HowTo ASIX Kerberos

Curs 2018-2019

Temes tractats

1. Instal·lació de Kerberos
2. Configuració de clients Kerberos
3. Autenticació pAM amb Kerberos
4. Aplicacions Kerberos aware.
5. Servidors Kerberos Aware

Documentació

Aquest document ha estat elaborant utilitzant com a eina de treball un sistema GNU/Linux Fedora 20.

- Documentació de les pàgines man de les ordres.
- Fedora Documentation, Fedora 19. Security Guide: 3.7 Kerberos.

Kerberos V5 System Administrator's Guide:
- /user/share/doc/kerberos-<server>
- /usr/share/doc/kerberos-<workstation>
- /usr/share/doc/kerberos-<libs>

MIT
- MIT Kerberos documentation
- Tutorial ** molt recomanable **

Ordres:
kdestroy(1), kinit(1), klist(1), kswitch(1), kpasswd(1), ksu(1), krb5.conf(5), kdc.conf(5), kadmin(1), kadmind(8), kdb5_util(8), krb5kdc(8)

Kerberos Server

Descripció General

Descripció

Fedora 19 Security Guide:

Kerberos is a network authentication protocol created by MIT, and uses symmetric-key cryptography to authenticate users to network services, which means passwords are

never actually sent over the network. Consequently, when users authenticate to network services using Kerberos, unauthorized users attempting to gather passwords by monitoring network traffic are effectively thwarted.

Kerberos depends on the following network services to function correctly:

• Approximate clock synchronization between the machines on the network.
• Domain Name Service (DNS). You should ensure that the DNS entries and hosts on the network are all properly configured.

Kerberos-aware services do not currently make use of Pluggable Authentication Modules (PAM) — these services bypass PAM completely.

However, **applications that use PAM can make use of Kerberos** for authentication if the **pam_krb5 module** (provided in the pam_krb5 package) is installed.

The pam_krb5 package contains sample configuration files that allow services such as login and gdm to authenticate users as well as obtain initial credentials using their passwords.

Kerberos Terminology

Fedora 19 Security Guide:

authentication server (AS)
A server that issues tickets for a desired service which are in turn given to users for access to the service. The AS responds to requests from clients who do not have or do not send credentials with a request. It is usually used to gain access to the *ticket-granting server (TGS)* service by issuing a *ticket-granting ticket (TGT)*. The AS usually runs on the same host as the *key distribution center (KDC)*.

ciphertext
Encrypted data.

client
An entity on the network (a user, a host, or an application) that can receive a ticket from Kerberos.

credentials
A temporary set of electronic credentials that verify the identity of a client for a particular service. Also called a ticket.

credential cache or ticket file
A file which contains the keys for encrypting communications between a user and various network services. Kerberos 5 supports a framework for using other cache types, such as shared memory, but files are more thoroughly supported.

crypt hash
A one-way hash used to authenticate users. These are more secure than using unencrypted data, but they are still relatively easy to decrypt for an experienced cracker.

GSS-API
The *Generic Security Service Application Program Interface* (defined in RFC-2743 published by The Internet Engineering Task Force) is a set of functions which provide security services. This API is used by clients and services to authenticate to each other without either program having specific knowledge of the underlying mechanism. If a network service (such as cyrus-IMAP) uses GSS-API, it can authenticate using Kerberos.

hash
Also known as a hash value. A value generated by passing a string through a hash function. These values are typically used to ensure that transmitted data has not been tampered with.

hash function
A way of generating a digital "fingerprint" from input data. These functions rearrange, transpose or otherwise alter data to produce a hash value.

key
Data used when encrypting or decrypting other data. Encrypted data cannot be decrypted without the proper key or extremely good fortune on the part of the cracker.

key distribution center (KDC)
A service that issues Kerberos tickets, and which usually run on the same host as the ticket-granting server (TGS).

keytab (or key table)
A file that includes an unencrypted list of principals and their keys. Servers retrieve the keys they need from keytab files instead of using kinit. The default keytab file is /etc/krb5.keytab. The KDC administration server, /usr/kerberos/sbin/kadmind, is the only service that uses any other file (it uses /var/kerberos/krb5kdc/kadm5.keytab).

kinit
The kinit command allows a principal who has already logged in to obtain and cache the initial ticket-granting ticket (TGT). Refer to the kinit man page for more information.

principal (or principal name)
The principal is the unique name of a user or service allowed to authenticate using Kerberos. A principal follows the form **root[/instance]@REALM**. For a typical user, the root is the same as their login ID. The instance is optional. If the principal has an instance, it is separated from the root with a forward slash ("/"). An empty string ("") is considered a valid instance (which differs from the default NULL instance), but using it can be confusing. All principals in a realm have their own key, which for **users is derived from a password** or is **randomly set for services**.

realm
A network that uses Kerberos, composed of one or more servers called KDCs and a

potentially large number of clients.

service
A program accessed over the network.

ticket
A temporary set of electronic credentials that verify the identity of a client for a particular service. Also called credentials.

ticket-granting server (TGS)
A server that issues tickets for a desired service which are in turn given to users for access to the service. The TGS usually runs on the same host as the KDC.

ticket-granting ticket (TGT)
A special ticket that allows the client to obtain additional tickets without applying for them from the KDC.

unencrypted password
A plain text, human-readable password.

Elements clau

realm
A network that uses Kerberos, composed of one or more servers KDC.
El reialme o “domini” gestionat per kerberos.

KDC key distribution center.
Issues Kerberos tickets.
Usually run on the same host as the ticket- granting server (TGS).
El servidor kerberos.

ticket-granting server (TGS)
A server that issues tickets for a desired service which are in turn given to users for access to the service.The TGS usually runs on the same host as the KDC.

ticket-granting ticket (TGT)
A special ticket that allows the client to obtain additional tickets without applying for them from the KDC.

ticket
A temporary set of electronic credentials that verify the identity of a client for a particular service. Also called **credentials**.

principal (or principal name)
The principal is the unique name of a user or service allowed to authenticate using Kerberos. A principal follows the form **root[/instance]@REALM**. For a typical user, the root is the same as their login ID.

The instance is optional. If the principal has an instance, it is separated from the root with a forward slash ("/"). An empty string ("") is considered a valid instance (which differs from the default NULL instance), but using it can be confusing.

All principals in a realm have their own key, which for **users is derived from a password** or is **randomly set for services**.

kadmind
: Dimoni del servei d'administració de kerberos.

krb5kdc
: Dimoni de kerberos encarregat de la distribució dels tickets (KDC).

kadmin
: Utilitat d'administració del kerberos. Es pot executar des de qualsevol màquina del reialme. Es comunica usant protocol kerberos amb el servidor kadmind. Només els usuaris autoritzats amb les ACLs pertinents poden fer modificacions a la base de dades.

kadmin.local
: Utilitat d'administració que només es pot executar en el propi host servidor. No usa kerberos sinó que accedeix a baix nivell directament a la base de dades.

Model de funcionament

Fedora 19 Security Guide:

Kerberos differs from username/password authentication methods. Instead of authenticating each user to each network service, Kerberos uses symmetric encryption and a trusted third party (a KDC), to authenticate users to a suite of network services.

When a **user authenticates to the KDC**, the KDC **sends a ticket specific** to that session back to the user's machine, and any **Kerberos-aware services look for the ticket** on the user's machine rather than requiring the user to authenticate using a password.

When a user on a Kerberos-aware network logs in to their workstation, their **principal** (is **sent to the KDC** as part of **a request for a TGT** from the Authentication Server. This request can be sent by the log-in program so that it is transparent to the user, or can be sent by the kinit program after the user logs in.

The KDC then checks for the principal in its database. If the principal is found, the KDC **creates a TGT**, which is encrypted using the user's key and returned to that user.

The login or kinit program on the client then decrypts the TGT using the user's key, which **it computes from the user's password**. The user's key is used only on the client machine and is not transmitted over the network.

The **TGT is set to expire after a certain period of time** (usually ten to twenty-four hours) and is stored in the client machine's credentials cache. An expiration time is set so that a compromised TGT is of use to an attacker for only a short period of time. After the TGT

has been issued, the user does not have to re-enter their password until the TGT expires or until they log out and log in again.

Whenever the user needs access to a network service, the client software uses the TGT to **request a new ticket for that specific service** from the TGS. The service ticket is then used to authenticate the user to that service transparently.

