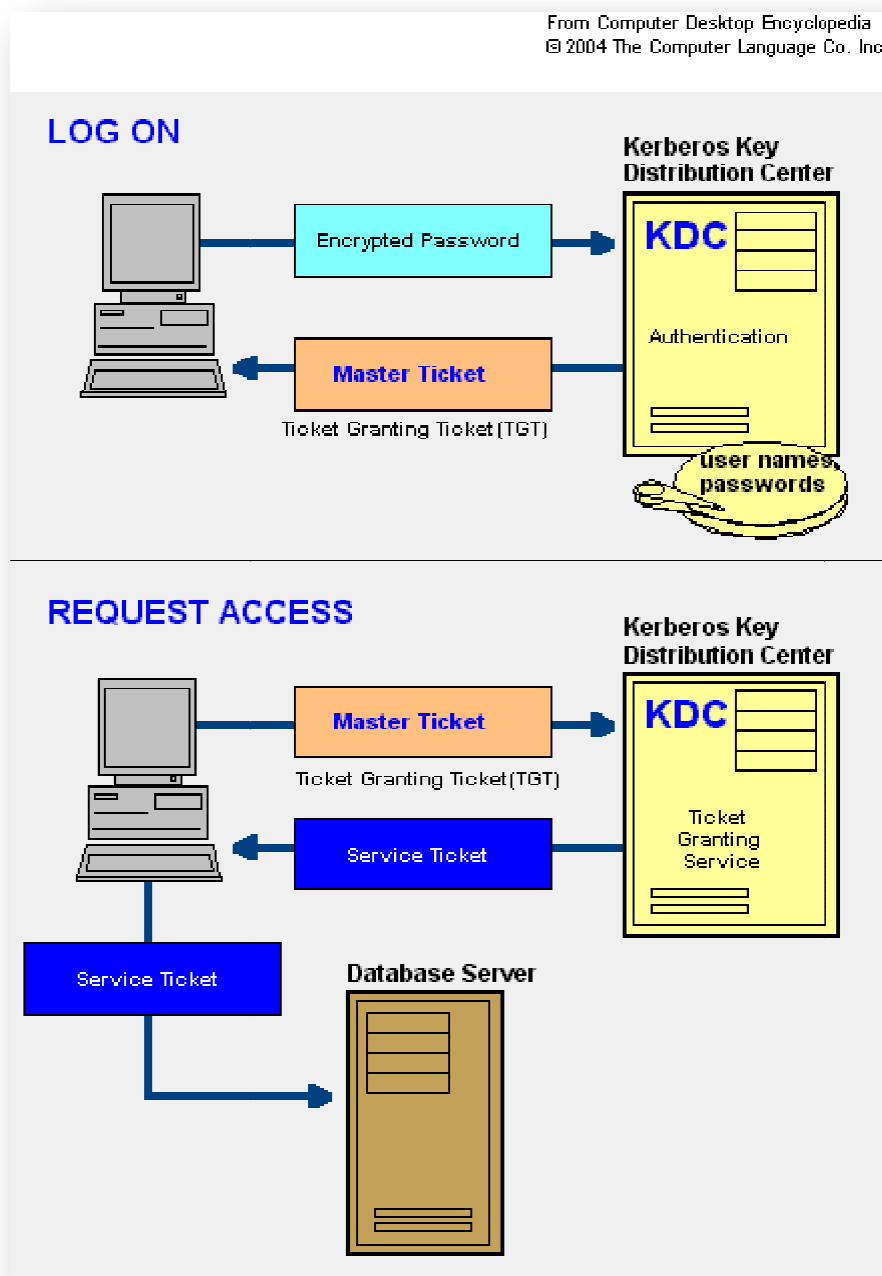


PROTOCOLLO KERBEROS

Il protocollo Kerberos può essere pensato come l'aggregato di tre componenti fondamentali:

1. Applicazione *Client*
2. Risorse di rete a cui il client tenta di accedere
3. Key Distribution Center (KDC)

Il KDC è eseguito su tutti i server e funge da intermediario tra il client e le risorse (server).



3.1 Aspetti fondamentali del sistema

Sulla base di quanto riportato nei precedenti capitoli, è possibile individuare una descrizione essenziale delle entità che costituiscono il sistema Kerberos. A tal proposito riportiamo le seguenti definizioni che verranno poi incontrate nel corso delle spiegazioni implementative:

- **Realm** = Definisce il concetto di una rete basata su Kerberos, formata da uno o più server e da un insieme più o meno grande di client.
- **Client** = Entità della rete (utente / host / applicazione) che riceve un ticket da Kerberos.
- **Principal** = Nome unico di un utente o un servizio abilitato ad autenticare usando Kerberos.
- **Ticket (Credenziali)** = Serie di dati elettronici che identificano temporaneamente l'identità di un client per un particolare servizio.
- **Credential cache (File del ticket)** = File che contiene le chiavi per la comunicazione cifrata fra un utente e vari servizi di rete.
- **Keytab (Tabella delle chiavi)** = File che contiene un elenco non cifrato dei principal e delle chiavi; i server recuperano le chiavi di cui necessitano dai file keytab.
- **KDC (Key distribution center)** = Servizio che emette i ticket Kerberos.
- **AS (Authentication server)** = Server che emette i ticket per un servizio (risponde alle richieste dei client) → Viene generalmente usato per ottenere accesso al servizio Ticket-Granting Server (TGS), emettendo un Ticket-Granting Ticket (TGT). AS e KDC generalmente vengono eseguiti sullo stesso host.
- **TGS (Ticket-granting service)** = emette i ticket per un servizio desiderato, i quali vengono dati a turno agli utenti per l'accesso al servizio.
- **TGT (Ticket-granting ticket)** = Ticket speciale che permette al client di ottenere ulteriori ticket senza richiederli al KDC.

3.2 Installazione e configurazione

L'installazione di Kerberos si divide in specifici passi che riguardano la configurazione del *master* e del *client* della rete. Verranno di seguito riportate le istruzioni per la configurazione del sistema di autenticazione in ambiente Linux.

In particolare la distribuzione Linux in cui sono stati effettuati i test è la *Ubuntu Feisty 7.04*.

FASE 1 – INSTALLAZIONE DEI PACCHETTI

Prima di tutto occorrerà installare i pacchetti relativi al protocollo Kerberos in modo da predisporre il sistema ad essere configurato per verificare il corretto funzionamento del processo di autenticazione.

Si svilupperanno i test sulla versione 5 del protocollo, in tal caso i principali pacchetti che è possibile installare sono:

krb5-admin-server → MIT Kerberos Master Server (kadmind)

krb5-clients → Predisposizione per l'uso dei servizi ftp, telnet e rsh con il MIT Kerberos

krb5-config → File di configurazione per il Kerberos Version 5







krb5-doc → Documentazione

krb5-kdc → MIT Kerberos Key Server (KDC)

krb5-user → Programmi base per l'autenticazione utilizzando il MIT Kerberos

Altri pacchetti, possibilmente installabili, che sfruttano ed integrano le caratteristiche del protocollo sono: *krb5-auth-dialog*, *krb5-ftpd*, *krb5-telnetd*, *krb5-rsh-server*, *libkrb53*, *libpam-krb5*.

GESTORE PACCHETTI SYNAPTIC PER UBUNTU

Gestore di pacchetti Synaptic				
 				
S	Pacchetto	Versione installata	Ultima versione	Descrizione
<input checked="" type="checkbox"/>	krb5-admin-server	1.4.3-9ubuntu1.1	1.4.3-9ubuntu1.1	MIT Kerberos master server (kadmind)
<input checked="" type="checkbox"/>	krb5-auth-dialog	0.6-0ubuntu1	0.6-0ubuntu1	dialog for reauthenticating kerberos tickets
<input checked="" type="checkbox"/>	krb5-clients	1.4.3-9ubuntu1.1	1.4.3-9ubuntu1.1	Secure replacements for ftp, telnet and rsh using MIT Kerberos
<input checked="" type="checkbox"/>	krb5-config	1.10	1.10	Configuration files for Kerberos Version 5
<input checked="" type="checkbox"/>	 krb5-doc	1.4.3-9ubuntu1.1	1.4.3-9ubuntu1.1	Documentation for MIT Kerberos
<input checked="" type="checkbox"/>	krb5-ftpd	1.4.3-9ubuntu1.1	1.4.3-9ubuntu1.1	Secure FTP server supporting MIT Kerberos
<input checked="" type="checkbox"/>	krb5-kdc	1.4.3-9ubuntu1.1	1.4.3-9ubuntu1.1	MIT Kerberos key server (KDC)
<input checked="" type="checkbox"/>	krb5-rsh-server	1.4.3-9ubuntu1.1	1.4.3-9ubuntu1.1	Secure replacements for rshd and rlogind using MIT Kerberos
<input checked="" type="checkbox"/>	krb5-telnetd	1.4.3-9ubuntu1.1	1.4.3-9ubuntu1.1	Secure telnet server supporting MIT Kerberos
<input checked="" type="checkbox"/>	krb5-user	1.4.3-9ubuntu1.1	1.4.3-9ubuntu1.1	Basic programs to authenticate using MIT Kerberos
<input checked="" type="checkbox"/>	libauthen-krb5-perl	1.5-2	1.5-2	Perl extension for Kerberos 5 API
<input type="checkbox"/>	libkrb5-17-heimdal		0.7.2.dfsg.1-3ubuntu1	Libraries for Heimdal Kerberos
<input checked="" type="checkbox"/>	 libkrb53	1.4.3-9ubuntu1.1	1.4.3-9ubuntu1.1	MIT Kerberos runtime libraries
<input type="checkbox"/>	 libkrb5-dbg		1.4.3-9ubuntu1.1	Debugging files for MIT Kerberos
<input type="checkbox"/>	 libkrb5-dev		1.4.3-9ubuntu1.1	Headers and development libraries for MIT Kerberos
<input checked="" type="checkbox"/>	libpam-krb5	2.4-1	2.4-1	PAM module for MIT Kerberos
<input type="checkbox"/>	openafs-krb5		1.4.1-4	AFS distributed filesystem Kerberos 5 integration

FASE 2 – CONFIGURAZIONE DEL SERVER

Una volta installati i pacchetti è necessario affrontare lo stadio di configurazione dei principali file che definiscono le proprietà del sistema. I file di configurazione su cui è necessario agire sono:

- */etc/krb5.conf*
- */etc/krb5kdc/kdc.conf*

Il file *krb5.conf* contiene le informazioni di configurazione necessarie al funzionamento del server, queste includono il *realm di default* e la posizione del *Key Distribution Center (KDC)* all'interno dei realms conosciuti.

