppp0v9NpJ5EP
S1X2jSfWJVJR2WLG4HyZGfJVOmYBayLjmR6433AhQbT+8yPirenjkvs7+EH1rICnMxACRS4GFWpjkc+EZPV+s+
4Xee2xhZwtw4i3imz4l/jJO2lfHnUXddLnpc4pWhvkuR4TY6IyCzam1lCCvWgBa6dKpHFDFdAgMBAAGjggE/MI
IBOzAfBgNVHSMEGDAWgBTRbNlqXldCVUQNKGu66SHbUA7efTA7BggrBgEFBQcBAQQvMC0wKwYIKwYBBQUHMAGG
H2h0dHA6Ly9pcGEtY2EudGVzdDMuYWx0L2NhL29jc3AwDgYDVR0PAQH/BAQDAgTwMB0GA1UdJQQWMBQGCCsGAQ
UFBwMBBggrBgEFBQcDAjB0BgNVHR8EbTBrMGmgMaAvhi1odHRwOi8vaXBhLWNhLnRlc3QzLmFsdC9pcGEvY3Js
L01hc3RlckNSTC5iaW6iNKQyMDAxDjAMBgNVBAoMBWlwYWNhMR4wHAYDVQQDDBVDZXJ0aWZpY2F0ZSBBdXRob3
JpdHkwHQYDVR0OBBYEFOKE4VGJ5KepoGfhDsC1eIMIK8M+MBcGA1UdEQQQMA6BDGpjQHRlc3QzLmFsdDANBgkq
hkiG9w0BAQsFAAOCAYEAZ2BHv4jNC2A1mjQ+XQWGpgvcwOaKqgAqrgN8gWVFG1SEdQ0yUVpAzLJUWdofp4quvi
pkmia6bH/7paniEK6GqPEgljRw9HQ+rOHeXbPuxODDpsiT1ItWTk3l9biE8EOxewJM10jgNpkdOZ4IrWJ/LcSl
97fMHfaxdZvJaNqGVP4AW0LrA3PWfD3S0PMOjvSJLhEO2TIJSdjfrkqEfQQntEvGZfNMTeaXT5JLNEwHcoYWNx
HtF4SXtXUwugYLsn45CC7hyTrX7Cv3hE37Q1XeOTzCH8sVaYsneJX9uKsQ5O1zDvpVsHoSvUulKWeLIBuLRsOS
ECmut1vqi+Lixfi4eHg3ocghqdmS9pNdNRTFafoW4CK1PGD4cyXsBYJ40mVwj0GjXMGCmo0f3EF8f97tqlROre
UH3y3W/bD1jiYM2pMEE/8wngA+LXW1dP/vdrTJ2OujqsLphiN74HKFoooME0CkXbAVMz3zw80AAcmEbypcGc+V
UD9vHx78L8ZE6BB0
  Subject: CN=jc,O=TEST3.ALT
  Subject email address: jc@test3.alt
  Issuer: CN=Certificate Authority,O=TEST3.ALT
  Not Before: Wed Apr 13 16:19:20 2022 UTC
  Not After: Sat Apr 13 16:19:20 2024 UTC
  Serial number: 17
  Serial number (hex): 0x11
  Revoked: False
  Owner user: jc
```

Можно проверить сопоставление экспортированного сертификата с пользователем (на примере **jc**):

```
user@ipa ~ $ ipa certmap-match jc.pem
--------------
1 user matched
--------------
  Domain: TEST3.ALT
  User logins: jc
----------------------------
Number of entries returned 1
----------------------------
```

При записи сертификата на токен необходимо указывать идентификатор (id), с которым он будет храниться на токене. Этот идентификатор **должен** совпадать с идентификаторами закрытого и открытого ключей на токене для сохранения целостности цепочки.

Записываем сертификат пользователя **rt** на токен Rutoken ECP:

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/p11-kit-proxy.so --login \
    --pin 12345678 --write-object rt.pem --type cert --id 7001 --label rt_2fa_ipa
Using slot 32 with a present token (0x30)
Created certificate:
Certificate Object; type = X.509 cert
  label:      rt_2fa_ipa
  subject:    DN: O=TEST3.ALT, CN=rt
  ID:         7001
```

Записываем сертификат пользователя **es** на токен ESMART Token:

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/p11-kit-proxy.so --login \
    --pin 12345678 --write-object es.pem --type cert --id 7002 --label es_2fa_ipa
Using slot 0 with a present token (0x10)
Created certificate:
Certificate Object; type = X.509 cert
  label:      es_2fa_ipa
  subject:    DN: O=TEST3.ALT, CN=es
  ID:         7002
```

Записываем сертификат пользователя **jc** на токен JaCarta-2 SE:

```
user@ipa ~ $ pkcs11-tool --module /usr/lib64/p11-kit-proxy.so --login \
    --pin 11111111 --write-object jc.pem --type cert --id 7003 --label jc_2fa_ipa \
    --token-label JC2SE-Laser
Created certificate:
Certificate Object; type = X.509 cert
  label:      jc_2fa_ipa
  subject:    DN: O=TEST3.ALT, CN=jc
  ID:         7003
```

4. Настраиваем PKINIT на клиенте

4.1. Настройка и проверка Kerberos

В конфигурационном файле Kerberos установим фильтр, чтобы использовать сертификаты выданные для домена `test3.alt`. Для этого добавим правило `pkinit_cert_match = <ISSUER>.*TEST3.ALT.*` в секцию `[realms]` файла `/etc/krb5.conf`:

Итого получается следующий конфигурационный файл Kerberos:

```
client ~ # cat /etc/krb5.conf | grep -viE '(^#|^$)'
includedir /etc/krb5.conf.d/
includedir /var/lib/sss/pubconf/krb5.include.d/
[libdefaults]
  default_realm = TEST3.ALT
  dns_lookup_realm = true
  dns_lookup_kdc = true
  rdns = false
  dns_canonicalize_hostname = false
  ticket_lifetime = 24h
  forwardable = true
  udp_preference_limit = 0
  default_ccache_name = KEYRING:persistent:%{uid}
[realms]
  TEST3.ALT = {
    pkinit_anchors = FILE:/var/lib/ipa-client/pki/kdc-ca-bundle.pem
    pkinit_pool = FILE:/var/lib/ipa-client/pki/ca-bundle.pem
    pkinit_cert_match = <ISSUER>.*TEST3.ALT.*
  }
[domain_realm]
  .test3.alt = TEST3.ALT
  test3.alt = TEST3.ALT
  client.test3.alt = TEST3.ALT
```

4.1.1. Rutoken ECP

Проверяем получение билета Kerberos для пользователя **rt**:

```
user@client ~ $ kinit -V -X X509_user_identity='PKCS11:librtpkcs11ecp.so' rt
Using existing cache: persistent:0:0
Using principal: rt@TEST3.ALT
PA Option X509_user_identity = PKCS11:librtpkcs11ecp.so
r_token_ecp2                    PIN:
Authenticated to Kerberos v5
user@client ~ $ klist
Ticket cache: KEYRING:persistent:0:0
Default principal: rt@TEST3.ALT
```

```
Valid starting       Expires              Service principal
04/05/2022 16:34:56  04/06/2022 16:34:48  krbtgt/TEST3.ALT@TEST3.ALT
```

`X509_user_identity='PKCS11:librtpkcs11ecp.so'` — указываем, какую библиотеку PKCS#11 использовать.

4.1.2. ESMART Token

Проверяем получение билета Kerberos для пользователя **es**:

```
user@client ~ $ kinit -V \
    -X X509_user_identity='PKCS11:libisbc_pkcs11_main.so' es
Using new cache: persistent:0:krb_ccache_DNikg01
Using principal: es@TEST3.ALT
PA Option X509_user_identity = PKCS11:libisbc_pkcs11_main.so
esmart_64                       PIN:
Authenticated to Kerberos v5
user@client ~ $ klist
Ticket cache: KEYRING:persistent:0:krb_ccache_DNikg01
Default principal: es@TEST3.ALT

Valid starting       Expires              Service principal
04/05/2022 17:04:02  04/06/2022 17:03:51  krbtgt/TEST3.ALT@TEST3.ALT
```

4.1.3. JaCarta-2 SE

Здесь необходимо дополнительно указать нужный слот, так как токен мультислотовый.

Варианты `pkcs11:<URL>` службы Kerberos (`man krb5.conf`):
`PKCS11:[module_name=]modname[:slotid=slot-id][:token=token-label][:certid=cert-id][:certlabel=cert-label]`

Для указания слота подходят два варианта:

- `token-label` — метка токена (предпочтительный вариант)
- `slot-id` — идентификатор слота. Указан в скобках после индекса слота при использовании утилиты `pkcs11-tool` — **"Slot 2 (0x3ffff)"**. В `krb5.conf` данный параметр необходимо указывать в десятеричном формате.

Проверяем получение билета Kerberos для пользователя **jc**:

```
user@client ~ $ kinit -V \
    -X X509_user_identity='PKCS11:libjcPKCS11-2.so:token=JC2SE-Laser' jc
Using default cache: persistent:500:500
Using principal: jc@TEST3.ALT
PA Option X509_user_identity = PKCS11:libjcPKCS11-2.so:token=JC2SE-Laser
JC2SE-Laser                     PIN:
```

```
Authenticated to Kerberos v5
user@client ~ $ klist
Ticket cache: KEYRING:persistent:500:500
Default principal: jc@TEST3.ALT

Valid starting       Expires              Service principal
04/14/2022 11:41:24  04/15/2022 11:41:05  krbtgt/TEST3.ALT@TEST3.ALT
```

4.1.4. Отладка Kerberos

Если при проверке Kerberos возникают ошибки, то для более полной информации по процессу PKINIT добавьте опцию вывода отладочной информации в консоль — `KRB5_TRACE=/dev/stdout`. Команда получения билета Kerberos с отладкой:

```
user@client ~ $ KRB5_TRACE=/dev/stdout \
    kinit -X X509_user_identity='PKCS11:librtpkcs11ecp.so' rt
```

4.2. Настройка sssd

После ввода ПК клиента в домен Samba DC за аутентификацию отвечает служба `sssd`. Утилита `p11_child`, входящая в состав пакета `sssd`, обеспечивает доступ к сертификату на токене.

Копируем сертификат УЦ на ПК доменного пользователя, в каталог `/etc/pki/tls/certs`. В нём служба sssd будет искать сертификат УЦ.

