JaCarta: Astra Linux Directory

- Kerberos
- kdc
.

:

- Astra Linux Special Edition .10015-01 (1.7), .10015-10
- Astra Linux Special Edition .10015-17
- Astra Linux Special Edition .10015-37 (7.7)
- Astra Linux Special Edition .10015-03 (7.6)
- Astra Linux Special Edition .10152-02 (4.7)
- Astra Linux Special Edition .10015-01 (1.6) 20190222SE16 (2)
- Astra Linux Special Edition .10015-16 . 1
- Astra Linux Special Edition .10015-16 . 2
- Astra Linux Special Edition .10265-01 (8.1)
- Astra Linux Common Edition 2.12

:

- ();
- ;
- ;

.

, ;

, Astra Linux Directory (ALD) , , , (ntp).

Kerberos

- (ticket) – , , ;
- (client) – (,), Kerberos;
- (key distribution center, KDC) – , Kerberos;
- (realm) – , Kerberos, KDC . realm , ;
- (principal) – , Kerberos. root[/instance]@REALM.

, (, , ,) jcpkcs11-2. Astra Linux. : https://www.aladdin-rd.ru.

1. jcpkcs11-2 ;
2. Astra Linux:

```
sudo apt install -y opensc libengine-pkcs11-openssl1.1 pcsc-tools
```

OpenSSL — SSL/TLS. RSA, DH, DSA, X.509, , CSR CRT. SMARTCARD.ALD. , :

- SMARTCARD.ALD;
- - kdc;
- - client.

, .

:

1. () :

```
sudo mkdir /etc/ssl/CA
```

   .

2. :

```
cd /etc/ssl/CA
```

3. , -subst:

> ⚠ **Common name** (CN) . : SMARTCARD.ALD

```
sudo openssl genrsa -out cakey.pem 2048
sudo openssl req -batch -new -key cakey.pem  -out cacert.pem -subj '
/C=RU/ST=MO/L=Moscow/O=Astra/OU=Wiki/emailAddress= /CN=SMARTCARD.ALD' -
x509 -days 3650
```

> ⚠ days .

kdc

1. KDC, -subst:

> ⚠ Common name (CN) kdc.

```
sudo openssl genrsa -out kdckey.pem 2048
sudo openssl req -batch -new -key kdckey.pem -out kdc.req    -subj '
/C=RU/ST=MO/L=Moscow/O=Astra/OU=Wiki/emailAddress= /CN=kdc'
```

2. . , , :

```
export REALM=SMARTCARD.ALD
export CLIENT=kdc
```

3. , :

```
env | grep -wE "REALM|CLIENT"
```

4. **pkinit_extensions** : **pkinit_extensions.**

```
[ kdc_cert ]
basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
keyUsage = nonRepudiation, digitalSignature, keyEncipherment, keyAgreement

#Pkinit EKU
extendedKeyUsage = 1.3.6.1.5.2.3.5

subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer

# Copy subject details

issuerAltName=issuer:copy

# Add id-pkinit-san (pkinit subjectAlternativeName)
subjectAltName=otherName:1.3.6.1.5.2.2;SEQUENCE:kdc_princ_name

[kdc_princ_name]
realm = EXP:0, GeneralString:${ENV::REALM}
principal_name = EXP:1, SEQUENCE:kdc_principal_seq

[kdc_principal_seq]
name_type = EXP:0, INTEGER:1
name_string = EXP:1, SEQUENCE:kdc_principals

[kdc_principals]
princ1 = GeneralString:krbtgt
princ2 = GeneralString:${ENV::REALM}

[ client_cert ]

# These extensions are added when 'ca' signs a request.

basicConstraints=CA:FALSE

keyUsage = digitalSignature, keyEncipherment, keyAgreement

extendedKeyUsage =  1.3.6.1.5.2.3.4
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer


subjectAltName=otherName:1.3.6.1.5.2.2;SEQUENCE:princ_name


# Copy subject details

issuerAltName=issuer:copy

[princ_name]
realm = EXP:0, GeneralString:${ENV::REALM}
principal_name = EXP:1, SEQUENCE:principal_seq

[principal_seq]
name_type = EXP:0, INTEGER:1
name_string = EXP:1, SEQUENCE:principals

[principals]
princ1 = GeneralString:${ENV::CLIENT}
```