Il file *kdc.conf* specifica i dati di configurazione per il realm, usati dal servizio di autenticazione Kerberos (AS) e dal KDC.

In tali file è possibile definire un proprio dominio di default cui far riferimento per l'accesso o l'eventuale gestione del servizio di autenticazione; nel caso esaminato si è fatto riferimento ad un nuovo dominio denominato "EPILUKE.IT".

Di seguito sono riportati i file di configurazione opportunamente settati:

```
[libdefaults]
    default_realm = EPILUKE.IT

# The following krb5.conf variables are only for MIT Kerberos.
    krb4_config = /etc/krb.conf
    krb4_realms = /etc/krb.realms
    kdc_timesync = 1
    ccache_type = 4
    forwardable = true
    proxiable = true

# The following encryption type specification will be used by MIT Kerberos
# if uncommented. In general, the defaults in the MIT Kerberos code are
# correct and overriding these specifications only serves to disable new
# encryption types as they are added, creating interoperability problems.
# default_tgs_etypes = aes256-cts arcfour-hmac-md5 des3-hmac-sha1 des-cbc-crc des-cbc-md5
# default_tkt_etypes = aes256-cts arcfour-hmac-md5 des3-hmac-sha1 des-cbc-crc des-cbc-md5
# permitted_etypes = aes256-cts arcfour-hmac-md5 des3-hmac-sha1 des-cbc-crc des-cbc-md5
```

```
# The following libdefaults parameters are only for Heimdal Kerberos.
```

```
v4_instance_resolve = false
v4_name_convert = {
    host = {
        rcmd = host
        ftp = ftp
    }
    plain = {
        something = something-else
    }
}
fcc-mit-ticketflags = true
```

```
[realms]
```

```
EPILUKE.IT = {
    kdc = lukesky.epiluke.it:88
    admin_server = lukesky.epiluke.it:749
}
.
.
.
```

```
[domain_realm]
```

```
.epiluke.it = EPILUKE.IT
epiluke.it = EPILUKE.IT
.
.
.
```

```
[logging]
```

```
kdc = FILE:/var/log/krb5kdc/krb5kdc.log
admin_server = FILE:/var/log/krb5kdc/kadmin.log
default = FILE:/var/log/krb5kdc/krb5lib.log
```

```
[login]
```

```
krb4_convert = true
krb4_get_tickets = false
```

krb5.conf

```

[kdcdefaults]
    kdc_ports = 750,88

[realms]
    EPILUKE.IT = {
        database_name = /var/lib/krb5kdc/principal
        admin_keytab = FILE:/etc/krb5kdc/kadm5.keytab
        acl_file = /etc/krb5kdc/kadm5.acl
        key_stash_file = /etc/krb5kdc/stash
        kadmin_port = 749
        max_life = 10h 0m 0s
        max_renewable_life = 7d 0h 0m 0s
        master_key_type = des3-hmac-sha1
        supported_encetypes = des3-hmac-sha1:normal des-cbc-crc:normal des:normal des:v4
        des:norealm des:onlyrealm des:afs3
        default_principal_flags = +preauth
    }

```

kdc.conf

Questi file riflettono il nome del realm e le mappature del dominio al realm. Per convenzione, tutti i nomi di realm sono scritti in lettere maiuscole mentre tutti gli hostname DNS e i nomi di dominio sono in formato minuscolo.

Come è facilmente intuibile dai file sopra riportati il KDC è identificato dall'URL *lukesky.epiluke.it* in corrispondenza della porta 88, mentre l'admin server è in corrispondenza della porta 749.

Una possibile configurazione del file */etc/hosts* che permette di risolvere lo specifico **indirizzo IP** relativo all'*hostname* di riferimento è la seguente:

```

127.0.1.1 laptop
169.254.6.202 lukesky.epiluke.it
127.0.0.1 localhost localhost.localdomain

```

/etc/hosts

```

lukesky.epiluke.it

```

/etc/hostname

Nel file *krb5.conf* è visibile una sezione denominata *logging* in cui sono predisposti dei file di log che consentono di avere a disposizione una **registrazione cronologica** delle operazioni che man

mano che vengono eseguite dal sistema; come si può vedere i file di log definiti sono tre, di cui vengono di seguito riportati alcuni semplici esempi di messaggi registrati:

```
Feb 15 15:43:05 lukesky.epiluke.it krb5kdc[6372](info): AS_REQ (7 etypes {18 17 16 23 1 3 2}) 169.254.137.61:
NEEDED_PREAUTH: lukesky@EPILUKE.IT for krbtgt/EPILUKE.IT@EPILUKE.IT, Additional pre-authentication required
Feb 15 15:43:05 lukesky.epiluke.it krb5kdc[6372](info): AS_REQ (7 etypes {18 17 16 23 1 3 2}) 169.254.137.61:
ISSUE: authtime 1171550585, etypes {rep=16 tkt=16 ses=16}, lukesky@EPILUKE.IT for
krbtgt/EPILUKE.IT@EPILUKE.IT
Feb 15 15:43:49 lukesky.epiluke.it krb5kdc[6372](info): TGS_REQ (1 etypes {1}) 169.254.137.61: ISSUE:
authtime 1171550046, etypes {rep=16 tkt=16 ses=1}, lukesky@EPILUKE.IT for
host/lukesky.epiluke.it@EPILUKE.IT
krb5kdc: Interrupted system call - while selecting for network input(1)
Feb 15 15:44:59 lukesky.epiluke.it krb5kdc[6372](info): shutting down
Feb 18 15:32:59 lukesky.epiluke.it krb5kdc[4062](info): setting up network...
Feb 18 15:32:59 lukesky.epiluke.it krb5kdc[4062](info): skipping unrecognized local address family 17
Feb 18 15:32:59 lukesky.epiluke.it krb5kdc[4062](info): skipping unrecognized local address family 17
Feb 18 15:32:59 lukesky.epiluke.it krb5kdc[4062](info): set up 0 sockets
krb5kdc: no sockets set up?
Feb 18 16:04:50 lukesky.epiluke.it krb5kdc[6681](info): setting up network...
Feb 18 16:04:50 lukesky.epiluke.it krb5kdc[6681](info): skipping unrecognized local address family 17
Feb 18 16:04:50 lukesky.epiluke.it krb5kdc[6681](info): skipping unrecognized local address family 17
Feb 18 16:04:50 lukesky.epiluke.it krb5kdc[6681](info): listening on fd 7: udp 192.168.2.3.88
Feb 18 16:04:50 lukesky.epiluke.it krb5kdc[6681](info): listening on fd 8: udp 192.168.2.3.750
```

krb5kdc.log

```
Feb 15 15:30:55 lukesky.epiluke.it kadmind[6307](debug): Got signal to request exit
Feb 15 15:30:55 lukesky.epiluke.it kadmind[6307](info): finished, exiting
Feb 15 15:30:56 lukesky.epiluke.it kadmind[6380](info): No dictionary file specified, continuing without one.
Feb 15 15:30:56 lukesky.epiluke.it kadmind[6381](info): Seeding random number generator
Feb 15 15:31:03 lukesky.epiluke.it kadmind[6381](info): starting
Feb 15 15:36:33 lukesky.epiluke.it kadmind[6381](Notice): Request: kadm5_init, krbadm/admin@EPILUKE.IT,
success, client=krbadm/admin@EPILUKE.IT, service=kadmin/admin@EPILUKE.IT, addr=169.254.137.61, flavor=6
Feb 15 15:37:10 lukesky.epiluke.it kadmind[6381](Notice): Request: kadm5_randkey_principal,
lukesky@EPILUKE.IT, success, client=krbadm/admin@EPILUKE.IT, service=kadmin/admin@EPILUKE.IT,
addr=169.254.137.61
Feb 15 15:37:10 lukesky.epiluke.it kadmind[6381](Notice): Request: kadm5_get_principal, lukesky@EPILUKE.IT,
success, client=krbadm/admin@EPILUKE.IT, service=kadmin/admin@EPILUKE.IT, addr=169.254.137.61
Feb 15 15:40:39 lukesky.epiluke.it kadmind[6381](Notice): Request: kadm5_init, krbadm/admin@EPILUKE.IT,
success, client=krbadm/admin@EPILUKE.IT, service=kadmin/admin@EPILUKE.IT, addr=169.254.137.61, flavor=6
```

kadmin.log

```
Feb 18 17:15:53 lukesky.epiluke.it krb524d[4065](info): No dictionary file specified, continuing without one.
krb524d: service entry `krb524' not found, using 4444
```

krb5lib.log

FASE 3 – CREAZIONE DEL DATABASE

Una volta messi a punto i settaggi di base è possibile creare il *database* e lo *stash file*, ossia il file che contiene una copia locale della chiave principale del database. Lo *stash file* è usato dal KDC per autenticare se stesso prima che i demoni `kadmind` e `krb5kdc` siano partiti, ed è molto importante quindi che sia protetto accuratamente.