```
client ~ # scp user@ipa:/home/user/ca.pem /etc/pki/tls/certs/
```

4.2.1. p11_child. Проверка работы sssd с токенами.

Убедимся, что служба sssd видит сертификат на токене.

Rutoken ECP

```
user@client ~ $ /usr/libexec/sssd/p11_child \
    --ca_db=/etc/pki/tls/certs/ca.pem --pre
r_token_ecp2
/usr/lib64/pkcs11/librtpkcs11ecp.so
7001
rt_2fa_ipa
MIIEkTCCAvmgAwIBAgIBCzANBgkqhkiG9w0BAQsFADA0MRIwEAYDVQQKDAlURVNUMy5BTFQxHjAcBgNVBAMMFU
NlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0yMjAzMzAxMjQ4NTVaFw0yNDAzMzAxMjQ4NTVaMCExEjAQBgNVBAoM
CVRFU1QzLkFMVDELMAkGA1UEAwwCcnQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDixyi3OLL16b
8dxldRWIPLWVDtWS5yBtyiPJw7akrO0Vuhz7+Ei280Pnubsjv/q1h2OKZ1YbHJhuQayLk+JPBTzn4AEoARHUfW
U3XIZtSF1Fu/udTRi1ANIeoKyuSL2AQiz217dsU9Zfne15bWTudJCHQJ2CTLDtYV/mhapSwoxNsrBAKLGgv4xZ
XPI9AQ1aA/ZgP1FFedHzkzh89VDuMQlpHROXDS5jFDhsC9AMZcslettypr3/hcbQoJJmOZE28NtAKb/15EqIFk
```

```
t+qalz4jsNBYsW7EZ9bJQV1rhotQLj8EdE1U3UwDwYFTDRaIDS4slL3le0/hk96xVO28lNeTAgMBAAGjggE/MI
IBOzAfBgNVHSMEGDAWgBTRbNlqXldCVUQNKGu66SHbUA7efTA7BggrBgEFBQcBAQQvMC0wKwYIKwYBBQUHMAGG
H2h0dHA6Ly9pcGEtY2EudGVzdDMuYWx0L2NhL29jc3AwDgYDVR0PAQH/BAQDAgTwMB0GA1UdJQQWMBQGCCsGAQ
UFBwMBBggrBgEFBQcDAjB0BgNVHR8EbTBrMGmgMaAvhi1odHRwOi8vaXBhLWNhLnRlc3QzLmFsdC9pcGEvY3Js
L01hc3RlckNSTC5iaW6iNKQyMDAxDjAMBgNVBAoMBWlwYWNhMR4wHAYDVQQDDBVDZXJ0aWZpY2F0ZSBBdXRob3
JpdHkwHQYDVR0OBBYEFCFX+7558EUtkCOET2oPWR17nTy3MBcGA1UdEQQQMA6BDHJ0QHRlc3QzLmFsdDANBgkq
hkiG9w0BAQsFAAOCAYEAXSzbY7CrpoI1Tv6Ejmsvan545DU+2FXICE9vUFlP9601yUU4HNRQobPc8huSBkqZUk
dAtERHBUZ/JrwY777+G/J8S1au6SiDK9LwuRwABTV7jJ3hhWB7qxTrWhu1ErdoCToN1kRsek5W/ske+LWM/Ubx
9q2vdW/eUjK69GtwbLAWgWZFA1U7PfpDrhk1RPJ7RPPU+o0qToKhz/02JZRsZOZE1/KR5hpGcgOWeE0cG2aAID
xGu4wN2KlHqY4bcefIiLLnwXyZ1Z3jhwYa4A/0qlhqUN7wn3lve/+MctFzh7HA3ULt+egUz8q8Ub1BfN33i1nb
dFtxbK6LllJaszxtILIMpu62XeZt33qMpm4ZvuiEJRTF3ewz+lhWYJwLoZLgxHLhtjbht2/Xebd4i2g31jsMus
J556Yq0chpFEw293q+FXjpbHJGiCtgUEsVU77ufKD7+gmqjMGZTkPyvlhZDaXZpKkZeAtXcH39yOs0w6haTBMS
FA+HszrseRkSVV2u
```

- `r_token_ecp2` — метка (label) токена
- `/usr/lib64/pkcs11/librtpkcs11ecp.so` — библиотека, которая подходит для данного токена
- `7001` — идентификатор (id) сертификата
- `rt_2fa_ipa` — метку (label) сертификата
- `MIIEkTC···kSVV2u` — сертификат

ESMART Token

```
user@client ~ $ /usr/libexec/sssd/p11_child \
    --ca_db=/etc/pki/tls/certs/ca.pem --pre
esmart_64
/usr/lib64/pkcs11/libisbc_pkcs11_main.so
7002
es_2fa_ipa
MIIEkDCCAvigAwIBAgIBDDANBgkqhkiG9w0BAQsFADA0MRIwEAYDVQQKDAlURVNUMy5BTFQxHjAcBgNVBAMMFU
NlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0yMjAzMzAxMjU1NTNaFw0yNDAzMzAxMjU1NTNaMCExEjAQBgNVBAoM
CVRFU1QzLkFMVDELMAkGA1UEAwwCZXMwggEhMA0GCSqGSIb3DQEBAQUAA4IBDgAwggEJAoIBAHB7rEXrujSWW3
OmUWqHXDxbFKNFU2j76pRQmK+5UbnnopoWfY6qGdb3EiYXnf9bBn1J06QU0FuDS8vZhNqYeb6Oc3zMKWMHtK+0
EJXx/9BA+X7Yu8TsvKEAPKqHE8s44BsF/SYCyQ3Psoo550hDEnoy8b/gl2r8bs6YnFXns5sYDvL2O8frNtrN36
jZIyqo7cE+OU+AXYyId+iXEnWEhUzoNeRAhoA6BV/gnnjsBdE7xkR2WoM6tlFROP5cXbL3rS/pwWR35uQlevgr
rijnQS2a0ZJQ4pITm2wseefe0e0fyZwP+ot0xtENIRK2Ks08J70X21lDaIIQEPtdNzpYpo8CAwEAAaOCAT8wgg
E7MB8GA1UdIwQYMBaAFNFs2WpeV0JVRA0oa7rpIdtQDt59MDsGCCsGAQUFBwEBBC8wLTArBggrBgEFBQcwAYYf
aHR0cDovL2lwYS1jYS50ZXN0My5hbHQvY2Evb2NzcDAOBgNVHQ8BAf8EBAMCBPAwHQYDVR0lBBYwFAYIKwYBBQ
UHAwEGCCsGAQUFBwMCMHQGA1UdHwRtMGswaaAxoC+GLWh0dHA6Ly9pcGEtY2EudGVzdDMuYWx0L2lwYS9jcmwv
TWFzdGVyQ1JMLmJpbqI0pDIwMDEOMAwGA1UECgwFaXBhY2ExHjAcBgNVBAMMFUNlcnRpZmljYXRlIEF1dGhvcm
l0eTAdBgNVHQ4EFgQU/vz0WK5daVHKe2KGpl6scCRkGXswFwYDVR0RBBAwDoEMZXNAdGVzdDMuYWx0MA0GCSqG
SIb3DQEBCwUAA4IBgQA9tFqv/4F5AFJafw9rYYba1L5czU6Z3dwlSe65nGfbMKZXckkR3a7zta0UoR1FHJS16m
q477ycS/RUKNt2BczPsT2B2lN3A32gII46TTsplK8pRPFnDTX5iCDBY3XIrgIFFtGCaNn9Cdj55aKHj9oVgary
+V8mxehoPZF3Y0X9Y1EE7ji1fNsDrJneX/RYNUJZVKmkPeM2tTccthK5k+ftsykEMFKadIzA9QbDnw6YQ7RxHT
Ik5sUP004qJZFMVDzWiUhd1ruvoSUz/sgLNXfl1avokPIumojZuP5ZPy15Gm3Q2vXsEGm0qROHVBWUTVBhps5R
gdvMF77nGppPTwo82MFQCinVS/oKFt8geQ1EMx8TTHLMxHDF5FiSg6QXl4fPyfI1wMWgssIkNBP6Y4nwygERJt
tW7cHs2cBufyjFmm1xfh2I6nmFJUIXyIZU+llKLIlmT2GIK+hzBXwb82zvv++lDsP6hmNX6DcF7AUGwxEab/R8
zLRCBxGSVHnBsjg=
```

JaCarta-2 SE

Для проверки работы службы sssd с токеном JaCarta-2 SE необходимо дополнительно указать слот. Сделать это можно при помощи схемы `pkcs11:<URI>`, которая поддерживается в sssd. С полным перечнем параметров `pkcs11:<URI>` можно ознакомиться в RFC7512 (https://datatracker.ietf.org/doc/html/rfc7512#section-2.1).

В данном случае мы будем использовать схему с указанием модели слота — `uri=pkcs11:model=JaCarta%20Laser`.

