Foundations of Information Assurance

Lab Assignment 4 Report

Team 20

Sonam Ghatode

Vishal Maurya

10.12.2019
CY5010

Index

Why is time synchronization important in Kerberos? What are the consequences of not synchronizing the clocks in realm?

When a client wants to get a service from a service server in Kerberos, he/she has to first send a request to the Authentication Server, which authenticates the clients and either gives a Ticket Granting Ticket(TGT) or denies the request(failed authentication). This ticket has a timestamp and is valid only till 24 hours from the time of issue. Although TGT timestamp is not that big issue, issue arises when client wants a service. Client sends this TGT to the Ticket Granting Server, which checks the TGT for Timestamp and either provides a Service Ticket or denies the request. The Service Ticket is valid only for 5 minutes. So, if the clocks in realm(VM and Container in our case) is not synchronized, we might be able to get the TGT but might be denied service if we failed to present the Service Ticket within the server time frame.

Screenshot of /etc/krb5kdc/kdc.conf :

```
root@server:~# cat /etc/krb5kdc/kdc.conf
[kdcdefaults]
    kdc_ports = 750,88

[realms]
    CY5010.COM = {
        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
        kdc_ports = 750,88
        max_life = 10h 0m 0s
        max_renewable_life = 7d 0h 0m 0s
        master_key_type = des3-hmac-sha1
        supported_enctypes = aes256-cts:normal arcfour-hmac:normal des3-hmac-sha1:normal des-cbc-crc:normal des:normal des:v4 des:norealm des:onlyrealm des:afs3
        default_principal_flags = +preauth
    }
root@server:~#
```

Significance of keytab file in Kerberos Authentication:

A keytab is a file containing pairs of Kerberos principals and encrypted keys (which are derived from the Kerberos password). Keytab file is to authenticate to various remote systems using Kerberos without entering a password.

Keytab files are commonly used to allow scripts to automatically authenticate using Kerberos, without requiring keyboard password input. The script is then able to use the acquired credentials to access files stored on a remote system. Anyone with read permission on a keytab file can use all the keys in the file. To prevent misuse, access permissions for any keytab files are restricted. This was the reason we were asked to change the permission of the keytab file to be readable and writable only by the root user.

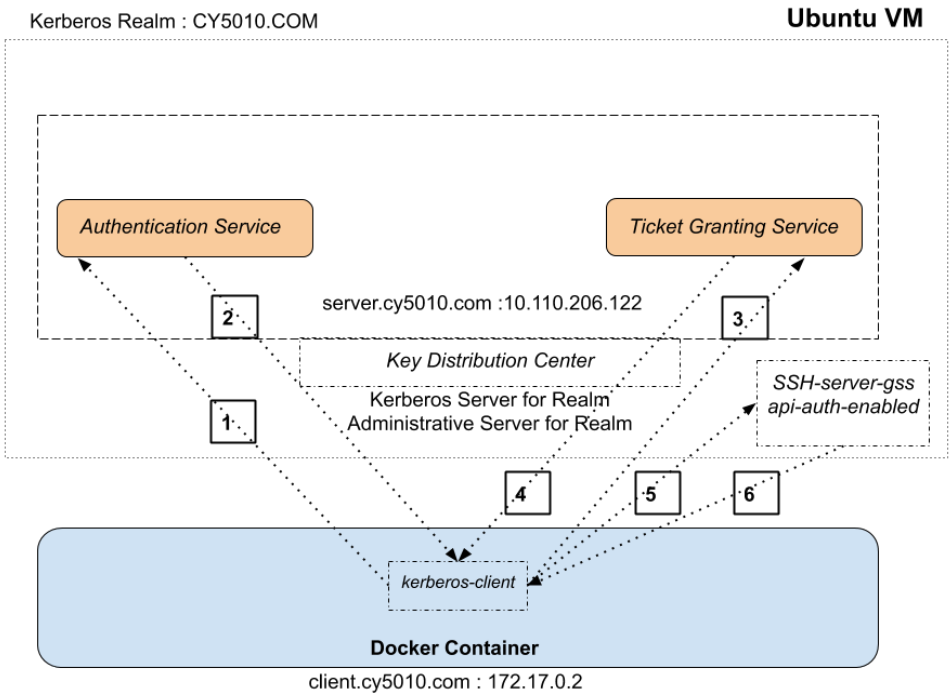
Attacks on Kerberos System:

There are two possible ways to attack Kerberos System:

1. If attacker can get his hands on the Service Ticket, he/she might use the ticket for availing any service. However, the context of the attacker is limited only to the service he got the Service Ticket for.
2. If the attacker gets his hands on the Ticket Granting Tickets(TGT), he has the power to get any service he wants from the Ticket Granting Server. In the best case, if the attacker has administrative privileges, he can issue a TGT valid for 10 years and continue exploiting the vulnerability.

