Kerberos

*Què és Kerberos, com funciona i quins avantatges té?:

When a server (or a PC's login system) needs to know whether to grant access to a resource (or to enter into the session) to a specific user, the server/PC simply makes a query from the LDAP database, and uses the results (username and password retrieved, group membership which determines the role, etc...) to make a decision. But in a modern domain controller, the LDAP system is complemented by Kerberos: instead of using LDAP both for authenticating and identifying users, Kerberos is used for the former task so LDAP remain only as a identity/authorization system. That's is: a client could be configured to use PAM to connect to Kerberos instead of LDAP but it should keep NSS configured to read LDAP information. So we can conclude that while LDAP stores all the information about you, Kerberos is responsible for telling services on the network who you are. Why this scheme? Because Kerberos gives a centralized user authentication system with two advantages:

-Passwords never aren't sent through the net. Kerberos uses a secure symmetric-key system for user authentication which doesn't use any PKI system like TLS but an specific protocol relaying in "tickets" shared between three points: the Kerberos server (also called KDC or "Key Distribution Center"), the third-party final servers (LDAP, SSH, NFS, IMAP etc) which must be previously configured to be "Kerberos-acknowledged", and the client (which must be previously configured too). More on this later

-The same user can log in many times into several third-party final servers (LDAP, SSH, NFS, IMAP, etc) with the same "remembered" credentials retrieved from the first time. That's called "Single Sign-On" (more on this later)

When a user logs in, they're actually having their credentials checked by Kerberos, which then issues them a "grainting ticket". From that point on, until the user logs out, whenever the user connects to some third-party server, this "granting ticket" will be used to retrieve from Kerberos server another specific "session ticket", only related to that connection; this "session ticket" will then be used by the third-party server to determine who is willing to connect. More specifically:

1.- Client logs on and receive an special "ticket" from KDC called "TGT" ("Ticket Granting Ticket")
2.- For each resource to be accessed, client first presents TGT to KDC to receive a "session ticket" called "TGS" ("Ticket Granting Service") related to that resource.
3.- Client presents the TGS to third-party final server at connection setup
4.- Third-party final server verifies TGS issued by KDC (without having to connect to it)

NOTA: Los KDC realmente están compuestos por un "Authentication Server" (encargado de repartir los TGT) y un "Ticket Granting Server" (encargado de repartir los TGS)

Let's take an example: when you log into your PC in the morning, if the Kerberos system is responsible for verifying your username and password, your PC will get what's called a special "ticket" called TGT that will be used to identify you in the future. This means that if, say, your email server (or your SSH server or your NFS server, etc) needs to verify you really are who you say you are before giving you access to your email/terminal/mount, all your PC needs to do is send the TGT to the KDC to get a TGS in return, which will be delivered to the email/ssh/nfs server so that server will know it's you. Without a Kerberos system, to enter into your email/ssh/nfs server you should send a username and password and then this final server should verify the correctness of this information on its own (maybe doing a request to a LDAP server). Or, in other (more detailed) words:

1.- You enter your username and password to log in your computer
2.- Your computer obtains from the KDC a TGT for that username and password, if they're valid. If they're not, then you're told you can't log in.
3.- Your computer now obtains information about you from LDAP and ensures your account is set up correctly. Finally, you get a desktop. Then, you double click on your email/terminal/folder icon.
4.- The email/ssh/nfs client first contacts the KDC with the (same) TGT and the name of the service to get a particular TGS, different for every final server.
5.- The email/ssh/nfs client sends its particular TGS to the email/ssh/nfs server.
6.- The email/ssh/nfs server sees that the TGS is for you, and verifies that the ticket was issued to your

computer and that it hasn't expired (all of this without contacting the Kerberos server: this entire information is in the TGS, together with a signature that proves that TGS hasn't been forged or tampered with). Then, your email client opens up and shows you the email on the mail server (or the ssh client enters into a remote terminal session or the nfs client mounts a remote folder, etc)

So:

-With Kerberos you can safely access your email/terminal/mount without you having to give your username and password anytime. Ideally, Kerberos is transparent to the user and eliminates the need to type passwords over and over again. Unforturately, this requires Kerberos support in all client and server software used, and proper configuration. Finding Kerberos patches for all of your software can be a daunting task.

-Sending login credentials and checking them directly against LDAP creates more load, and more opportunities to break the LDAP server. If the LDAP server is down for a minute, all network services will shut down as soon as they require any kind of authentication. By comparisons, the tickets issued by Kerberos can be checked without going back to any other servers. On the other hand, a KDC remains a single point of failure anyway, but most implementations support replication.

-Sending login credentials is inherently insecure. What if your PC sent your username and password to a computer masquerading as the final server, but run by someone trying to hack into your network? Kerberos's tickets don't contain any secure information (user passwords are never transmitted over the network, nor are they cached on the client machine), they expire after a certain period of time (by default 10h), and they can only be used for network services delivered to the computer that they were issued to (that is the "mutual authentication", i.e. both sides prove their identity to the other party, not just the user to the server).