```
user@client ~ $ /usr/libexec/sssd/p11_child \
    --ca_db=/etc/pki/tls/certs/ca.pem --pre --uri=pkcs11:model=JaCarta%20Laser
JC2SE-Laser
/usr/lib64/pkcs11/libjcPKCS11-2.so
7003
jc_2fa_ipa
MIIEkTCCAvmgAwIBAgIBETANBgkqhkiG9w0BAQsFADA0MRIwEAYDVQQKDAlURVNUMy5BTFQxHjAcBgNVBAMMFU
NlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0yMjA0MTMxNjE5MjBaFw0yNDA0MTMxNjE5MjBaMCExEjAQBgNVBAoM
CVRFU1QzLkFMVDELMAkGA1UEAwwCamMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCilMBH0MxHRt
gRkDr17Vqtk3e028xHEhwzSflyYNo4L0hKqa3NYduzw9JdWQ6DqvMaypeCpVEfMy6mIMQQqGJd8sqxsBJZEh+Q
6yuKZuZpLjg1fm7NNi74WJMjaSoh+C8n4U42LmGvMPrYQiX5f0K+mj5qi3zhAh1GbtIEoipZKPppp0v9NpJ5EP
S1X2jSfWJVJR2WLG4HyZGfJVOmYBayLjmR6433AhQbT+8yPirenjkvs7+EH1rICnMxACRS4GFWpjkc+EZPV+s+
4Xee2xhZwtw4i3imz4l/jJO2lfHnUXddLnpc4pWhvkuR4TY6IyCzam1lCCvWgBa6dKpHFDFdAgMBAAGjggE/MI
IBOzAfBgNVHSMEGDAWgBTRbNlqXldCVUQNKGu66SHbUA7efTA7BggrBgEFBQcBAQQvMC0wKwYIKwYBBQUHMAGG
H2h0dHA6Ly9pcGEtY2EudGVzdDMuYWx0L2NhL29jc3AwDgYDVR0PAQH/BAQDAgTwMB0GA1UdJQQWMBQGCCsGAQ
UFBwMBBggrBgEFBQcDAjB0BgNVHR8EbTBrMGmgMaAvhi1odHRwOi8vaXBhLWNhLnRlc3QzLmFsdC9pcGEvY3Js
L01hc3RlckNSTC5iaW6iNKQyMDAxDjAMBgNVBAoMBWlwYWNhMR4wHAYDVQQDDBVDZXJ0aWZpY2F0ZSBBdXRob3
JpdHkwHQYDVR0OBBYEFOKE4VGJ5KepoGfhDsC1eIMIK8M+MBcGA1UdEQQQMA6BDGpjQHRlc3QzLmFsdDANBgkq
hkiG9w0BAQsFAAOCAYEAZ2BHv4jNC2A1mjQ+XQWGpgvcwOaKqgAqrgN8gWVFG1SEdQ0yUVpAzLJUWdofp4quvi
pkmia6bH/7paniEK6GqPEgljRw9HQ+rOHeXbPuxODDpsiT1ItWTk3l9biE8EOxewJM10jgNpkdOZ4IrWJ/LcSl
97fMHfaxdZvJaNqGVP4AW0LrA3PWfD3S0PMOjvSJLhEO2TIJSdjfrkqEfQQntEvGZfNMTeaXT5JLNEwHcoYWNx
HtF4SXtXUwugYLsn45CC7hyTrX7Cv3hE37Q1XeOTzCH8sVaYsneJX9uKsQ5O1zDvpVsHoSvUulKWeLIBuLRsOS
ECmut1vqi+Lixfi4eHg3ocghqdmS9pNdNRTFafoW4CK1PGD4cyXsBYJ40mVwj0GjXMGCmo0f3EF8f97tqlROre
UH3y3W/bD1jiYM2pMEE/8wngA+LXW1dP/vdrTJ2OujqsLphiN74HKFoooME0CkXbAVMz3zw80AAcmEbypcGc+V
UD9vHx78L8ZE6BB0
```

4.2.2. sssd.conf. Настройка службы sssd.

Добавим в секцию `[pam]` следующие параметры для 2ФА по сертификатам:

```
[pam]
pam_cert_auth = True
pam_p11_allowed_services = +mate-screensaver, +lightdm, +kf5-screenlocker, +sddm
pam_cert_db_path = /etc/pki/tls/certs/ca.pem
```

- `pam_cert_auth` — определяем, что следует проводит аутентификацию по сертификату
- `pam_p11_allowed_services` — определяем, каким службам sssd предоставит доступ к сертификату на токене

- `pam_cert_db_path` — указываем расположение сертификата УЦ

Так как аутентификация по токену занимает больше времени чем по паролю (поиск сертификата на токене, проверка сертификата по протоколу OCSP и т.п.) необходимо увеличить таймауты для Kerberos и для p11_child. Добавим следующие параметры:

```
[domain/test3.alt]
...
krb5_auth_timeout = 60
...
[pam]
...
p11_child_timeout = 60
```

В итоге конфигурационный файл sssd примет следующий вид:

```
client ~ # cat /etc/sssd/sssd.conf | grep -viE '(^#|^$|^;)'
[domain/test3.alt]
id_provider = ipa
ipa_server = srv, ipa.test3.alt
ipa_domain = test3.alt
ipa_hostname = client.test3.alt
auth_provider = ipa
chpass_provider = ipa
access_provider = ipa
cache_credentials = True
ldap_tls_cacert = /etc/ipa/ca.crt
krb5_store_password_if_offline = True
krb5_auth_timeout = 60
[sssd]
config_file_version = 2
services = nss, pam, ssh, sudo
user = _sssd
domains = test3.alt
[nss]
[ssh]
[sudo]
[pam]
pam_cert_auth = True
pam_p11_allowed_services = +mate-screensaver, +lightdm
pam_cert_db_path = /etc/pki/tls/certs/ca.pem
p11_child_timeout = 60
```

После изменения конфигурации перезапустим службу, очистив при этом кеш службы sssd:

```
client ~ # systemctl stop sssd && sss_cache -E \
    && systemctl start sssd
```

5. Проверяем 2ФА в текстовой консоли

При аутентификации по сертификату на токене, до приглашения ввести пароль ожидается приглашение ввода PIN-кода. Если PIN-код верный, аутентификация должна пройти успешно, иначе — отказ. При успешной аутентификации выдаётся действующий билет Kerberos.

5.1. Rutoken ECP

```
client login: rt
PIN for r_token_ecp2:
rt@client ~ $ klist
Ticket cache: KEYRING:persistent:1450000001:krb_ccache_v41ZW1c
Default principal: rt@TEST3.ALT

Valid starting       Expires              Service principal
06.04.2022 11:47:38  07.04.2022 11:47:34  krbtgt/TEST3.ALT@TEST3.ALT
rt@client ~ $ id
uid=1450000001(rt) gid=1450000001(rt) группы=1450000001(rt)
```

Проверяем отказ в аутентификации при вводе некорректного PIN-кода:

```
client login: rt
PIN for r_token_ecp2:   вводим неверный PIN-код
Login incorrect
```

5.2. ESMART Token

```
client login: es
PIN for esmart_64:
Last login: Fri Apr  1 13:32:07 MSK 2022 on ttyS0
es@client ~ $ klist
Ticket cache: KEYRING:persistent:1450000004:krb_ccache_947lzAG
Default principal: es@TEST3.ALT

Valid starting       Expires              Service principal
06.04.2022 11:53:40  07.04.2022 11:53:36  krbtgt/TEST3.ALT@TEST3.ALT
es@client ~ $ id
uid=1450000004(es) gid=1450000004(es) группы=1450000004(es)
```

Проверяем отказ в аутентификации при вводе некорректного PIN-кода:

