Kerberos-

Related links

- https://help.ubuntu.com/community/Kerberos
- https://help.ubuntu.com/10.04/serverguide/kerberos.html
- : http://k5wiki.kerberos.org/wiki/Pkinit_configuration

- Key Distribution Center (KDC) -
- Admin server - kerberos. KDC admin server
- Realm - "",
- Principal - , .

1. USB- .
2. USB- :

```
$ lsusb
```

USB-:

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

, : **Aktiv Rutoken ECP**

- Ubuntu 12.10 x86
- Astra Linux 1.3 x64

: . .

Ubuntu

- `<username> = testuser`
- `<realm> = AKTIV-TEST`
- `<server> = aktiv-test.ru`

- `krb5-kdc, krb5-admin-server, krb5-pkinit`

- Kerberos realm: AKTIV-TEST, aktiv-test.ru (/etc/hosts)
  :
- : testuser@AKTIV-TEST

- krb5-user, krb5-config, krb5-pkinit
- default realm: AKTIV-TEST
- (kdc, admin) IP- (/etc/hosts)

Astra Linux 1.3

- <username> = test1
- <realm> = RUSBITECH.RU
- <server> = server.rusbitech.ru

(openct, opensc)

- Kerberos realm: RUSBITECH.RU, server.rusbitech.ru (/etc/hosts)
- : test1@RUSBITECH.RU

ald/kerberos krb5-pkinit (krb5-pkinit_1.10.1+dfsg-3_amd64.deb, Debian Wheezy):

```
$ dpkg -i --force-depends krb5-pkinit_1.10.1+dfsg-3_amd64.deb
```

1. /* Ubuntu */ realm

Astra Linux krb5-pkinit

1.1

```
$ sudo apt-get install krb5-kdc krb5-admin-server krb5-pkinit
#   :
# realm = AKTIV-TEST
#  = aktiv-test.ru
$ sudo krb5_newrealm
#
```

1.2

```
$ sudo apt-get install krb5-user libpam-krb5 libpam-ccreds auth-client-config krb5-pkinit
#   :
# realm = AKTIV-TEST
#  = aktiv-test.ru
$ sudo dpkg-reconfigure krb5-config
```

1.3 [domain_realm] /etc/krb5.conf

```
[domain_realm]
        .aktiv-test.ru = AKTIV-TEST
        aktiv-test.ru = AKTIV-TEST
```

2. /* Ubuntu */ kadmin.local

```
$ sudo kadmin.local
# username = testuser
# password = test
kadmin.local:$ addprinc <username>
# ...
kadmin.local:$ quit
```

3. ,

```
$ kinit <username>
...
$ klist
...
$ kdestroy
```

4. , ,

```
$ kinit <username>@<realm>
...
$ klist
...
$ kdestroy
```

5.

5.1

5.1.1 CA:

```
$ openssl genrsa -out cakey.pem 2048
$ openssl req -key cakey.pem -new -x509 -out cacert.pem
```

5.1.2 KDC:

```
$ openssl genrsa -out kdckey.pem 2048
#
$ openssl req -new -out kdc.req -key kdckey.pem
#
$ REALM=<realm>; export REALM
$ CLIENT=<server>; export CLIENT
#   pkinit_extensions
$ openssl x509 -req -in kdc.req -CAkey cakey.pem -CA cacert.pem -out kdc.pem -extfile pkinit_extensions -
extensions kdc_cert -CAcreateserial
```

5.1.3 /var/lib/krb5kdc/:

- kdc.pem
- kdckey.pem
- cacert.pem

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}
```

5.1.4 preauth .

Astra Linux: kdcdefaults :

/etc/krb5kdc/kdc.conf

```
[kdcdefaults]
    kdc_tcp_ports = 88
    pkinit_identity = FILE:/var/lib/krb5kdc/kdc.pem,/var/lib/krb5kdc/kdckey.pem
    pkinit_anchors = FILE:/var/lib/krb5kdc/cacert.pem
[realms]
    AKTIV-TEST = {
        database_name = /var/lib/krb5kdc/principal
        admin_keytab = FILE:/etc/krb5kdc/kadm5.keytab
        acl_file = /etc/krb5kdc/kadm5.acl
        key_stash_file = /etc/krb5kdc/stash
        max_life = 10h 0m 0s
        max_renewable_life = 7d 0h 0m 0s
        master_key_type = des3-hmac-sha1
        supported_enctypes = aes256-cts:normal arcfour-hmac:normal des3-hmac-sha1:normal des-cbc-crc:normal des:
normal des:v4 des:norealm des:onlyrealm des:afs3
        default_principal_flags = +preauth
    }
```

5.1.5 preauth

```
$ kadmin.local
kadmin.local$: modprinc +requires_preauth <username>
```

5.2

5.2.1 CA (cacert.pem) c /etc/krb5/

5.2.2

```
$ pkcs15-init --erase-card -p rutoken_ecp
$ pkcs15-init --create-pkcs15 --so-pin "87654321" --so-puk ""
$ pkcs15-init --store-pin --label "User PIN" --auth-id 02 --pin "12345678" --puk "" --so-pin "87654321" --
finalize
```

5.2.3

```
#    ID!
$ pkcs15-init -G rsa/2048 --auth-id 02 --id 42 --label "testuser's key" --public-key-label "testuser's public
key"
# ...

$ openssl
# NB: !  multiarch- opensc-pkcs11.so  engine_pkcs11.so
OpenSSL> engine dynamic -pre SO_PATH:/usr/lib/openssl/engines/engine_pkcs11.so -pre ID:pkcs11 -pre LIST_ADD:1 -
pre LOAD -pre MODULE_PATH:opensc-pkcs11.so
(dynamic) Dynamic engine loading support
[Success]: SO_PATH:/usr/lib/openssl/engines/engine_pkcs11.so
[Success]: ID:pkcs11
[Success]: LIST_ADD:1
[Success]: LOAD
[Success]: MODULE_PATH:opensc-pkcs11.so
Loaded: (pkcs11) pkcs11 engine
OpenSSL> req -engine pkcs11 -new -key 1:42 -keyform engine -out client.req -subj "/C=RU/ST=Moscow/L=Moscow
/O=Aktiv/OU=dev/CN=testuser/emailAddress=testuser@mail.com"
engine "pkcs11" set.
PKCS#11 token PIN:
OpenSSL> quit
```

5.3

5.3.1 (client.req) CA:

```
$ REALM=<realm>; export REALM
$ CLIENT=<username>; export CLIENT
$ openssl x509 -CAkey cakey.pem -CA cacert.pem -req -in client.req -extensions client_cert -extfile
pkinit_extensions -out client.pem
```

5.3.2 kdc:

```
 $ /etc/init.d/krb5-admin-server restart
 $ /etc/init.d/krb5-kdc restart
```

5.4

5.4.1 (client.pem) /etc/krb5/

5.4.2

```
$ pkcs15-init --store-certificate client.pem --auth-id 02 --id 42 --format pem
```

5.4.3 kerberos

/etc/krb5.conf

```
[libdefaults]
        default_realm = <realm>
        pkinit_anchors = FILE:/etc/krb5/cacert.pem
#
#        pkinit_identities = FILE:/etc/krb5/client.pem,/etc/krb5/clientkey.pem
#
        pkinit_identities = PKCS11:/usr/lib/opensc-pkcs11.so
```

5.4.4

```
$ kinit <username>
```