Bhartiya Vidya Bhavan's
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Autonomous Institute Affiliated to the University of Mumbai)
Academic Year: 2024 -25

Experiment No. 10

Semester: V **Batch:** A4

Name	Saayna Narvekar
Class	TE CSE – A4
UID	2023800067
Subject	Cryptography and Network Security

Aim:

- Understand how Kerberos authentication works (tickets, KDC, TGT, etc.).
- Set up a simple Kerberos Key Distribution Center (KDC).
- Configure a client and service to use Kerberos for authentication.
- Test the Kerberos authentication process.

"server-kdc" vm commands 192.168.64.12

`ip a` find server's ip address

`sudo nano /etc/hosts` edit hosts file to map names to ips

`sudo apt install krb5-kdc krb5-admin-server krb5-user` install all kerberos server & client packages

`sudo nano /etc/krb5.conf` edit the main kerberos config file

`sudo krb5_newrealm` create the new kerberos database (realm)

`sudo kadmin.local` start the local kerberos admin tool

`kadmin.local: addprinc admin/admin` add the main admin user

`kadmin.local: quit` exit the admin tool

`sudo systemctl enable krb5-kdc krb5-admin-server` make the kdc services start on boot

`sudo systemctl start krb5-kdc krb5-admin-server` start the kdc services now

`sudo systemctl status krb5-kdc` check if the kdc service is running

`sudo kadmin.local` start the local kerberos admin tool again

`kadmin.local: addprinc student` add the 'student' user

`kadmin.local: addprinc -randkey host/server.lab.example.com` add the 'host' service for the server (with a random key)

`kadmin.local: ktadd host/server.lab.example.com` save the host's key into the server's key file

`kadmin.local: quit` exit the admin tool

`sudo adduser student` add a matching local linux user for 'student'

`sudo apt install openssh-server` install the ssh server software

`sudo nano /etc/ssh/sshd_config` edit the ssh server's config file

`sudo systemctl restart sshd` restart the ssh server to apply changes

`sudo journalctl -u krb5-kdc.service -n 30 --no-pager` view the kdc logs

"client" vm commands 192.168.64.11

`ip a` find client's ip address

`sudo nano /etc/hosts` edit hosts file to map names to ips

`ping server.lab.example.com` test connection to the server by name

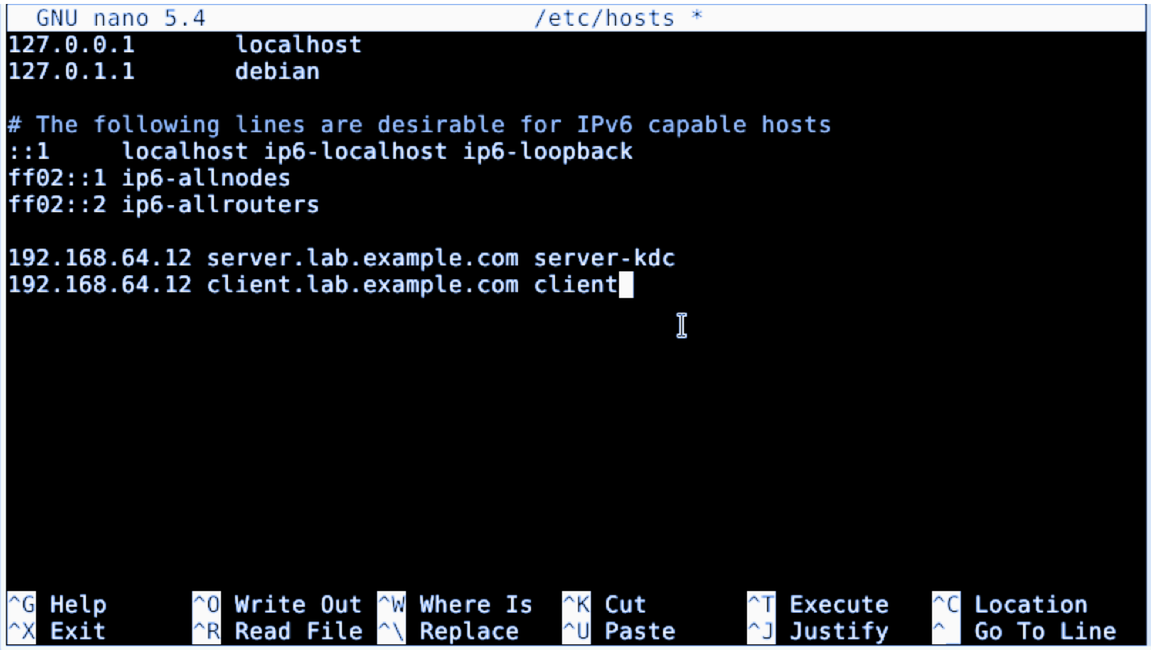
`sudo apt install krb5-kdc krb5-admin-server krb5-user` install kerberos client packages

`sudo nano /etc/krb5.conf` edit the main kerberos config file (to match the server)