Descripció del model de funcionament:

Petició client

- Un usuari fa login en un sistema client. L'aplicació de login està configurada per fer us del kerberos.
- Un altre exemple és usar l'ordre client *kinit* per iniciar una sessió d'usuari kerberos. Com a resultat s'obtindrà el tiket del servidor KDC.
- El client (per exemple login o kinit) envia una petició d'autenticació. Identifica a l'usuari amb el seu ID d'usuari que correspon en llenguatge kerberos a un **principal**. Un principal és un usuari o una màquina.
- Així si en pere es vol autenticar en fer login s'envia pere@REAILME-EDT.ORG com a principal si pertany al reialme "REIALME-EDT.ORG".
- L'objectiu és obtenir un ticket TGT. És un ticket que dona dret a obtenir tickets individuals per a serveis concrets.

Generació de TGT

- En el servidor KDC es busca a la base de dades si existeix aquest principal (aquest usuari). Si és així s'encripta un TGT (ticket-grating) usant el password de l'usuari.
- Fixeu-vos que el password ja és a la base de dades, no l'envia l'usuari.
- El TGT es retorna al client.

Verificació del ticket TGT

- El client (l'aplicació login o Kinit) rep el TGT i el desencripta usant el password de l'usuari. Ara si se li demana localment el password a l'usuari.
- Si la desencriptació és correcte és que l'autenticació ha sigut correcte.
- El ticket TGT és vàlid només per un període de temps finit.
- Observar que el passwd de l'usuari pere no viatja per la xarxa. S'utilitza per desxifrar/validar el TGT rebut.

Instal·lació kerberos Server

Podeu trobar els docker i el git d'aquest apartat a:

- ❏ dockerhub: **edtasixm11/k18**
  https://cloud.docker.com/u/edtasixm11/repository/docker/edtasixm11/k18
- ❏ github: **edtasixm11/k18**
  https://github.com/edtasixm11/k18

Descripció del procediment a seguir:

1. Instal·lar els paquets de kerberos: krb5-{workstation,libs,server}
2. Configurar el fitxer /etc/krb5.conf incorporant el reialme apropiat. En l'exemple s'ha usat EDT.ORG.
3. Configurar el fitxer /var/kerberos/krb5kdc//kdc.conf incorporant el reialme apropiat.
4. Definir una entrada DNS que identifiqui per nom (si no es fa per IP) el servidor KDC. Per exemple editant el */etc/hosts*. El servidor s'anomena *kdc.edt.org* i pertany al domini *edt.org*. Tant el KDC com el Servidor d'administració són el mateix host.
5. Crear la base de dades de *principals* (usuaris i màquines) usant la utilitat kdb5_util.
6. Definir les ACLs dels usuaris (*principals*) amb dret d'administració del servidor kerberos *kadmin*.
7. Crear el primer principal (primer usuari) de la base de dades de kerberos.
8. Engegar el servei que utilitza el dimoni *kadmind*.
9. Gestió de principals: crear / eliminar / llistar / Modificar.

1) Instal·lar kerberos

Repassar els paquets instal·lats i instal·lar els que faltin.

```
# rpm -qa | grep krb5
krb5-workstation-1.10.2-12.fc17.i686
krb5-libs-1.10.2-12.fc17.i686
pam_krb5-2.3.14-1.fc17.i686

# yum install krb5-server
```

Observar els directoris:

```
# tree /var/kerberos/
/var/kerberos/
├── krb5
│   └── user
└── krb5kdc
    ├── kadm5.acl
    └── kdc.conf

# ll /etc/krb5.conf
-rw-r--r--. 1 root root 495 Nov  5 18:37 /etc/krb5.conf
```

2) Configurar krb5.conf

Contingut original del fitxer de configuració /etc/krb5.conf

```
# cat /etc/krb5.conf
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log
```

```
[libdefaults]
 dns_lookup_realm = false
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true
 rdns = false
# default_realm = EXAMPLE.COM
 default_ccache_name = KEYRING:persistent:%{uid}

[realms]
# EXAMPLE.COM = {
#  kdc = kerberos.example.com
#  admin_server = kerberos.example.com
# }

[domain_realm]
# .example.com = EXAMPLE.COM
# example.com = EXAMPLE.COM
```

Configurar kerberos al reialme *EDT.ORG*

```
# vim /etc/krb5.conf
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 dns_lookup_realm = false
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true
 default_realm = EDT.ORG

[realms]
EDT.ORG = {
  kdc = krb.edt.org
  admin_server = krb.edt.org
}

[domain_realm]
.edt.org = EDT.ORG
edt.org = EDT.ORG
```

3) Configurar kdc.conf

Contingut original del fitxer /var/kerberos/krb5kdc/kdc.conf

```
# cat /var/kerberos/krb5kdc/kdc.conf
[kdcdefaults]
  kdc_ports = 88
  kdc_tcp_ports = 88

[realms]
  EXAMPLE.COM = {
    #master_key_type = aes256-cts
    acl_file = /var/kerberos/krb5kdc/kadm5.acl
    dict_file = /usr/share/dict/words
    admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
    supported_enctypes = aes256-cts:normal aes128-cts:normal des3-hmac-sha1:normal\
      arcfour-hmac:normal camellia256-cts:normal camellia128-cts:normal \
      des-hmac-sha1:normal des-cbc-md5:normal des-cbc-crc:normal
 }
```

Configurar el reialme EDT.ORG al fitxer /var/kerberos/krb5kdc/kdc.conf

```
(fer despres de la ordre krb5_util)

# vim /var/kerberos/krb5kdc/kdc.conf
[kdcdefaults]
 kdc_ports = 88
 kdc_tcp_ports = 88

[realms]
EDT.ORG = {
    #master_key_type = aes256-cts
    acl_file = /var/kerberos/krb5kdc/kadm5.acl
    dict_file = /usr/share/dict/words
    admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
    supported_enctypes = aes256-cts:normal aes128-cts:normal des3-hmac-sha1:normal
arcfour-hmac:normal des-hmac-sha1:normal des-cbc-md5:normal des-cbc-crc:normal
 }
```

4) Configurar el KDC i el Server Admin al DNS

Editar/Comprovar que hi ha les entrades corresponents al host que realitza les dues funcions:

```
# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.37 krb.edt.org
```

5) Crear la Base de dades

Cal crear la base de dades que emmagatzema les keys dels principals (usuaris i màquines) de Kerberos.

```
# /usr/sbin/kdb5_util create -s
Loading random data
Initializing database '/var/kerberos/krb5kdc/principal' for realm 'EDT.ORG',
master key name 'K/M@EDT.ORG'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key: masterkey
Re-enter KDC database master key to verify: masterkey
```

Descripció de l'ordre **kdb5_util**:

The *create* command creates the database that stores keys for the Kerberos realm.
The -s switch forces creation of a stash file in which the master server key is stored. If no stash file is present from which to read the key, the Kerberos server (krb5kdc) prompts the user for the master server password (which can be used to regenerate the key) every time it starts.

NAME
kdb5_util - Kerberos database maintenance utility

SYNOPSIS
kdb5_util [-r realm] [-d dbname] [-k mkeytype] [-M mkeyname] [-kv mkeyVNO] [-sf stashfilename] [-m] command [command_options]

DESCRIPTION
kdb5_util allows an administrator to perform maintenance procedures on the KDC database. Databases can be created, destroyed, and dumped to or loaded from ASCII files. kdb5_util can create a Kerberos master key stash file or perform live rollover of the master key.

When kdb5_util is run, it attempts to acquire the master key and open the database. However, execution continues regardless of whether or not kdb5_util successfully opens the database, because the database may not exist yet or the stash file may be corrupt. Note that some KDC database modules may not support all kdb5_util commands.

COMMANDS
create destroy stash dump load ark add_mkey use_mkey list_mkeys purge_mkeys update_princ_encryption

L'ordre anterior genera la base de dades. Podem observar els principals creats al directori kerberos:

```
# tree /var/kerberos/
/var/kerberos/
├── krb5
│   └── user
└── krb5kdc
    ├── kadm5.acl
```

```
├── kdc.conf
├── principal
├── principal.kadm5
├── principal.kadm5.lock
└── principal.ok
```

6) Definir ACLs d'usuaris amb dret d'administració de kadmin

Editar el fitxer **kadmin5.acl** que denifeix les ACLs dels usuaris (*principals*) amb drets d'administració del servei ***kadmin***.

Contingut original del fitxer /var/kerberos/krb5kdc/kadmin5.acl

```
# cat /var/kerberos/krb5kdc/kadm5.acl
*/admin@EXAMPLE.COM     *
```

Contingut del fitxer editat:

```
# cat /var/kerberos/krb5kdc/kadm5.acl
*/admin@EDT.ORG    *
ecanet@EDT.ORG     *
superuser@EDT.ORG  *
```

Són administradors del servidor kerberos **kadmin** els usuaris *ecanet* i *superuser*. També qualsevol usuari/instància on la instància sigui *admin*. Per exemple *pere/admin*, *marta/admin*. Tots ells del reialme EDT.ORG.