5. KDC:

```
sudo openssl x509 -req -in kdc.req -CAkey cakey.pem -CA cacert.pem -out kdc.pem -extfile pkinit_extensions -extensions kdc_cert -CAcreateserial -days 365
```

1. , . (PKI GOST), (libjcPKCS11-2.so libASEP11.so). , :

```
dpkg -L jcpkcs11-2 | egrep "(libjcPKCS11-2.so|libASEP11.so)"
```

```
/usr/lib/libASEP11.so
/usr/lib/libjcPKCS11-2.so.2.7.3
```

, /usr/lib/.

2. JaCarta:

   a. libjcPKCS11-2.so (JaCarta PKI//FLASH):

   ```
   pkcs11-tool --module /usr/lib/libjcPKCS11-2.so -T
   ```

   b. libASEP11.so (JaCarta PKI):

   ```
   pkcs11-tool --module /usr/lib/libASEP11.so -T
   ```

3. :

   ! JaCarta PKI .

   ```
   pkcs11-tool --slot 0x1ffff --init-token --so-pin 00000000 --label
   'JaCarta PKI' --module /usr/lib/libjcPKCS11-2.so
   ```

4. - :

   ```
   pkcs11-tool --slot 0x1ffff --init-pin --so-pin 00000000 --login --pin
   11111111 --module /usr/lib/libjcPKCS11-2.so
   ```

   --slot 0x1ffff — . , 0, – 1,2 ..;
   --init-token – ;
   --pin - JaCarta. 11111111;
   --so-pin 00000000 – JaCarta PKI. 00000000;
   --label 'JaCarta PKI' - () ;
   --module - libjcPKCS11-2.so

5. :

   ```
   pkcs11-tool --slot 0x1ffff --login --pin 11111111 --keypairgen --key-
   type rsa:2048 --id 33 --label “2fa_test1_key” --module /usr/lib/libjcPKC
   S11-2.so
   ```

   -keypairgen --key-type rsa:2048 — , RSA 2048 ;
   --id 33 — CKA_ID . CKA_ID ;
   --label “test1 key” — CKA_LABEL() . ;

6. openssl. :

Astra Linux Special Edition .10015-01 (1.6) pkcs11 libengine-pkcs11-openssl 1.0.2 libjcPKCS11-2.so. , :

- libengine-pkcs11-openssl1.1 0.4.4-4 Astra Linux Special Edition .10015-01 (1.6): libengine-pkcs11-openssl1.1_0.4.4-4_amd64.deb
- Astra Linux Special Edition .10015-01 (1.6)

<__>:<id>. 0x1, 0x2 .., 0x1ffff, 0x2ffff .. 0x1ffff 131071 ( ):

```
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/libjcPKCS11-2.so
(dynamic) Dynamic engine loading support
[Success]: SO_PATH:/usr/lib/x86_64-linux-gnu/engines-1.1/pkcs11.so
[Success]: ID:pkcs11
[Success]: LIST_ADD:1
[Success]: LOAD
[Success]: MODULE_PATH:/usr/lib/libjcPKCS11-2.so
Loaded: (pkcs11) pkcs11 engine
OpenSSL> req -engine pkcs11 -new -key 131071:33 -keyform engine -out client.req
engine "pkcs11" set.
Enter PKCS#11 token PIN for JaCarta ECP <AstraLinux>:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:RU
State or Province Name (full name) [Some-State]:Moscow
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]: Rusbitech
Organizational Unit Name (eg, section) []: Astra
Common Name (e.g. server FQDN or YOUR name) []:test1 (!_!)
Email Address []:*********

OpenSSL> exit
```

-new -key 131071:33, **131071**— (0x1ffff) , **33**— CKA_ID .

