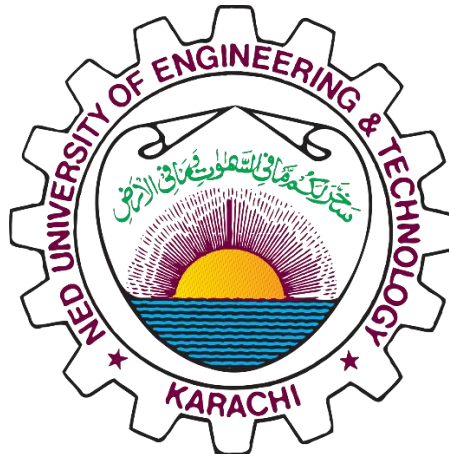


# **CT-541 – NETWORK SECURITY**

MS-IS 004 2019/20 – Evening Fall 2019

## **CT-541 NS Assignment-04**

**Kerberos KDC, Server, Client, SSH Connection**



**STUDENT NAME: MUHAMMAD UMAR TARIQ**

**ROLL NUMBER : IS 004 2019/20**

**COURSE INSTRUCTOR: Dr. Mubashir M Khan**

**DEPARTMENT OF COMPUTER SCIENCE &  
INFORMATION TECHNOLOGY**

**NED UNIVERSITY, KARACHI CAMPUS**

Kerberos authentication has three components KDC, server and client that will connect using Kerberos authentication.

Each Server has to be assigned a fully qualified domain name.

**VM1 is the server in our case, so we change VM1 hostname.**

**VM2 is going to be a Kerberos client that will authenticate from VM1.**

```
[08/14/20]seed@VM:~$ hostname
VM
[08/14/20]seed@VM:~$ hostname -f
VM
[08/14/20]seed@VM:~$
```

To change the FQDN:

**sudo hostnamectl set-hostname kdc.example.com**

```
[08/14/20]seed@VM:~$ sudo hostnamectl set-hostname kdc.example.com
[08/14/20]seed@VM:~$ hostname
kdc.example.com
[08/14/20]seed@VM:~$
```

**Since we are implementing the default MIT Kerberos system, the time synchronization must be less than equal to 5 minutes between every machine we want to authenticate. So set the time to local time**



**We also have to add a manual entry for the dns lookups, if we do not have dns server available we can do**

**sudo gedit /etc/hosts**

If the server does not already have a FQDN assigned to it and DNS services are not available, name resolution can be implemented by editing the local hosts file (typically this is located in /etc) on each server and client, adding the following line:

127.0.0.1 linuxwork.example.com localhost linuxwork

**Hosts File ON VM1 [KDC SERVER]:**

```
127.0.0.1      localhost
127.0.1.1      VM
192.168.56.102 clientvm2.example.com clientvm2
192.168.56.101 kdc.example.com      kdc
```

**Hosts File ON VM2 [CLIENT of Kerberos Authentication]:**

```
127.0.0.1      localhost
127.0.1.1      VM
192.168.56.102 clientvm2.example.com clientvm2
192.168.56.101 kdc.example.com      kdc
```

Open the file as sudo , and save, and then restart vm after the time settings change and hosts change.

Now we need to perform DNS Lookup to verify that our KDC server has a FQDN assigned to it.

### **KDC SERVER And Kerberos Client NSLookup**

```
[08/14/20]seed@kdc:~$ nslookup clientvm2.example.com
Server:          127.0.1.1
Address:         127.0.1.1#53

** server can't find clientvm2.example.com: NXDOMAIN

[08/14/20]seed@kdc:~$ nslookup kdc.example.com
Server:          127.0.1.1
Address:         127.0.1.1#53

** server can't find kdc.example.com: NXDOMAIN

[08/14/20]seed@kdc:~$
```

For reverse domain lookups we will set rdns variable to false on Kerberos client machines krb5.conf files.

To verify connectivity, ping each hosts FQDN.

Before pinging, add the Host-Only Mode Private ip addresses to the hosts file of the client.