El KDC solamente reconocerá como válidas (es decir, gestionables por él) las máquinas de la red que estén configuradas para pertenecer a su mismo "reino" (*realm*). Por comodidad el valor de este dato suele hacerse coincidir con el dominio DNS de la organización pero en mayúsculas. Por ejemplo, "EXAMPLE.COM". De esta manera, en la base de datos que maneja el KDC las definiciones de los usuarios (definiciones que se llaman genéricamente "principales" porque también pueden ser definiciones de equipos o de servicios que necesiten igualmente autenticarse contra Kerberos) aparecerán así: "usuario@EXAMPLE.COM". La sintaxis general de los nombres de los principales, de hecho, es simpre *nombre/instancia@REINO* donde la "instancia" es opcional.

> NOTA: Kerberos requires the clocks of the involved hosts to be synchronized; authentication will fail if they aren't (so it will be necessary some network time server like NTP to achieve that). Moreover, some piece of software (belonging to the same realm, moreover) must be installed on each client machine (and third-party final servers) to be recognized by the KDC. Finally, all hosts must have a name. This makes Kerberos a protocol especially designed for local network operation, having some difficulties to work properly through firewalls, over the Internet, or on mobile computers.

There's two main open-source Kerberos' software stand-alone implementations:

-MIT Kerberos (http://web.mit.edu/kerberos/www): It includes a KDC, libraries for client applications, and libraries for network service daemons. In Fedora can be installed by "krb5-server" (for the KDC) and "krb5-workstation" packages (for the clients and third-party final servers) and in Ubuntu by "krb5-kdc" and "krb5-user", respectively. However, on clients where user first locally logins, a PAM module should also be installed and configured to get the TGT, (this module could be the standalone one -called "pam_krb5", which is obsolete though- or the one shipped inside the SSSD framework)

-Heimdal Kerberos (http://www.h5l.org): It's an alternative software but API-compatible with MIT Kerberos implementation. In Fedora can be installed by "heimdal-server" (for the KDC) and "heimdal-workstation" packages (for the clients and third-party final servers) and in Ubuntu by "heimdal-kdc" and "heimdal-clients", respectively.

We will study the MIT's one on a Fedora system

*Configuració d'un servidor Kerberos (KDC):

Passos previs:

1.-Canviar el nom local de la màquina (l'anomenarem "kdc.midominio.local"). Això es pot fer editant directament l'arxiu "/etc/hostname" o bé, equivalentment, fent ús de la comanda sudo *hostnamectl set-hostname kdc.midominio.local*

2.-Afegir dins de l'arxiu "/etc/hosts" el nom "kdc.midominio.local" a la llista de noms equivalents a la IP 127.0.0.1

3.-Instal.lar un servidor NTP (com per exemple el paquet "chrony"). Per configurar-lo només cal assegurar-se d'afegir a l'arxiu "/etc/chrony.conf" (a Fedora) o "/etc/chrony/chrony.conf" (a Ubuntu) les següents línies: *"local stratum 8"* i *"allow X.X.X.X/Y"* (on "X.X.X.X/Y" representa la IP i màscara de la xarxa local -per exemple, 192.168.3.0/24- mantenint com estant les altres línies que hi puguin haver; per més informació, consultar *man chrony.conf*). Un cop fet aquest canvi, només caldrà posar en marxa el servei (*sudo systemctl enable chronyd && sudo systemctl start chronyd*):

NOTA: The "local" directive enables a local reference mode, which allows chronyd operating as an NTP server to appear synchronised to real time (from the viewpoint of clients polling it),even when it was never synchronised or the last update of the clock happened a long time ago. This directive is normally used in an isolated network, where computers are required to be synchronised to one another, but not necessarily to real time. The server can be kept vaguely in line with real time by manual input. This directive can have several options, among them there is the "stratum" one; this option sets the stratum (the "importance", the "relevance") of the server which will be reported to clients: the specified value is in the range 1 (most rellevant) through 15 (less rellevant); the default value is 10 but it should be larger than the maximum expected stratum in the network when external NTP servers are accessible. Stratum 1 indicates a computer that has a true real-time reference directly connected to it (e.g. GPS, atomic clock, etc.), such computers are expected to be very close to real time. Stratum 2 computers are those which have a stratum 1 server; stratum 3 computers have a stratum 2 server and so on. A value of 10 indicates that the clock is so many hops away from a reference clock that its time is fairly unreliable.

NOTA: The address in "allow" directive represents the network/subnet/host IP address from which the clients are allowed to connect to NTP server (the default is that no clients are allowed access). If this directive is used, chronyd will be both a client of its servers (specified by "pool" or "server" directives), and a server to other clients.