CN =

:

```
pkcs11-tool --module /usr/lib/libjcPKCS11-2.so -L
```

openssl , , : 131071:33

7. :
   a. :

```
export REALM=<_>
export CLIENT=<__>
```

b. :

```
env | grep -wE "REALM|CLIENT"
```

c. :

```
sudo openssl x509 -CAkey cakey.pem -CA cacert.pem -req -in client.
req -extensions client_cert -extfile pkinit_extensions -out client.
pem -days 365
```

d. PEM DER:

```
sudo openssl x509 -in client.pem -out client.cer -inform PEM -
outform DER
```

e. :

```
pkcs11-tool --slot 0x1ffff --login --pin 11111111 --write-object
client.cer --type 'cert' --label 'test1' --id 33 --module /usr/lib
/libjcPKCS11-2.so
```

⚠ **--write-object ./client.cer**— , ;
--type 'cert'— , - ;
'cert' --label 'test1'— CKA_LABEL() . ;

1. kdc.pem, kdckey.pem, cacert.pem /var/lib/krb5kdc/ ;
2. /etc/krb5kdc/kdc.conf /etc/krb5kdc/kdc.conf, [kdcdefaults] :

```
pkinit_identity = FILE:/var/lib/krb5kdc/kdc.pem,/var/lib/krb5kdc/kdckey.pem
pkinit_anchors = FILE:/var/lib/krb5kdc/cacert.pem
```

, . :

```
sudo sed -i.`date +%Y-%m%d-%H:%M:%S` "/\[kdcdefaults\]/a \
pkinit_anchors = FILE:/var/lib/krb5kdc/cacert.pem" /etc/krb5kdc/kdc.conf
sudo sed -i                              "/\[kdcdefaults\]/a \
pkinit_identity = FILE:/var/lib/krb5kdc/kdc.pem,/var/lib/krb5kdc/kdckey.
pem" /etc/krb5kdc/kdc.conf
```

3. :

```
sudo systemctl restart krb5-admin-server
sudo systemctl restart krb5-kdc
```

1. /etc/krb5/:

```
sudo mkdir /etc/krb5/
```

2. /etc/krb5/ (cacert.pem) c , ;
3. /etc/krb5.conf [libdefaults] :

```
[libdefaults]
default_realm = SMARTCARD.ALD
pkinit_anchors = FILE:/etc/krb5/cacert.pem
#
pkinit_identities = PKCS11:/usr/lib/libjcPKCS11-2.so
```

4. ;
5. :

```
kinit
```

6. PIN- - PIN-;
7. , Kerberos , :

```
klist
```

8. :

```
kdestroy
```

⚠ kinit :

```
env KRB5_TRACE=/dev/stdout kinit <_>
```

.

Login , Password <PIN >. , , <PIN >:

```
login <_>
```

---

pam_krb5.so **/etc/pam.d/common-auth** pam_krb5.so**:**

```
# here are the per-package modules (the "Primary" block)
auth    [success=4 default=ignore]      pam_krb5.so minimum_uid=2500 use_pkinit
auth    [success=1 default=ignore]      pam_succeed_if.so quiet user ingroup astra-admin
auth    [success=ignore default=die]    pam_tally.so per_user deny=8
auth    [success=1 default=ignore]      pam_unix.so nullok_secure try_first_pass
# here's the fallback if no module succeeds
```

– try_pkinit — PKCS-11, Kerberos ;
– use_pkinit — PKCS-11, ;
– pkinit_prompt — PKCS-11 .

⚠ pam-auth-update, pkinit . , , /usr/share/pam-configs/krb5 Auth-Initial .

```
Name: Kerberos authentication
Default: yes
Priority: 704
Conflicts: krb5-openafs
Auth-Type: Primary
Auth:
        [success=end default=ignore]    pam_krb5.so minimum_uid=2500 try_first_pass
Auth-Initial:
        [success=end default=ignore]    pam_krb5.so minimum_uid=2500 use_pkinit
```

. man.

- Kerberos
-