```
login: es
PIN for esmart_64:  вводим неверный PIN-код
```

```
Login incorrect
```

5.3. JaCarta-2 SE

Для проверки работы службы sssd с токеном JaCarta-2 SE необходимо дополнительно указать слот. Добавьте в секцию `[pam]` параметр `p11_uri=pkcs11:model=JaCarta%20Laser`, указывающий слот по его модели, и перезапустите службу `sssd`.

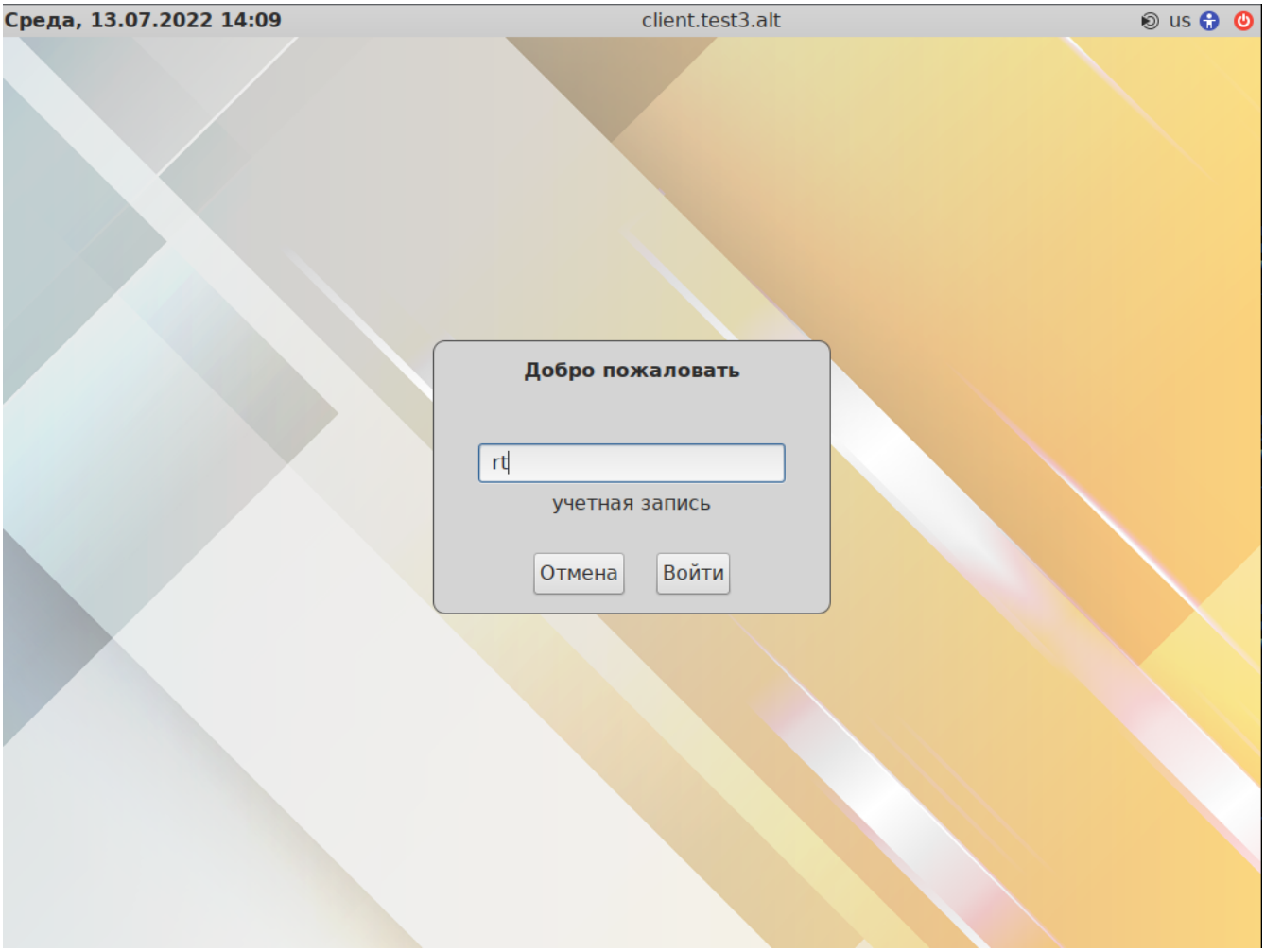
```
client login: jc
PIN for JC2SE-Laser:
Last login: Thu Apr 14 12:57:06 MSK 2022 on ttyS0
jc@client ~ $ klist
Ticket cache: KEYRING:persistent:1450000003:krb_ccache_52q9mhr
Default principal: jc@TEST3.ALT