Eseguire quindi dalla shell di Linux il seguente comando:

```
$ /usr/sbin/kdb5_util create -r EPILUKE.IT -s
```

attraverso questa utility si crea il database che sarà utilizzato per conservare le chiavi per il realm Kerberos. La specifica `-s` impone la creazione di un file *stash* che conserverà la chiave del server master → se non esiste alcun file *stash* da cui leggere la chiave, ad ogni avvio il server Kerberos (`krb5kdc`) richiederà all'utente la password del server master (utilizzabile per rigenerare la chiave).

FASE 4 – CREAZIONE ACCESS CONTROL LIST

A questo punto si crea un ACL di default nel file `/etc/krb5kdc/kadm5.acl` in modo che ogni principal senza istanza di admin può comunque fare richieste al database. Il `kadm5.acl` è quindi utilizzato da `kadmind` per stabilire quali principal hanno accesso amministrativo al database di Kerberos nonché al loro livello di accesso.

Il file ACL in tal caso definito è il seguente:

```
*/admin@EPILUKE.IT      *
*/*@EPILUKE.IT         i
```

kadm5.acl

Con questa configurazione gli utenti che hanno un secondo principal come ad esempio *admin* (per esempio `nomeutente/admin@EPILUKE.IT`) avranno pieni poteri sul database del realm di Kerberos.

FASE 5 – AGGIUNTA DEI PRINCIPAL E RELATIVE POLICY

La flessibilità dei file di configurazione permette di creare delle policy diverse per amministratori e utenti in modo da forzare durata e robustezza delle password per i due ruoli.

Si aggiungono quindi l'amministratore di sistema e un utente non privilegiato usufruendo del comando *kadmin.local* :

```
$ sudo kadmin.local
>: addpol -maxlife "180 days" -minlength 8 -minclasses 3 -history 3 user
>: addpol -maxlife "90 days" -minlength 10 -minclasses 3 -history 6 admin
>: addprinc -policy admin +requires_preauth krbadm/admin
>: addprinc -policy user lukesky
>: ktadd -k /etc/krb5kdc/kadm5.keytab kadmin/admin kadmin/changepw
```

attraverso quest'ultima istruzione si aggiunge la keytab di *kadmind*, cioè la chiave che i demoni di amministrazione *kadmind4* e *v5passwd* usano per decifrare i ticket Kerberos di amministratori o client per determinare se questi hanno o no accesso al database.

krbadm/admin → *amministratore di sistema*

lukesky → *utente generico*

L'utility *kadmin* comunica con il server *kadmind* attraverso la rete e utilizza Kerberos per gestire l'autenticazione. Per questa ragione, il primo principal deve essere già esistente, prima di potersi connettere al server attraverso la rete per amministrarlo. Il primo principal è stato creato infatti utilizzando il comando *kadmin.local*, il quale è concepito proprio per essere utilizzato sullo stesso host di KDC e non si serve di Kerberos per l'autenticazione.

Una volta che *kadmind* viene avviato sul server, qualunque utente è in grado di accedere ai suoi servizi sulla base delle regole di accesso precedentemente definite.

FASE 6 – VERIFICA FUNZIONAMENTO DEL SISTEMA

A questo punto il sistema è operativo e ciò è verificabile eseguendo specifici comandi:

- *kinit* → comando che permette l'acquisizione delle credenziali. Attraverso questo comando è possibile verificare che il server stia emettendo i ticket, eseguendo *kinit* si ottiene un ticket che successivamente verrà immagazzinato in un credential cache (o file dei ticket). I ticket acquisiti sono verificabili utilizzando l'istruzione *klist* (per effettuare il test digitare da shell *kinit -p krbadm/admin*).

Attraverso il comando *kinit* quindi un principal può ottenere e depositare il Ticket-Granting Ticket (TGT) iniziale.

- *kadmin* → comando per verificare il corretto funzionamento del server (per effettuare il test digitare da shell *kadmin -p krbadm/admin*).

Se l'esecuzione di questi comandi da esito positivo allora il server è configurato correttamente ed è quindi possibile passare all'impostazione del client per poter sperimentar effettivamente il regolare funzionamento del protocollo.

RISORSE DISPONIBILI – DOCUMENTAZIONI:

Per maggiori informazioni sul Kerberos è possibile, in primo luogo, consultare la documentazione fornita; inoltre si può far riferimento alle *pagine man* relative ai comandi eseguibili, valutando ed esaminando le molteplici opzioni di configurazione messe a disposizione.

Di seguito sono riportati esempi di *pagine man* associate ai comandi precedentemente citati:

KINIT(1)

NAME

kinit - obtain and cache Kerberos ticket-granting ticket

SYNOPSIS

```
kinit [-5] [-4] [-V] [-l lifetime] [-s start_time] [-r renewable_life]
      [-p | -P] [-f | -F] [-a] [-A] [-v] [-R] [-k [-t keytab_file]]
      [-c cache_name] [-S service_name] [principal]
```

DESCRIPTION

kinit obtains and caches an initial ticket-granting ticket for principal. The typical default behavior is to acquire only Kerberos 5 tickets. However, if kinit was built with both Kerberos 4 support and with the default behavior of acquiring both types of tickets, it will try to acquire both Kerberos 5 and Kerberos 4 by default. Any documentation particular to Kerberos 4 does not apply if Kerberos 4 support was not built into kinit.

OPTIONS

- 5 get Kerberos 5 tickets
- 4 get Kerberos 4 tickets.
- V display verbose output.
- l lifetime
- s start_time
- r renewable_life
- f request forwardable tickets. (Not applicable to Kerberos 4.)
- F do not request forwardable tickets. (Not applicable to Kerberos 4.)
- p request proxiable tickets. (Not applicable to Kerberos 4.)
- P do not request proxiable tickets. (Not applicable to Kerberos 4.)

- a request tickets with the local address[es]. (Not applicable to Kerberos 4.)
- A request address-less tickets. (Not applicable to Kerberos 4.)
- v requests that the ticket granting ticket in the cache (with the invalid flag set) be passed to the kdc for validation. (Not applicable to Kerberos 4.)
- R requests renewal of the ticket-granting ticket.
- k [-t keytab_file] requests a host ticket, obtained from a key in the local host's keytab file.
- c cache_name use cache_name as the Kerberos 5 credentials cache name and location.
- S service_name specify an alternate service name to use when getting initial tickets.

ENVIRONMENT

Kinit uses the following environment variables:

KRB5CCNAME Location of the Kerberos 5 credentials (ticket) cache.

KRBTKFILE Filename of the Kerberos 4 credentials (ticket) cache.

FILES

/tmp/krb5cc_[uid] default location of Kerberos 5 credentials cache ([uid] is the decimal UID of the user).

/tmp/tkt[uid] default location of Kerberos 4 credentials cache ([uid] is the decimal UID of the user).

/etc/krb5.keytab default location for the local host's keytab file.

SEE ALSO

klist(1), kdestroy(1), kerberos(1)

man kinit

KADMIN(8)

NAME

kadmin - Kerberos V5 database administration program

SYNOPSIS

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

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

DESCRIPTION

kadmin and kadmin.local are command-line interfaces to the Kerberos V5 KADM5 administration system. Both kadmin and kadmin.local provide identical functionalities; the difference is that kadmin.local runs on the master KDC and does not use Kerberos to authenticate to the database. Except as explicitly noted otherwise, this man page will use kadmin to refer to both versions. kadmin provides for the maintenance of Kerberos principals, KADM5 policies, and service key tables (keytabs).

The remote version uses Kerberos authentication and an encrypted RPC, to operate securely from anywhere on the network. It authenticates to the KADM5 server using the service principal kadmin/admin. If the credentials cache contains a ticket for the kadmin/admin principal, and the -c credentials_cache option is specified, that ticket is used to authenticate to KADM5. 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 kadmin/admin Kerberos service ticket from the KDC, and uses that service ticket to authenticate to KADM5.

The local client kadmin.local, is intended to run directly on the master KDC without Kerberos authentication. The local version provides all of the functionality of the now obsolete kdb5_edit(8), except for database dump and load, which is now provided by the kdb5_util(8) utility.

OPTIONS

- r realm Use realm as the default database realm.
- p principal Use principal to authenticate.
- k Use a keytab to decrypt the KDC response.
- t keytab Use keytab to decrypt the KDC response.
- c credentials_cache Use credentials_cache as the credentials cache.

- w password Use password instead of prompting for one on the TTY.

- q query pass query directly to kadmin, which will perform query and then exit.

- d dbname Specifies the name of the Kerberos database.

- s admin_server[:port] Specifies the admin server which kadmin should contact.

- m Do not authenticate using a keytab.

- e enc:salt_list Sets the list of encryption types and salt types to be used for any new keys created.

- O Force use of old AUTH_GSSAPI authentication flavor.

- N Prevent fallback to AUTH_GSSAPI authentication flavor.

DATE FORMAT

Various commands in kadmin can take a variety of date formats, specifying durations or absolute times. Examples of valid formats are:

- 1 month ago
- 2 hours ago
- 400000 seconds ago
- last year
- this Monday
- next Monday
- yesterday
- tomorrow
- now
- second Monday
- a fortnight ago
- 3/31/92 10:00:07 PST
- January 23, 1987 10:05pm
- 22:00 GMT

Dates which do not have the "ago" specifier default to being absolute dates, unless they appear in a field where a duration is expected. In that case the time specifier will be interpreted as relative. Specifying "ago" in a duration may result in unexpected behavior.