Both of the client and the KDC server.

```
[08/14/20]seed@clientvm2:~$ ping clientvm2.example.com
PING clientvm2.example.com (192.168.56.102) 56(84) bytes of data.
64 bytes from clientvm2.example.com (192.168.56.102): icmp_seq=1 ttl=64 time=0.013 ms
64 bytes from clientvm2.example.com (192.168.56.102): icmp_seq=2 ttl=64 time=0.018 ms
^C
--- clientvm2.example.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 0.013/0.015/0.018/0.004 ms
[08/14/20]seed@clientvm2:~$ ping kdc.example.com
PING kdc.example.com (192.168.56.101) 56(84) bytes of data.
64 bytes from kdc.example.com (192.168.56.101): icmp_seq=1 ttl=64 time=0.247 ms
64 bytes from kdc.example.com (192.168.56.101): icmp_seq=2 ttl=64 time=0.284 ms
64 bytes from kdc.example.com (192.168.56.101): icmp_seq=3 ttl=64 time=0.226 ms
^C
--- kdc.example.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 0.226/0.252/0.284/0.027 ms
[08/14/20]seed@clientvm2:~$
```

On KDC Server VM, run:

`sudo apt-get update`

`sudo apt-get install aptitude`

`sudo aptitude install krb5-kdc krb5-admin-server`

for dependencies issue, first select no and then select Y for downgrading the dependencies.

Enter the domain (Realm):

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:

EXAMPLE.COM

<Ok>

Enter FQDN of KDC:

Package configuration

### Configuring Kerberos Authentication

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

Kerberos servers for your realm:

kdc.example.com

<Ok>

Enter the password changing admin server for Our Realm:

Package configuration

### Configuring Kerberos Authentication

Enter the hostname of the administrative (password changing) server for the EXAMPLE.COM Kerberos realm.

Administrative server for your Kerberos realm:

kdc.example.com

<Ok>

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>

After the package is built check the krb5.conf file and check the FQDN:

```
[libdefaults]
    default_realm = EXAMPLE.COM

# 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
```

And also check the realm, if this is not properly configured, the clients will not authenticate.

```
[realms]
    EXAMPLE.COM = {
        kdc = kdc.example.com
        admin_server = kdc.example.com
    }
    ATHENA.MIT.EDU = {
        kdc = kerberos.mit.edu:88
        kdc = kerberos-1.mit.edu:88
        kdc = kerberos-2.mit.edu:88
        admin_server = kerberos.mit.edu
        default_domain = mit.edu
    }
    MEDIA-LAB.MIT.EDU = {
        kdc = kerberos.media.mit.edu
        admin_server = kerberos.media.mit.edu
    }
    ZONE.MIT.EDU = {
        kdc = casio.mit.edu
        kdc = seiko.mit.edu
        admin_server = casio.mit.edu
    }
    MOOSE.MIT.EDU = {
```



Now we will create a new **KRB5** realm:

With the command:

```
sudo krb5_newrealm
```

```
[08/15/20]seed@kdc:~$ sudo krb5_newrealm
This script should be run on the master KDC/admin server to initialize
a Kerberos realm. It will ask you to type in a master key password.
This password will be used to generate a key that is stored in
/etc/krb5kdc/stash. You should try to remember this password, but it
is much more important that it be a strong password than that it be
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 'EXAMPLE.COM',
master key name 'K/M@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:
```

```
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.
```

Now we are going to add the principal

```
sudo kadmin.local
```

```
[08/15/20]seed@kdc:~$ sudo kadmin.local
Authenticating as principal root/admin@EXAMPLE.COM with password.
kadmin.local: addprinc seed/admin
WARNING: no policy specified for seed/admin@EXAMPLE.COM; defaulting to no policy
Enter password for principal "seed/admin@EXAMPLE.COM":
Re-enter password for principal "seed/admin@EXAMPLE.COM":
Principal "seed/admin@EXAMPLE.COM" created.
```

The admin user needs to have proper Access Control List Permissions, so we edit kadm5.acl file

```
sudo gedit /etc/krb5kdc/kadm5.acl
```

```
# This file is the access control list for krb5 administration.
# When this file is edited run /etc/init.d/krb5-admin-server restart to activate
# One common way to set up Kerberos administration is to allow any principal
# ending in /admin is given full administrative rights.
# To enable this, uncomment the following line:

# Any principal in the ATHENA.MIT.EDU realm with an admin instance has all administrative privileges.
*seed/admin@example.com *
```

Restart Kerberos server for ACL to take effect.

sudo systemctl restart krb5-admin-server.service

```
[08/15/20]seed@kdc:~$ kinit seed/admin
Password for seed/admin@EXAMPLE.COM:
[08/15/20]seed@kdc:~$
```

Use klist utility to check the TGT:

```
[08/15/20]seed@kdc:~$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: seed/admin@EXAMPLE.COM

Valid starting      Expires            Service principal
08/15/2020 13:34:43  08/15/2020 23:34:43  krbtgt/EXAMPLE.COM@EXAMPLE.COM
        renew until 08/16/2020 13:34:39
[08/15/20]seed@kdc:~$
```

## Kerberos Client Installation and Configuration :

```
sudo aptitude install krb5-user libpam-krb5 libpam-ccreds auth-client-config
```

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:

EXAMPLE.COM

<0k>

Package configuration

### Configuring Kerberos Authentication

Enter the hostname of the administrative (password changing) server for the EXAMPLE.COM Kerberos realm.