Valid starting       Expires              Service principal
14.04.2022 12:58:07  15.04.2022 12:57:54  krbtgt/TEST3.ALT@TEST3.ALT
jc@client ~ $ id
uid=1450000003(jc) gid=1450000003(jc) группы=1450000003(jc)
```

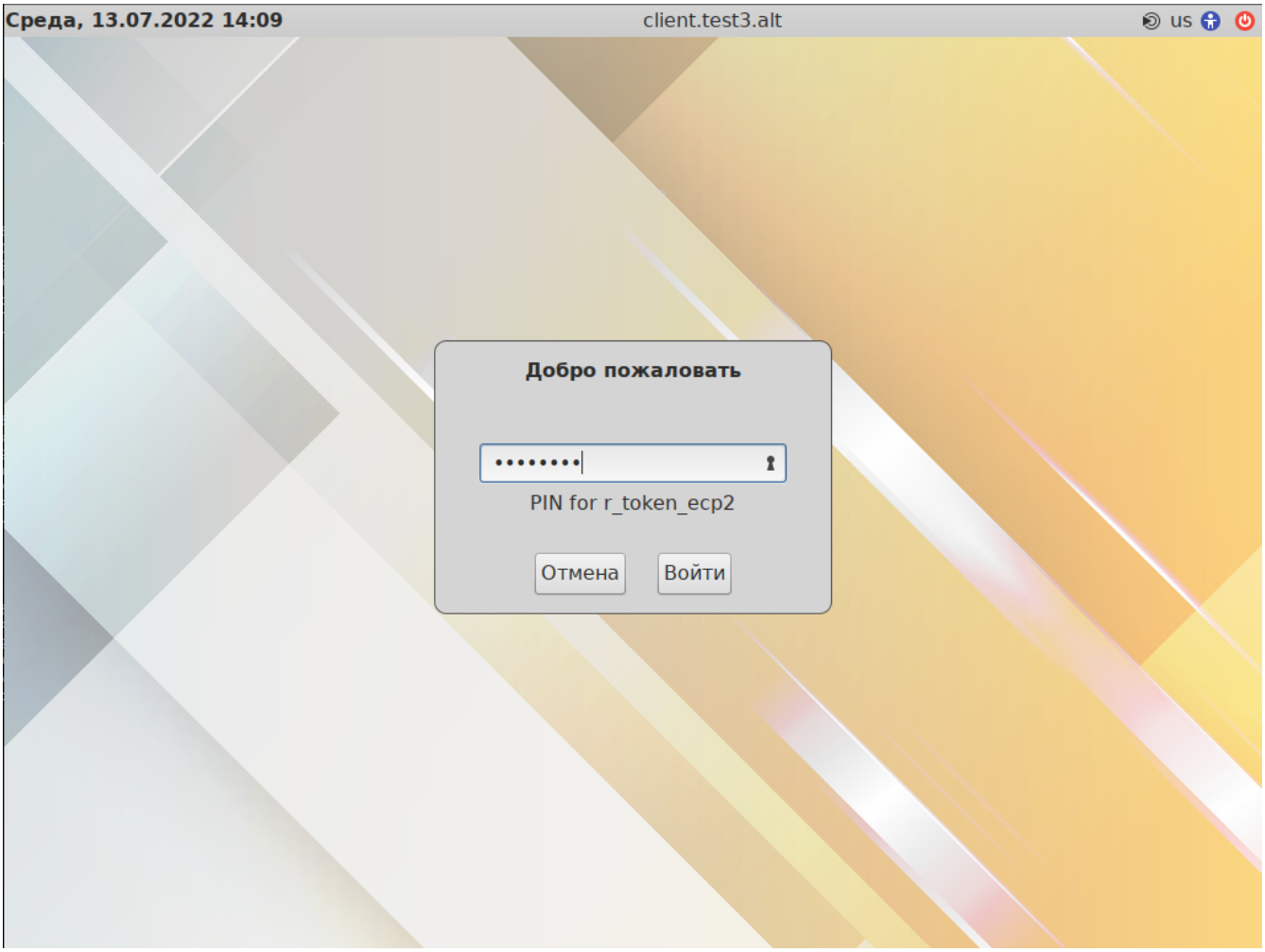
Проверяем отказ в аутентификации при вводе некорректного PIN-кода:

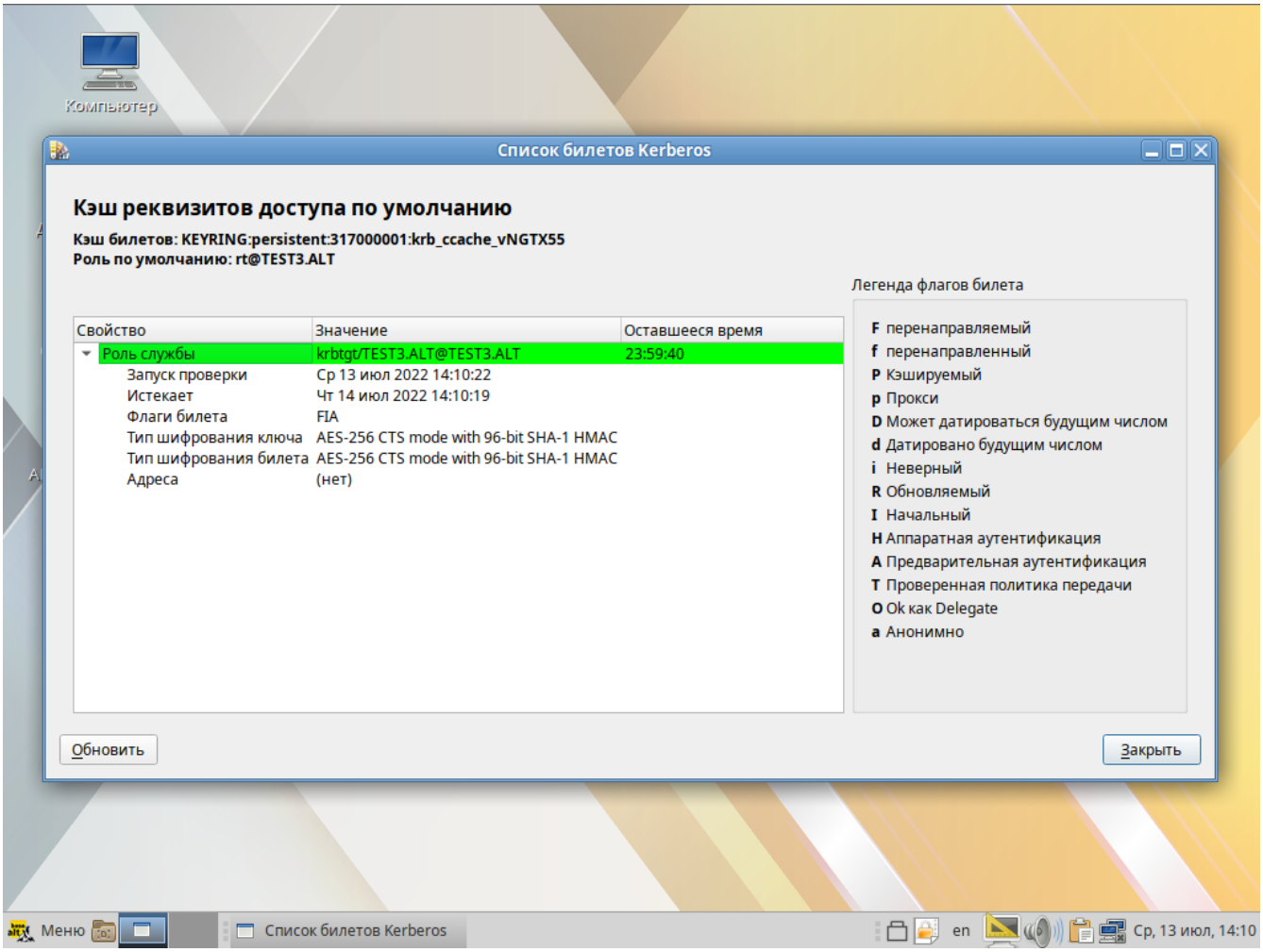
```
client login: jc
PIN for JC2SE-Laser:    вводим неверный PIN-код
Login incorrect
```

6. Проверяем 2ФА в графическом сеансе

6.1. Rutoken ECP

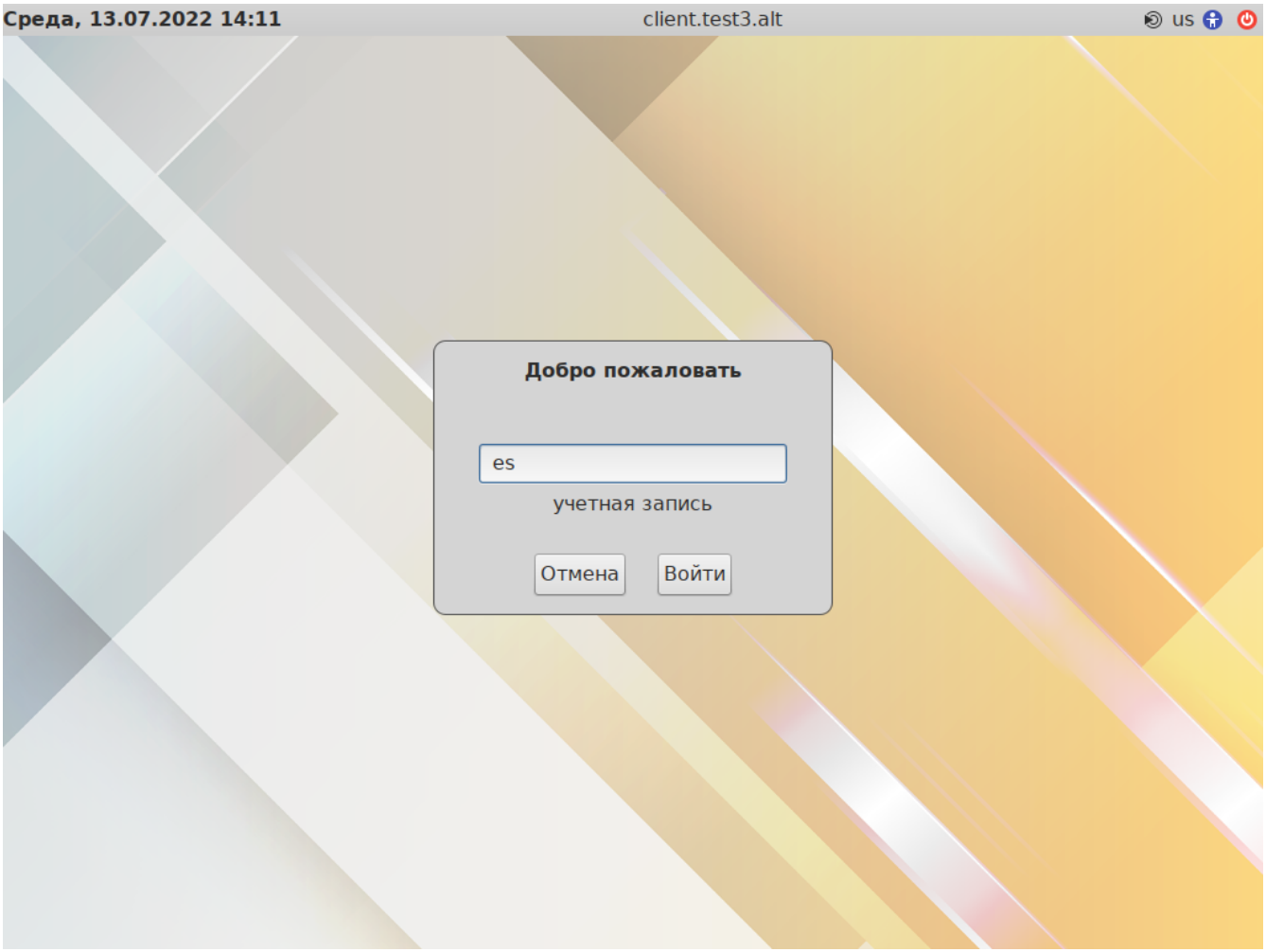


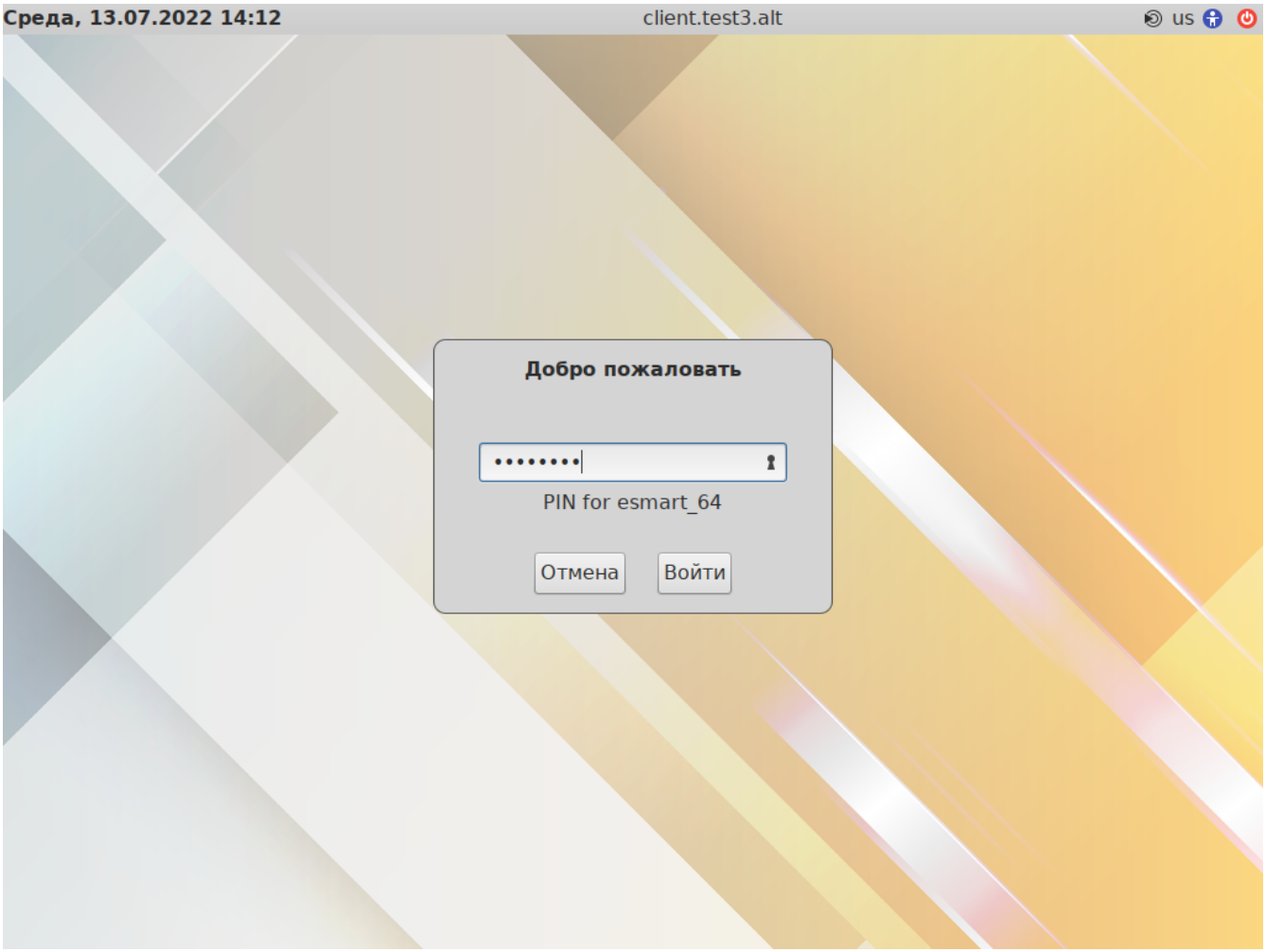

Среда, 13.07.2022 14:09
client.test3.alt
us
Добро пожаловать
PIN for r_token_ecp2
Отмена
Войти

