Настройка Kerberos-аутентификации

Пользователям Astra Linux

→ Настройка двухфакторной аутентификации в домене Astra Linux Directory

Введение

Related links

- https://help.ubuntu.com/community/Kerberos
- Про аутентификацию с использованием сертификатов здесь: http://k5wiki.kerberos.org/wiki/Pkinit_configuration

Основные понятия

- Key Distribution Center (KDC) - хранилище информации о паролях пользователей.
- Admin server - основной сервер kerberos. У нас KDC и admin server находятся на одной машине.
- Realm - "среда", в которой производится аутентификация.
- Principal - пользователь или сервис, участвующий в механизме аутентификации. Мы пока рассматриваем только пользователей.

Перед настройкой

Центральной частью схемы аутентификации Kerberos является третья доверенная сторона - **Key Distribution Center (KDC)**, которая является централизованным хранилищем информации о пользователях. Перед разворачиванием Kerberos, должен быть выбран сервер, который будет выполнять роль KDC. Физическая и сетевая безопасность критичны для этого сервера, так как его компрометация ведет к компрометации всего realm.

Выбор хорошего имени для realm так же важен. По правилам, имя realm это доменное имя сайта в верхнем регистре. Например, для сайта или доменной зоны example.com рекомендуется выбрать EXAMPLE.COM в качестве имени realm.

Все серверы и клиенты, которые входят в realm Kerberos должны иметь возможность взаимодействовать между собой. Время между устройствами в realm должно быть синхронизовано. Далее описано как этого добиться.

Host Names

Каждый сервер внутри Kerberos realm должен иметь **Fully Qualified Domain Name (FQDN)**.

Kerberos так же ожидает, что FQDN сервера является reverse-resolvable. Если выяснение доменного имени по IP недоступно, то установите значение переменной rdns в значение false на клиентах в файле krb5.conf.

Active Directory сильно зависит от DNS, поэтому весьма вероятно что ваш Active Directory Domain Controller уже имеет роль DNS. В этом случае убедитесь в том, что каждый сервер имеет свое FQDN перед выполнением тестов, описанных ниже в этом разделе.

Если сервер уже имеет назначенное FQDN, проверьте корректность обнаружения forward и reverse выполнив на клиенте следующие команды:

```
$ nslookup server.example.com
$ nslookup <server ip address>
```

Если вы используете Astra Linux (или другой дистрибутив), то для установки программы nslookup, вам необходимо установить пакет dnsutils.

Вы можете воспользоваться Synaptic Package Manager или выполнить из командной строки $ apt-get install dnsutils

Вывод первой команды должен содержать IP адрес сервера. Вывод второй команды должен содержать FQDN сервера.

Если у сервера нет назначенного FQDN и сервис DNS не доступен, то вы можете отредактировать локальные файлы hosts (обычно они находятся в /etc) на сервере добавив туда следующую строку:

```
127.0.0.1 server.aktiv-test.ru localhost server
```

А на каждом клиенте добавить строку

```
<IP-address> server.aktiv-test.ru <IP-address> server
```

Где IP-address - это IP адрес сервера. В нашем примере это будет 10.0.0.1.

После этого проверьте работу локальных DNS имен используя команду nslookup как показано выше.

Наличие соединения

Для проверки соединения между хостами, выполните ping для каждого хоста по его FQDN:

```
$ ping server.aktiv-test.ru
PING server.aktiv-test.ru (10.0.0.1) 56(84) bytes of data.
64 bytes from server.aktiv-test.ru (10.0.0.1): icmp_seq=1 ttl=128 time=0.176ms
```

Вывод команды ping показывает успешное определение IP адреса по FQDN, и простой ответ от сервера. Ответ от сервера является подтверждением того, что между хостом и сервером есть соединение.

Проблемы при работе ping указывают на проблемы настройки сервера или клиента.

Синхронизация времени

Протокол Kerberos требует синхронизации времени сервера и клиента: если системные часы клиентов и сервера расходятся, то аутентификация не будет выполнена. Простейший способ синхронизировать системные часы - использование **Network Time Protocol (NTP)** сервера. Некоторый линуксы, например, Astra Linux по-умолчанию синхронизирует время с российскими NTP-серверами. Для настройки собственного NTP-сервера смотрите документацию на ваш дистрибутив (например, UbuntuTime для Ubuntu).

Active Directory Domain Controllers обычно так же являются NTP серверами.

Брандмауэры

Так же как и все остальные сетевые службы, Kerberos должен иметь возможность проходить через любые брандмауэры между хостами. Инструкция Kerberos System Administration Manual имеет детальное описание портов, которые необходимо открыть при настройке брандмауэров.