7) Crear els principals de la base de dades de kerberos

Crear els *principals* corresponents als usuaris que es volen incorporar al kerberos. Almenys crear els *principals* dels usuaris administradors del servidor *kadmin*. Per crear els usuaris s'utilitza l'ordre **kadmin.local**, que cal executar des del servidor kerberos.

Crear els principals ecanet, admin/admin, superuser i pere/admin, per exemple:

```
# /usr/sbin/kadmin.local -q "addprinc ecanet"
Authenticating as principal root/admin@EDT.ORG with password.
WARNING: no policy specified for ecanet@EDT.ORG; defaulting to no policy
Enter password for principal "ecanet@EDT.ORG": ecanet
Re-enter password for principal "ecanet@EDT.ORG": ecanet
Principal "ecanet@EDT.ORG" created.

# /usr/sbin/kadmin.local -q "addprinc admin/admin"
Authenticating as principal root/admin@EDT.ORG with password.
WARNING: no policy specified for admin/admin@EDT.ORG; defaulting to no policy
Enter password for principal "admin/admin@EDT.ORG":  admin
Re-enter password for principal "admin/admin@EDT.ORG": admin
```

```
Principal "admin/admin@EDT.ORG" created.

# /usr/sbin/kadmin.local -q "addprinc superuser"
# /usr/sbin/kadmin.local -q "addprinc pere/admin"
```

Ara en el sistema existeixen els usuaris o *principals*: ecanet@EDT.ORG, admin/admin@EDT.ORG, superuser@EDT.ORG i pere/admin@EDT.ORG. Tots ells són administradors del servidor (uns explícitament per la ACL *ecanet* i *superuser* i els altres per la ACL que identifica les instàncies *admin* com a usuaris administradors).

Llistar els *principals* creats:

```
# kadmin.local -q "list_principals"
Authenticating as principal ecanet/admin@EDT.ORG with password.
K/M@EDT.ORG
admin/admin@EDT.ORG
ecanet@EDT.ORG
kadmin/admin@EDT.ORG
kadmin/changepw@EDT.ORG
kadmin/portatil.localdomain@EDT.ORG
krbtgt/EDT.ORG@EDT.ORG
...
```

8) Engegar el servei kerberos

Engegar els serveis de kerberos usant *systemctl*. Cal engegar els serveis:

- **krb5kdc**: servei que realitza la tasca de KDC Key Distribution Center.
- **kadmin**: administració del servei kerberos.

After **kadmind** has been started on the server, any **user** can access its services by running kadmin on any of the clients or servers in the realm. However, **only** users listed in the **kadm5.acl** file can **modify** the database in any way, **except** for **changing** their own **passwords**.

The **kadmin** utility communicates with the **kadmind** server over the network, and **uses Kerberos** to handle authentication. Consequently, the **first principal must already** exist before connecting to the server over the network to administer it. Create the first principal with the **kadmin.local** command, which is specifically designed **to be used on the same host** as the KDC and does **not use** Kerberos for authentication.

```
# systemctl start krb5kdc.service

# systemctl status krb5kdc.service
krb5kdc.service - Kerberos 5 KDC
    Loaded: loaded (/usr/lib/systemd/system/krb5kdc.service; disabled)
    Active: active (running) since Mon, 09 Feb 2015 20:27:30 +0100; 13s ago
```

```
    Process: 7781 ExecStart=/usr/sbin/krb5kdc -P /var/run/krb5kdc.pid $KRB5KDC_ARGS
(code=exited, status=0/SUCCESS)
   Main PID: 7782 (krb5kdc)
     CGroup: name=systemd:/system/krb5kdc.service
             └ 7782 /usr/sbin/krb5kdc -P /var/run/krb5kdc.pid

# systemctl start kadmin.service

# systemctl status kadmin.service
kadmin.service - Kerberos 5 Password-changing and Administration
     Loaded: loaded (/usr/lib/systemd/system/kadmin.service; disabled)
     Active: active (running) since Mon, 09 Feb 2015 20:28:16 +0100; 7s ago
    Process: 7788 ExecStart=/usr/sbin/kadmind -P /var/run/kadmind.pid
$KADMIND_ARGS (code=exited, status=0/SUCCESS)
   Main PID: 7789 (kadmind)
     CGroup: name=systemd:/system/kadmin.service
             └ 7789 /usr/sbin/kadmind -P /var/run/kadmind.pid
```

En resum:

- Hi ha dos daemon engegats, el servidor KDC i el servidor kadmin. Un fa la tasca de KDC (generar/distribuir tickets) i l'altre administrar.
- La utilitat **kadmin** es comunica amb el dimoni **kadmind** usant el protocol kerberos. Per poder-ho fer cal que existeixi prèviament almenys un principal, el **first principal**.
- Per poder crear el first principal s'utilitza la utilitat **kadmin.local**. Aquesta utilitat només es pot usar en el propi host on hi ha el servidor executant-se i no es connecta per xarxa ni via kerberos, sinó que accedeix directament a la base de dades de kerberos.

Ports oberts per els serveis kadmind i krb5kdc:

```
# nmap localhost
Starting Nmap 6.01 ( http://nmap.org ) at 2015-02-10 19:57 CET
Nmap scan report for krb (192.168.1.41)
Host is up (0.000021s latency).
Not shown: 992 closed ports
PORT  STATE SERVICE
...
88/tcp  open  kerberos-sec
464/tcp open  kpasswd5
749/tcp open  kerberos-adm
```

9) Gestió de principals: crear / llistar / connectar

Verificar

Les següents ordres permetran verificar els passos anteriors de posar en marxa un sistema d'autenticació kerberos:

- **kinit** permet a un usuari indicat o a l'usuari actual obtenir un ticket kerberos (TGT).
- **klist** permet llistar les dades del ticket obtingut.
- **kdestroy** elimina el ticket. Ja no serà vàlid.

Observar que l'usuari *pau* del sistema Gnu/Linux no pot obtenir un ticket kerberos perquè no se n'ha creat el compte ala base de dades de *principals*:

```
[pau@portatil ~]$ kinit pau
kinit: Client not found in Kerberos database while getting initial credentials
```

L'usuari *ecanet* pot obtenir un ticket, llistar les propietats del ticket. Destruir el ticket i observar que ja no es llista:

```
[ecanet@portatil ~]$ kinit
Password for ecanet@EDT.ORG: ecanet

[ecanet@portatil ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: ecanet@EDT.ORG
Valid starting   Expires          Service principal
02/09/15 20:29:58  02/10/15 20:29:58  krbtgt/EDT.ORG@EDT.ORG
 renew until 02/09/15 20:29:58

[ecanet@portatil ~]$ kdestroy

[ecanet@portatil ~]$ klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_1000)
```

Llistar les dades d'un principal

Llistar les dades de la base de dades de principals corresponets a un usuari concret (provar-ho un cop obtingut un ticket). És una ordre d'administració (root).

```
# kadmin.local -q "get_principal ecanet"
Authenticating as principal root/admin@EDT.ORG with password.
Principal: ecanet@EDT.ORG
Expiration date: [never]
Last password change: Mon Feb 09 20:23:23 CET 2015
Password expiration date: [none]
Maximum ticket life: 1 day 00:00:00
Maximum renewable life: 0 days 00:00:00
Last modified: Mon Feb 09 20:23:23 CET 2015 (root/admin@EDT.ORG)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 4
Key: vno 1, aes256-cts-hmac-sha1-96, no salt
```

```
Key: vno 1, aes128-cts-hmac-sha1-96, no salt
Key: vno 1, des3-cbc-sha1, no salt
Key: vno 1, arcfour-hmac, no salt
MKey: vno 1
Attributes:
Policy: [none]
```

Crear principals

Crear usuaris o principals (d'usuaris o de màquines) del reialme de kerberos que s'està configurant.

Crear un usuari "normal" anomenat *pau*.

```
# kadmin.local -q "addprinc pau"
Authenticating as principal root/admin@EDT.ORG with password.
WARNING: no policy specified for pau@EDT.ORG; defaulting to no policy
Enter password for principal "pau@EDT.ORG": pau
Re-enter password for principal "pau@EDT.ORG": pau
Principal "pau@EDT.ORG" created.
```

Utilitats d'administració: kadmin.local / kadmin

Kadmin i kadmin.local proporcionen la mateixa utilitat, administrar el servidor kerberos. És en com ho fan (via xarxa i protocol kerberos o localment) on varia el seu funcionament.

- ***Kadmin.local***: utilitat d'administració de la base de dades de principals que accedeix a baix nivell directament als fitxers de dades. Només es pot executar des del servidor coma usuari amb drets administratius (root). No usa el protocol Kerberos ni la xarxa, accedeix directament al backend.
- ***Kadmin***: utilitat d'administració que connecta amb el servidor per xarxa sant el protocol Kerberos. Es pot usar des de qualsevol host (client o server) i estableix una connexió de xarxa segura. Per defecte espera que es connectiu un usuari tipus admin (nomuser/admin). Les ACLs d'accés que s'han definit determinaran els privilegis de què disposa l'usuari per administrar.