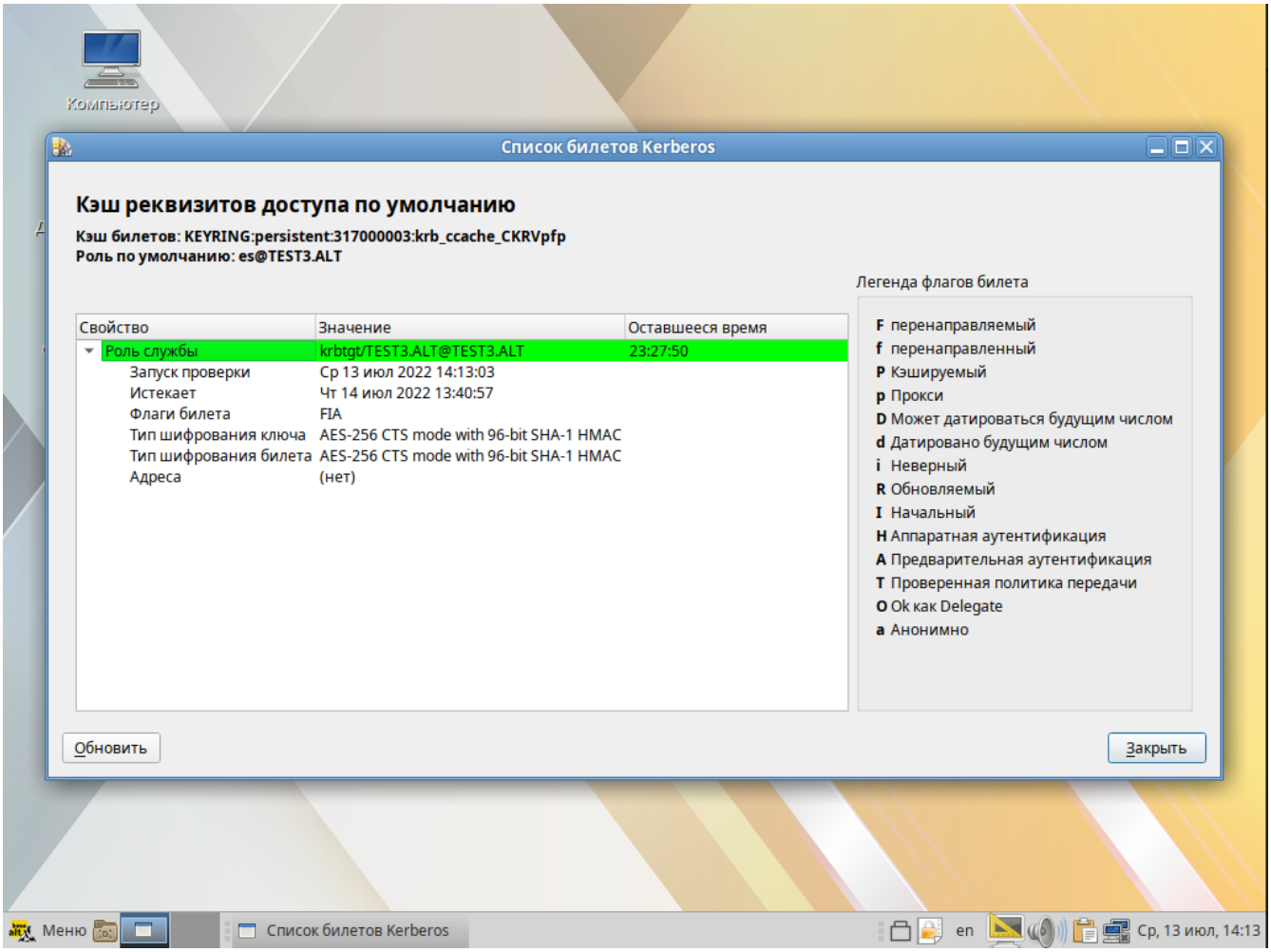


Разблокировка графического сеанса MATE по сертификату:

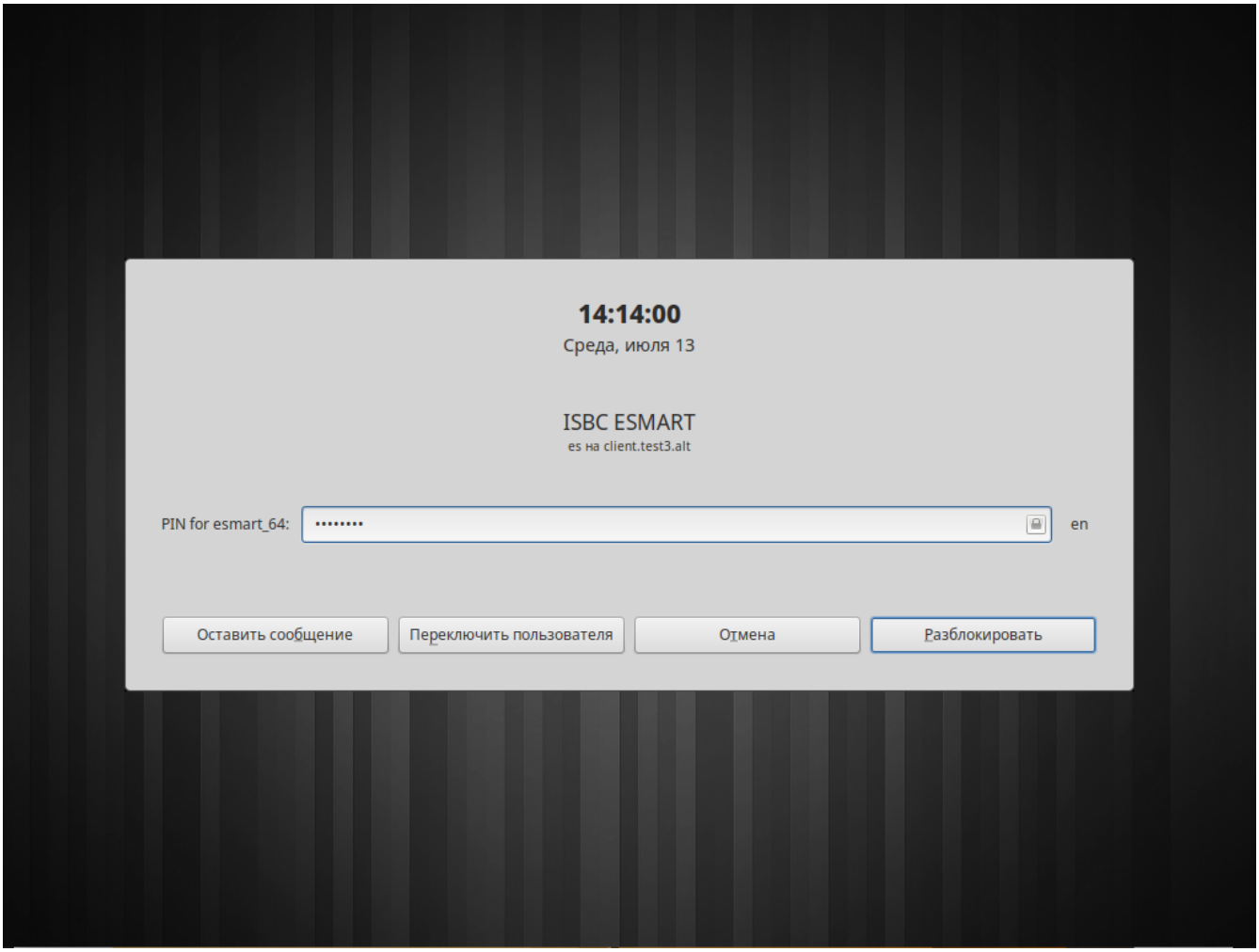
6.2. ESMART Token


Среда, 13.07.2022 14:11
client.test3.alt
us
Добро пожаловать
es
учетная запись
Отмена
Войти


Среда, 13.07.2022 14:12
client.test3.alt
us
Добро пожаловать
PIN for esmart_64
Отмена
Войти

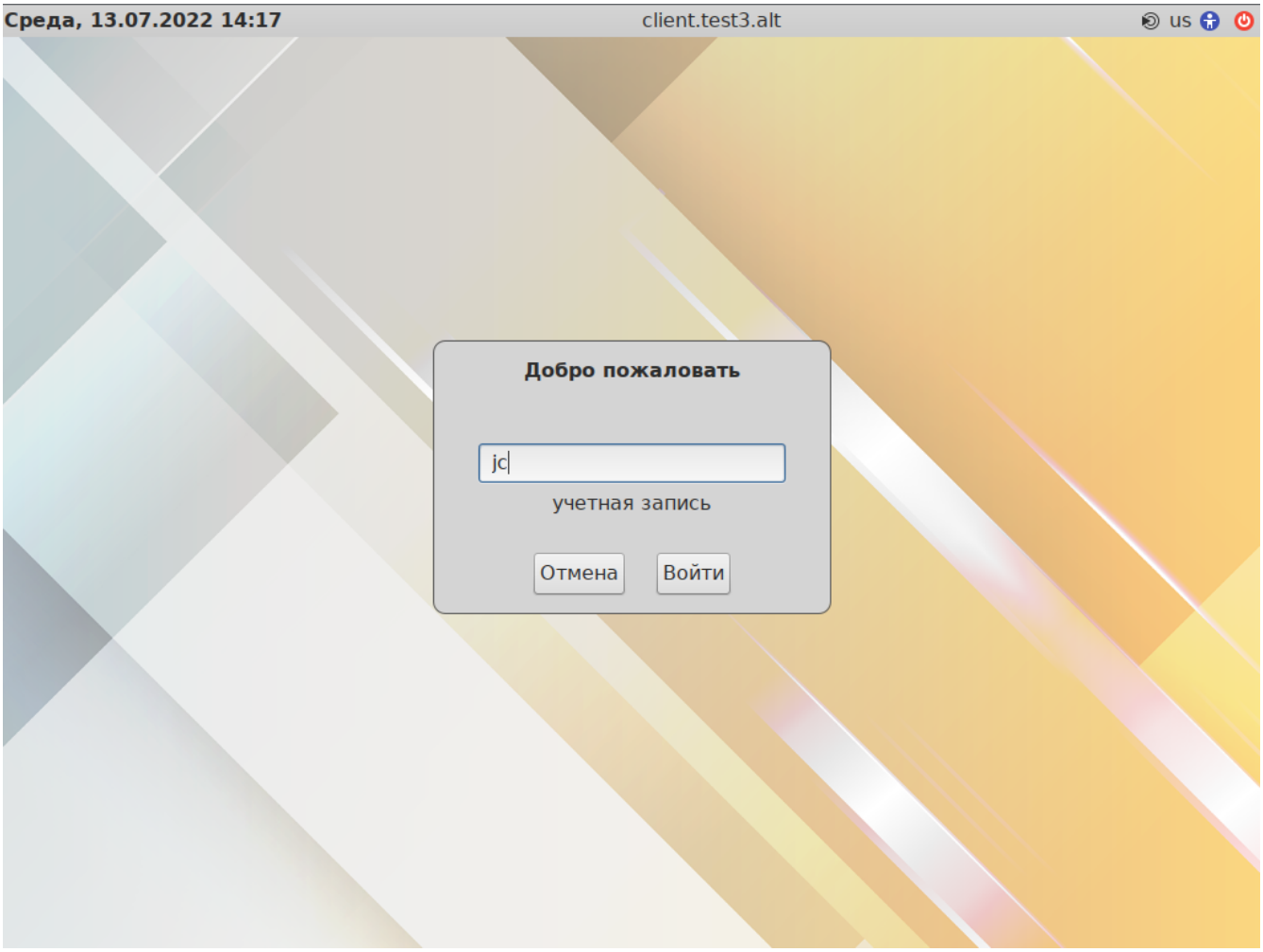


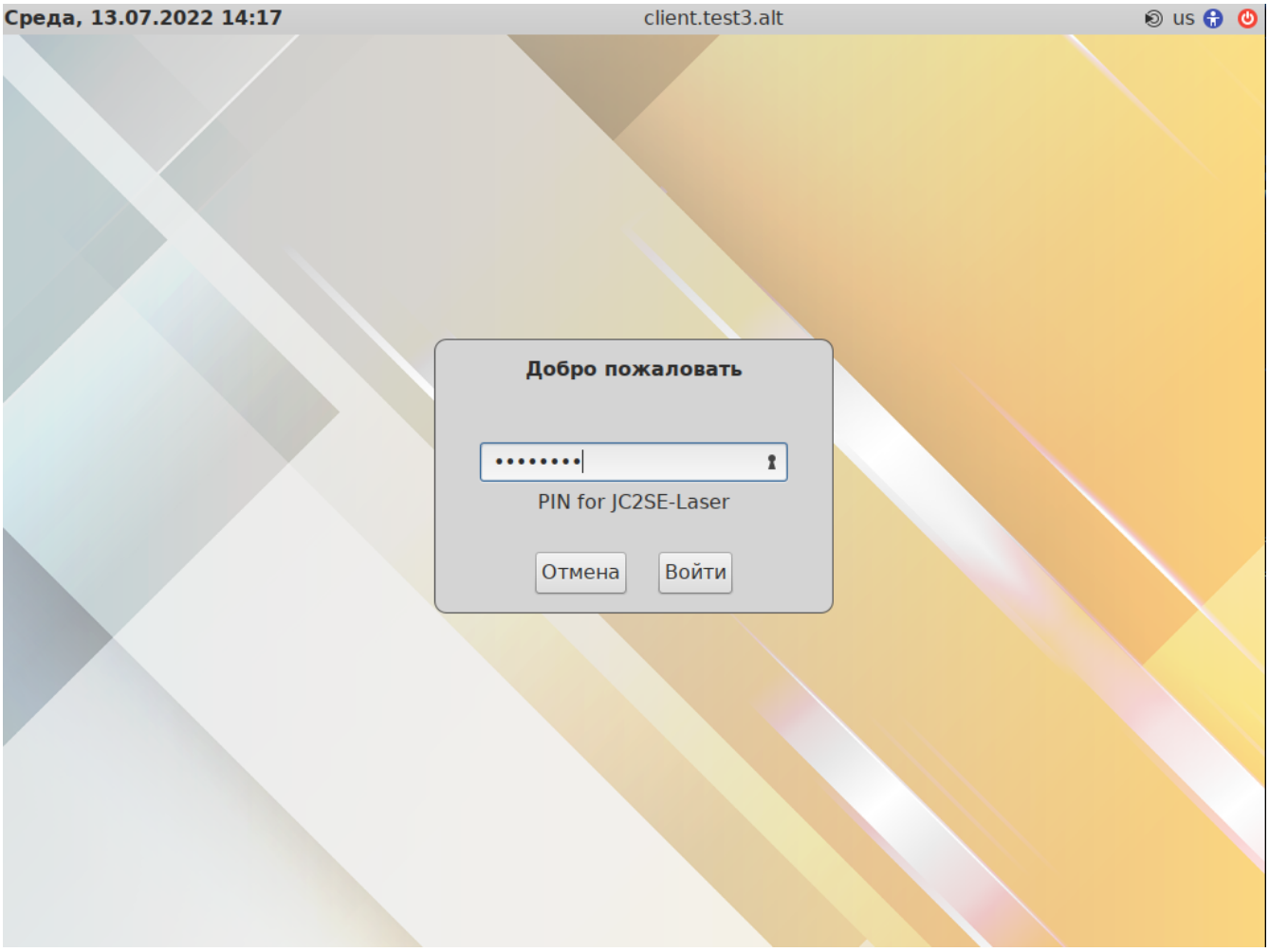
Разблокировка графического сеанса MATE по сертификату:

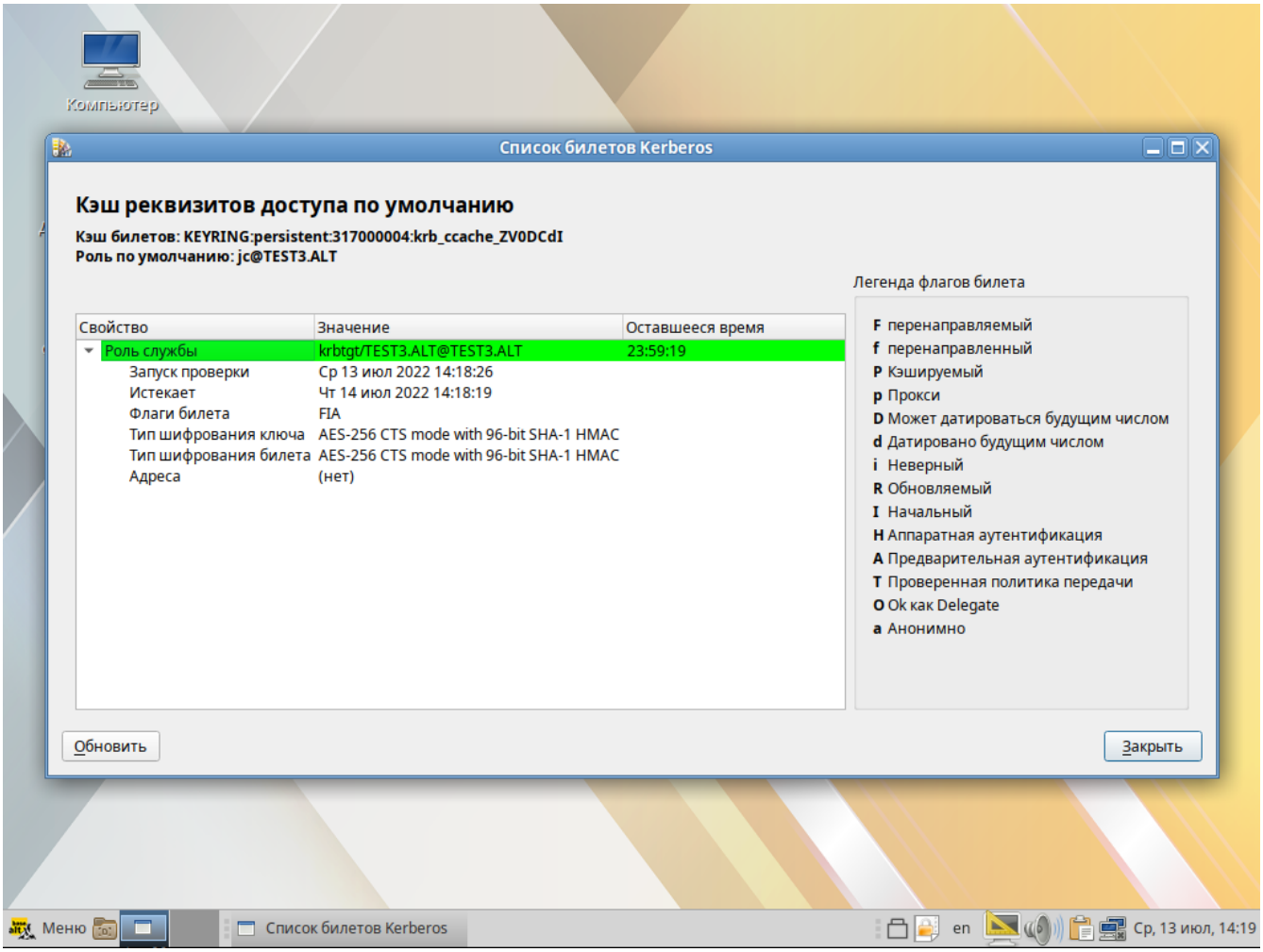


6.3. JaCarta-2 SE

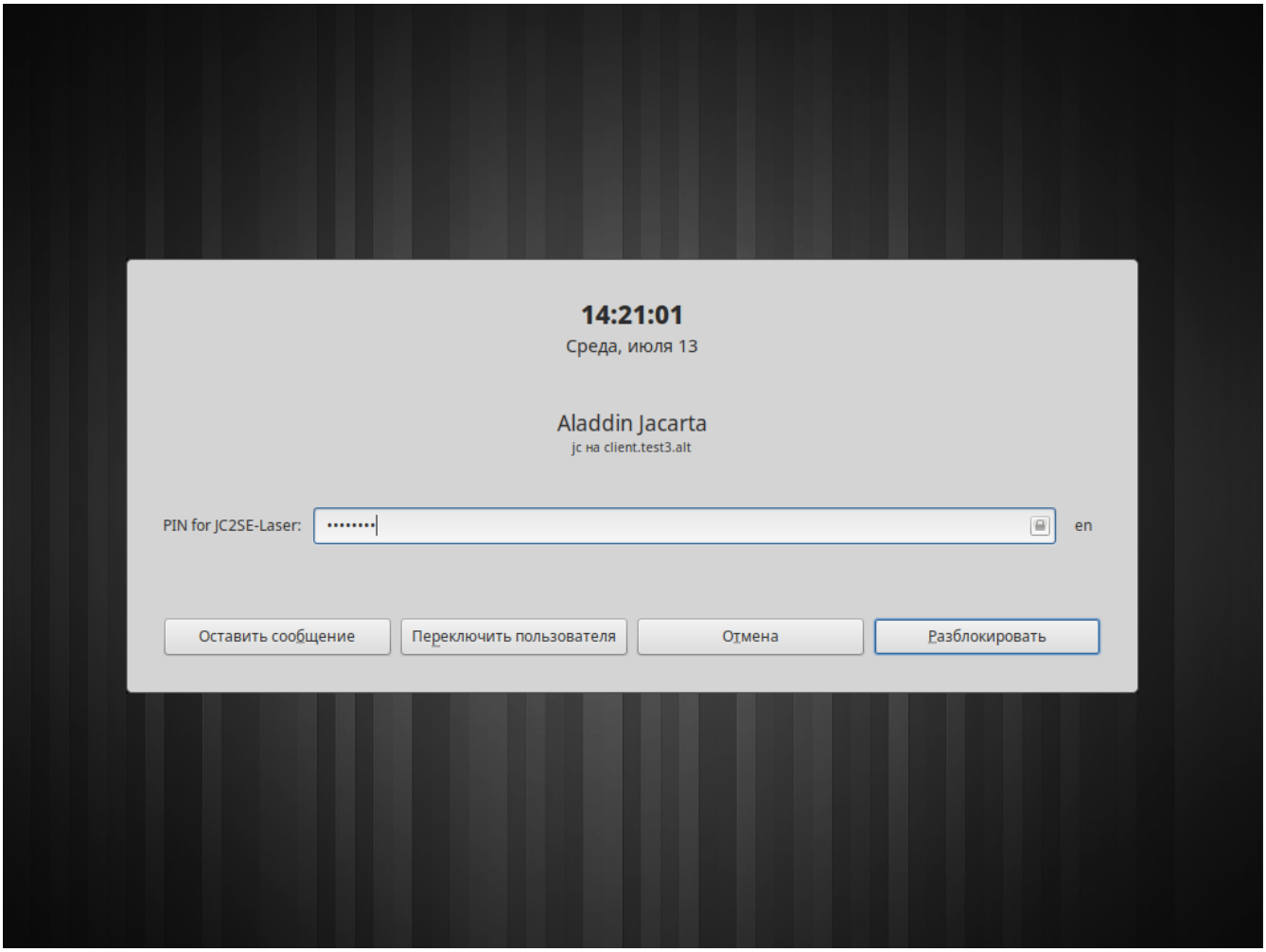
Для проверки работы службы sssd с токеном JaCarta-2 SE необходимо дополнительно указать слот. Добавьте в секцию `[pam]` параметр `p11_uri=pkcs11:model=JaCarta%20Laser`, указывающий слот по его модели, и перезапустите службу `sssd`.


Среда, 13.07.2022 14:17
client.test3.alt
us
Добро пожаловать
jc
учетная запись
Отмена
Войти


Среда, 13.07.2022 14:17
client.test3.alt
us
Добро пожаловать
PIN for JC2SE-Laser
Отмена
Войти



Разблокировка графического сеанса MATE по сертификату:


14:21:01
Среда, июля 13
Aladdin Jacarta
jc на client.test3.alt
PIN for JC2SE-Laser:
en
Оставить сообщение
Переключить пользователя
Отмена
Разблокировать

7. Проверка отзыва сертификата

Отзываем сертификат на сервере FreeIPA (на примере пользователя **rt**, Rutoken ECP), указывая в запросе серийный номер сертификата:

