```
-> CAMBIOS EN LA CONFIGURACIÓN DE /etc/ssh/sshd_config PARA QUE SE PUEDAN
   CONECTAR A NOSOTROS MEDIANTE NUESTRO SERVIDOR SSH KERBERIZADO.

-> EN EL SERVIDOR...

-> Lo primero: borramos todos los TGT que tengamos...

[root@servidor ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: alumno@LAB2701.ASO

Valid starting     Expires            Service principal
04/20/12 18:19:39  04/21/12 18:19:39  krbtgt/LAB2701.ASO@LAB2701.ASO
    renew until 04/20/12 18:19:39
04/20/12 18:20:29  04/21/12 18:19:39  host/pc2.lab2701.aso@LAB2701.ASO
    renew until 04/20/12 18:19:39
[root@servidor ~]# kdestroy
[root@servidor ~]# klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)

-> Ahora procedemos con el cambio de configuración...

[root@servidor ~]# hostname
servidor.lab2701.aso
[root@servidor ~]# ping pc2.lab2701.aso
PING pc2.lab2701.aso (155.54.225.2) 56(84) bytes of data.
64 bytes from pc2.lab2701.aso (155.54.225.2): icmp_req=1 ttl=64 time=0.118 ms
64 bytes from pc2.lab2701.aso (155.54.225.2): icmp_req=2 ttl=64 time=0.107 ms
^C
--- pc2.lab2701.aso ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.107/0.112/0.118/0.011 ms
[root@servidor ~]# cd /etc/ssh
[root@servidor ssh]# cp sshd_config sshd_config_20120420
[root@servidor ssh]# vi ./sshd_config

[root@servidor ssh]# cat /etc/ssh/sshd_config | grep -i kerberos
# Kerberos options
KerberosAuthentication yes
```

```
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no
#KerberosUseKuserok yes
[root@servidor ssh]#

[root@servidor ssh]# /etc/init.d/sshd restart
Restarting sshd (via systemctl):                           [  OK  ]

-> EN EL CLIENTE...

[root@pc2 ~]# klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)
[root@pc2 ~]# ssh alumno@servidor.lab2701.aso
alumno@servidor.lab2701.aso's password:
Last login: Fri Apr 20 18:24:46 2012 from pc2.lab2701.aso
[alumno@servidor ~]$ hostname
servidor.lab2701.aso
[alumno@servidor ~]$ pwd
/home/alumno
[alumno@servidor ~]$ exit
logout
Connection to servidor.lab2701.aso closed.

* * *

-> COMENTARIOS SOBRE EL TGT OBTENIDO CON SSH:

Cuando desde "servidor" hacemos un ssh contra "pc2" autenticándonos
como usuario "alumno" (contraseña "alumno2012"), el usuario que realmente
se está autenticando es el usuario "alumno" de "pc2", con lo cuál, es
éste, y no el "alumno" de "servidor" quien recibe el TGT.

Esto lo veremos claramente en la siguiente secuencia:

1a. SITUACIÓN INICIAL EN EL SERVIDOR:

[alumno@servidor ~]$ hostname
servidor.lab2701.aso
```

```
[alumno@servidor ~]$ klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_500)

1b. SITUACIÓN INICIAL EN EL CLIENTE:

[alumno@pc2 ~]$ hostname
pc2.lab2701.aso
[alumno@pc2 ~]$ klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_500)

2. CONEXIÓN, DESDE "servidor", A "pc2", COMO USUARIO "alumno" de "pc2":

[alumno@servidor ~]$ ssh alumno@pc2.lab2701.aso
alumno@pc2.lab2701.aso's password:
Last login: Sat Apr 21 08:08:20 2012 from servidor.lab2701.aso
[alumno@pc2 ~]$ hostname
pc2.lab2701.aso
[alumno@pc2 ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_500_KFzYrl2106
Default principal: alumno@LAB2701.ASO

Valid starting     Expires            Service principal
04/21/12 08:15:58  04/22/12 08:15:58  krbtgt/LAB2701.ASO@LAB2701.ASO
    renew until 04/21/12 08:15:58

-> NÓTESE, además, que, como era de esperar, una vez que nos hemos conectado
a "pc2" mediante ssh, puesto que el "alumno" de "pc2" ya cuenta con su TGT,
nada le impide llevar a cabo una conexión ssh a "servidor" SIN TENER QUE
INTRODUCIR SU CONTRASEÑA...