NOTA: There is a "manual" directive which enables support to modify the time of NTP server manually at run-time via the *chronyc settime "hh:mm:ss"* command. It could be useful if NTP server has no "pool"/"server" directives defined or functioning.

1.-Instal.lar els paquets necessaris

A Fedora: *sudo dnf install krb5-server krb5-workstation*
A Ubuntu: *sudo apt install krb5-kdc krb5-admin-server krb5-user*

2.-Executar les següents comandes per establir el regne i el domini DNS al que pertanyerà la màquina servidora:

sudo sed -i "s/example.com/midominio.local/g" /etc/krb5.conf
sudo sed -i "s/EXAMPLE.COM/MIDOMINIO.LOCAL/g" /etc/krb5.conf

La primera comanda canvia el regne de la màquina establert per defecte (el qual apareix escrit a diferents línies de l'arxiu */etc/krb5.conf* en majúscules; concretament és "EXAMPLE.COM") pel regne que desitjem (en el nostre cas, "MIDOMINIO.LOCAL"). La segona comanda canvia el domini DNS associat al regne anterior (el qual apareix escrit a diferents línies de l'arxiu en minúscules; concretament és "example.com") pel domini DNS adient (en el nostre cas, "midominio.local").

Executant les comandes anteriors el nom DNS "fully qualified" del servidor KDC, (que apareix configurat a la línia *kdc=* sota la secció *[realms]*) serà "kdc.midominio.local" (el qual ja quadra amb el valor que vam establir als "passos previs") però aquest es podria editar perfectament a mà per a què fos qualsevol altre (com ara "elmeuservidor.midominio.local" o "maquina1.midominio.local", etc) encara que, evidentment, llavors hauríem d'especificar el valor concret escollit en els "passos previs" indicats adalt.

El fitxer *krb5.conf* té un conjunt de directives relacionades amb la configuració de la màquina com a membre d'un regne, indistintament del seu paper dins d'aquest (servidor KDC, client, etc). Algunes de les directives més interessants que hi podem trobar són (per més informació, consultar *man krb5.conf*):

*Dins de la secció *[libdefaults]*: Apareixen múltiples línies, entre les quals podem destacar:

-Línia *default_realm=* : Indica el regne escollit de la màquina (entre els possibles regnes definits sota la secció *[realms]*

-Línia *ticket_lifetime=* : Indica la duració (en unitats temporals: dies -"...d"-, hores -"...h"-, minuts -"...m"- o segons "...s"-) dels eventuals TGTs que pugui rebre la màquina d'altres servidors Kerberos. Per defecte el seu valor és d'un dia.

*Dins de la secció *[realms]*: Apareix una subsecció (o més), cadascuna anomenades com un regne possible de la màquina (a escollir en la línia *default_realm=* indicada anteriorment). Cada subsecció conté, entre claus, les següents línies:

-Línia *kdc=* : Indica el nom DNS "fully qualified" del servidor KDC que hauran de fer servir els clients per connectar amb ell. Poden haver-hi vàries línies *kdc=*

-Línia *admin_server=* : Indica el nom DNS "fully qualified" del servidor d'administració que ens permetrà gestionar remotament el servidor KDC i la seva base de dades de principals (l'anomenat servei *kadmin/kpasswd*). Aquest servidor ve incorporat "de sèrie" amb el propi servidor KDC, així que, si es vol fer servir, normalment el valor d'aquesta línia serà el mateix que el de la línia *kdc=*

*Dins de la secció *[domain_realm]*: Apareixen dues línies: la que comença per un punt indica que totes les màquines remotes que tinguin aquest domini seran reconegudes com a pertanyents al regne indicat. L'altra indica que una eventual màquina remota anomenada només com el domini també pertanyeria al regne indicat. Es poden afegir altres noms FQDN que siguin fins i tot de dominis DNS diferents i associar-los al mateix regne.

*Dins de la secció *[logging]*: Apareixen vàries línies que indiquen els destins (normalment en forma de fitxers "*.log") dels diferents missatges generats pel subsistema Kerberos (genèrics, relacionades amb el servidor KDC pròpiament dit, relacionades amb el servidor d'administració del KDC...)

NOTA: For Ubuntu/Debian, the setup of the default realm for the KDC and KDC Admin hostnames is interactively performed during the KDC server install. You can re-run setup executing *dpkg-reconfigure krb5-kdc*. Therefore, this step is not needed for Ubuntu/Debian.