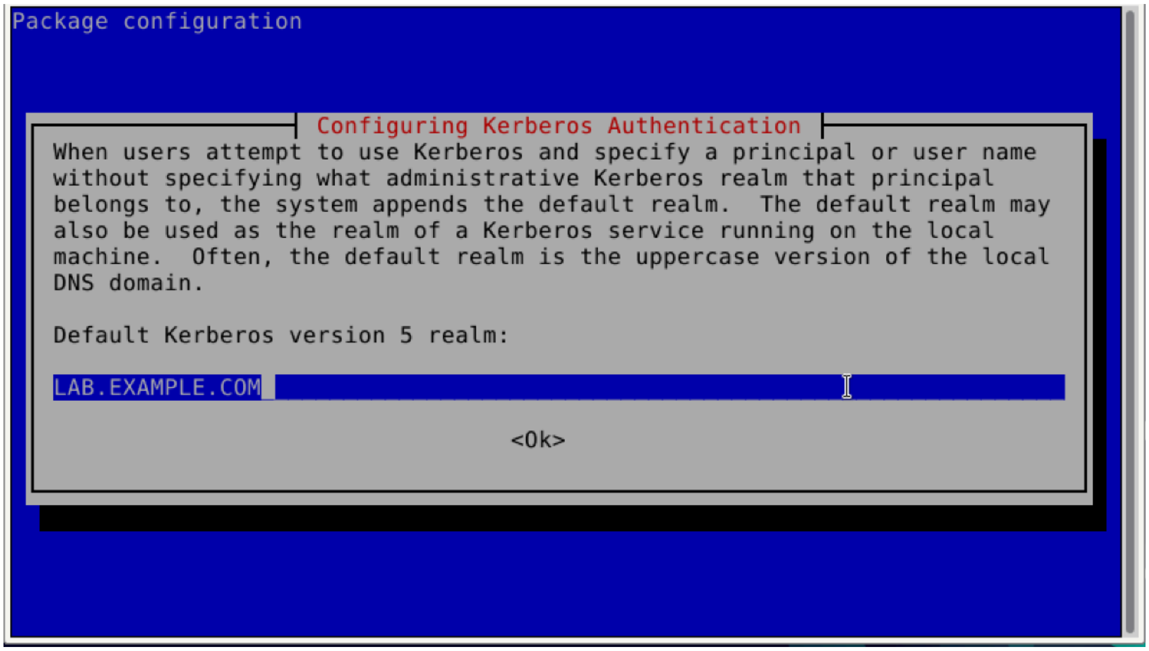
`kinit admin/admin` get a ticket for the admin user (tests the kdc)

`klist` list the tickets we have

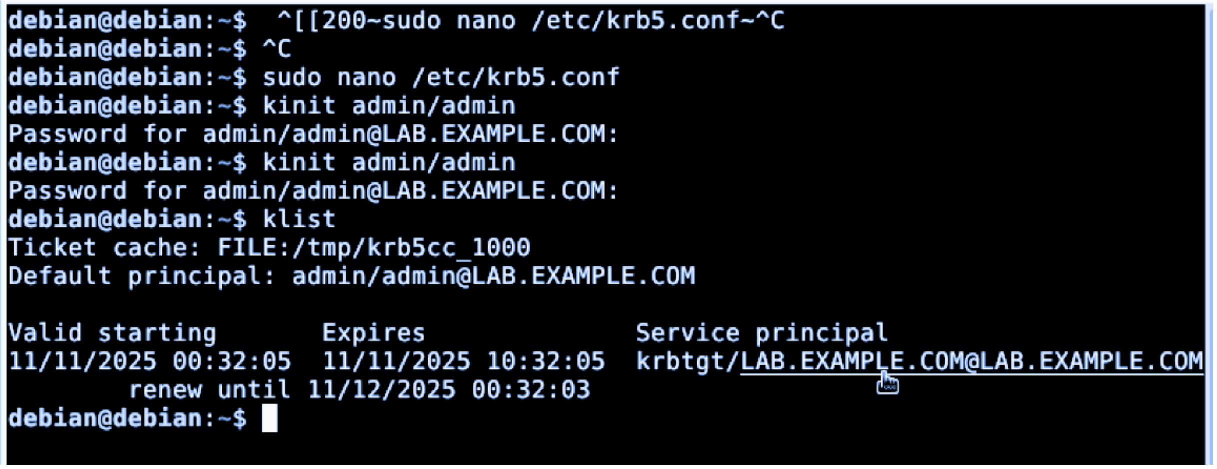
`kdestroy` destroy all current tickets
`klist` verify that tickets are gone
`kinit student` get a ticket for the 'student' user
`klist` list the ticket for 'student' (for lab report)
`ssh -o GSSAPIAuthentication=yes student@server.lab.example.com` log in to the server using the ticket (no password)
`exit` log out of the ssh session