[alumno@pc2 ~]$ ssh alumno@servidor.lab2701.aso
Last login: Sat Apr 21 08:07:18 2012 from pc2.lab2701.aso
[alumno@servidor ~]$ hostname
servidor.lab2701.aso

-> SALIDA DE LOS TERMINALES:

[alumno@pc2 ~]$ exit
logout
```

```
Connection to pc2.lab2701.aso closed.
[alumno@servidor ~]$ hostname
servidor.lab2701.aso
[alumno@servidor ~]$ klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_500)

* * *

-> OTRA COSA DISTINTA ES QUE... antes de hacer el primer ssh, nos cojamos
el TGT con "kinit". En ese caso, ya no se nos pediría la contraseña al
hacer ssh, porque ya tendríamos el TGT. En este caso el TGT lo tendría el
usuario "alumno" de "servidor" porque es el usuario "alumno" de "servidor"
el que se está autenticando (NO con ssh, SINO con "kinit"):

[alumno@servidor ~]$ hostname
servidor.lab2701.aso
[alumno@servidor ~]$ klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_500)
[alumno@servidor ~]$ kinit -p alumno@LAB2701.ASO
Password for alumno@LAB2701.ASO:
[alumno@servidor ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_500
Default principal: alumno@LAB2701.ASO

Valid starting     Expires            Service principal
04/21/12 08:36:43  04/22/12 08:36:43  krbtgt/LAB2701.ASO@LAB2701.ASO
    renew until 04/21/12 08:36:43
[alumno@servidor ~]$ hostname
servidor.lab2701.aso
[alumno@servidor ~]$ ssh alumno@pc2.lab2701.aso
Last login: Sat Apr 21 08:15:58 2012 from servidor.lab2701.aso
[alumno@pc2 ~]$ hostname
pc2.lab2701.aso
[alumno@pc2 ~]$ who
alumno   tty1         2012-04-21 08:02 (:0)
alumno   pts/0        2012-04-21 08:03 (:0)
alumno   pts/1        2012-04-21 08:37 (servidor.lab2701.aso)
[alumno@pc2 ~]$ exit
logout
```

```
Connection to pc2.lab2701.aso closed.
[alumno@servidor ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_500
Default principal: alumno@LAB2701.ASO

Valid starting     Expires            Service principal
04/21/12 08:36:43  04/22/12 08:36:43  krbtgt/LAB2701.ASO@LAB2701.ASO
    renew until 04/21/12 08:36:43
04/21/12 08:37:01  04/22/12 08:36:43  host/pc2.lab2701.aso@LAB2701.ASO
    renew until 04/21/12 08:36:43

-> COMO vemos en el último "klist" después del cierre de la conexión
ssh, seguimos conservando el TGT que hemos adquirido con "kinit", y
así seguirá siendo durante las siguientes 24 horas (ver campo
"Expires"), o bien hasta que nosotros, voluntariamente, lo eliminemos
con "kdestroy":

[alumno@servidor ~]$ kdestroy
[alumno@servidor ~]$ klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_500)
[alumno@servidor ~]$

* * *

-> SOBRE EL PERÍODO DE VIDA...

Es posible modificar el período máximo de validez del TGT. Como
puede apreciarse en cualquiera de los "klist" anteriores, éste es,
por defecto, de 24 horas. Si quisiéramos cambiarlo a, por ejemplo,
12 horas, habría que modificar el fichero principal de configuración
de Kerberos (/etc/krb5.conf) y reiniciar el servicio:

[alumno@servidor ~]$ su -
Contraseña:
[root@servidor ~]# cat /etc/krb5.conf | grep -i "ticket_lifetime"
 ticket_lifetime = 24h
[root@servidor ~]# vi /etc/krb5.conf
[root@servidor ~]# cat /etc/krb5.conf | grep -i "ticket_lifetime"
 ticket_lifetime = 12h
```

```
[root@servidor ~]# /etc/init.d/krb5kdc restart
Restarting krb5kdc (via systemctl):                        [  OK  ]
[root@servidor ~]# kinit -p alumno@LAB2701.ASO
Password for alumno@LAB2701.ASO:
[root@servidor ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: alumno@LAB2701.ASO

Valid starting     Expires            Service principal
04/21/12 08:43:27  04/21/12 20:43:27  krbtgt/LAB2701.ASO@LAB2701.ASO
    renew until 04/21/12 08:43:27
[root@servidor ~]# kdestroy
[root@servidor ~]# klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)

* * *

-> AHORA VAMOS A "kerberizar" PAM:

[root@servidor ~]# ls -l /etc/pam.d/system-auth
lrwxrwxrwx. 1 root root 14 sep  7  2011 /etc/pam.d/system-auth -> system-auth-ac
[root@servidor ~]# cp /etc/pam.d/system-auth-ac /etc/pam.d/system-auth-ac_20120420
[root@servidor ~]# cat /etc/pam.d/system-auth
#%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_fprintd.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 500 quiet
account     required      pam_permit.so

password    requisite     pam_cracklib.so try_first_pass retry=3 type=
password    sufficient    pam_unix.so sha512 shadow nullok try_first_pass use_authtok
```

```
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     required      pam_unix.so
[root@servidor ~]# vi /etc/pam.d/system-auth
[root@servidor ~]# cat /etc/pam.d/system-auth
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.