3.-Executar la següent comanda per configurar la màquina com a servidor KDC:

sudo sed -i "s/EXAMPLE.COM/MIDOMINIO.LOCAL/g" /var/kerberos/krb5kdc/kdc.conf

El fitxer *kdc.conf* té un conjunt de directives relacionades amb la configuració de la màquina com a servidor KDC. Algunes de les directives més interessants que hi podem trobar són (per més informació, consultar *man kdc.conf*):

*Dins de la secció *[kdcdefaults]*: Apareixen múltiples línies, entre les quals podem destacar:

-Línia *kdc_ports* = : Indica el port UDP per on escoltarà el servidor KDC (cal especificar-ho; normalment s'indica el 88). També existeix la línia *kdc_tcp_ports* = , que indica el port TCP per on escoltarà (per defecte Kerberos no escolta en cap)

*Dins de la secció *[realms]*: Apareix una subsecció (o més), cadascuna anomenades com un regne possible a gestionar pel KDC. Cada subsecció pot contenir, entre claus, línies com:

-Línia *acl_file=* : Indica la ubicació del fitxer que el servidor d'administració remota del KDC (el servei *kadmin/kpasswd*) utilitzarà per determinar quins principals tenen associats quins permisos sobre la base de dades Kerberos. Normalment aquest valor és "/var/kerberos/krb5kdc/kadm5.acl" (a Fedora) o "/etc/krb5kdc/kadm5.acl" (a Ubuntu) i no cal canviar-lo. Per més informació sobre aquest fitxer, consultar *man kadm5.acl*

4.-Executar la següent comanda per generar la base de dades de principals d'un determinat regne en el servidor KDC (per defecte es crearà en forma de fitxer binari -concretament, en format "BerkeleyDB"- anomenat "/var/kerberos/krb5kdc/principal"). S'ens demanarà una contrasenya: aquesta contrasenya protegeix aquesta base de dades i l'haurem de fer servir cada cop que volguem consultar/manipular la informació que hi conté, així que ¡no es pot oblidar mai! :

sudo kdb5_util create -r MIDOMINIO.LOCAL -s

NOTA: Si només hi ha un regne definit a la secció *[realms]* de l'arxiu "kdc.conf", s'agafarà aquest per defecte i no caldrà especificar llavors el paràmetre *-r*

NOTA: A més de la base de dades "principal" (i "principal.ok"), la comanda anterior també crea altres fitxers, com ara the Kerberos administrative database file, "principal.kadm5"; the administrative database lock file, "principal.kadm5.lock" or the access control list, kadm5.acl. The *–s* argument creates 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 prompts the user for the master server password (which can be used to regenerate the key) every time it starts.

NOTA: En Debian/Ubuntu la comanda anterior hauria de ser *krb5_newrealm*

NOTA: Per revertir aquest pas es pot fer *sudo kdb5_util destroy -f* o directament esborrar tots els arxius "principal*" de dins de la carpeta "/var/kerberos/krb5kdc"

5.-Iniciar el servei KDC:

A Fedora: *sudo systemctl start krb5kdc*
A Ubuntu: *sudo systemctl start krb5-kdc*

NOTA: També podríem posar en marxa opcionalment el servidor d'administració remota del KDC *kadmin/kpasswd* (per si volem gestionar el KDC des d'una altra màquina) amb les comandes *sudo systemctl start kadmin* (a Fedora) o *sudo systemctl start krb5-admin-server* (a Ubuntu).

NOTA: Atenció, cal obrir els ports 88 (UDP i/o TCP) per a fer accessible el servidor KDC pròpiament dit (i també els ports 749 UDP i TCP per fer accessible el servidor *kadmin/kpasswd*). Això a Fedora es pot fer amb les comandes: *sudo firewall-cmd --permanent –add-service=kerberos && sudo firewall-cmd --reload*

6.-Afegir els "principals" que necessitem. Començarem per afegir els usuaris que podran demanar TGTs en loguejar-se. Això es pot fer executant la comanda...:

sudo kadmin.local

...i dins del shell que s'obre, executar llavors la comanda interna següent: *addprinc pepito* (on "pepito" representa el nom d'usuari). Per sortir del shell intern només cal escriure *exit*. També ho podríem fer tot d'una tacada executant *sudo kadmin.local -q "addprinc pepito"* ; en qualsevol cas, la contrasenya associada al principal afegit es preguntarà interactivament.

NOTA: Una altra ordre interessant del shell intern de les comandes *kadmin.local/kadmin* és *listprincs*, la qual opcionalment pot admetre un paràmetre actuant com a filtre de recerca (el qual pot contenir comodins: *, ?, []). O *getprinc pepito*, per obtenir totes les dades relacionades amb el principal indicat. També existeix *delprinc pepito*. Una altra ordre es *getprivs*, per conèixer què pot fer l'usuari amb què s'ha accedit a la base de dades. Per conèixer totes les ordres possibles (entre les quals hi ha per establir polítiques de contrasenyes o gestionar arxius keytab -en parlarem més endavant-), consulteu *man kadmin.local*

NOTA: Cal saber que encara que el nom del principal serà en realitat "pepito@MIDOMINIO.LOCAL" (i així s'emmagatzemarà a la base de dades del KDC), com que el domini per defecte establert a l'arxiu "/etc/kdc.conf" és aquest, en general no caldrà especificar-ho