COMMANDS

```

add_principal [options] newprinc
delete_principal [-force] principal
modify_principal [options] principal
change_password [options] principal
get_principal [-terse] principal
list_principals [expression]
add_policy [options] policy
delete_policy [-force] policy
modify_policy [options] policy
get_policy [-terse] policy
list_policies [expression]
ktadd [-k keytab] [-q] [-e keysaltlist] [principal | -glob princ-exp] [...]
    Adds a principal or all principals matching princ-exp to a keytab.
ktremove [-k keytab] [-q] principal [kvno | all | old]
    Removes entries for the specified principal from a keytab.

```

FILES

```

principal.db      default name for Kerberos principal database

<dbname>.kadm5    KADM5 administrative database. (This would be "principal.kadm5", if you use
                  the default database name.) Contains policy information.

<dbname>.kadm5.lock  lock file for the KADM5 administrative database. This file works backwards
                    from most other lock files. I.e., kadmin will exit with an error if this file
                    does not exist.

kadm5.acl         file containing list of principals and their kadmin administrative privileges.

kadm5.keytab      keytab file for kadmin/admin principal.

kadm5.dict        file containing dictionary of strings explicitly disallowed as passwords.

```

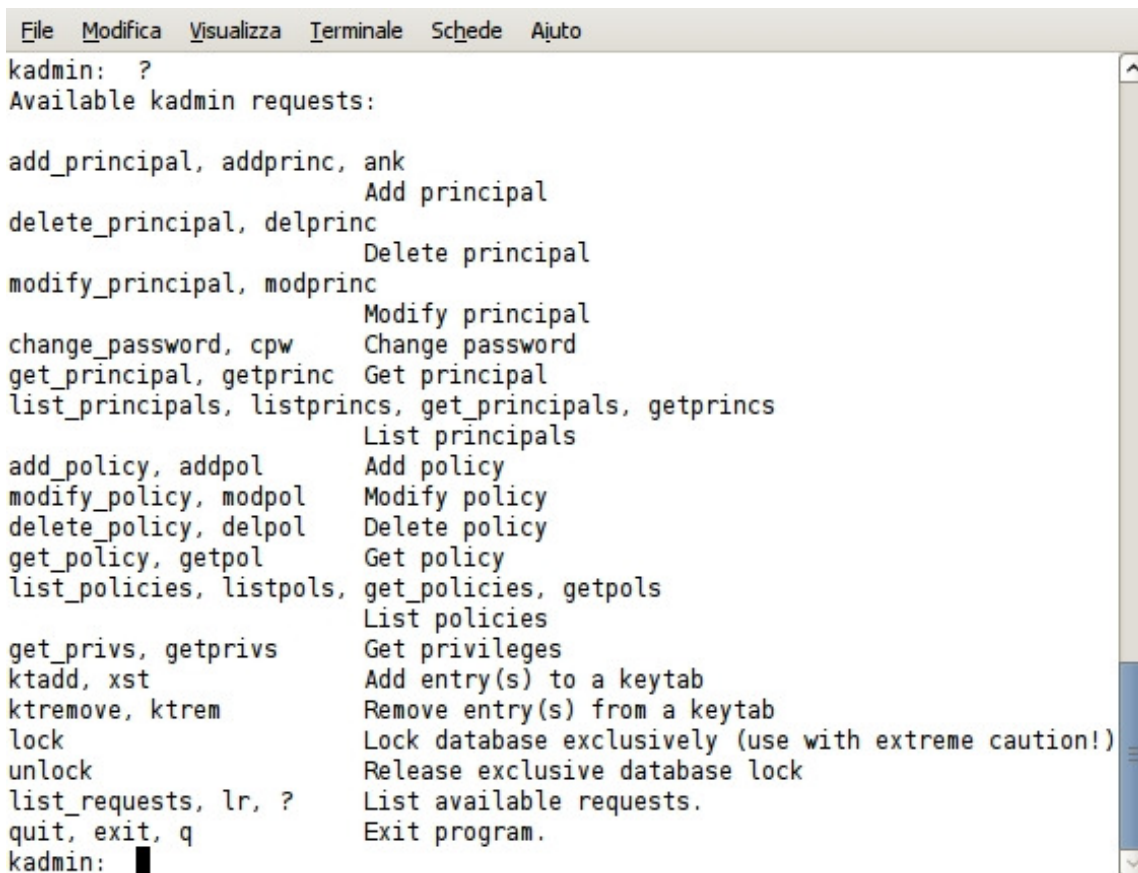
HISTORY

The kadmin program was originally written by Tom Yu at MIT, as an interface to the OpenVision Kerberos administration program.

SEE ALSO

kerberos(1), kpasswd(1), kadmind(8)

BUGS Command output needs to be cleaned up.

NOTA: LISTA OPZIONI CONSOLE KADMIN


```

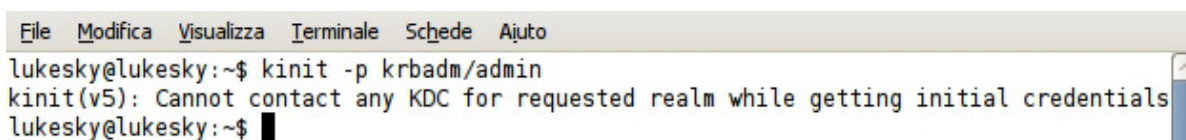
File  Modifica  Visualizza  Terminale  Schede  Ajuto
kadmin: ?
Available kadmin requests:

add_principal, addprinc, ank          Add principal
delete_principal, delprinc           Delete principal
modify_principal, modprinc           Modify principal
change_password, cpw                 Change password
get_principal, getprinc              Get principal
list_principals, listprincs, get_principals, getprincs  List principals
add_policy, addpol                   Add policy
modify_policy, modpol                Modify policy
delete_policy, delpol                Delete policy
get_policy, getpol                   Get policy
list_policies, listpols, get_policies, getpols          List policies
get_privs, getprivs                  Get privileges
ktadd, xst                            Add entry(s) to a keytab
ktremove, ktrem                       Remove entry(s) from a keytab
lock                                   Lock database exclusively (use with extreme caution!)
unlock                                 Release exclusive database lock
list_requests, lr, ?                  List available requests.
quit, exit, q                          Exit program.
kadmin: █

```

NOTA: ERRORI COMUNI

Un errore abbastanza comune che è possibile incontrare durante il testing del corretto funzionamento del sistema Kerberos è il seguente:



```

File  Modifica  Visualizza  Terminale  Schede  Ajuto
lukesky@lukesky:~$ kinit -p krbadm/admin
kinit(v5): Cannot contact any KDC for requested realm while getting initial credentials
lukesky@lukesky:~$ █

```

Questo errore può verificarsi quando si tenta di richiedere il ticket al KDC; se ciò accade è buona norma ricontrollare la corretta mappatura degli indirizzi IP nel file *hosts*, in modo da accertarsi che l'hostname del KDC sia raggiunto con successo, e successivamente effettuare un riavvio dei servizi krb per aggiornare la configurazione.

NOTA: AVVIO SERVER KERBEROS

Dalla shell di Linux, se necessario è possibile ri-avviare o arrestare i demoni kerberos aggiungendoli o rimuovendoli dal *run level*. In particolar modo i comandi da eseguire per agire sui processi sono:

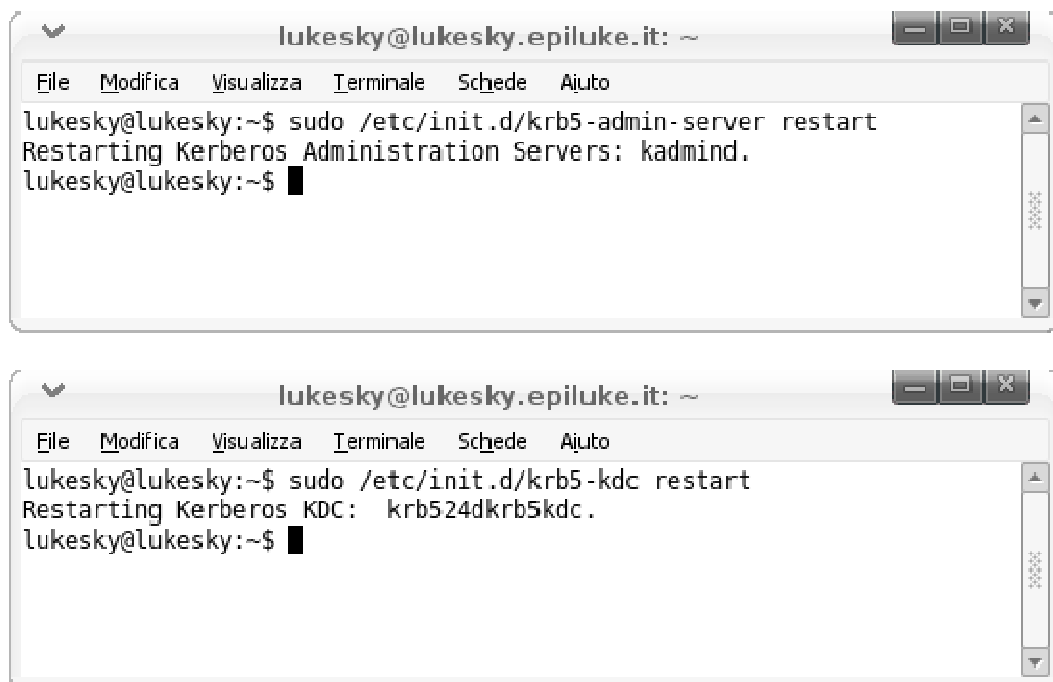
```
$ /etc/init.d/krb5-kdc OPERAZIONE  
  
( dove OPERAZIONE può essere: start / restart / stop )
```

per il Kdc Server;

```
$ /etc/init.d/krb5-admin-server OPERAZIONE  
  
( dove OPERAZIONE può essere: start / restart / stop )
```

per l' Admin-Server.