Recull d'opcions de l'ordre *kadmin.local* per administrar localment des del host servidor kerberos la base de dades de principals:

- change_password
- list_principals
- get_principal
- addprinc

```
# kadmin.local -q "change_password ecanet"
Authenticating as principal ecanet/admin@EDT.ORG with password.
Enter password for principal "ecanet@EDT.ORG":
```

```
Re-enter password for principal "ecanet@EDT.ORG":
Password for "ecanet@EDT.ORG" changed.

# kadmin.local -q "list_principals"
Authenticating as principal ecanet/admin@EDT.ORG with password.
K/M@EDT.ORG
admin/admin@EDT.ORG
ecanet@EDT.ORG
kadmin/admin@EDT.ORG
kadmin/changepw@EDT.ORG
kadmin/portatil.localdomain@EDT.ORG
krbtgt/EDT.ORG@EDT.ORG
```

Extret del man kadmin.local

```
KADMIN(1)                                MIT Kerberos

NAME
kadmin - Kerberos V5 database administration program

SYNOPSIS
kadmin [-O|-N] [-r realm] [-p principal] [-q query] [[-c cache_name]|[-k [-t keytab]]|-n] [-w
password] [-s admin_server[:port]]

kadmin.local [-r realm] [-p principal] [-q query] [-d dbname] [-e enc:salt ...] [-m] [-x
db_args]

DESCRIPTION
kadmin and kadmin.local are command-line interfaces to the Kerberos V5
administration system. They provide nearly identical functionalities; the difference is that
kadmin.local directly accesses the KDC database, while kadmin performs operations
using kadmind(8). Except as explicitly noted otherwise, this man page will use "kadmin"
to refer to both versions. kadmin provides for the maintenance of Kerberos principals,
password policies, and service key tables (keytabs).

The remote kadmin client uses Kerberos to authenticate to kadmind using the service
principal kadmin/ADMINHOST (where ADMINHOST is the fully-qualified hostname of
the admin server) or kadmin/admin. If the credentials cache contains a ticket for one of
these principals, and the -c credentials_cache option is specified, that ticket is used to
authenticate to kadmind. Otherwise, the -p and -k options are used to specify the client
Kerberos principal name used to authenticate. Once kadmin has determined the principal
name, it requests a service ticket from the KDC, and uses that service ticket to
authenticate to kadmind.

Since kadmin.local directly accesses the KDC database, it usually must be run directly on
the master KDC with sufficient permissions to read the KDC database. If the KDC
database uses the LDAP database module, kadmin.local can be run on any host which
can access the LDAP server.
```

Utilitats client

Des d'un host client (membre d'un Kingdom) Kerberos es disposa de vàries utilitats (cal instal·lar el paquet kerb5-workstation). En el proper capítol es detalla més el funcionament client.

Odres client:

- ❏ kinit
- ❏ klist
- ❏ kdestroy
- ❏ kpasswd

```
# kinit ecanet
Password for ecanet@EDT.ORG:

# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: ecanet@EDT.ORG

Valid starting   Expires          Service principal
02/10/15 18:29:09  02/11/15 18:29:09  krbtgt/EDT.ORG@EDT.ORG
   renew until 02/10/15 18:29:09

# kdestroy ecanet
```

```
# kpasswd ecanet
Password for ecanet@EDT.ORG:
Enter new password:
Enter it again:
Password changed.
```

Kerberos Client

Per configurar un host com a client de kerberos cal:

1. Instal·lar els paquets de kerberos: krb5-{libs,workstation}
2. Proporcionar el fitxer de configuració krb5.conf. Usualment el mateix fitxer de configuració que en el servidor. És on es configura el reialme.
3. [Si aquest host client ha de ser un servidor d'aplicacions també cal afegir el host client a la llista de principals del servidor.]

1) Instal·lar / Llistar els paquets del kerberos

```
# rpm -qa | grep krb5
krb5-workstation-1.7.1-19.fc13.i686
krb5-libs-1.7.1-19.fc13.i686
```

2) Editar el fitxer de configuració krb5.conf

```
# vim /etc/krb5.conf
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]

 default_realm = EDT.ORG
 dns_lookup_realm = false
 dns_lookup_kdc = false
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true

[realms]
 EDT.ORG = {
  kdc = krb.edt.org
  admin_server = krb.edt.org
 }

[domain_realm]
.edt.org = EDT.ORG
edt.org = EDT.ORG
```

3) Verificar la connectivitat de kadmin

El client i el servidor es connectaran per xarxa usant el protocol kerberos. Cal verificar la connectivitat del client amb el servidor. De fet cal verificar **sempre** el següent:

- El servidor connecta amb el propi servidor. Verificar que l'ordre kadmin és capaç de contactar des del servidor amb ell mateix (es va igualment via xarxa en una conexió local).
  Aquest punt és important per detectar si hi ha firewalls, tcpWrappers o altres elements que impedeixen contactar amb el servidor.
  Evidentment per fer aquest pas cal instal3lar en el servidor el software de kerberos workstation.

- Verificar que el client pot accedir al servei i connectar remotament amb el servidor.

Server 2 Server

Comprovar que des del servidor tenim conectivitat al propi servidor. ULL: desactivar els firewalls!!

Si no s'ha fet caldrà instal3lar en el servidor el software de workstation (el paquet krb5-workstation). Només cal fer aquest pas si es vol poder treballar en el propi servidor de kerberos com a client kerberos, i poder usar kadmin, kinit, etc.

Verificar iptables i ports del servidor:

```
# systemctl stop iptables
# systemctl stop ip6tables
# systemctl stop firewalld

# examinar ports del servidor
# nmap 192.168.1.41
Starting Nmap 6.01 ( http://nmap.org ) at 2015-02-10 19:57 CET
Nmap scan report for krb (192.168.1.41)
Host is up (0.000021s latency).
Not shown: 992 closed ports
PORT  STATE SERVICE
22/tcp  open  ssh
88/tcp  open  kerberos-sec
111/tcp open  rpcbind
143/tcp open  imap
464/tcp open  kpasswd5
749/tcp open  kerberos-adm
873/tcp open  rsync
993/tcp open  imaps
Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
```

Verificar que l'ordre d'administració kadmin (executada des del mateix servidor) és capaç de contactar amb el servidor via xarxa usant el protocol kerberos. S'utilitza una sessió interactiva de la ordre:

```
# kadmin [-p principal]
Authenticating as principal root/admin@EDT.ORG with password.
Password for root/admin@EDT.ORG:
kadmin:  listprincs
192.168.1.33/notebook1.edt.org@EDT.ORG
K/M@EDT.ORG
admin/admin@EDT.ORG
ecanet/admin@EDT.ORG
ecanet@EDT.ORG
...
```

```
kadmi:wq:n:  exit
```

Client 2 Server

Verificar que el client pot contactar amb el servidor usant kadmin. Exemple de sessió interactiva amb l'utilitat kadmin:

```
# kadmin
Authenticating as principal root/admin@EDT.ORG with password.
Password for root/admin@EDT.ORG:
kadmin:  listprincs
192.168.1.33/notebook1.edt.org@EDT.ORG
K/M@EDT.ORG
admin/admin@EDT.ORG
ecanet/admin@EDT.ORG
ecanet@EDT.ORG
…
kadmin:  exit
```

Obtenir tiquet d'un usuari en una màquina client

```
[client]$ kinit jan
Password for jan@EDT.ORG:

[client]$ klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: jan@EDT.ORG
Valid starting   Expires          Service principal
02/10/15 20:13:10  02/11/15 20:13:10  krbtgt/EDT.ORG@EDT.ORG
renew until 02/10/15 20:13:10

# kdestroy
```

Kerberos Application server

Un host que fa la funció de servidor d'algun tipus de servei kerberitzat l'anomenem un kerberos application server. Atenció, no és un servidor de kerberos, és un servidor de sshd, httpd, imap, etc que utilitza autenticació kerberos.

De fet es tracta d'un client o workstation kerberos i cal configurar-lo com a tal. Però com que ofereix un servei kerberitzat, caldrà fer un pas més consistent en crear un principal del host o del servei en la base de dades de principals.

També caldrà extreure aquest "principasl" (la clau corresponent) i desar-la en el host en el seu fitxer /etc/krb5.keytab. Aquesta és la clau que identifica el host o servei com a principal i que utilitza com a clau per parlar amb el servidor kerberos a l'hore de gestionar els tikets del servei que ofereix.