Kerberos principals can be created either on the KDC machine itself (with the *kadmin.local* command, as we have explained above) or through the network (with the *kadmin* command by means of an "admin" principal). By using *kadmin.local* command on the KDC machine you can create principals without needing to create a separate "admin" principal before you start because you can access directly to KDC database (using *sudo*). However, by using *kadmin* command you are connecting to the *kadmind* server, which perform database operations only through the "admin" principal. So you should create this special principal before you can use *kadmin* command from any remote machine (belonging to the realm). To do so you should first execute *sudo kadmin.local -q "addprinc manolito/admin"* (note the "/admin" tail...it's called the "role"; there's some predefined ones and "/admin" is one of them). Then you must also confirm in the KDC ACL that this specific "admin" principal has all permissions: to do this, you should open the "/var/kerberos/krb5kdc/kadm5.acl" file (in Fedora) or "/etc/krb5kdc/kadm5.acl" file (in Ubuntu) with a text editor and ensure that this file includes an entry so to allow the admin principal to administer the KDC for your specific realm, like this: *"*/admin@MIDOMINIO.LOCAL *"* After editing and saving the file (to know which values can be written into it, you can see *man kadm5.acl*), you must restart the *kadmin/krb5-admin-server* service. Since then, you will be able to administer Kerberos server from another remote machine (belonging to the same realm) simply executing *sudo kadmin -p manolito/admin*

*Configuració d'una màquina client Kerberos:

Passos previs:

1.-Establir el nom local de la màquina client per a què pertanyi al mateix domini que el servidor KDC (l'anomenarem "client.midominio.local"). Això es pot fer editant directament l'arxiu "/etc/hostname" o bé, equivalentment, fent ús de la comanda sudo *hostnamectl set-hostname client.midominio.local*

2.-Resoldre el nom "kdc.midominio.local" al servidor KDC. Això es pot aconseguir configurant un servidor DNS que faci aquesta feina (i establint-lo com a servidor DNS a usar a cada màquina client, ja sigui "a mà" o bé a través de DHCP) o bé editant directament l'arxiu "/etc/hosts" de cada client per a què hi aparegui una línia indicant l'associació entre aquest nom i la IP visible a la LAN del servidor KDC.

3.-Posar en marxa un client NTP (com per exemple el paquet "chrony"). Per configurar-lo només cal assegurar-se de comentar a l'arxiu "/etc/chrony.conf" (a Fedora) o "/etc/chrony/chrony.conf" (a Ubuntu) les línies que comencin per "*pool ...*" i, en canvi, afegir la línia *"server kdc.midominio.local iburst prefer"* (mantenint com estant les altres línies que hi puguin haver; per més informació, consultar *man chrony.conf*). Un cop fet aquest canvi, caldrà posar en marxa el servei (*sudo systemctl enable chronyd && sudo systemctl start chronyd*).

NOTA: The NTP clients needs to know which NTP servers it should contact to get the current time. We can specify the NTP servers in the "server" or "pool" directive in the NTP configuration file. The syntax of "pool" directive is similar to that for the "server" directive, except that it is used to specify a pool of NTP servers rather than a single NTP server; the pool name is expected to resolve to multiple addresses which might change over time. Both directives accept the same options; for instance, the "iburst" option is used to speed up the initial synchronisation; the "prefer" option is used to specify the preferred server/pool, if it were more than one listed in config file. The only option is different among these two directives is "maxsources" (only available in "pool"), which refers to the maximum number of NTP sources can be used from the pool

NOTA: Chrony has a command line utility named "chronyc" to control and monitor the chrony daemon (*chronyd*). For instance:
chronyc tracking : Checks if chrony is synchronized
chronyc sources -v : Verifies the current time sources that chrony uses
chronyc sourcestats : Finds the statistics of each sources, such as drift rate and offset estimation process
chronyc activity : Verifies the status of your NTP sources
chronyc offline : Notifies Chrony that the system is not connected to the Internet to avoid misunderstandings; to return to sinchronyze, *chronyc online*

1.-Instal.lar els paquets necessaris

A Fedora: *sudo dnf install krb5-workstation*
A Ubuntu: *sudo apt install krb5-user*

2.-Executar les següents comandes per establir el regne i el domini DNS al que pertanyerà la màquina client:

sudo sed -i "s/example.com/midominio.local/g" /etc/krb5.conf
sudo sed -i "s/EXAMPLE.COM/MIDOMINIO.LOCAL/g" /etc/krb5.conf

La primera comanda canvia el regne de la màquina establert per defecte, el qual apareix escrit a diferents línies de l'arxiu */etc/krb5.conf* en majúscules (concretament és "EXAMPLE.COM") pel regne que desitjem (en el nostre cas, "MIDOMINIO.LOCAL"). La segona comanda canvia el domini DNS associat al regne anterior (el qual apareix escrit a diferents línies de l'arxiu en minúscules, concretament és "example.com") pel domini DNS adient (en el nostre cas, "midominio.local").