ESEMPIO RESTART SERVER



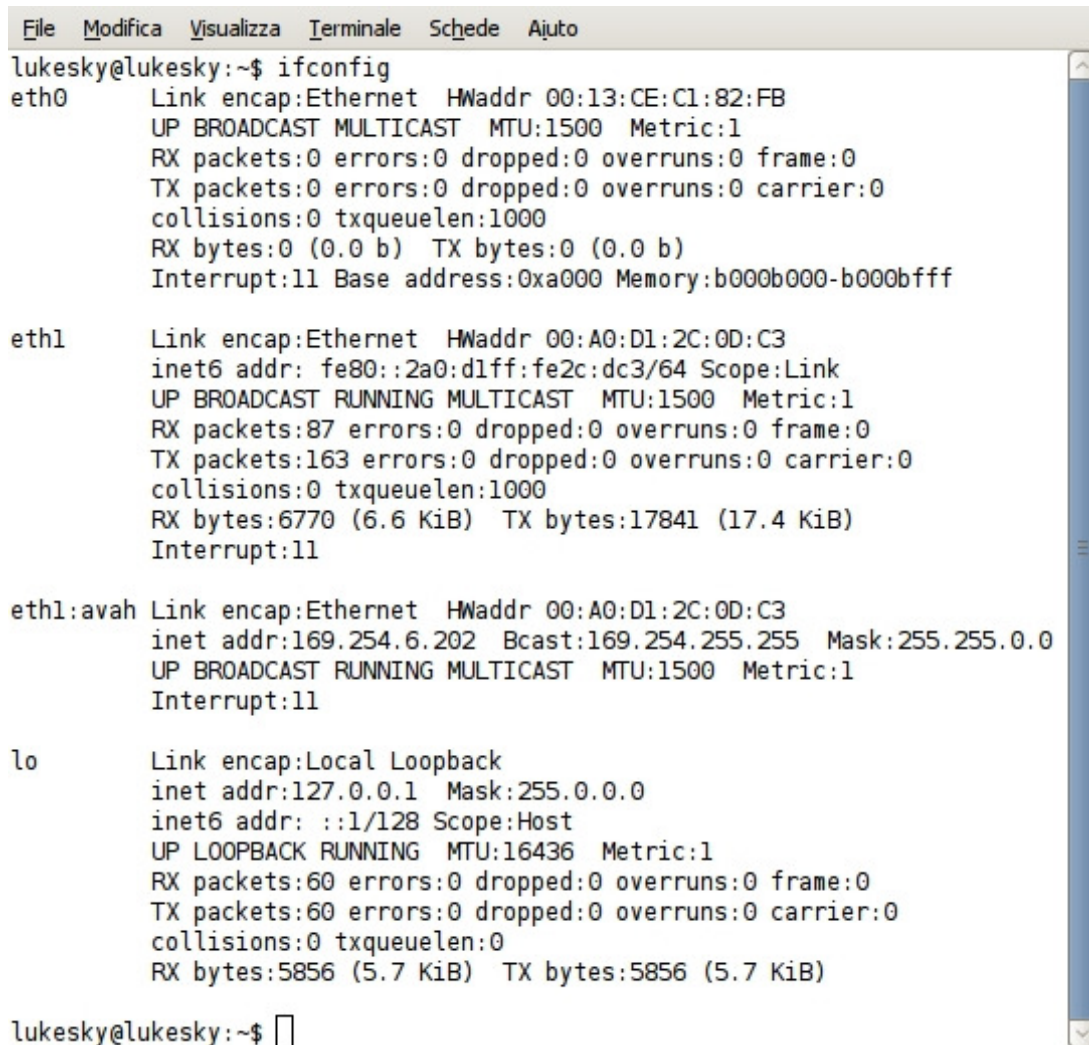
The image shows two terminal windows from a user named 'lukesky' on a machine named 'lukesky.epiluke.it'. The top window shows the command 'sudo /etc/init.d/krb5-admin-server restart' being executed, resulting in the output 'Restarting Kerberos Administration Servers: kadmind.'. The bottom window shows the command 'sudo /etc/init.d/krb5-kdc restart' being executed, resulting in the output 'Restarting Kerberos KDC: krb524dkrb5kdc.'.

```
lukesky@lukesky.epiluke.it: ~  
File Modifica Visualizza Terminale Schede Aiuto  
lukesky@lukesky:~$ sudo /etc/init.d/krb5-admin-server restart  
Restarting Kerberos Administration Servers: kadmind.  
lukesky@lukesky:~$ █
```

```
lukesky@lukesky.epiluke.it: ~  
File Modifica Visualizza Terminale Schede Aiuto  
lukesky@lukesky:~$ sudo /etc/init.d/krb5-kdc restart  
Restarting Kerberos KDC: krb524dkrb5kdc.  
lukesky@lukesky:~$ █
```

NOTA: INFORMAZIONI SULLA RETE LOCALE

Per avere informazioni sugli indirizzi IP della rete e settare correttamente la mappatura degli hosts è possibile eseguire il comando *ifconfig*:



```
lukesky@lukesky:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:13:CE:C1:82:FB
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:11 Base address:0xa000 Memory:b000b000-b000bfff

eth1      Link encap:Ethernet  HWaddr 00:A0:D1:2C:0D:C3
          inet6 addr: fe80::2a0:d1ff:fe2c:dc3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:87 errors:0 dropped:0 overruns:0 frame:0
          TX packets:163 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6770 (6.6 KiB)  TX bytes:17841 (17.4 KiB)
          Interrupt:11

eth1:avah Link encap:Ethernet  HWaddr 00:A0:D1:2C:0D:C3
          inet addr:169.254.6.202 Bcast:169.254.255.255 Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:11

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:60 errors:0 dropped:0 overruns:0 frame:0
          TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5856 (5.7 KiB)  TX bytes:5856 (5.7 KiB)

lukesky@lukesky:~$
```

In questo caso l'identificativo *eth1* punta alla scheda di rete LAN.

QUADRO RIASSUNTIVO

- Applicazioni del client: *kinit*, *kdestroy*, *klist*
- Applicazioni amministrative: *kadmin*, *kdb5_util*
- Applicazioni del server: *krb5kdc*, *kadmind*
- File di configurazione: *krb5.conf*, *kdc.conf*

FASE 7 – CONFIGURAZIONE DI UN CLIENT

La configurazione di un client Kerberos risulta meno impegnativa rispetto a quella del server. In primo luogo occorre installare i pacchetti client e fornire a ciascun client un file di configurazione *krb5.conf* valido; ciò significa che il client deve aver installati i pacchetti che consentono di usufruire delle applicazioni client (es: *kinit*) ed, inoltre, deve avere lo stesso file di configurazione *krb5.conf* del server.

Una volta configurato, il client può consentire allo specifico utente l'accesso al servizio kerberizzato qualora ne abbia il permesso.

3.3 Attivazione servizi ad accesso kerberizzato

A questo punto è interessante definire il modo in cui il protocollo Kerberos può essere integrato nei differenti servizi di rete per offrire comunicazioni sicure ed accessi autenticati.

Come precedentemente affermato il sistema *Kerberos*, come anche i protocolli *SSL* ed *HTTPS*, permette di consolidare il grado di sicurezza nelle comunicazioni di rete.



Il protocollo Kerberos può essere integrato in differenti servizi di rete che devono essere prima di tutto impostati ed avviati. Un elenco dei possibile servizi kerberizzati più comuni è il seguente:

- *TELNET*

E' un protocollo di rete con l'obiettivo di fornire un supporto per le comunicazioni sufficientemente generalizzato, bidirezionale ed orientato al byte. E' solitamente utilizzato per fornire all'utente sessioni di login remoto di tipo linea di comando tra host su internet.

Ci sono tre problemi principali legati a Telnet, che lo rendono una brutta scelta per sistemi moderni dal punto di vista della sicurezza:

- Nei daemon telnet comunemente usati sono state trovate nel corso degli anni molte vulnerabilità, e probabilmente altre esistono tuttora.
- Telnet non cripta i dati inviati tramite la connessione (nemmeno le password) ed è quindi banale catturare i dati scambiati ed usare la password per scopi malevoli.

- A Telnet manca uno schema di autenticazione che renda sicura la comunicazione tra due host e non intercettabile.

In ambienti dove la sicurezza è importante, come la rete pubblica Internet, telnet non dovrebbe essere usato. Queste falle hanno fatto sì che l'uso del protocollo telnet cadesse rapidamente a favore del più sicuro protocollo SSH.

- *FTP (File Transfer Protocol)*

Protocollo applicativo standard TCP/IP per trasferire file da un host ad un altro; è un servizio che fornisce gli elementi fondamentali per la condivisione di file tra host.

Gli obiettivi dell'FTP sono:

1. promuovere la condivisione di file (programmi o dati)
2. incoraggiare l'uso indiretto o implicito (tramite programma) di computer remoti
3. salvaguardare l'utente al variare dei sistemi di stoccaggio file, tra un host e l'altro
4. trasferire dati in maniera affidabile ed efficiente

Attualmente si tende a ricorrere sempre meno all'utilizzo di server FTP in quanto presentano alcune vulnerabilità.

- *HTTP (Hyper Text Transfer Protocol)*

È il protocollo di trasferimento di un ipertesto, usato come principale sistema per la trasmissione di informazione sul web.

- *LDAP (Lightweight Directory Access Protocol)*

È un protocollo standard per l'interrogazione e la modifica dei *servizi di directory*. LDAP definisce un semplice protocollo basato su TCP per la ricerca e l'aggiornamento di informazioni contenute all'interno di un *directory service*.