Així doncs, caldria:

1. Instal·lar els paquets de kerberos: kerb5-{libs,workstation}
2. Proporcionar el fitxer de configuració krb5.conf. Usualment el mateix fitxer de configuració que en el servidor. És on es configura el reialme.
3. Tot servidor kerberitzat (per exemple un servidor sshd que vol autenticar amb kerberos) ha de ser un principal a la base de dades de kerberos. És a dir, hi ha principals de usuaris i principals de hosts. Tot servidor kerberitzat ha de tenir creada una entrada com a principal del tipus "host/nom.domini.cat" o "ssh/nom.domini.cat".
4. En el servidor kerberitzat, per exemple sshd, cal disposar del fitxer /etc/krb5.keytab que conté el principal del servidor. És a dir, cal extreure de la base de dades del servidor kerberos una còpia de les claus del servdior ssh (del seu principal).
5. Condifurar el servidor per acceptar l'automatització kerberitzada.

1) Afegir el host client a la BD de principals

No en el host client sinó en el servidor kerberos, afegir el host com a principal usant la utilitat *kadmin.local*. en el servidor o via *kadmin* en el propi admin server.

```
# kadmin.local -q "addprinc -randkey host/sshserver.edt.org"
Authenticating as principal ecanet/admin@EDT.ORG with password.
WARNING: no policy specified for host/sshserver.edt.org@EDT.ORG; defaulting to no
policy
Principal "192.168.1.33/sshserver.edt.org@EDT.ORG" created.

# kadmin.local -q "list_principals"
Authenticating as principal ecanet/admin@EDT.ORG with password.
host/sshserver.edt.org@EDT.ORG
K/M@EDT.ORG
admin/admin@EDT.ORG
ecanet@EDT.ORG
kadmin/admin@EDT.ORG
kadmin/changepw@EDT.ORG
kadmin/portatil.localdomain@EDT.ORG
krbtgt/EDT.ORG@EDT.ORG
```

Cada servidor quepertany al reailment de kerberos ha de tenir el corresponent principal. El nom del principal és del tipus **servei/fqdn**. Servei és el nom del servei, com per exemple sshd, imapd, httpd, etc. Es pot usar el nom general ***host***. Cal identificar el servidor amb el seu Fully Qualified Domain Name.

Quan es creen els principals usuaris s'estableix un password que identifica l'usuari. En el cas de serveis i hosts aquest password es genera usualment de manera aleatòria.

2) Generar el ketab local

Al server d'aplicacions cal generar-hi o afegir-hi la clau del host. Fixem-nos que els usuaris s'identifiquen instroduint el seu password, però no tindria sentit que el servidor d'aplicacions requeris cada dos per tres escriure el password del servei (que de fet hem generat aleatòriament).

És per això que al host servidor d'aplicacions cal desar-hi la clau del seu corresponent princiapl. Usualment des del propi host servidor d'aplicacions via kadmin s'exteru una còpia del seu propi principal i es desa locament al fitxer ***/etc/krb5.keytab***.

Usualment s'utilitza l'ordre ***ktadd*** de kadmin:

```
# kadmin
kadmin:  ktadd -k /etc/krb5.keytab host/sshserver.edt.org
```

3) Servidor sshd amb autenticació kerberos

Per instal·lar un servidor sshd que realitzi autenticació kerberos es faran els següents passos:

- Generar un docker per al servidor sshd que anomenem sshserver.
- Configurar-lo com a membre del domini kerberos (/etc/krb5.conf i /etc/hosts). Verificar la ruta FQDN al propi host i al servidor kserver.
- Generar al servidor l'entrada de principal d'aquest servidor d'aplicacions: host/sshserver.edt.org
- Extreure una còpia del principal anterior i desar-lo al servidor sshdserver en el fitxer /etc/krb5.keytab. Utilitzar l'opció ktadd de l'aplicació kadmin en el host servidor d'aplicacions sshdserver.
- Crear usuaris locals al servidor sshdserver perquè els usuaris han de tenir compte d'usuari (és a dir, uid, gid, homedir, shell, etc). Però no assignar-los passwd, ja que l'autenticació la farà kerberos.
  Observar que no podem iniciar sessió local com un d'aquests usuaris al docker sshserver, perquè no té password assifnat).
- Nota: verificar que al servidor kerberos kserver hi ha els principals dels usuaris creats i al /etc/hosts està correctament definida l'entrada de sshserver.edt.org.
- Configurar el servei sshd i engegar-lo. De la configuració per defecte cal unicament modificar: ***KerberosAuthentication=yes, KerberosTiketCleanup=yes***.
- Et vôilà!

Llistat dels usuaris locals (recordeu que sense passwd)

```
[root@sshserver ~]# tail -8 /etc/passwd
pere:x:1002:1002::/home/pere:/bin/bash
marta:x:1003:1003::/home/marta:/bin/bash
anna:x:1004:1004::/home/anna:/bin/bash
pau:x:1005:1005::/home/pau:/bin/bash
ramon:x:1006:1006::/home/ramon:/bin/bash
julia:x:1007:1007::/home/julia:/bin/bash
jordi:x:1008:1008::/home/jordi:/bin/bash
jan:x:1009:1009::/home/jan:/bin/bash
```

Modificar la configuració de sshd: ***/etc/ssh/sshd_config***

```
[root@sshserver ~]# grep Kerberos /etc/ssh/sshd_config
# Kerberos options
KerberosAuthentication yes
#KerberosOrLocalPasswd yes
KerberosTicketCleanup yes
#KerberosGetAFSToken no
#KerberosUseKuserok yes
GSSAPIAuthentication yes
GSSAPICleanupCredentials no
```

** si el docker indica problemes amb pam_nologin oenseu a esborrar /var/run/nologin **

Engegar el servei sshd i connectar-hi com a usuari pere:

```
[root@sshserver ~]# ssh pere@sshserver.edt.org
pere@sshserver.edt.org's password: kpere
Last login: Fri Mar 10 20:47:30 2017 from 172.17.0.5

[pere@sshserver ~]$ id
uid=1002(pere) gid=1002(pere) groups=1002(pere)
exit
```

Observeu que en l'exemple anterior root connecta al host sshserver.edt.org com l'usuari pere. Com que no s'ha validat abans li demana el passwd per autenticar-lo, el passwd de kerberos. Un cop entrat inicia una sessió ssh com a pere al sshserver.edt.org.

En l'exemple de sota podeu veure un altre cas. Root demana un tiket com a pere i es valida al kerberos. Llavors es connecta al sshserver.edt.org com a pere i ja no li demana el passwd perquè des del punt de vista del kerberos ka està autenticat (te tiket de pere). SSH envia les credencials de kerberos obtingudes en fer la connexió.

Un cop autenticat connectar per ssh:

```
[root@sshserver ~]# kinit pere
Password for pere@EDT.ORG: kpere
[root@sshserver ~]# ssh pere@sshserver.edt.org
Last login: Sat Mar 11 12:49:27 2017 from 172.17.0.4
```

```
[pere@sshserver ~]$ id
uid=1002(pere) gid=1002(pere) groups=1002(pere)
[pere@sshserver ~]$
```

Fixeu-vos que com que l'autenticació local amb PAM és unix (no l'hem modificada), i com que els comptes d'usuaris unix no tenen passwd, no és possible iniciar una sessió local amb els usuaris.

```
[root@sshserver ~]# login pere
Password:
Login incorrect
```

4) Client SSH

Cal configurar el client ssh per tal de que en usar l'ordre client ssh transmeti automàticament les credencilas de Kerberos (si en té). Per fer-ho cal configurar el fitxer del client ssh /etc/ssh/ssh_config i activar les opcions de GSSAPI.

/etc/ssh/ssh_config

```
GSSAPIAuthentication yes
GSSAPIDelegateCredentials yes
```

5) Resolució DNS

Des del client cal identificar clarament el FQDN del servidor sshd, kerberos i ldap, això ho farem modificant el /etc/hosts i configurant l'adreça IP dels servidors i el seu FQDN.
Ara bé, si estem fent containers Docker i desplegem per exemple tots els serveis en un únic host per execmple en una AMI de AWS EC2 llavors cal posar *atenció*:

- definirem al /etc/hosts una sola adreça IP amb múltiples noms o àlies.
- **important** el FQDN del servidor sshd.edt.org ha de ser el primer per tal de que sigui el nom canònic.
  Recordem com funciona dns: podem assignar a una ip múltiples noms amb entrades tipus A i tipus CNAME, així a un host el podem identificar per varis noms que es resolen en una mateixa adreça IP. Però la resolució inversa resol una adreça IP a un ***sol*** nom. Un de sol. En una entrada dedel /etc/hosts de múltiples àlies serà el primer d'ells.
  Molts serveis i protocols quan connecten a servidors realitzen una verificació de resolució inversa, aquest és el cas del sshd.