# "kerberizamos" el servicio PAM, para hacer posible la autenticación
# de usuarios de ESTE equipo mediante contraseñas que pertenecen al
# mismo usuario del reini Kerberos.
auth        sufficient    pam_krb5.so forwardable

auth        required      pam_env.so
auth        sufficient    pam_fprintd.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 500 quiet
account     required      pam_permit.so

password    requisite     pam_cracklib.so try_first_pass retry=3 type=
password    sufficient    pam_unix.so sha512 shadow nullok try_first_pass use_authtok
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     required      pam_unix.so
[root@servidor ~]# locate pam_krb5.so
```

```
/lib/security/pam_krb5.so

-> Si ahora abrimos un terminal de texto (ctrl + alt +f2, por ejemplo),
comprobaremos que ya nos podemos autenticar como "alumno" con la contraseña
"alumno2012" (la de Kerberos).

-> SIN EMBARGO...

Esta autenticación todavía no funciona desde el terminal gráfico.
Deberíamos ejecutar el comando "authconfig" con los argumentos
"enablekrb5" y "update". Este comando ya se encarga de actualizar por
nosotros el fichero de configuración de PAM (/etc/pam.d/system-auth),
para que sea compatible con la autenticación Kerberos.

[root@servidor ~]# grep -i "krb5" /etc/pam.d/system-auth
auth        sufficient    pam_krb5.so forwardable
[root@servidor ~]# authconfig --enablekrb5 --update
[root@servidor ~]# grep -i "krb5" /etc/pam.d/system-auth
auth        sufficient    pam_krb5.so use_first_pass
account     [default=bad success=ok user_unknown=ignore] pam_krb5.so
password    sufficient    pam_krb5.so use_authtok
session     optional      pam_krb5.so

-> Ahora ya sí que nos funcionaría la autenticación de "alumno"
mediante la contraseña "alumno2012" desde el cuadro de diálogo de
introducción de credenciales de Fedora.

Ahora bien, la contraseña del usuario "alumno" que hay en /etc/passwd
("alumno") sigue estando vigente, y de hecho también podríamos entrar
con ella. Esto es porque PAM está configurado así; fijarse en la línea

auth        sufficient    pam_krb5.so use_first_pass

que indica que primero debe comprobarse la autenticación Kerberos,
pero sin descartar la autenticación local.

Ahora bien, siempre podemos bloquear la contraseña local de "alumno",
con lo que ya sólo sería posible la autenticación kerberos.
```

```
* * *

-> VAMOS AHORA A CREAR UN NUEVO PRINCIPAL DEL REINO "LAB2701.ASO",
correspondiente al usuario "eduardo", PERO SIN CREAR LA CUENTA EN
EN EL KDC ("servidor.lab2701.aso"). Lo que sí haremos es crear la
cuenta "eduardo" en el host cliente ("pc2.lab2701.aso"), aunque
bloqueando su contraseña local, para estar seguros de que la autenticación
que se realiza es la de Kerberos.

EN EL SERVIDOR...

[alumno@servidor ~]$ kadmin.local
Couldn't open log file /var/log/kadmind.log: Permission denied
Authenticating as principal alumno/admin@LAB2701.ASO with password.
kadmin.local: Permission denied while initializing kadmin.local interface
[alumno@servidor ~]$ su -
Contraseña:
[root@servidor ~]# kadmin.local
Authenticating as principal root/admin@LAB2701.ASO with password.
kadmin.local:  listprincs
K/M@LAB2701.ASO
alumno@LAB2701.ASO
host/pc2.lab2701.aso@LAB2701.ASO
host/servidor.lab2701.aso@LAB2701.ASO
kadmin/admin@LAB2701.ASO
kadmin/changepw@LAB2701.ASO
kadmin/servidor.lab2701.aso@LAB2701.ASO
krbtgt/LAB2701.ASO@LAB2701.ASO
root/admin@LAB2701.ASO
kadmin.local:  addprinc eduardo
NOTICE: no policy specified for eduardo@LAB2701.ASO; assigning "default"
Enter password for principal "eduardo@LAB2701.ASO":
Re-enter password for principal "eduardo@LAB2701.ASO":
Principal "eduardo@LAB2701.ASO" created.