Administrative server for your Kerberos realm:

kdc.example.com

<Ok>

We are on a Kerberos Client now, so we will obtain the service from Kerberos KDC:

```
[08/15/20]seed@clientvm2:~$ hostname  
clientvm2.example.com  
[08/15/20]seed@clientvm2:~$ hostname -f  
clientvm2.example.com  
[08/15/20]seed@clientvm2:~$
```

Here we have to write the FQDN i.e domain name in capital, otherwise it will give a credential error. The client should obtain the admin configuration from the access control list settings and then generate a ticket.

```
[08/15/20]seed@clientvm2:~$ kinit seed/admin@EXAMPLE.COM  
Password for seed/admin@EXAMPLE.COM:  
[08/15/20]seed@clientvm2:~$
```

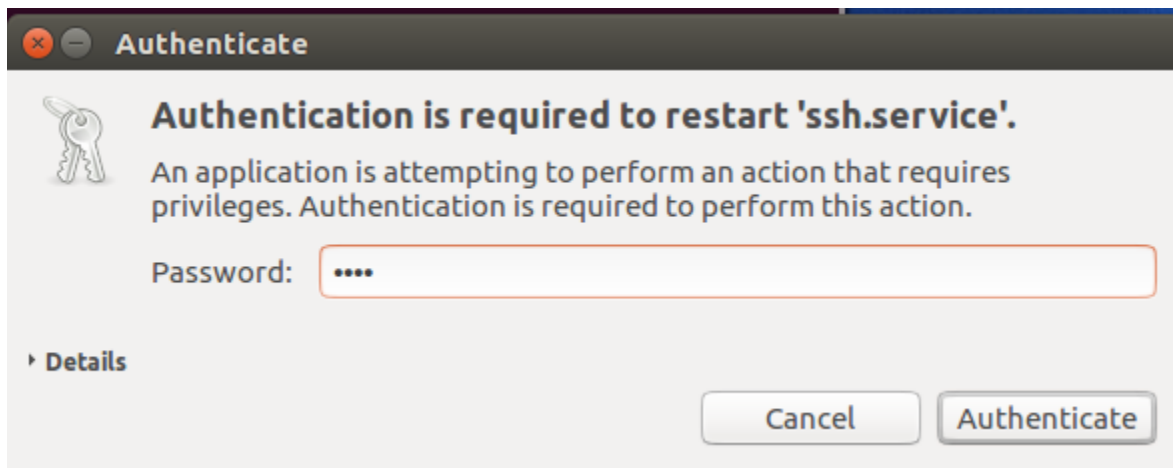
```
[08/15/20]seed@clientvm2:~$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: seed/admin@EXAMPLE.COM

Valid starting    Expires          Service principal
08/15/2020 13:47:27  08/15/2020 23:47:27  krbtgt/EXAMPLE.COM@EXAMPLE.COM
        renew until 08/16/2020 13:46:21
```

On the SSH server **[KDC]** and the **Kerberos** client, edit /etc/ssh/sshd\_config

```
# GSSAPI options
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
```

Do service ssh restart on the server and client:



We also have to enable Kerberos Authentication on the server ssh configuration

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

# GSSAPI options
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
```

We use the SSH version 1 to connect to our KDC server.

```
[08/15/20]seed@kdc:~$ sudo kadmin.local
Authenticating as principal root/admin@EXAMPLE.COM with password.
kadmin.local: addprinc root/admin
WARNING: no policy specified for root/admin@EXAMPLE.COM; defaulting
to no policy
Enter password for principal "root/admin@EXAMPLE.COM":
Re-enter password for principal "root/admin@EXAMPLE.COM":
Principal "root/admin@EXAMPLE.COM" created.
kadmin.local: addprinc -randkey host/kdc.example.com
WARNING: no policy specified for host/kdc.example.com@EXAMPLE.COM;
defaulting to no policy
Principal "host/kdc.example.com@EXAMPLE.COM" created.
kadmin.local: ktadd host/kdc.example.com
Entry for principal host/kdc.example.com with kvno 2, encryption ty
pe aes256-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/kdc.example.com with kvno 2, encryption ty
pe arcfour-hmac added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/kdc.example.com with kvno 2, encryption ty
pe des3-cbc-sha1 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/kdc.example.com with kvno 2, encryption ty
pe des-cbc-crc added to keytab FILE:/etc/krb5.keytab.
kadmin.local:
```

```
[08/15/20]seed@kdc:~$ sudo nano /etc/krb5kdc/kadm5.acl
[08/15/20]seed@kdc:~$ sudo systemctl restart krb5-admin-server.serv
ice
```

To make ssh connection we have to add a non-admin user in server and client. And add the principals for them.

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

# GSSAPI options
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
```

We have to modify these parameters and restart sshd service everytime. So that when we use SSH connection we will not have to enter password everytime.