- *IMAP (Internet Message Access Protocol)*

E' un protocollo di comunicazione per la ricezione di e-mail; Il protocollo è stato inventato come alternativa più moderna al protocollo POP.

- *CVS (Concurrent Versions System)*

E' un servizio che implementa un sistema di controllo versione, utilizza un'architettura client-server: un server immagazzina la versione corrente di un progetto e la sua storia, ed il client si connette al server per verificare l'ultima versione disponibile del software ed utilizzarla. Tipicamente, client e server si connettono su una LAN o su Internet.

- *SSH (Secure Shell)*

E' un protocollo che permette di stabilire una sessione remota cifrata ad interfaccia a linea di comando con un altro host. SSH offre tutte le funzioni di telnet più una sicura criptazione, che previene l'intercettazione dei dati scambiati, ed un'autenticazione a chiave pubblica, che assicura l'identità del server remoto. Il client SSH ha una interfaccia a linea di comando simile a quella di *telnet* e *rlogin*, ma l'intera comunicazione (ovvero sia l'autenticazione che la sessione di lavoro) avviene in maniera cifrata.

3.3.1 Servizio TELNET

Come esempio applicativo prendiamo in considerazione il servizio Telnet e configuriamo il server in modo da predisporre un accesso di tipo kerberizzato.

- PACCHETTO NECESSARIO: krb5-telnetd

In primo luogo occorre aggiungere un *principal host* alla workstation sul *KDC*, cioè si deve definire il “principal” del nodo server per riconoscere il servizio:

```
$ sudo kadmin -p krbadm/admin
kadmin: addprinc host/lukesky.epiluke.it
kadmin: ktadd -k /etc/krb5.keytab host/lukesky.epiluke.it
```

A questo punto occorre avviare il servizio; si dovrà quindi creare un file di definizione del servizio da associare al demone *xinetd* (*NB: controllare che il pacchetto xinetd sia installato sul sistema*).

DEMONO XINETD

Il demone *xinetd* è utilizzato per la gestione di servizi di rete; in sostanza è un demone che gestisce altri demoni.

Le sue funzioni sono molteplici: controllo sull’accesso, funzionalità di log, possibilità di associare un servizio a specifiche interfacce di rete, limitazione di possibili attacchi Denial of Service. Dopo aver letto il suo file di configurazione (*/etc/xinetd.conf*), *xinetd* si mette in ascolto sulle porte indicate avviando il servizio richiesto, per testare se vengono superati i controlli. Il file */etc/xinetd.conf* contiene solo alcuni valori di default che si applicano a tutti i servizi e un’istruzione che rimanda alla directory */etc/xinetd.d/*, che contiene i file di configurazione di ogni singolo servizio.

Nella directory */etc/xinetd.d/* sono contenuti i file di definizione per ogni servizio gestito da *xinetd* ed i nomi dei file relativi al servizio (il formato dei file nella directory */etc/xinetd.d/* usa le stesse convenzioni di */etc/xinetd.conf*). Come con *xinetd.conf*, questa directory viene letta solo quando il servizio *xinetd* è avviato. Per confermare qualsiasi cambiamento, l’amministratore deve riavviare il

servizio xinetd (`sudo /etc/init.d/xinetd restart`). La ragione principale per la quale la configurazione di ogni servizio viene conservata in file separati, è quella di permettere una personalizzazione più facile con minore probabilità di condizionare altri servizi. Il file di configurazione valido da definire per il caricamento del servizio telnet kerberizzato sarà il seguente:

```
service telnet
{
    socket_type    = stream
    wait          = no
    nice          = 10
    user          = root
    server        = /usr/sbin/telnetd
    server_args   = -h
}
```

telnet

Ogni istruzione riportata nel file definisce un preciso aspetto del servizio, in generale:

- **service** - Definisce il nome del servizio, generalmente usando un nome elencato nel file `/etc/services`.
- **flags** - Imposta un numero di attributi per il collegamento. REUSE indica a xinetd di usare nuovamente il socket per un collegamento Telnet.
- **socket_type** - Imposta il tipo di rete socket per *stream*.
- **wait** - Definisce se il servizio è singolo “single-threaded” (*yes*) o multiplo “multi-threaded” (*no*).
- **user** - Definisce con quale user ID verrà eseguito il processo.
- **server** - Definisce il binario eseguibile da lanciare.
- **log_on_failure** - Definisce i parametri di logging per *log_on_failure* in aggiunta a quelli già definiti in *xinetd.conf*.
- **disable** - Definisce se il server è attivo.

A questo punto non resterà che riavviare *xinetd* e testare l’accesso al servizio *telnet* tramite client.

3.3.2 Servizio FTP

Gli stessi principi di funzionamento valgono per il servizio FTP; riadattiamo, quindi, i passi precedentemente esposti per integrare il protocollo Kerberos ad *FTP*.

- PACCHETTO NECESSARIO: krb5-ftp

1 – AGGIUNTA PRINCIPAL

```
kadmin: addprinc ftp/lukesky.epiluke.it
kadmin: ktadd -k /etc/krb5.keytab ftp/lukesky.epiluke.it
```

2 – CONFIGURAZIONE SERVIZIO

```
service ftp
{
    socket_type    = stream
    wait          = no
    nice          = 10
    user          = root
    server        = /usr/sbin/ftpd
}
```

ftp

Il comando *klist* ci consente di verificare la cache delle credenziali osservando i ticket registrati relativi ai servizi.

```
lukesky@lukesky.epiluke.it: ~
File Modifica Visualizza Terminale Schede Aiuto
lukesky@lukesky:~$ sudo klist -kte
Keytab name: FILE:/etc/krb5.keytab
KVNO Timestanp Principal
-----
 2 02/19/07 11:47:03 host/lukesky.epiluke.it@EPILUKE.IT (Triple DES cbc mode with HMAC/sha1)
 2 02/19/07 11:47:03 host/lukesky.epiluke.it@EPILUKE.IT (DES cbc mode with CRC-32)
 2 02/19/07 11:47:08 ftp/lukesky.epiluke.it@EPILUKE.IT (Triple DES cbc mode with HMAC/sha1)
 2 02/19/07 11:47:08 ftp/lukesky.epiluke.it@EPILUKE.IT (DES cbc mode with CRC-32)
lukesky@lukesky:~$ █
```

in particolare le opzioni *-kte* individuano i seguenti settaggi:

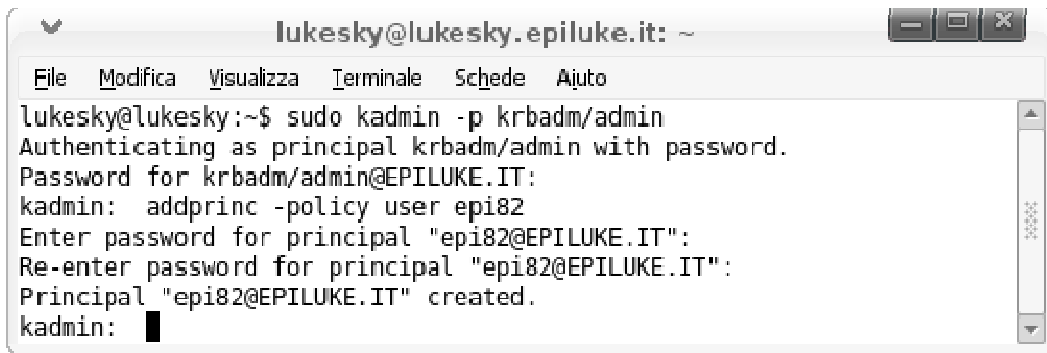
- e → Displays the encryption types of the session key and the ticket for each credential in the credential cache, or each key in the keytab file.
- k → List keys held in a keytab file.
- t → Display the time entry timestamps for each keytab entry in the keytab file.

REGISTRAZIONE USER

Uno specifico utente può sfruttare un host della rete, opportunamente configurato come client Kerberos, per accedere ai vari servizi. L'utente dovrà essere riconosciuto dal sistema sulla base delle proprie credenziali; in tal senso sarà necessario sia registrare un nuovo principal per il sistema Kerberos (in modo da abilitare l'utente all'autenticazione tramite Kerberos), sia aggiungere un nuovo user per il server (in modo da riconoscere l'utente). Dal server di amministrazione si assocerà un nuovo *principal* corrispondente allo *user* che intende usufruire dei servizi.