kadmin.local:  listprincs
K/M@LAB2701.ASO
alumno@LAB2701.ASO
eduardo@LAB2701.ASO
host/pc2.lab2701.aso@LAB2701.ASO
```

```
host/servidor.lab2701.aso@LAB2701.ASO
kadmin/admin@LAB2701.ASO
kadmin/changepw@LAB2701.ASO
kadmin/servidor.lab2701.aso@LAB2701.ASO
krbtgt/LAB2701.ASO@LAB2701.ASO
root/admin@LAB2701.ASO
kadmin.local:  q
[root@servidor ~]#
```

-> NOTAS:

* A "eduardo" le hemos puesto la contraseña kerberos "eduardo2012".

* Nótese que "kadmin.local" y "kadmin" sólo se pueden ejecutar como "root"
(no hay, por defecto, permisos de escritura para /var/log/kadmind.log).
Además, si ejecutamos "kadmin" se nos pedirá la contraseña del usuario
administrativo de la base de datos principal ("root/admin@LAB2701.ASO",
según lo indicado en /var/kerberos/krb5kdc/kadm5.acl). Si ejecutamos
"kadmin.local" no se nos pide contraseña alguna, porque se supone que
ese programa sólo lo podemos ejecutar desde el KDC, y además sólo
como "root".

```
[root@servidor ~]# kadmin
Authenticating as principal root/admin@LAB2701.ASO with password.
Password for root/admin@LAB2701.ASO:
kadmin:  listprincs
K/M@LAB2701.ASO
alumno@LAB2701.ASO
eduardo@LAB2701.ASO
host/pc2.lab2701.aso@LAB2701.ASO
host/servidor.lab2701.aso@LAB2701.ASO
kadmin/admin@LAB2701.ASO
kadmin/changepw@LAB2701.ASO
kadmin/servidor.lab2701.aso@LAB2701.ASO
krbtgt/LAB2701.ASO@LAB2701.ASO
root/admin@LAB2701.ASO
kadmin:  q
```

-> SEGUIMOS CON LA AUTENTICACIÓN KERBEROS DE "eduardo",

```
AHORA EN EL CLIENTE...

[alumno@pc2 ~]$ su -
Contraseña:
[root@pc2 ~]# useradd eduardo
[root@pc2 ~]# id eduardo
uid=501(eduardo) gid=502(eduardo) grupos=502(eduardo)
[root@pc2 ~]# ls /home/eduardo
ls: no se puede acceder a /home/eduardo: No existe el fichero o el directorio
[root@pc2 ~]# cat /etc/passwd | grep -i eduardo
eduardo:x:501:502::/home/eduardo:/bin/bash
[root@pc2 ~]# cat /etc/shadow | grep -i eduardo
eduardo:!!:15451:0:99999:7:::
[root@pc2 ~]# grep -i "krb5" /etc/pam.d/system-auth
[root@pc2 ~]# authconfig --enablekrb5 --update
[root@pc2 ~]# grep -i "krb5" /etc/pam.d/system-auth
auth        sufficient    pam_krb5.so use_first_pass
account     [default=bad success=ok user_unknown=ignore] pam_krb5.so
password    sufficient    pam_krb5.so use_authtok
session     optional      pam_krb5.so
[root@pc2 ~]# klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)

-> DESDE EL CLIENTE, YA PODEMOS AUTENTICARNOS COMO "eduardo"
(nótese que la única contraseña posible es la Kerberos).

-> Además, tenemos el TGT de "eduardo", con lo que teóricamente
podríamos autenticarnos directamente contra cualquier servicio kerberizado,
como, por ejemplo, el ssh de "servidor" (SIEMPRE Y CUANDO, CLARO ESTÁ,
en el servidor el usuario tenga algún tipo de permiso).

[eduardo@pc2 ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_501_HCJlqT
Default principal: eduardo@LAB2701.ASO

Valid starting     Expires            Service principal
04/21/12 09:40:38  04/22/12 09:40:38  krbtgt/LAB2701.ASO@LAB2701.ASO
    renew until 04/21/12 09:40:38
[eduardo@pc2 ~]$ ssh eduardo@servidor.lab2701.aso
```

```
The authenticity of host 'servidor.lab2701.aso (155.54.225.1)' can't be established.