**The default versions use SSH ver. 1 but it is better to use SSH version2.**

```
[08/15/20]seed@clientvm2:~$ sudo kadmin
Authenticating as principal root/admin@EXAMPLE.COM with password.
Password for root/admin@EXAMPLE.COM:
kadmin: addprinc -randkey host/clientvm2.example.com
WARNING: no policy specified for host/clientvm2.example.com@EXAMPLE.COM; defaulting to no policy
add_principal: Principal or policy already exists while creating "host/clientvm2.example.com@EXAMPLE.COM".
kadmin: ktadd host/clientvm2.example.com
Entry for principal host/clientvm2.example.com with kvno 3, encryption type aes256-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/clientvm2.example.com with kvno 3, encryption type arcfour-hmac added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/clientvm2.example.com with kvno 3, encryption type des3-cbc-sha1 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/clientvm2.example.com with kvno 3, encryption type des-cbc-crc added to keytab FILE:/etc/krb5.keytab.
```

```
[08/15/20]seed@kdc:~$ sudo useradd -m -s /bin/bash umar
[08/15/20]seed@kdc:~$ sudo kadmin.local
Authenticating as principal root/admin@EXAMPLE.COM with password.
kadmin.local: addprinc umar
WARNING: no policy specified for umar@EXAMPLE.COM; defaulting to no policy
Enter password for principal "umar@EXAMPLE.COM":
Re-enter password for principal "umar@EXAMPLE.COM":
Principal "umar@EXAMPLE.COM" created.
kadmin.local:
```

```
[08/15/20]seed@kdc:~$ sudo nano /etc/ssh/sshd_config
[08/15/20]seed@kdc:~$ sudo systemctl restart sshd
[08/15/20]seed@kdc:~$
```

```
[08/15/20]seed@clientvm2:~$ sudo useradd -m -s /bin/bash umar
[08/15/20]seed@clientvm2:~$ sudo su - umar
umar@clientvm2:~$
```



```

umar@clientvm2:~$ kinit
Password for umar@EXAMPLE.COM:
umar@clientvm2:~$ klist
Ticket cache: FILE:/tmp/krb5cc_1002
Default principal: umar@EXAMPLE.COM

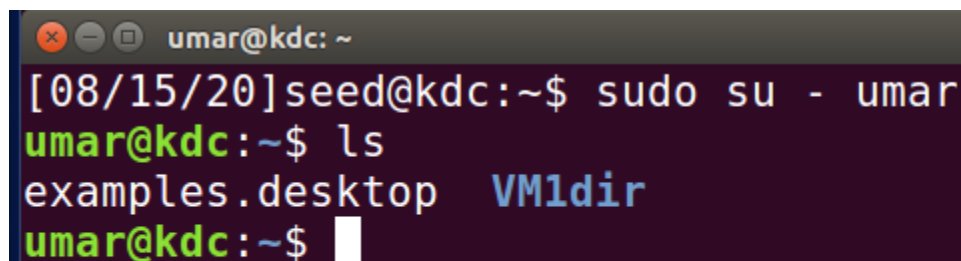
Valid starting      Expires            Service principal
08/15/2020 16:02:32  08/16/2020 02:02:32  krbtgt/EXAMPLE.COM@EXAMPLE.COM
        renew until 08/16/2020 16:02:27
umar@clientvm2:~$ ssh kdc.example.com
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

3 packages can be updated.
0 updates are security updates.

Last login: Sat Aug 15 16:00:29 2020 from 192.168.56.102
umar@kdc:~$ ls
examples.desktop
umar@kdc:~$ cd Documents
-bash: cd: Documents: No such file or directory
umar@kdc:~$ ls
examples.desktop
umar@kdc:~$ mkdir VM1dir
umar@kdc:~$ cd VM1dir
umar@kdc:~/VM1dir$ ls
umar@kdc:~/VM1dir$

```



```

[08/15/20]seed@kdc:~$ sudo su - umar
umar@kdc:~$ ls
examples.desktop  VM1dir
umar@kdc:~$

```

```

umar@kdc:~/VM1dir$ Connection to kdc.example.com closed by remote host.
Connection to kdc.example.com closed.
umar@clientvm2:~$

```

We made a SSH connection from Kerberos Client, to the principal user we added in the Kerberos Server Virtual Machine.