Per això és important configurar el /etc/hosts com per exemple:

```
# tres serveis en una AMI de AWS EC2
2.3.4.5 sshd.edt.org kserver.edt.org  ldap.edt.org
```

Teoria / model de treball

Teoria

Autenticaction Provider AP
Kerberos propoerciona el servei de proveïdor d'autenticació. No emmagatzema informació dels comptes d'usuari com el uid, git, shell, etc. Simplement emmagatzema i gestiona els passwords dels usuaris, en entrades anomenades *principals* en la seva base de dades. Coneixem els següents AP:

- */etc/passwd* que conté els password (AP) i també la informació dels comptes d'usuari (IP).
- *ldap* el servei de directori ldap conté informació dels comptes d'usuari (IP) i també els seus passwords (AP).
- *kerberos* que únicament actua de AP i no de IP.

Information Provider IP
Els serveis que emmagatzemen la informació dels comptes d'usuari s'anomenen Information providers. Aquests serveis proporcionen el uid, gid, shell, gecos, etc. Els clàssics són /etc/passwd i ldap.

Model de pràctiques

El model que mantindrem a tot el mòsul ASIX M11-SAD ´es el següent:

- **ldap** al servidor ldap tenim els usuaris habituals pere, marta, anna, julia, pau, jordi. El seu password és el seu propi nom.
- **/etc/passwd** en els containers hi ha els usuaris locals local01, local02 i local03 que tenen assignat com a passeord el seu mateix nom.
- **kerberos + IP** els usuaris user01, user02 i user03 són principals de kerberos amb passwords tipus kuser01, kuser02 i kuser03. La informació del seu compte d'usuari és local al */etc/passwd* on **no** tenen password assignat.
- **kerberos + ldap** Al servidor kerberos hi ha també principals per als usuaris usuals ldap pere, marta, anna, julia, jordi i pau. Els seus passwords són del tipus kpere, kmarta, kanna, kjulia, kjordi i kpau.

Es resum, podem verificar l'accés/autenticació d'usuaris locals usant el prototipus *local01*, podem fer test de la connectivitat kerberos amb comptes locals amb usuaris tipus *user01*. I finalment podem verificar l'autenticació d'usuaris kerberos amb IP ldap amb els clàssics pere (kpere).

Accés kerberitzat / Accés normal

Feu atenció al significat d'accés kerberitzat! Si l'usuari user01 és un usuari vàlid en el host sshd.edt.org (provingi el seu information privider de /etc/passwd o de ldap) i la seva autenticació és kerberos (el seu password està desat al kerberos), en un accés kerberitzat, si l'usuari ja disposa de ticket podrà accedir al serveidor sshd sense cap password. **atenció** usualment quan això no va és causat per el keytab, no s'ha exportat bé, no s'ha creat bé, o no coincideixen els noms assignats al host.

Exemple-1
Des del propi container l'usuari local01, sense estar en posessió de cap ticket, realitza l'ordre *ssh user01@sshd.edt.org*, el servidor li demanarà el password, en tractar-se d'un usuari de *AP* kerberos, verificarà l'autebnticació contra el servidor kerberos (el *IP* l'ha obtingut de /etc/passwd).

Exemple-2
Des del propi container l'usuari local01 sol·licita un ticket de user02 amb l'ordre *kinit user02*. Si s'autentica correctament amb el password de kerberos obté un tiket. Seguidament l'usuari local01 realitza l'ordre *ssh user01@sshd.edt.org* i conecta automàticament al servidor ssh sense que se li demnai el password.
Perquè? perquè està jà en possesió d'un ticket kerberos vàlid que el servidor sshd verifica i li permet iniciar sessió ssh sense necessitat de demanar-li el password (similar a l'accés per clau pública).

```
# ssh user01@172.21.0.3
The authenticity of host '172.21.0.3 (172.21.0.3)' can't be established.
ECDSA key fingerprint is SHA256:FakX5h5J4mbjss2v3b4F4vqPllFn+AWXLj7f8ivdeAs.
ECDSA key fingerprint is MD5:bb:50:c9:26:1b:16:df:5c:91:b3:5a:b3:7d:69:82:7a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.21.0.3' (ECDSA) to the list of known hosts.
user01@172.21.0.3's password: kuser01
Last failed login: Fri Feb 22 16:49:32 UTC 2019 from 172.21.0.1 on ssh:notty

[user01@sshd ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_1003_h55yoBfeGG
Default principal: user01@EDT.ORG
Valid starting     Expires            Service principal
02/22/19 16:49:35  02/23/19 16:49:35  krbtgt/EDT.ORG@EDT.ORG

[user01@sshd ~]$ ssh user01@sshd.edt.org
The authenticity of host 'sshd.edt.org (172.21.0.3)' can't be established.
ECDSA key fingerprint is SHA256:FakX5h5J4mbjss2v3b4F4vqPllFn+AWXLj7f8ivdeAs.
ECDSA key fingerprint is MD5:bb:50:c9:26:1b:16:df:5c:91:b3:5a:b3:7d:69:82:7a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'sshd.edt.org,172.21.0.3' (ECDSA) to the list of known hosts.
Last login: Fri Feb 22 16:49:35 2019 from 172.21.0.1
```

```
[user01@sshd ~]$ klist
klist: No credentials cache found (filename: /tmp/krb5cc_1003)

[user01@sshd ~]$ exit
logout
Connection to sshd.edt.org closed.

[user01@sshd ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_1003_h55yoBfeGG
Default principal: user01@EDT.ORG
Valid starting     Expires            Service principal
02/22/19 16:49:35  02/23/19 16:49:35  krbtgt/EDT.ORG@EDT.ORG
02/22/19 16:49:56  02/23/19 16:49:35  host/sshd.edt.org@EDT.ORG

[user01@sshd ~]$ ssh user01@sshd.edt.org
Last login: Fri Feb 22 16:49:56 2019 from 172.21.0.3
```

Observeu com en la sessió actual de user01 a més a més del seu ticket té el ticket del servidor sshd, que li permet iniciar sessió ssh de manera desatesa.

Integració amb PAM

Exemple amb chfn

***nota** podeu saltar aquest pas i anar directament a la autenticació amb system-auth.*
Com a exemple d'integració de PAM utilitzant kerberos es redefiniran les regles PAM de l'ordre *chfn*.

Observar la configuració PAM actual per a l'ordre chfn:

```
# cat /etc/pam.d/chfn
#%PAM-1.0
auth       sufficient  pam_rootok.so
auth       include     system-auth
account    include     system-auth
password   include     system-auth
session    include     system-auth
```

Modificar la configuració PAM de chfn per autoritzar modificar el finger de l'usuari validant via kerberos:

```
# cat /etc/pam.d/chfn
#%PAM-1.0
auth            sufficient   pam_krb5.so
account         sufficient   pam_krb5.so
password        include         system-auth
session          include         system-auth
```

Observar que si l'usuari si s'identifica amb el password corresponent al seu compte local de la màquina no es permet que modifiqui el seu finger. En canvi, si s'identifica amb el password del seu compte de Kerberos llavors obté el permís

```
$ chfn
S'està canviant la informació del finger per a l'usuari ecanet.
Contrasenya: secret
Permís denegat

$ chfn
S'està canviant la informació del finger per a l'usuari ecanet.
Contrasenya: kerberos-passwd
Nom []:
Office []:
Office Phone []:
Home Phone []:
OK!
```

```
$ su -  jan
Contrasenya:
[jan@notebook1 ~]$ chfn
Changing finger information for jan.
Password: secret
Permission denied

[jan@notebook1 ~]$ chfn
Changing finger information for jan.
Password: kerberos-passwd
Name []:
Office []:
....
```

Exemple amb login

***nota** podeu saltar aquest pas i anar directament a la autenticació amb system-auth.*
Un exemple més complert és realitzar l'autenticació complerta d'usuaris via kerberos. És a dir, que els usuaris del sistema s'autentifiquen no amb el passwd del passwd/shadow sinó amb kerberos.

Per fer això es simplificarà el model i treballarem amb:

- Un docker host client de kerberos.
- Amb usuaris locals al /etc/passwd als que no s'ha assignat password. Calen els usuaris locals per disposar dels comptes dels usuaris (uid, shell, homedir, etc). Però no els assignem passwd per verificar que la autenticació la fa kerberos
- Modificar el PAM de login (fer-ne una copia) establint el mòdul pam_krb5.so com a mecanisme d'autenticació.