RSA key fingerprint is d4:1c:7c:22:a0:37:d3:b3:c0:0c:4e:79:03:cc:75:53.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'servidor.lab2701.aso,155.54.225.1' (RSA) to the list of known hosts.
eduardo@servidor.lab2701.aso's password:
Permission denied, please try again.
eduardo@servidor.lab2701.aso's password:
Received disconnect from 155.54.225.1: 2: Too many authentication failures for eduardo
```

-> COMO LA CUENTA "eduardo" NO EXISTE EN "servidor.lab2701.aso", no podemos conectarnos con ssh. Pero, ¿y si nos hubiésemos autenticado como "alumno" mediante Kerberos? ("alumno" sí que tiene cuenta en "servidor.lab2701.aso").

```
[alumno@pc2 ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_500_VZnqTf
Default principal: alumno@LAB2701.ASO

Valid starting     Expires            Service principal
04/21/12 09:46:14  04/22/12 09:46:14  krbtgt/LAB2701.ASO@LAB2701.ASO
    renew until 04/21/12 09:46:14
[alumno@pc2 ~]$ hostname
pc2.lab2701.aso
[alumno@pc2 ~]$ ssh alumno@servidor.lab2701.aso
Last login: Sat Apr 21 08:53:31 2012
[alumno@servidor ~]$ hostname
servidor.lab2701.aso
[alumno@servidor ~]$ who
alumno   tty7         2012-04-21 09:13 (:0)
alumno   pts/0        2012-04-21 09:46 (pc2.lab2701.aso)
[alumno@servidor ~]$ exit
logout
Connection to servidor.lab2701.aso closed.
```

-> FIJARSE EN QUE EL TGT DEL ÚLTIMO PASO ES DISTINTO DEL TGT OBTENIDO CON "eduardo" (EL TGT ES POR USUARIO Y MÁQUINA).

-> POR ÚLTIMO, AHORA VAMOS A DAR DE ALTA "root" TAMBIÉN DENTRO DEL REINO "LAB2701.ASO" (en ese reino está "root/admin@LAB2701.ASO", que es un principal diferente del que nos disponemos a dar de alta, que es, según

```
lo dicho, "root@LAB2701.ASO").

OBJETIVO: Que "root" se pueda autenticar, como siempre, en cada máquina
con la contraseña local que en esa máquina tenga "root", pero que además
también exista otra persona que pueda actuar como "root" EN TODAS LAS MÁQUINAS
DEL REINO conociendo una contraseña común, DIFERENTE DE LAS CONTRASEÑAS
LOCALES de root en cada máquina. Esta otra contraseña común podría utilizarse
para conectarse mediante ssh, como root, a cualquier equipo de
LAB2701.ASO.

Esto, además, lo vamos a hacer desde otra máquina diferente del KDC,
en este caso desde "pc2.lab2701.aso" (también, lógicamente, lo podríamos
hacer desde el KDC con "kadmin.local").

[alumno@pc2 ~]$ su -
Contraseña:
[root@pc2 ~]# kadmin
Authenticating as principal root/admin@LAB2701.ASO with password.
Password for root/admin@LAB2701.ASO:
kadmin:  listprincs
K/M@LAB2701.ASO
alumno@LAB2701.ASO
eduardo@LAB2701.ASO
host/pc2.lab2701.aso@LAB2701.ASO
host/servidor.lab2701.aso@LAB2701.ASO
kadmin/admin@LAB2701.ASO
kadmin/changepw@LAB2701.ASO
kadmin/servidor.lab2701.aso@LAB2701.ASO
krbtgt/LAB2701.ASO@LAB2701.ASO
root/admin@LAB2701.ASO
kadmin:  addprinc root
NOTICE: no policy specified for root@LAB2701.ASO; assigning "default"
Enter password for principal "root@LAB2701.ASO":
Re-enter password for principal "root@LAB2701.ASO":
Principal "root@LAB2701.ASO" created.
kadmin:  listprincs
K/M@LAB2701.ASO
alumno@LAB2701.ASO
eduardo@LAB2701.ASO
```

```
host/pc2.lab2701.aso@LAB2701.ASO
host/servidor.lab2701.aso@LAB2701.ASO
kadmin/admin@LAB2701.ASO
kadmin/changepw@LAB2701.ASO
kadmin/servidor.lab2701.aso@LAB2701.ASO
krbtgt/LAB2701.ASO@LAB2701.ASO
root/admin@LAB2701.ASO
root@LAB2701.ASO
kadmin:  q
```