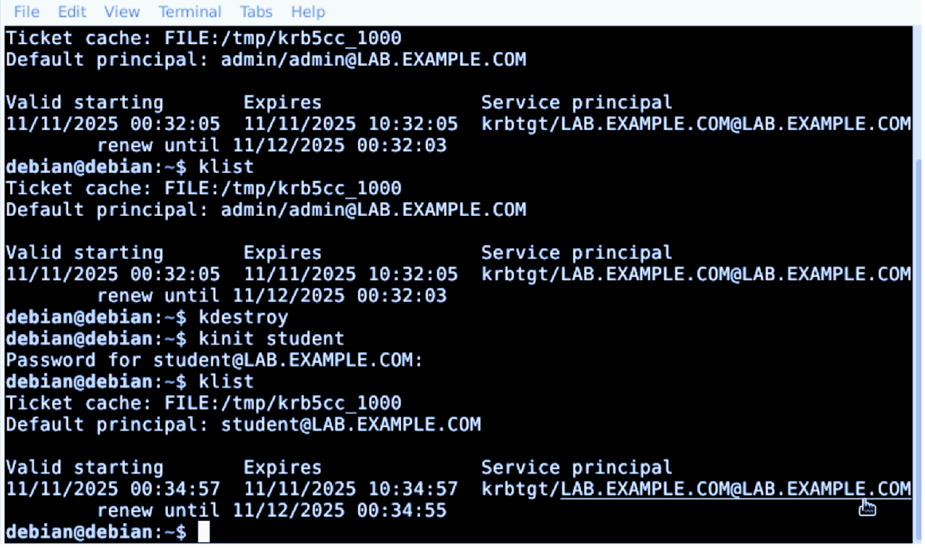
```
  GNU nano 5.4                          /etc/hosts *
127.0.0.1       localhost
127.0.1.1       debian

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

192.168.64.12 server.lab.example.com server-kdc
192.168.64.12 client.lab.example.com client
```



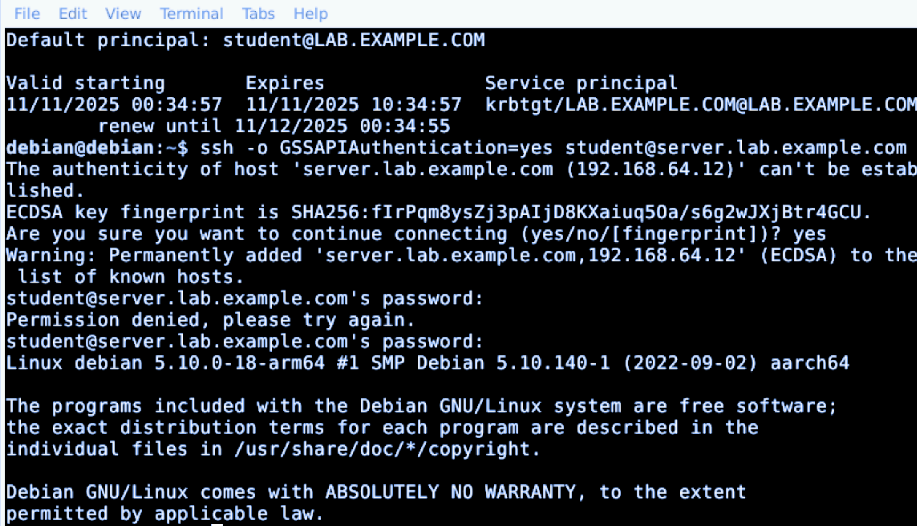
```
Package configuration

              Configuring Kerberos Authentication
  When users attempt to use Kerberos and specify a principal or user name
  without specifying what administrative Kerberos realm that principal
  belongs to, the system appends the default realm.  The default realm may
  also be used as the realm of a Kerberos service running on the local
  machine.  Often, the default realm is the uppercase version of the local
  DNS domain.

  Default Kerberos version 5 realm:

  LAB.EXAMPLE.COM
```



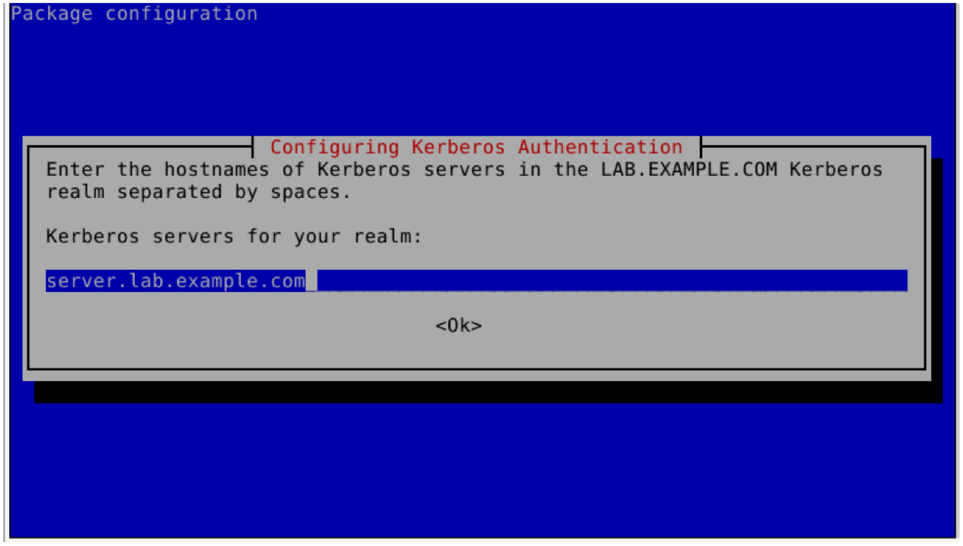
```
debian@debian:~$  ^[[200~sudo nano /etc/krb5.conf~^C
debian@debian:~$ ^C
debian@debian:~$ sudo nano /etc/krb5.conf
debian@debian:~$ kinit admin/admin
Password for admin/admin@LAB.EXAMPLE.COM:
debian@debian:~$ kinit admin/admin
Password for admin/admin@LAB.EXAMPLE.COM:
debian@debian:~$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: admin/admin@LAB.EXAMPLE.COM

Valid starting       Expires              Service principal
11/11/2025 00:32:05  11/11/2025 10:32:05  krbtgt/LAB.EXAMPLE.COM@LAB.EXAMPLE.COM
        renew until 11/12/2025 00:32:03
debian@debian:~$ 
```