Observem els usuaris creats i recordeu que no tenen passwd assignat:

```
[root@khostpam ~]# tail -8 /etc/passwd
pere:x:1000:1000::/home/pere:/bin/bash
marta:x:1001:1001::/home/marta:/bin/bash
anna:x:1002:1002::/home/anna:/bin/bash
julia:x:1003:1003::/home/julia:/bin/bash
ramon:x:1004:1004::/home/ramon:/bin/bash
jordi:x:1005:1005::/home/jordi:/bin/bash
jan:x:1006:1006::/home/jan:/bin/bash
pau:x:1007:1007::/home/pau:/bin/bash
```

El pam de login (un cop salvaguardat queda, **/etc/pam.d/login**:

```
#%PAM-1.0
auth    sufficient   pam_krb5.so
account         sufficient   pam_krb5.so
session         sufficient   pam_krb5.so
```

En el docker khostpam root fa login i esdevé pere. Fallen algunes coses degut en part al docker i en part al PAM que l'hem retallat molt!

```
[root@khostpam ~]# login pere
Password: kpere
-bash: cannot set terminal process group (-1): Inappropriate ioctl for device
-bash: no job control in this shell

[pere@khostpam ~]$ id
uid=1000(pere) gid=1000(pere) groups=1000(pere)
```

Exemple amb system-auth

Donat un host kerberitzat podem molt fàcilment configurar l'autenticació dels usuaris amb kerberos, simplemenmt cal configurar apropiadament system-auth. Podem configurar system-auth manualment o utilitzant *authconfig*.

Models a implementar:

- ❏ Un primer exemple és fer un host kerberos que permeti amb PAM l'autenticació d'usuaris locals i d'usuaris kerberos (recordeu que la informació del compte d'usuari

serà local al /etc/passwd però el password serà kerberos).
Podeu trobar aquest exemple a github *edtasixm11/k18:khostp*.

- ❏ Un segon model és fer un host completament similar als hosts dels alumnes de l'escola, amb autenticació PAM que permeti usuaris locals i usuaris de ldap, utilitzant kerberos. En aquest model tindrem usuaris locals (tipus local01), usuaris amb el compte d'usuari local i el passwd de kerberos (tipus user01) i usuaris de ldap amb passwd kerberos (tipus pere).
  Podeu trobar aquest exemple a github *edtasixm11/k18:khostpl*.

Host PAM Kerberos

El procés per configurar l'autenticació PAM usant pam unix i pam kerberos és molt senzill, cal:

- Instal·lar el paquet pam_krb5 per poder usar el mòdul pam_kerb5.so.
- Configurar system-auth (manualmen o via *authconfig*) per usar pam_unix i pam_kerb5.so.

Exemple de configuració pam amb pam-unix i pam-kerberos:

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        sufficient    pam_unix.so try_first_pass nullok
auth        sufficient    pam_krb5.so
auth        required      pam_deny.so

account     sufficient    pam_unix.so
account     sufficient    pam_krb5.so

password requisite   pam_pwquality.so try_first_pass local_users_only retry=3 authtok_type=
password sufficient  pam_unix.so try_first_pass use_authtok nullok sha512 shadow
password sufficient  pam_krb5.so
password required    pam_deny.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
-session    optional      pam_systemd.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session     sufficient    pam_krb5.so
session sufficient pam_unix.so
```

Host PAM Kerberos Ldap

Per configurar un host com els de les aules (d'alumnes) que permeten l'autenticació loc al i via ldap amb kerberos, cal fer:

- Instal·lar pam_krb5 per poder disposar de pam_krb5.so
- Configurar tota la part de client ldap (descrita al HowTo-ASIX-PAM) que com recordeu implica configurar el client ldap, nslcd, cscd i nsswitch.
- Configurar l'autenticació PAM de system-auth manualment o usant *authconfig*.

Si la configuració és com de la de classe podem partir de l'exemple de configuració de authconfig que hi ha a les instruccions d'instal·lació dels hosts de classe a fedora@inf. La diferència és com es diu la base de dades ldap i el kingdom de kerberos.

Exemple de configuració amb authconfig:

```
authconfig  --enableshadow --enablelocauthorize --enableldap \
          --ldapserver='ldap.edt.org' --ldapbase='dc=edt,dc=org' \
          --enablekrb5 --krb5kdc='kserver.edt.org' \
          --krb5adminserver='kserver.edt.org' --krb5realm='EDT.ORG' \
          --updateall
```

Host PAM Kerberos Ldap amb mkhomedir i pam_mount

L'exemple anterior es pot ampliar amb:

- Afegir al system-auth pam_mkhomedir.so per assegurar-se de crear els homes dels usuaris de ldap.

- Afegir al system-auth pam_mount.so i la configuració pertinent de pam_mount.conf.xml per muntar els homes dels usuaris (o els homes dins els homes) via NFS o Samba (Exercicis fets en el HowTo-ASIX-Samba).

Pràctiques

Pràctica proposada 2018

edtasixm11/k18

Proposta conceptes docker a exposar usant kerberos:

- volumes, bind-mounts, tmpfs
- docker-compose, service, stack, swarm.

Propostes de pràctiques Kerberos:

- kserver servidor kerberos.
- khost host simple per obtenir tickets.
- ksshd servidor sshd amb usuaris locals i kerberos,
- khostp host amb pam que autentica kerberos usuaris passwd.
- khostpl host amb pam que autentica usuaris locals i ldap, amb kerberos.
- sshdpl servidor sshd amb pam d'usuaris locals i ldap i autenticació kerberos
- script-kerberos: fer un script que obté els noms de usuari de ldap i crea els principals i passwd klogin.
- script-mkhome: fer un script que obté de ldap el login,uid i gid dels usuaris i crea els seus homes assignant la propietat i grup.
- VolumeUser que crea (?) usar un container d'infraestructura tipus volum + script els homes dels usuaris.

Desplegaments (compose/AWS)

> ***Nota:***
> per fer aquests exercicis abans s'ha parcticat una mica els conceptes / tecnologies de:
> - docker-compose: serveis i apps.
> - docker stack
> - docker swar.
> - nodes
> - AWS EC2.

Un cop construïts tots els components descrits a les pràctiques podem passar a desplegar-los no un a un sinó en conjunt, tots de cop. Per això veurem per sobre com fer-ho utilitzant:

- *docker-compose* per desplegar localment al nostre host una app amb els serveis kserver, sshd, ldap. Verificar localment el funcionament de l'autenticació.

- *AWS EC2*: desplegar a Amazon AWS EC2 el docker-compose anterior i observar com usant un client local al nostre host podem realitzar l'autenticació contra els serveis desplegats amb docker-compose dins del núvol.

Desplegament Docker-compose

Objectiu / Descripció:

- Desplegar els containers kserver, sshd i ldap tots tres en una xarxa mynet amb redirecció dels ports de ldap (389) ssh (1022) i kerberos (88,464, 749).
- Això farà que el nostre host tingui mapejats els ports 389, 1022, 88, 464 i 749, de manera que accedint a ells s'accedeix realment als serveis que hi ha a l'interior en els containers.
- Es tracta de verificar que un host client, configurat per atacar el nostre host com a servidor (kerberos, ssh i ldap) pot usar els serveis que en realitat proporcionen els containers.
- Podriem configuarar un host de l'aula però per no desconfigurar-lo de la configuració actual, és més senzill engegar un container khostpl que actuarà com un host client.
- Aquest host client (el container) ha d'atacar els serveis que proporciona kserver.edt.org, sshd.edt.org i ldap.edt.org.
  Per tant es tracta de definir al seu /etc/hosts el lligam entre aquests noms de domini i la adreça IP real del host on hem desplegat els containers.
  nota: l'ordre de la línia del /etc/hosts és significatiu, posar primer el ssh!
- Verificar el funcionament dels serveis:
  - fer el test obtenint tiquets.

    - fer el test connectant al servei ssh amb usuari local (local01) i usuari kerberos (user01).
    - fer el test del servei ssh kerberitzat: obtenir ticket de user01 i llavors fer sessió ssh automatitzada.
- Verificar el funcionament de l'autenticació:
    - iniciar sessió local (recordeu que primer cal fer una primera sessió amb su).
    - iniciar sessió amb usuari kerberos.
    - iniciar sessió amb usuari ldap+kerberos.

Implementació

El servei:

Fitxer per al desplegament amb docker compose:

```
version: "2"
services:
  kserver:
        image: edtasixm11/k18:kserver
        container_name: kserver.edt.org
        hostname: kserver.edt.org
        ports:
        - "88:88"
        - "464:464"
        - "749:749"
        networks:
        - mynet
  sshd:
        image: edtasixm11/k18:sshd
        container_name: sshd.edt.org
        hostname: sshd.edt.org
        ports:
        - "1022:22"
        networks:
        - mynet
  ldap:
        image: edtasixm06/ldapserver:18group
        container_name: ldapserver.edt.org
        hostname: ldap.edt.org
        ports:
        - "389:389"
        networks:
        - mynet
networks:
  mynet:
```

Engegar/aturar el desplegament:

```
docker-compose up -d
docker-compose down
```

```
$ docker ps
CONTAINER ID        IMAGE                          COMMAND                  CREATED             STATUS
PORTS                                              NAMES
f494de571ecf        edtasixm11/k18:khostpl         "/opt/docker/start..."   36 seconds ago      Up 33 seconds
                    khost
0bfcde4a392a        edtasixm06/ldapserver:18group  "/opt/docker/start..."   About a minute ago  Up About a minute
0.0.0.0:389->389/tcp                               ldapserver.edt.org
a89d6d9e924f        edtasixm11/k18:kserver         "/opt/docker/start..."   About a minute ago  Up About a minute
0.0.0.0:88->88/tcp, 0.0.0.0:464->464/tcp, 0.0.0.0:749->749/tcp   kserver.edt.org
38ebb21ed601        edtasixm11/k18:sshd            "/opt/docker/start..."   About a minute ago  Up About a minute
0.0.0.0:1022->22/tcp                               sshd.edt.org
```

El host té oberts els ports 389, 1022, 88, 464 i 749 (entre altres). Aquí hem usat la ip de la xarxa interna xxxx_mynet creada per docker compose (obtinguda amb network inspect).