Executant les comandes anteriors el nom DNS "fully qualified" del servidor KDC (que apareix configurat a la línia *kdc=* sota la secció *[realms]*) serà "kdc.midominio.local" (el qual ja quadra -o hauria de quadrar- amb el valor que vam establir tant als "passos previs" de la configuració del servidor com a la pròpia configuració del fitxer *krb5.conf* del servidor) .

3.-A partir d'aquí ja es pot demanar un TGT al KDC anterior. Per fer-ho manualment, només cal executar la següent comanda (es demanarà la contrasenya corresponent de forma interactiva):

kinit pepito

NOTA: Per saber tota la informació sobre el TGT (o TGTs si s'han demanat varis per diferents usuaris-principals) i els eventuals TGS (corresponents a l'eventual accés a servidors finals) que l'usuari que ha executat la comanda *kinit* ha obtingut fins ara i que encara són vàlids, es pot escriure la comanda *klist* . Per eliminar-los tots (i, per tant, desfer el SSO), es pot executar la comanda *kdestroy* o bé, per un de sol, *kdestroy -p pepito*

Per fer que el TGT sigui demanat automàticament per la màquina client a cada inici de sessió local (via *login/gdm,* agafant com a usuari-principal el nom de l'usuari i com a contrasenya-principal la contrasenya respectiva, escrits ambdós al quadre d'inici de sessió), cal configurar el framework SSSD per a què utilitzi el servidor KDC anterior com "authentication provider". Això implica editar convenientment l'arxiu "sssd.conf" i activar l'ús dels mòduls PAM de SSSD al sistema. Però com que per iniciar sessió al sistema no només és necessari un servei d'autenticació sinó també un servei d'identificació, a més de Kerberos necessitarem tenir un servidor LDAP com a font d´identificació (treballant coordinadament amb Kerberos). Així doncs, abans de poder fer la configuració de SSSD haurem d'implementar prèviament la "coordinació" mútua entre el servidor LDAP i el servidor Kerberos (mitjançant l'ús, tal com explicarem a continuació, de TGS basats en la presència dels fitxers "keytab" adients). Per tant, primer començarem per això últim.

*Fitxers "keytab":

Per a què poguem integrar un determinat servidor de tercers (LDAP, SSH, NFS, HTTP, POP/IMAP, etc) en la infrastructura Kerberos (o dit d'una altra manera, per a què poguem delegar l'autenticació de tots aquests serveis en el KDC i els TGS emesos per aquest), primer cal que afegim aquests servidors concrets com a principals dins de la base de dades del regne, a l'igual que abans hem fet amb els usuaris.

Els principals que representen serveis s'han d'anomenar d'una forma específica:

*Si són servidors LDAP: ***ldap****/nomFQDNdelServidor@MIDOMINIO.LOCAL*
*Si són servidors SSH o clients: ***host****/nomFQDNdelaMaquina@MIDOMINIO.LOCAL*

*Si són servidors NFS: ***nfs****/nomFQDNdelServidor@MIDOMINIO.LOCAL*
*Si són servidors HTTP: ***http****/nomFQDNdelServidor@MIDOMINIO.LOCAL*
*Si són servidors IMAP: ***imap****/nomFQDNdelServidor@MIDOMINIO.LOCAL*
*Si són servidors POP: ***pop****/nomFQDNdelServidor@MIDOMINIO.LOCAL*

Una gran diferència entre els principals que representen persones i els que representen serveis del regne és que, en el primer cas, la clau simètrica utilitzada per generar els TGT dels primers està protegida amb la contrasenya (interactiva) de l'usuari en qüestió però en el cas del serveis, si la clau simètrica utilitzada per generar els TGS estigués protegida amb contrasenya interactiva estaríem perdent la capacitat de SSO. Per això, la clau simètrica associada als serveis de tercers se sol generar de forma aleatòria al KDC guardant-se encriptada en un fitxer binari (anomenat genèricament fitxer "keytab"), que haurà de ser copiat a totes les màquines clients que vulguin fer servir aquest servei de tercers. És a dir, els fitxers "keytab" són emprats per processos/serveis/tasques programades que estan configurats per delegar l'autenticació en el sistema de TGS de Kerberos i necessiten que aquesta autenticació es realitzi sense cap intervenció interactiva de l'usuari (a diferència de quan es demana el TGT).

NOTA: El fitxer "keytab" per defecte és "/etc/krb5.keytab", encara que això es podria modificar mitjançant la directiva *default_keytab_name* del fitxer "/etc/krb5.conf". The KDC administration server *kadmind* is the only service that uses any other file (it uses "/var/kerberos/krb5kdc/kadm5.keytab")

NOTA: A keytab file is a binary file which contains one or more entries, where each entry consists of a timestamp (indicating when the entry was written to the keytab), a principal name, a key version number, an encryption type, and the encryption key itself.