In definitiva la fase di registrazione di un nuovo utente si articola in due passi:

1) AGGIUNTA DI UN NUOVO PRINCIPAL NEL KERBEROS DATABASE

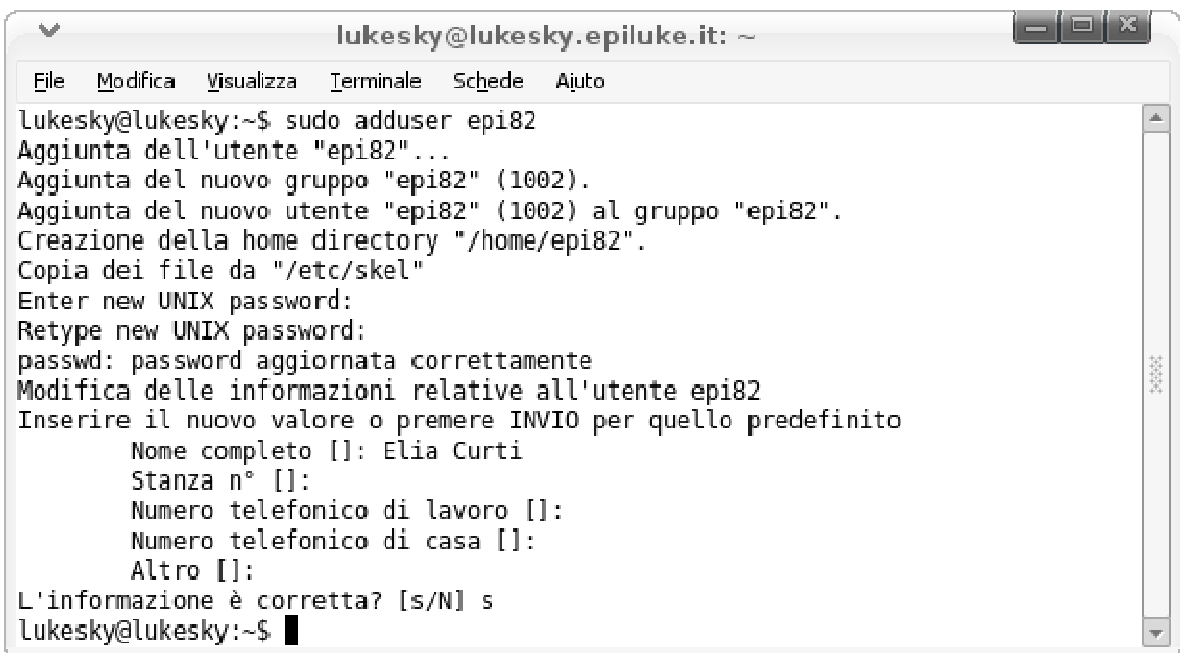


```

lukesky@lukesky.epiluke.it: ~
File Modifica Visualizza Terminale Schede Aiuto
lukesky@lukesky:~$ sudo kadmin -p krbadm/admin
Authenticating as principal krbadm/admin with password.
Password for krbadm/admin@EPILUKE.IT:
kadmin: addprinc -policy user epi82
Enter password for principal "epi82@EPILUKE.IT":
Re-enter password for principal "epi82@EPILUKE.IT":
Principal "epi82@EPILUKE.IT" created.
kadmin: █

```

2) AGGIUNTA DI UN NUOVO UTENTE AL SERVER



```

lukesky@lukesky.epiluke.it: ~
File Modifica Visualizza Terminale Schede Aiuto
lukesky@lukesky:~$ sudo adduser epi82
Aggiunta dell'utente "epi82"...
Aggiunta del nuovo gruppo "epi82" (1002).
Aggiunta del nuovo utente "epi82" (1002) al gruppo "epi82".
Creazione della home directory "/home/epi82".
Copia dei file da "/etc/skel"
Enter new UNIX password:
Retype new UNIX password:
passwd: password aggiornata correttamente
Modifica delle informazioni relative all'utente epi82
Inserire il nuovo valore o premere INVIO per quello predefinito
Nome completo []: Elia Curti
Stanza n° []:
Numero telefonico di lavoro []:
Numero telefonico di casa []:
Altro []:
L'informazione è corretta? [s/N] s
lukesky@lukesky:~$ █

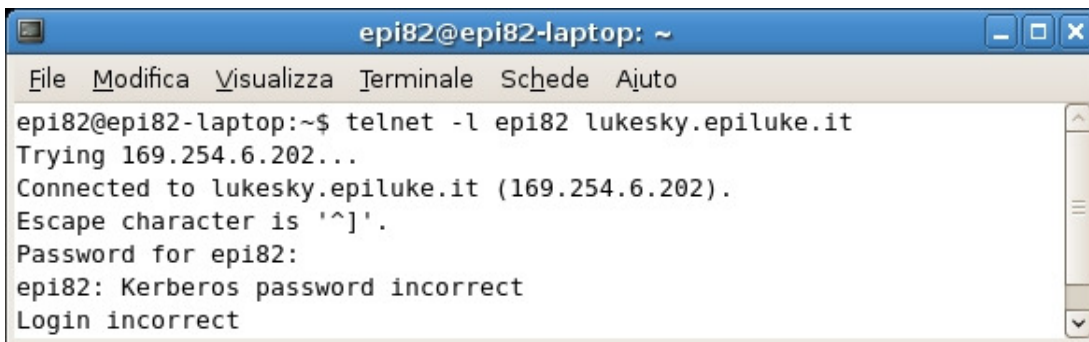
```

È stato aggiunto al server l'utente *epi82* per cui è stata abilitata la possibilità di autenticazione tramite Kerberos, a questo punto non resta che testare l'accesso ai servizi da parte del client.

(NOTA: LA UNIX PASSWORD IMPOSTATA PER EPI82 E' '3005')

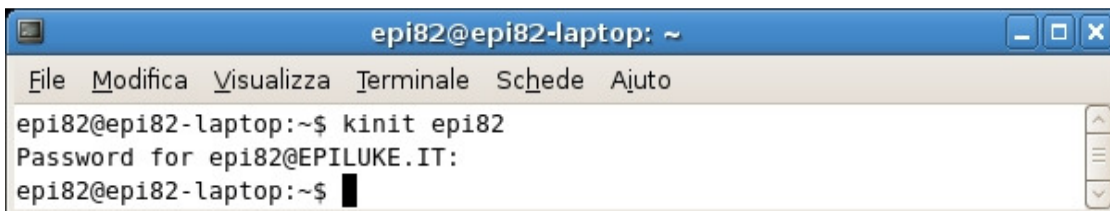
ACCESSO AI SERVIZI

ACCESSO TELNET SENZA RICHIESTA DEL TICKET



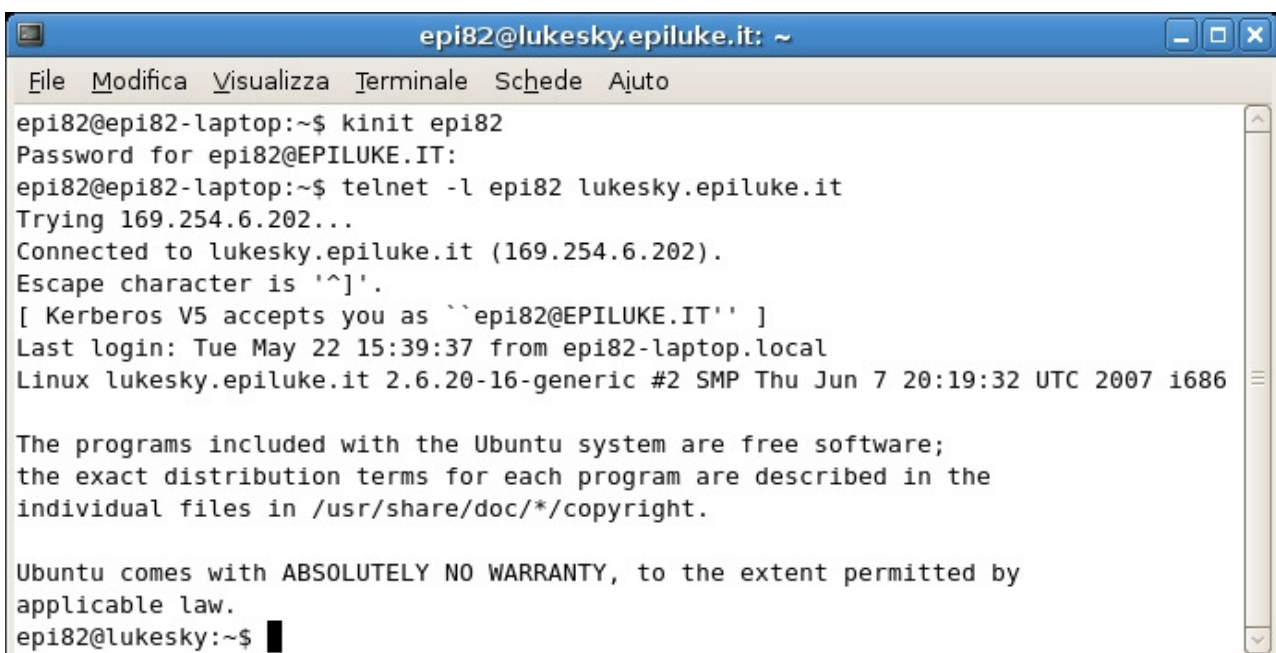
```
epi82@epi82-laptop: ~
File Modifica Visualizza Terminale Schede Ajuto
epi82@epi82-laptop:~$ telnet -l epi82 lukesky.epiluke.it
Trying 169.254.6.202...
Connected to lukesky.epiluke.it (169.254.6.202).
Escape character is '^]'.
Password for epi82:
epi82: Kerberos password incorrect
Login incorrect
```

RICHIESTA DEL TICKET



```
epi82@epi82-laptop: ~
File Modifica Visualizza Terminale Schede Ajuto
epi82@epi82-laptop:~$ kinit epi82
Password for epi82@EPILUKE.IT:
epi82@epi82-laptop:~$ █
```