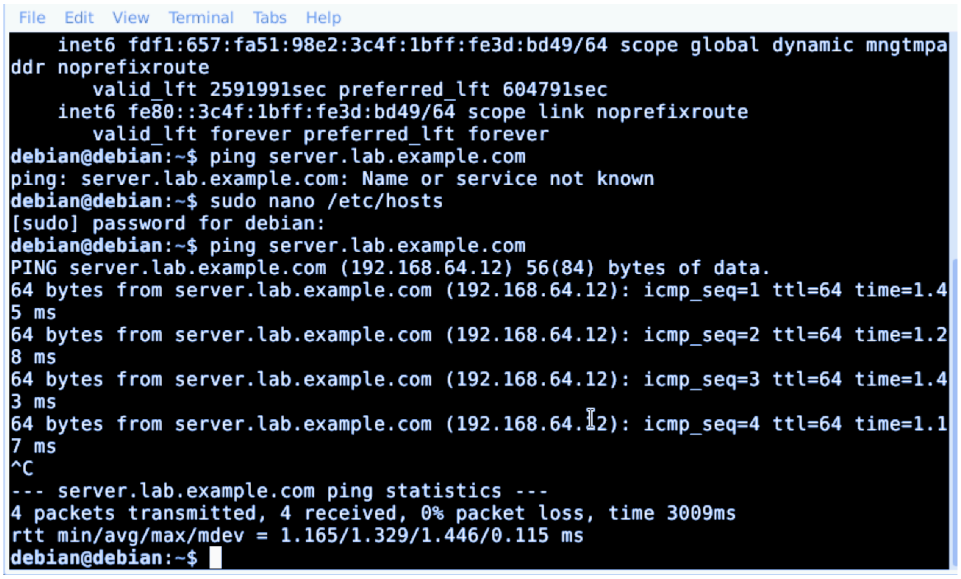
```
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: admin/admin@LAB.EXAMPLE.COM

Valid starting       Expires              Service principal
11/11/2025 00:32:05  11/11/2025 10:32:05  krbtgt/LAB.EXAMPLE.COM@LAB.EXAMPLE.COM
        renew until 11/12/2025 00:32:03
debian@debian:~$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: admin/admin@LAB.EXAMPLE.COM

Valid starting       Expires              Service principal
11/11/2025 00:32:05  11/11/2025 10:32:05  krbtgt/LAB.EXAMPLE.COM@LAB.EXAMPLE.COM
        renew until 11/12/2025 00:32:03
debian@debian:~$ kdestroy
debian@debian:~$ kinit student
Password for student@LAB.EXAMPLE.COM:
debian@debian:~$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: student@LAB.EXAMPLE.COM

Valid starting       Expires              Service principal
11/11/2025 00:34:57  11/11/2025 10:34:57  krbtgt/LAB.EXAMPLE.COM@LAB.EXAMPLE.COM
        renew until 11/12/2025 00:34:55
debian@debian:~$ 
```