```
$ nmap 172.19.0.1
PORT     STATE    SERVICE
22/tcp   open     ssh
88/tcp   filtered kerberos-sec
389/tcp  filtered ldap
464/tcp  filtered kpasswd5
749/tcp  filtered kerberos-adm
1022/tcp filtered exp2
```

El client:

Engegar el host que farà la funció de client:

```
docker run --rm --name khost.edt.org -h khost.edt.org --net mynet -it
edtasixm11/k18:khostpl
```

Configurar el host client per accedir als serveis que proporciona el host on s'ha fet el desplegament amb docker-compose:

```
vi /etc/hosts
A.B.C.D sshd.edt.org kserver.edt.org ldsap.edt.org
```

*****nota*****

És important l'ordre en que es fa l'associació dels noms de domini a l'adreça IP. Cal que el servidor sshd sigui el primer per poder fer ús del servei sshd kerberitzat (entrada automàtica quan disposem de un tocket). perquè?

Perquè en el procés de validació del ticket es fa una resolució del nom de domini i es mira quin és el nom del host A.B.C.D i si aquest és per exemple kserver.edt.org no coincideix amb el principal que té al keytab el servidor sshd, que és host/sshd.edt.org.

Ara tenim un container local khost que ataca els serveis kerberos, ssh i ldap que ofereix el host on hem desplegat el docker-compose.

```
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS                                   NAMES
```

```
f494de571ecf        edtasixm11/k18:khostpl        "/opt/docker/start..."   4 minutes ago       Up 4 minutes
            khost
...
```

Instal·lar-hi el openssh-clients i nmap

```
dnf -y install openssh-clients nmap
```

Configurar l'adreça IP dels serveis

```
127.0.0.1               localhost
::1             localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      khost
172.17.0.1 sshd.edt.org kserver.edt.org ldap.edt.org
```

Observeu que hem usat l'adreça del host amfitrió de docker perquè en aquest cas el desplegament dels containers del servei amb docker-compose i del client khostpl l'hem fet tot al mateix host.

Atenció: CAL que el primer nom de domini sigui el del sshd per poder usar el servei sshd kerberitzat apropiadament.

Comprovem que el host client khostpl té accés als serveis fent un test amb nmap a cada un dels noms de domini:

```
$ nmap sshd.edt.org
$ nmap kserver.edt.org
$ nmap sshd.edt.org
```

Verificació:

Verificar servei kserver

```
# kinit user01
Password for user01@EDT.ORG: kuser01

# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: user01@EDT.ORG
Valid starting       Expires              Service principal
02/03/19 12:17:32  03/03/19 12:17:32  krbtgt/EDT.ORG@EDT.ORG

$kinit pere
Password for pere@EDT.ORG: kpere

# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: pere@EDT.ORG
Valid starting       Expires              Service principal
02/03/19 12:18:00  03/03/19 12:18:00  krbtgt/EDT.ORG@EDT.ORG
```

Verificació servei sshd

```
# ssh -p 1022 user01@sshd.edt.org
user01@sshd.edt.org's password: kuser01
[user01@sshd ~]$
^d

**recordeu que pere no és un usuari vàlid ssh
```

Servei sshd kerberitzat (inici sessió automatitzat)

```
# kinit user01
Password for user01@EDT.ORG: kuser01

# ssh -p 1022 user01@sshd.edt.org
Last login: Sat Mar  2 11:20:23 2019 from 172.19.0.1
[user01@sshd ~]$
```

Verificació autenticació

```
su - local01

su - user01

su - pere
```

Desplegament AWS

Objectiu / Descripció:

- Repetir el mateix exercici anterior però dins del núvol, per exemple a AWS EC2. En un host AMI desplegar-hi el docker-compose.
- Obrir els ports del firewall de AWS EC2 dels ports 389, 1022, 88, 464 i 749.
- Des de un host client, per exemple el container khostpl executant-se en el nostre host de l'aula, verificar el funcionament de kerberos, ldap i sshd.
- Des del host client, un cop configurat el /etc/hosts per accedir als serveis de kserver.edt.org, sshd.edt.org i ldap.edt.org, verificar l'autenticació d'usuairs locals, kerberoos i ldap contra la màquina AMI de AWS EC2.
- En resum, hem desplegat a AWS EC2 un conjunt de serveis que serveixen per autenticar usuaris via kerberos+ldap i permetre l'accés ssh des de qualsevol màquina del món que actuï de client (ben configurada).

Annex

Dockers Kerberos (2016-2017)

Descarregar imatge Docker:
$ docker pull edtasixm11/kerberos

Altres dockers:

- edtasixm11/kserver
- edtasixm11/kclient
- edtasixm11/sshsever
- edtasixm11/khostpam

- edtasixm11/kappserver
- edtasixm11/kappclie

GitHub Kerberos 2017-2018

Al compte https://github.com/edtasixm11/kerberos trobareu exemples de dockerfiles per generar les següents imatges:

Kerberos

Repositori d'imatges de kerberos @edt ASIX-M11 Curs 2017-2018

Imatges:

kserver

Servidor kerberos del kingdom EDT.ORG. Incorpora els usuaris pere, marta, anna, pau, julia, jordi (password k) L'usuari pau és administrador i marta/admin també .

kclient

Simple client de kerberos del kingdom EDT.ORG. Aquest host no fa autenticació d'usuaris (PAM...) amb kerberos, simplement permet fer kinit / klist / kdestroy / kadmin. És a dir, treballar com a un client de kerberos.
nota per veure un host fent autenticació d'usuaris del sistema contra kerberos mireu la imatge khost.

khost

Host client de kerberos que realitza la autenticació d'usuaris PAM validant-los contra kerberos. S'ha modificat el PAM de chfn, login i system-auth perquè usin pam_krb5.so (kerberos). Observar que els usuaris validats es poden canviar el passwd amb la ordre passwd normal i en realitat modifiquen el passwd de kerberos.

kssh

Servidor SSH kerberos aware. Servidor ssh que permet l'autenticació d'usuaris amb kerberos (autenticació de clients ssh). Es creen comptes de usuaris locals unix però no se'ls assigna passwd per mostrar més clarament que l'autenticació és kerberos. Usuaris creats: pere, anna, marta, pau, julia, jordi

GitHub Kerberos 2018-2019

Al compte github edtasixm11 trobareu exemples de dockerfiles per generar les imatges treballades aquest any.

edtasixm11/k18:kserver servidor kerberos detach. Crea els usuaris pere pau (admin), jordi, anna, marta, marta/admin i julia. Assignar-li el nom de host: *kserver.edt.org*

edtasixm11/k18:khost host client de kerberos. Simplement amb eines kinit, klist i kdestroy (no pam). El servidor al que contacta s'ha de dir *kserver.edt.org*.

edtasixm11/k18:sshd Servidor SSHD *kerberitzat*. Servidor ssh que permet l'accés d'usuaris locals i usuaris locals amb autenticació kerberos. El servidor s'ha de dir sshd.edt.org.

edtasixm11/k18:khostp host amb PAM de kerberos. El servidor al que contacta s'ha de dir *kserver.edt.org*. Aquest host configura el *system-auth* de pam per usar el mòdul *pam_krb5.so*.

edtasixm11/k18:khostpl host amb PAM amb autenticació AP de kerberos i IP de ldap. El servidor kerberos al que contacta s'ha de dir *kserver.edt.org*. El servidor ldap s'anomena ldap.edt.org. Aquest host es configura amb authconfig .