ACCESSO TELNET KERBERIZZATO

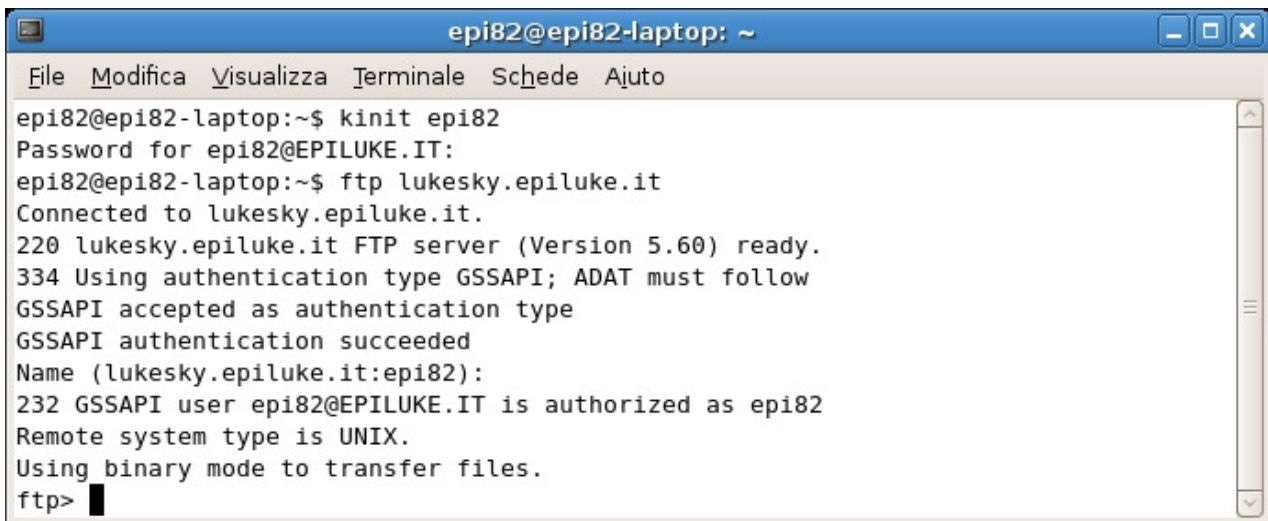


```
epi82@lukesky.epiluke.it: ~
File Modifica Visualizza Terminale Schede Ajuto
epi82@epi82-laptop:~$ kinit epi82
Password for epi82@EPILUKE.IT:
epi82@epi82-laptop:~$ telnet -l epi82 lukesky.epiluke.it
Trying 169.254.6.202...
Connected to lukesky.epiluke.it (169.254.6.202).
Escape character is '^]'.
[ Kerberos V5 accepts you as ``epi82@EPILUKE.IT'' ]
Last login: Tue May 22 15:39:37 from epi82-laptop.local
Linux lukesky.epiluke.it 2.6.20-16-generic #2 SMP Thu Jun 7 20:19:32 UTC 2007 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
epi82@lukesky:~$ █
```

ACCESSO FTP KERBERIZZATO



```
epi82@epi82-laptop: ~  
File Modifica Visualizza Terminale Schede Ajuto  
epi82@epi82-laptop:~$ kinit epi82  
Password for epi82@EPILUKE.IT:  
epi82@epi82-laptop:~$ ftp lukesky.epiluke.it  
Connected to lukesky.epiluke.it.  
220 lukesky.epiluke.it FTP server (Version 5.60) ready.  
334 Using authentication type GSSAPI; ADAT must follow  
GSSAPI accepted as authentication type  
GSSAPI authentication succeeded  
Name (lukesky.epiluke.it:epi82):  
232 GSSAPI user epi82@EPILUKE.IT is authorized as epi82  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp>
```

3.3.3 Servizio SSH

I pacchetti necessari da installare per poter configurare ed usufruire del servizio SSH sono i seguenti:

- *ssh*
- *openssh-server*
- *openssh-client*

non è necessario installare lo specifico pacchetto *ssh-krb5*; tale pacchetto rappresenta una precedente versione del servizio realizzata per la distribuzione Debian. Installando il servizio SSH sulla distribuzione Ubuntu attraverso i pacchetti precedentemente elencati sarà possibile utilizzare il protocollo Kerberos senza dover integrare ulteriori package, in quanto il servizio SSH impostato per Ubuntu offre già la possibilità di sfruttare questo tipo di autenticazione.

OpenSSH (Open Secure Shell)

è un insieme di programmi che rendono disponibili sessioni crittografate di comunicazione in una rete di computer usando il protocollo SSH.

CONFIGURAZIONE

Una volta che il servizio SSH è integrato nel sistema occorre configurarlo in modo opportuno predisponendolo all'utilizzo del protocollo Kerberos. Il file di configurazione da editare è:

/etc/ssh/sshd_config.

```
# Package generated configuration file
# See the sshd(8) manpage for details

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 768

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile      %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication
#IgnoreUserKnownHosts yes
```

```
# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Change to no to disable tunnelled clear text passwords
#PasswordAuthentication yes

# Kerberos options
KerberosAuthentication yes
#KerberosGetAFSToken no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes

# GSSAPI options
GSSAPIAuthentication yes
#GSSAPICleanupCredentials yes

X11Forwarding yes
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
TCPKeepAlive yes
#UseLogin no

#MaxStartups 10:30:60
#Banner /etc/issue.net

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

Subsystem sftp /usr/lib/openssh/sftp-server

UsePAM yes
```

sshd_config

come è facilmente intuibile dal file, tra le opzioni di configurazione è possibile abilitare l'autenticazione Kerberos; questo file di configurazione è necessario impostarlo per ogni client del sistema che vuole eseguire un accesso kerberizzato verso il server.

GSSAPI (Generic Security Services Application Programming Interface)

GSSAPI sono API generiche per autenticazioni client-server. Il motivo per cui è stata usata una interfaccia API per la sicurezza è di far sì che più sistemi diversi vengano sviluppati con interfaccia comune, permettendo loro di poter comunicare in maniera sicura (Kerberos 5 può essere integrato mediante l'interfaccia GSSAPI).

Il servizio SSH integra un controllo preliminare per la verifica del client; è necessario verificare l'host key del client per permettere l'esecuzione del *login* (viene impiegato il sistema di cifratura a chiave pubblica *RSA*). Questa caratteristica del servizio offre una protezione per attacchi di *IP Spoofing*, *DNS Spoofing* e *Routing Spoofing*.

Nel file `~/.ssh/known_hosts` sono contenute le informazioni per l'accesso degli hosts riconosciuti, un esempio del file è il seguente:

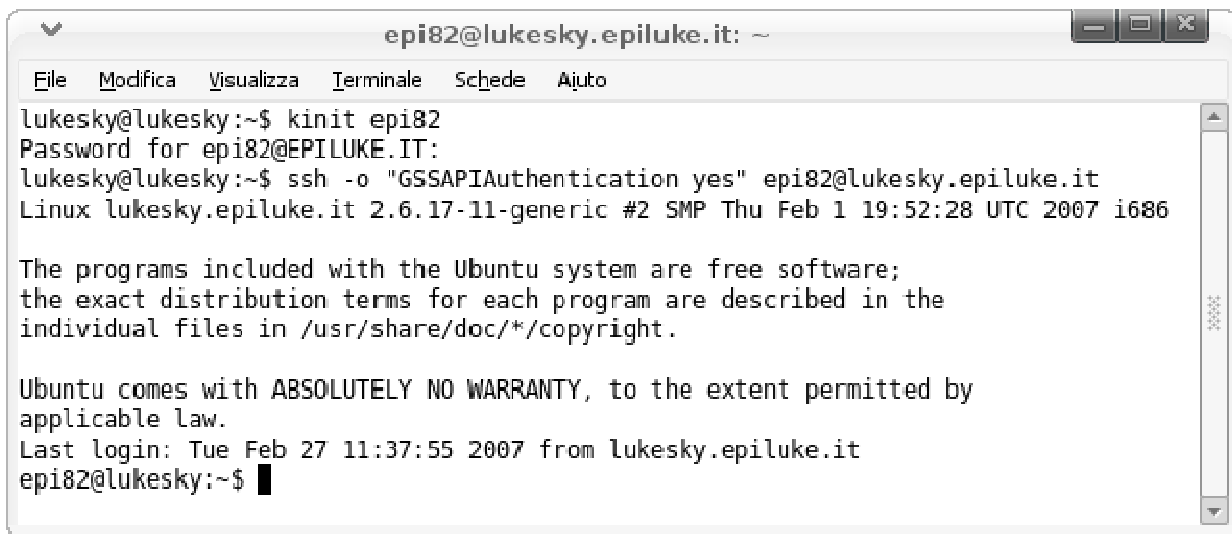
```
lukesky.epiluke.it,169.254.137.61 ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAyVb+q8DrIWp7R4HqhznMHsYh8gt9UZD5tC+ZZUIkXFND3koWSWII
vVTSRLZLrVEbUFBwwMYOw+hzyngyQnQcuPgcxn1zZD+QsasLC/mccNfHrQOghCNYATHPPskSawdaZABx
AYbKZEzESy2IVM7+DtT4ZVGvS8urCJGoWxJnS+BODyT1KGNXEyCZBMYXJXkQgqShtO10YZ1zNgIB2YX+
nq5seKszTRmJC0VFi23zkYkFF2RpYqsQplq7SjThukffakBrNFv8HNxptt2NY+en8vrSg2BgygriIhzhSxr19bH
3LjsR09Bm8h1GHw1M+3lzNgEFkkT4LRacoI0UmdjtjQ==
```

known_hosts

NOTA: RICARICARE IL SERVIZIO



```
lukesky@lukesky.epiluke.it: ~
File Modifica Visualizza Terminale Schede Aiuto
lukesky@lukesky:~$ sudo /etc/init.d/ssh reload
* Reloading OpenBSD Secure Shell server's configuration [ ok ]
lukesky@lukesky:~$ █
```

ACCESSO SSH KERBERIZZATO

```
epi82@lukesky.epiluke.it: ~
File  Modifica  Visualizza  Terminale  Schede  Ajuto
lukesky@lukesky:~$ kinit epi82
Password for epi82@EPILUKE.IT:
lukesky@lukesky:~$ ssh -o "GSSAPIAuthentication yes" epi82@lukesky.epiluke.it
Linux lukesky.epiluke.it 2.6.17-11-generic #2 SMP Thu Feb 1 19:52:28 UTC 2007 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
Last login: Tue Feb 27 11:37:55 2007 from lukesky.epiluke.it
epi82@lukesky:~$ █
```

Sintassi linea di comando: \$ ssh [opzioni] nomeutente@host [comando]