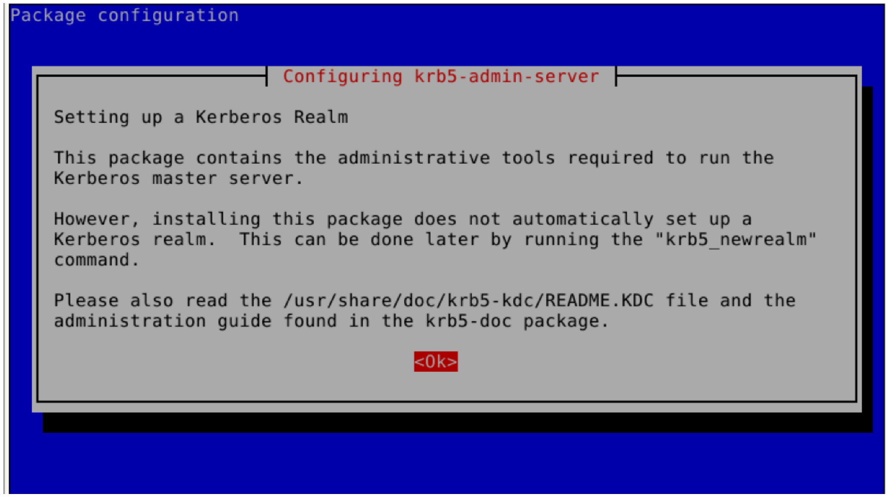
```
Default principal: student@LAB.EXAMPLE.COM

Valid starting       Expires              Service principal
11/11/2025 00:34:57  11/11/2025 10:34:57  krbtgt/LAB.EXAMPLE.COM@LAB.EXAMPLE.COM
        renew until 11/12/2025 00:34:55
debian@debian:~$ ssh -o GSSAPIAuthentication=yes student@server.lab.example.com
The authenticity of host 'server.lab.example.com (192.168.64.12)' can't be estab
lished.
ECDSA key fingerprint is SHA256:fIrPqm8ysZj3pAIjD8KXaiuq5Oa/s6g2wJXjBtr4GCU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server.lab.example.com,192.168.64.12' (ECDSA) to the
 list of known hosts.
student@server.lab.example.com's password:
Permission denied, please try again.
student@server.lab.example.com's password:
Linux debian 5.10.0-18-arm64 #1 SMP Debian 5.10.140-1 (2022-09-02) aarch64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```



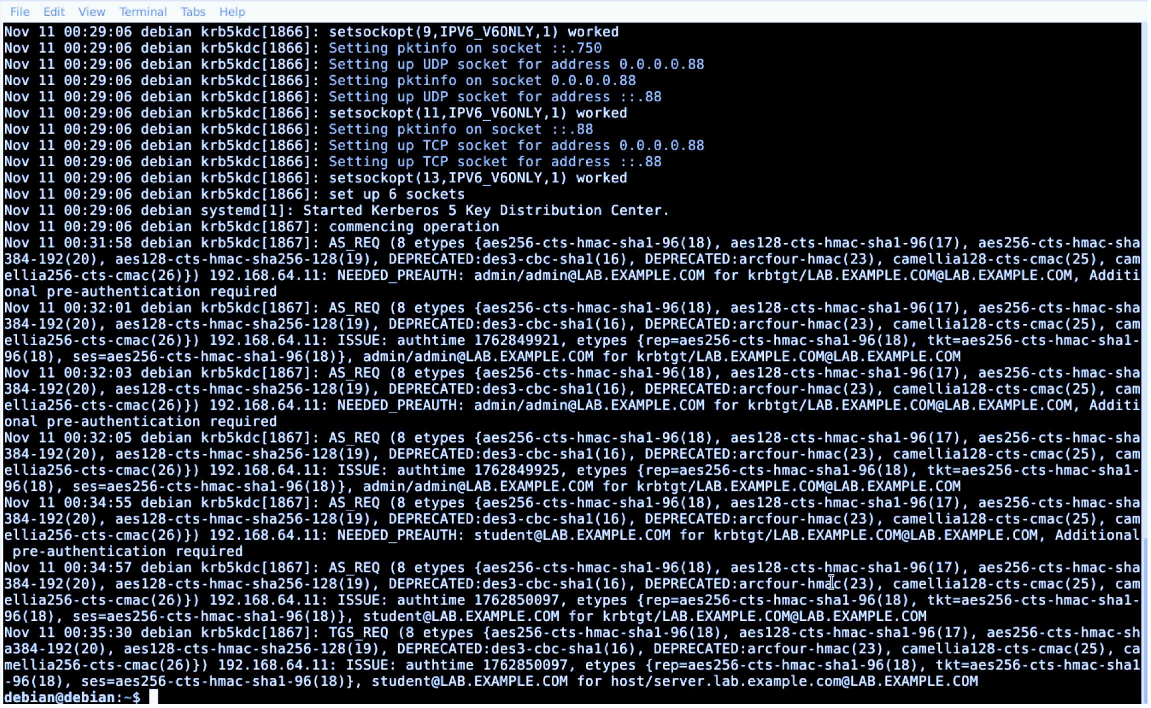
```
Package configuration

            Configuring Kerberos Authentication
 Enter the hostnames of Kerberos servers in the LAB.EXAMPLE.COM Kerberos
 realm separated by spaces.

 Kerberos servers for your realm:

 server.lab.example.com

                               <Ok>
```



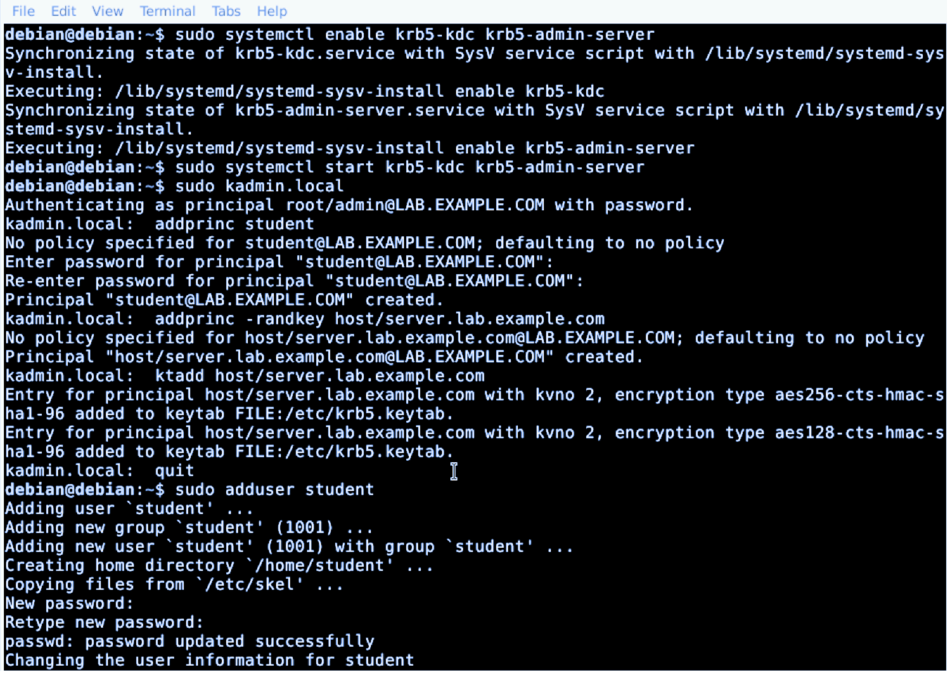
```
    inet6 fdf1:657:fa51:98e2:3c4f:1bff:fe3d:bd49/64 scope global dynamic mngtmpa
ddr noprefixroute
       valid_lft 2591991sec preferred_lft 604791sec
    inet6 fe80::3c4f:1bff:fe3d:bd49/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
debian@debian:~$ ping server.lab.example.com
ping: server.lab.example.com: Name or service not known
debian@debian:~$ sudo nano /etc/hosts
[sudo] password for debian:
debian@debian:~$ ping server.lab.example.com
PING server.lab.example.com (192.168.64.12) 56(84) bytes of data.
64 bytes from server.lab.example.com (192.168.64.12): icmp_seq=1 ttl=64 time=1.4
5 ms
64 bytes from server.lab.example.com (192.168.64.12): icmp_seq=2 ttl=64 time=1.2
8 ms
64 bytes from server.lab.example.com (192.168.64.12): icmp_seq=3 ttl=64 time=1.4
3 ms
64 bytes from server.lab.example.com (192.168.64.12): icmp_seq=4 ttl=64 time=1.1
7 ms
^C
--- server.lab.example.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3009ms
rtt min/avg/max/mdev = 1.165/1.329/1.446/0.115 ms
debian@debian:~$
```



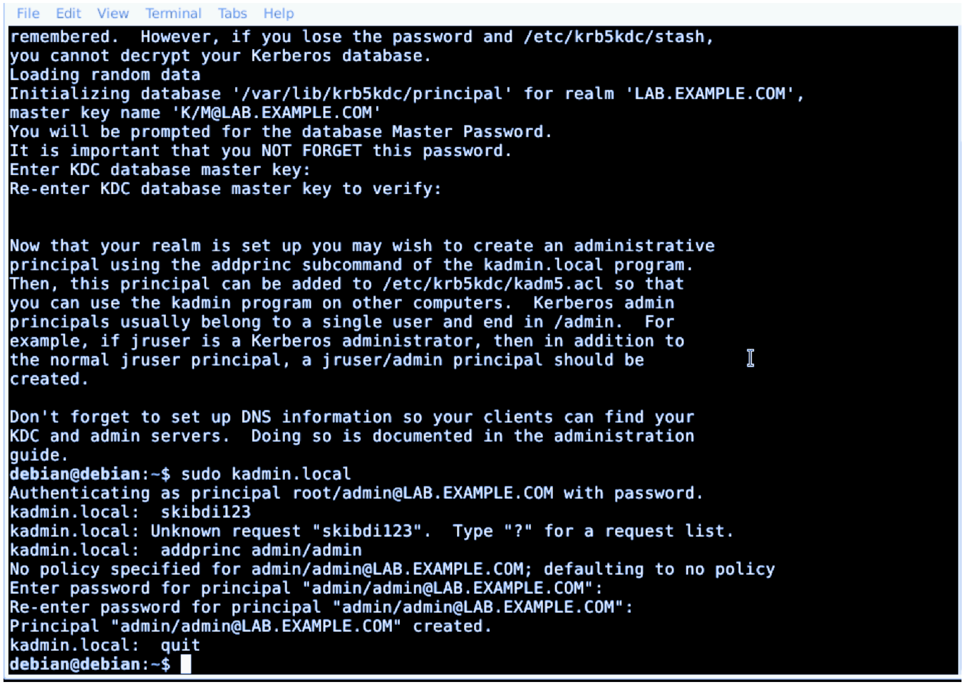
```
Package configuration

                Configuring krb5-admin-server

 Setting up a Kerberos Realm

 This package contains the administrative tools required to run the
 Kerberos master server.

 However, installing this package does not automatically set up a
 Kerberos realm.  This can be done later by running the "krb5_newrealm"
 command.

 Please also read the /usr/share/doc/krb5-kdc/README.KDC file and the
 administration guide found in the krb5-doc package.