Els passos a seguir per tal d'afegir un determinat servei (suposarem un servidor LDAP anomenat "miservidor.midominio.local") com a principal serien:

1.-Executar al KDC la següent comanda per afegir el principal que representa el servei de tercers...:

sudo kadmin.local -q "addprinc -randkey ldap/miservidor.midominio.local"

...i tots els clients que en faran ús (en aquest cas només en tindrem un):

sudo kadmin.local -q "addprinc -randkey host/client.midominio.local"

NOTA: Instead of setting a password for the new principal, the *-randkey* flag tells *kadmin* to generate a random key. This is used here because no user interaction is wanted for this principal.

2.-Executar al KDC la següent comanda per extreure de la base de dades de principals les claus aleatòries generades i associades als principals afegits en el pas anterior, i guardar-les en un fitxer "keytab" (el qual, si no s'especifica cap nom, serà "/etc/krb5.keytab" i només podrà ser llegit per "root")

sudo kadmin.local -q "ktadd ldap/miservidor.midominio.local"
sudo kadmin.local -q "ktadd host/client.midominio.local"

NOTA: Si es vol especificar la ruta i nom d'un altre fitxer "keytab" diferent del per defecte, només cal afegir el paràmetre *-k /ruta/fitxer.keytab* just després de la paraula *ktadd*

3.-Executar al KDC la següent comanda per afegir els principals que representen els possibles usuaris que voldrem fer servir per autenticar-nos en el moment de voler utilitzar el servei de tercers:

sudo kadmin.local -q "addprinc usu1ldap"

NOTA: El contenido del fichero "keytab" (que es binario) puede consultarse y modificarse interactivamente utilizando el comando *ktutil*. Por ejemplo, el primer subcomando que deberemos ejecutar casi siempre dentro de su shell propia es *rkt /ruta/fichero.keytab*, el cual sirve para seleccionar el fichero keytab sobre el cual se trabajará. A partir de aquí, se pueden ver todas las keys incluidas en el fichero mediante el subcomando *list* ; se puede añadir "a mano" una nueva key asociada a un determinado principal mediante el subcomando *addent -key*

-p ldap/miservidor.midominio.local -k nº , se puede eliminar una determinada key mediante el subcomando *delent nº* (o todas mediante el comando *clear*), etc Para más ejemplos, consultar *man ktutil*

NOTA: Un atacante que tuviese acceso a un fichero "keytab" podría autenticarse en la red con cualquiera de los principales que en el fichero hubiera, por lo que es realmente importante controlar el acceso a estos ficheros.

NOTA: A keytab can be displayed using the *sudo klist -kte [/ruta/fichero.keytab] [principal@REINO]* command (si no se indica ningún fichero "keytab" se asume el fichero por defecto. Si no se indica ningún principal, se asumen todos.

4.-The 389DS Server has a pre-defined "Kerberos UID" mapping rule to match a Kerberos principal name (with the form "user@EXAMPLE.COM") with the corresponding user inside the directory tree ("uid=user,dc=example,dc=com"). As you can see, the realm is used to define the search base, and the user ID (authid) defines the filter. This pre-defined mapping can be seen/edited inside the "dse.ldif" configuration file and specifically it's implemented in these following lines:

dn: cn=Kerberos uid mapping,cn=mapping,cn=sasl,cn=config
objectClass: top
objectClass: nsSaslMapping
cn: Kerberos uid mapping
nsSaslMapRegexString: \(.\)@\(.*\)\.\(.*\)*
nsSaslMapBaseDNTemplate: dc=\2,dc=\3
nsSaslMapFilterTemplate: (uid=\1)

Desgraciadament, tal i com es pot veure, el "mapping" per defecte no va bé pel directori que tenim al nostre servidor LDAP perquè el principal usu1ldap@MIDOMINIO.LOCAL el tradueix a "uid=usu1ldap,dc=midominio,dc=local" quan hauria de ser "uid=usu1ldap,ou=usuarios, dc=midominio,dc=local". Per sol.lucionar això sense haver d'alterar l'arbre de directoris podríem fer dues coses: o bé crear un arxiu .ldif amb el contingut mostrat a continuació i llavors integrar-lo a la configuració del servidor 389DS mitjançant l'execució de la comanda *ldapmodify* corresponent, o bé, directament modificar les línies anteriors per a què quedin igual que les mostrades a continuació (i a continuació reiniciar el servei dirsrv*@miservidor).*

dn: cn=Kerberos uid mapping,cn=mapping,cn=sasl,cn=config
objectClass: top
objectClass: nsSaslMapping
cn: Kerberos uid mapping
nsSaslMapRegexString: \(.\)@\(.*\)\.\(.*\)*
nsSaslMapBaseDNTemplate: ***ou=usuarios,****dc=\2,dc=\3*
nsSaslMapFilterTemplate: (uid=\1)

*Autenticació contra un servidor LDAP fent servir Kerberos:

Ara mateix, a la màquina client hauríem d'accedir a les dades del directori autenticant-nos directament amb algun dels usuaris presents al propi directori (a més de cn=admin). És a dir, una comanda com la següent ens hauria de retornar les dades de tots els usuaris existents al directori, autenticant-nos precisament amb d'ells, l'usuari "usu1ldap":

LDAPTLS_REQCERT=never ldapsearch -H ldaps://miservidor.midominio.local -LLL -b "dc=midominio,dc=local" -D "uid=usu1ldap,ou=usuarios,dc=midominoi,dc=local" -W

El nostre objectiu ara serà realitzar la mateixa acció però autenticant-nos mitjançant un principal emmagatzemat a Kerberos. Per aconseguir-ho hem de fer els passos següents:

1.-Copiar el fitxer "keytab" generat a l'apartat anterior a la carpeta "/etc/dirsrv/slapd-miservidor" del servidor 389DS (amb el nou nom de, per exemple, "elmeu.keytab"):

sudo cp /etc/krb5.keytab /etc/dirsrv/slapd-miservidor/elmeu.keytab

NOTA: Si el KDC i el servidor LDAP estiguessin en màquines diferents, aquesta còpia s'hauria de realitzar mitjançant algun mètode segur com ara *scp* o similar

2.-Escriure la línia *KRB5_KTNAME=/etc/dirsrv/slapd-miservidor/elmeu.keytab* en el fitxer "/etc/sysconfig/dirsrv-*miservidor*" del servidor LDAP (que segurament no existirà; l'hauràs de crear) i reiniciar el servidor 389DS

NOTA: El fitxer anterior s'utilitza per configurar variables d'entorn útils pel servei dirsrv@miservidor , tal com es pot veure executant *systemctl cat dirsrv@miservidor*

3.-Copiar el fitxer "keytab" generat a l'apartat anterior (assumirem que té el nom estàndar) a la carpeta "/etc" de la màquina client. Aquest pas caldria fer-lo de forma segura mitjançant *scp* o similar

NOTA: Si ja existís un fitxer "/etc/krb5.keytab" al client, es pot utilitzar la comanda *ktutil* per unit convenientment el contingut del fitxer preexistent amb el del nou

4.-A la màquina client, demanar el TGT pel principal usuari que volem que faci el SSO...:

kinit usu1ldap

...i a partir d'aquí, un cop obtingut el seu TGT, l'usuari en qüestió ja podrà realitzar totes les consultes que es desitgi al servidor LDAP autenticat automàticament. Ho podem provar si executem:

LDAPTLS_REQCERT=never ldapsearch -H ldaps://miservidor.midominio.local -LLL -b "dc=midominio,dc=local"

NOTA: Para comprobar que efectivament la consulta se ha hecho con el usuario que adquierió el TGT (usu1ldap) se puede ejecutar *LDAPTLS_REQCERT=never ldapwhoami -H ldaps://miservidor.midominio.local*

NOTA: Note that in Kerberos, authentication is always mutual. This means that not only have you authenticated yourself to the LDAP server, but also the LDAP server has authenticated itself to you. In particular, this means communication is with the desired LDAP server, rather than some bogus service set up by an attacker.

*Configuració de l'inici de sessió fent servir Kerberos/LDAP:

Un cop ja hem comprovat que la "coordinació" entre el servidor Kerberos i el servidor LDAP funciona si es demana des de la nostra màquina client, ara només caldrà indicar al servidor SSSD d'aquesta darrera que faci ús d'aquells dos servidors que ja treballen plegats. Concretament haurem de modificar el fitxer de configuració "/etc/sssd/sssd.conf" per a què tingui un contingut similar al següent (i seguidament reiniciar el servei SSSD):

```
[sssd]
domains=pepito
[domain/pepito]
auth_provider=krb5
krb5_server=kdc.midominio.local
krb5_realm=MIDOMINIO.LOCAL
id_provider=ldap
ldap_uri=ldaps://miservidor.midominio.local
ldap_search_base=dc=midominio,dc=local
ldap_tls_reqcert=allow
ldap_id_use_start_tls = true
ldap_sasl_mech = gss-spnego  #Indica la llibreria que usarà el client per enviar el TGS al LDAP
ldap_sasl_authid = host/client.midominio.local@MIDOMINIO.LOCAL
```

I ja està: a partir d'aquí, un cop l'usuari inicïi sessió (via gdm, *su -l usu1ldap*, etc) escrivint la seva contrasenya, obtindrà automàticament un TGT que farà servir tot seguit per demanar al KDC un TGS per fer-lo servir contra el servidor LDAP (concretament, per realitzar una consulta autenticada mitjançant la llibreria GSS-SPNEGO, que forma part del "framework" de seguretat SASL), obtenint així (sense haver-se d'autenticar de nou) la resta de dades identificatives que necessita per iniciar sessió.