```
user@ipa ~ $ kinit admin
Password for admin@TEST3.ALT:
user@ipa ~ $ ipa cert-revoke
Serial number: 11
  Revoked: True
```

При этом сам сертификат остается в LDAP. Проверка валидности сертификата осуществляется по протоколу OCSP.

Проверяем 2ФА в консоли:

```
client login: rt
Password:
```

Так как сертификат отозван, то запроса PIN-кода нет.
Увидеть это можно в `/var/log/sssd/p11_child.log`:

```
...
   *  (2022-04-06 14:35:34): [p11_child[5553]] [read_certs] (0x4000): found
cert[rt_2fa_ipa][/O=TEST3.ALT/CN=rt]
   *  (2022-04-06 14:35:34): [p11_child[5553]] [do_ocsp] (0x4000): Using OCSP URL
[http://ipa-ca.test3.alt/ca/ocsp].
   *  (2022-04-06 14:35:35): [p11_child[5553]] [do_ocsp] (0x4000): Nonce in OCSP
response is the same as the one used in the request.
   *  (2022-04-06 14:35:35): [p11_child[5553]] [do_ocsp] (0x0020): OCSP check failed
with [1][revoked].
...
(2022-04-06 14:35:35): [p11_child[5553]] [read_certs] (0x0040): Certificate
[rt_2fa_ipa][/O=TEST3.ALT/CN=rt] not valid, skipping.
```

8. Отладка sssd

Если аутентификация по токену не работает как ожидается, то для более полной информации по этому процессу добавьте опцию вывода отладочной информации в файлы логов службы sssd. Логи службы sssd находятся в директории `/var/log/sssd/*`. Добавление опции вывода отладочной информации необходимо добавить в те секции конфигурационного файла `sssd.conf`, которые необходимо исследовать. Выглядеть это может, примерно, так:

```
[domain/test3.alt]
debug_level = 8
...
[sssd]
debug_level = 8
...
[nss]
[ssh]
[sudo]
[pam]
debug_level = 8
...
```

`debug_level = 8` — уровень отладки при котором выводится содержимое внутренних переменных функций службы sssd.

Более подробно об уровнях отладки см. `man sssd.conf`.

9. Полезные ссылки

- https://www.freeipa.org/page/Howto/Client_Certificate_Authentication_with_LDAP
- https://github.com/OpenSC/libp11
- https://sssd.io/troubleshooting/basics.html
- https://floblanc.wordpress.com/2017/06/02/freeipa-troubleshooting-smartcard-authentication/
- https://datatracker.ietf.org/doc/html/rfc7512#section-2.1
- https://www.mankier.com/5/sss-certmap
- https://sssd.io/troubleshooting/basics.html
- https://www.freeipa.org/page/Troubleshooting/Kerberos
- https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/linux_domain_identity_authentication_and_policy_guide/trouble-authentication