                               <Ok>
```



```
Nov 11 00:29:06 debian krb5kdc[1866]: setsockopt(9,IPV6_V6ONLY,1) worked
Nov 11 00:29:06 debian krb5kdc[1866]: Setting pktinfo on socket ::.750
Nov 11 00:29:06 debian krb5kdc[1866]: Setting up UDP socket for address 0.0.0.0.88
Nov 11 00:29:06 debian krb5kdc[1866]: Setting pktinfo on socket 0.0.0.0.88
Nov 11 00:29:06 debian krb5kdc[1866]: Setting up UDP socket for address ::.88
Nov 11 00:29:06 debian krb5kdc[1866]: setsockopt(11,IPV6_V6ONLY,1) worked
Nov 11 00:29:06 debian krb5kdc[1866]: Setting pktinfo on socket ::.88
Nov 11 00:29:06 debian krb5kdc[1866]: Setting up TCP socket for address 0.0.0.0.88
Nov 11 00:29:06 debian krb5kdc[1866]: Setting up TCP socket for address ::.88
Nov 11 00:29:06 debian krb5kdc[1866]: setsockopt(13,IPV6_V6ONLY,1) worked
Nov 11 00:29:06 debian krb5kdc[1866]: set up 6 sockets
Nov 11 00:29:06 debian systemd[1]: Started Kerberos 5 Key Distribution Center.
Nov 11 00:29:06 debian krb5kdc[1867]: commencing operation
Nov 11 00:31:58 debian krb5kdc[1867]: AS_REQ (8 etypes {aes256-cts-hmac-sha1-96(18), aes128-cts-hmac-sha1-96(17), aes256-cts-hmac-sha
384-192(20), aes128-cts-hmac-sha256-128(19), DEPRECATED:des3-cbc-sha1(16), DEPRECATED:arcfour-hmac(23), camellia128-cts-cmac(25), cam
ellia256-cts-cmac(26)}) 192.168.64.11: NEEDED_PREAUTH: admin/admin@LAB.EXAMPLE.COM for krbtgt/LAB.EXAMPLE.COM@LAB.EXAMPLE.COM, Additi
onal pre-authentication required
Nov 11 00:32:01 debian krb5kdc[1867]: AS_REQ (8 etypes {aes256-cts-hmac-sha1-96(18), aes128-cts-hmac-sha1-96(17), aes256-cts-hmac-sha
384-192(20), aes128-cts-hmac-sha256-128(19), DEPRECATED:des3-cbc-sha1(16), DEPRECATED:arcfour-hmac(23), camellia128-cts-cmac(25), cam
ellia256-cts-cmac(26)}) 192.168.64.11: ISSUE: authtime 1762849921, etypes {rep=aes256-cts-hmac-sha1-96(18), tkt=aes256-cts-hmac-sha1-
96(18), ses=aes256-cts-hmac-sha1-96(18)}, admin/admin@LAB.EXAMPLE.COM for krbtgt/LAB.EXAMPLE.COM@LAB.EXAMPLE.COM
Nov 11 00:32:03 debian krb5kdc[1867]: AS_REQ (8 etypes {aes256-cts-hmac-sha1-96(18), aes128-cts-hmac-sha1-96(17), aes256-cts-hmac-sha
384-192(20), aes128-cts-hmac-sha256-128(19), DEPRECATED:des3-cbc-sha1(16), DEPRECATED:arcfour-hmac(23), camellia128-cts-cmac(25), cam
ellia256-cts-cmac(26)}) 192.168.64.11: NEEDED_PREAUTH: admin/admin@LAB.EXAMPLE.COM for krbtgt/LAB.EXAMPLE.COM@LAB.EXAMPLE.COM, Additi
onal pre-authentication required
Nov 11 00:32:05 debian krb5kdc[1867]: AS_REQ (8 etypes {aes256-cts-hmac-sha1-96(18), aes128-cts-hmac-sha1-96(17), aes256-cts-hmac-sha
384-192(20), aes128-cts-hmac-sha256-128(19), DEPRECATED:des3-cbc-sha1(16), DEPRECATED:arcfour-hmac(23), camellia128-cts-cmac(25), cam
ellia256-cts-cmac(26)}) 192.168.64.11: ISSUE: authtime 1762849925, etypes {rep=aes256-cts-hmac-sha1-96(18), tkt=aes256-cts-hmac-sha1-
96(18), ses=aes256-cts-hmac-sha1-96(18)}, admin/admin@LAB.EXAMPLE.COM for krbtgt/LAB.EXAMPLE.COM@LAB.EXAMPLE.COM
Nov 11 00:34:55 debian krb5kdc[1867]: AS_REQ (8 etypes {aes256-cts-hmac-sha1-96(18), aes128-cts-hmac-sha1-96(17), aes256-cts-hmac-sha
384-192(20), aes128-cts-hmac-sha256-128(19), DEPRECATED:des3-cbc-sha1(16), DEPRECATED:arcfour-hmac(23), camellia128-cts-cmac(25), cam
ellia256-cts-cmac(26)}) 192.168.64.11: NEEDED_PREAUTH: student@LAB.EXAMPLE.COM for krbtgt/LAB.EXAMPLE.COM@LAB.EXAMPLE.COM, Additional
 pre-authentication required
Nov 11 00:34:57 debian krb5kdc[1867]: AS_REQ (8 etypes {aes256-cts-hmac-sha1-96(18), aes128-cts-hmac-sha1-96(17), aes256-cts-hmac-sha
384-192(20), aes128-cts-hmac-sha256-128(19), DEPRECATED:des3-cbc-sha1(16), DEPRECATED:arcfour-hmac(23), camellia128-cts-cmac(25), cam
ellia256-cts-cmac(26)}) 192.168.64.11: ISSUE: authtime 1762850097, etypes {rep=aes256-cts-hmac-sha1-96(18), tkt=aes256-cts-hmac-sha1-
96(18), ses=aes256-cts-hmac-sha1-96(18)}, student@LAB.EXAMPLE.COM for krbtgt/LAB.EXAMPLE.COM@LAB.EXAMPLE.COM
Nov 11 00:35:30 debian krb5kdc[1867]: TGS_REQ (8 etypes {aes256-cts-hmac-sha1-96(18), aes128-cts-hmac-sha1-96(17), aes256-cts-hmac-sh
a384-192(20), aes128-cts-hmac-sha256-128(19), DEPRECATED:des3-cbc-sha1(16), DEPRECATED:arcfour-hmac(23), camellia128-cts-cmac(25), ca
mellia256-cts-cmac(26)}) 192.168.64.11: ISSUE: authtime 1762850097, etypes {rep=aes256-cts-hmac-sha1-96(18), tkt=aes256-cts-hmac-sha1
-96(18), ses=aes256-cts-hmac-sha1-96(18)}, student@LAB.EXAMPLE.COM for host/server.lab.example.com@LAB.EXAMPLE.COM
debian@debian:~$
```