Проверка модели устройства

1. Подключите USB-токен к компьютеру.
2. Для определения названия модели USB-токена откройте **Терминал** и введите команду:

```
$ lsusb
```

В результате в окне Терминала отобразится название модели USB-токена:

```
[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
```

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

Стенд

- две виртуальные машины с Ubuntu

Важно: время на клиенте и сервере должно быть синхронизировано. Невыполнение этого требования может привести к возникновению проблем.

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

Сервер

- Установлены `krb5-kdc, krb5-admin-server, krb5-pkinit`
- Kerberos realm: `AKTIV-TEST`, доменное имя `aktiv-test.ru` (прописано в /etc/hosts на клиенте)
  Примечание: доменное имя стоит делать минимум второго уровня для избежания ошибок
- Пользователи: `testuser@AKTIV-TEST`

Клиент

- Установлены `krb5-user, libpam-krb5, libpam-ccreds, auth-client-config, krb5-pkinit, opensc, libengine-pkcs11-openssl`
- default realm: `AKTIV-TEST`
- серверы (kdc, admin) указаны по IP-адресу (лучше указать их в /etc/hosts)

Первичная настройка

Сервер

Установить пакеты и создать новый realm

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

В файле конфигурации сервера /etc/krb5.conf указать

/etc/krb5.conf

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

Создать на сервере нового пользователя

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

На сервере проверить, что для этого пользователя можно получить тикет

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

Клиент

Загрузим rtengine для openssl из sdk и поместим в папку с энджинами

```
$ wget https://download.rutoken.ru/Rutoken/SDK/sdk-180919-80c054.zip
$ unzip -q sdk-180919-80c054.zip
$ sudo cp -P sdk/openssl/rtengine/bin/linux_glibc-x86_64/lib/librtengine.so* /usr/lib/x86_64-linux-gnu/engines-1.1/
```

Установим pkcs11 модуль rtpkcs11ecp.so

Установим необходимые пакеты и сконфигурируем kerberos

```
$ sudo apt-get install krb5-user libpam-krb5 libpam-ccreds auth-client-config krb5-pkinit opensc libengine-
pkcs11-openssl
#   :
# realm = AKTIV-TEST
#  = aktiv-test.ru
$ sudo dpkg-reconfigure krb5-config
```

В файле конфигурации клиента /etc/krb5.conf указать

/etc/krb5.conf

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

Проверим, что пользователь может аутентифицироваться по паролю

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

Настройка аутентификации по Рутокену

Сервер

Создадим ключ и самоподписанный сертификат УЦ

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

Создадим файл pkinit_extensions со следующим содержимым

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}
```

Создадим ключ и сертификат 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
```

Переместим файлы kdc.pem, kdckey.pem, cacert.pem в директорию /etc/krb5/

```
$ sudo mkdir /etc/krb5
$ sudo cp kdc.pem kdckey.pem cacert.pem /etc/krb5/
```

Включим preauth на сервере. Для этого опишем realm AKTIV-TEST в файле /etc/krb5kdc/kdc.conf

/etc/krb5kdc/kdc.conf

```
[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
        pkinit_anchors = FILE:/etc/krb5/cacert.pem
        pkinit_identity = FILE:/etc/krb5/kdc.pem,/etc/krb5/kdckey.pem
    }
```

Включим preauth для пользователя

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

Клиент

Отформатируйте токен с помощью утилиты rtadmin

Сгенерируем ключевую пару клиента. Создаем заявку на сертификат.

```
#    ID!
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so --keypairgen --key-type rsa:2048 -l --id 45
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:librtpkcs11ecp.so
...
OpenSSL> req -engine pkcs11 -new -key 45 -keyform engine -out client.req -subj "/C=RU/ST=Moscow/L=Moscow/O=Aktiv
/OU=dev/CN=testuser/emailAddress=testuser@mail.com"
```

Сервер

Копируем заявку на сертификат (client.req). И подписываем ее.

```
$ 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
```

Перезапустим сервер и KDC

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

Обратно клиент

Получаем сертификат и записываем его на токен. Также нужно положить корневой сертификат в /etc/krb5/

```
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so -l -y cert -w ./client.pem --id 45
$ sudo cp cacert.pem /etc/krb5/cacert.pem
```

Изменим файл конфигурации /etc/krb5.conf

/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/librtpkcs11ecp.so
```

Пробуем аутентифицироваться

```
$ kinit <username>
```

Примечание

Если по каким-то причинам не удалось аутентифицироваться, то можно узнать об причине неисправности с помощью логирования. Для этого в файле настройки /etc/krb5.conf и /etc/krb5kdc/kdc.conf.

```
[logging]
        default = FILE:/var/log/krb5.log
        kdc = FILE:/var/log/krb5kdc.log
        admin_server = FILE:/var/log/kadmin.log
```

При этом после этого нужно перезагрузить KDC с помощью команд:

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