```
debian@debian:~$ sudo systemctl enable krb5-kdc krb5-admin-server
Synchronizing state of krb5-kdc.service with SysV service script with /lib/systemd/systemd-sys
v-install.
Executing: /lib/systemd/systemd-sysv-install enable krb5-kdc
Synchronizing state of krb5-admin-server.service with SysV service script with /lib/systemd/sy
stemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable krb5-admin-server
debian@debian:~$ sudo systemctl start krb5-kdc krb5-admin-server
debian@debian:~$ sudo kadmin.local
Authenticating as principal root/admin@LAB.EXAMPLE.COM with password.
kadmin.local:  addprinc student
No policy specified for student@LAB.EXAMPLE.COM; defaulting to no policy
Enter password for principal "student@LAB.EXAMPLE.COM":
Re-enter password for principal "student@LAB.EXAMPLE.COM":
Principal "student@LAB.EXAMPLE.COM" created.
kadmin.local:  addprinc -randkey host/server.lab.example.com
No policy specified for host/server.lab.example.com@LAB.EXAMPLE.COM; defaulting to no policy
Principal "host/server.lab.example.com@LAB.EXAMPLE.COM" created.
kadmin.local:  ktadd host/server.lab.example.com
Entry for principal host/server.lab.example.com with kvno 2, encryption type aes256-cts-hmac-s
ha1-96 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/server.lab.example.com with kvno 2, encryption type aes128-cts-hmac-s
ha1-96 added to keytab FILE:/etc/krb5.keytab.
kadmin.local:  quit
debian@debian:~$ sudo adduser student
Adding user `student' ...
Adding new group `student' (1001) ...
Adding new user `student' (1001) with group `student' ...
Creating home directory `/home/student' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for student
```



```
remembered.  However, if you lose the password and /etc/krb5kdc/stash,
you cannot decrypt your Kerberos database.
Loading random data
Initializing database '/var/lib/krb5kdc/principal' for realm 'LAB.EXAMPLE.COM',
master key name 'K/M@LAB.EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:


Now that your realm is set up you may wish to create an administrative
principal using the addprinc subcommand of the kadmin.local program.
Then, this principal can be added to /etc/krb5kdc/kadm5.acl so that
you can use the kadmin program on other computers.  Kerberos admin
principals usually belong to a single user and end in /admin.  For
example, if jruser is a Kerberos administrator, then in addition to
the normal jruser principal, a jruser/admin principal should be
created.

Don't forget to set up DNS information so your clients can find your
KDC and admin servers.  Doing so is documented in the administration
guide.
debian@debian:~$ sudo kadmin.local
Authenticating as principal root/admin@LAB.EXAMPLE.COM with password.
kadmin.local:  skibdi123
kadmin.local: Unknown request "skibdi123".  Type "?" for a request list.
kadmin.local:  addprinc admin/admin
No policy specified for admin/admin@LAB.EXAMPLE.COM; defaulting to no policy
Enter password for principal "admin/admin@LAB.EXAMPLE.COM":
Re-enter password for principal "admin/admin@LAB.EXAMPLE.COM":
Principal "admin/admin@LAB.EXAMPLE.COM" created.
kadmin.local:  quit
debian@debian:~$
```

Conclusion:

In this experiment, the Kerberos authentication mechanism was successfully implemented and tested to understand its key components and workflow. A Kerberos Key Distribution Center (KDC) was set up on the server, and both client and service configurations were completed to enable secure authentication. Through the use of tickets, Ticket Granting Tickets (TGT), and the KDC, the authentication process was verified without requiring direct password transmission. The successful SSH login using Kerberos confirmed that the setup was functioning correctly. Overall, the experiment demonstrated how Kerberos ensures secure, ticket-based authentication and centralized access control in a networked environment.