Other attacks include getting the access of the Ticket Granting Server, which leaves the attacker with enormous power so as to affect other clients as well and also single Key Distribution Server is the single point of failure.

Working of Kerberos for this Lab:



The above diagram shows the working setup of kerbererizing ssh service to allow a user to authenticate ssh service using kerberos where the ssh-server, Kerberos KDC, Authentication Service are running on Ubuntu VM (server.cy5010.com) which will resolve to 10.110.206.122 due to local DNS resolution using /etc/hosts file. The client is a docker container running inside the VM which is used as the kerberos client (client.cy5010.com) which will resolves to 172.17.0.2.

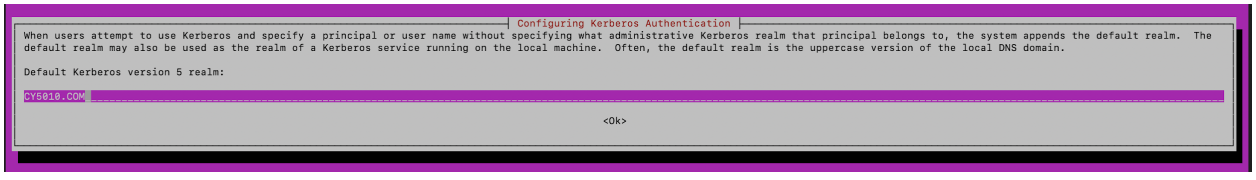
Steps involved in the Kerberos Authentication (Each box in the diagram with number):

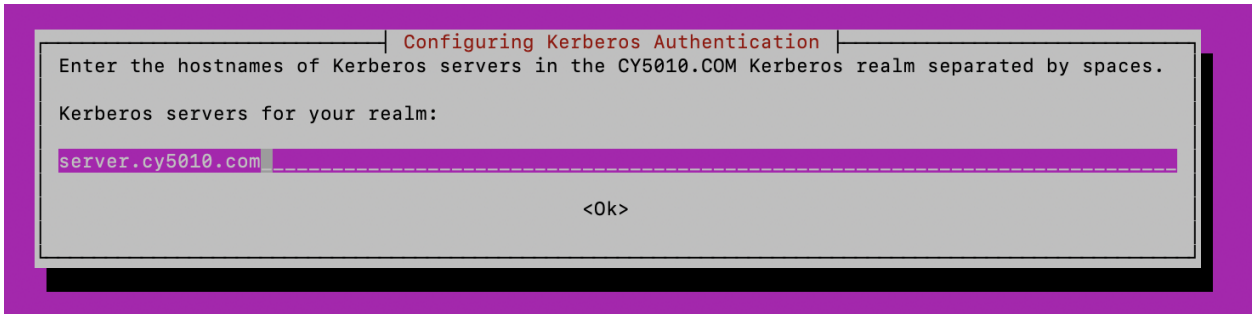
1. **AS_REQ** : Requesting Authentication Service for a TGT(Ticket Granting Ticket), so that we can use this TGT to request for a ticket to use ssh service later (**We make this request with kinit <username> command in our lab**).
2. **AS_REP** : The Authentication Service replies with a TGT and a session key and also mentions the validity of the ticket, which in our case is 24 hours. The Authentication service encrypts this response with the encryption key of Ticket Granting Service.
3. **TGS_REQ** : Now that we have a TGT we request the Ticket granting service for a ticket to allow us to use the resource(ssh-auth-service) we add a timestamp to this request and add our identity as the individual who is requesting access.
4. **TGS_REP** : The Ticket granting service replies with a ticket to allow us to use the ssh service only by encrypting it with the key of ssh service.
5. **AP_REQ** : We now request the resource we have are trying to access, we append our client IP and again add a timestamp of sending this request. This request is still encrypted with the key of SSH service.
6. On receiving this request, the SSH-Auth service decrypts the request and authenticates the user so that he/she can use the services provided by SSH service.

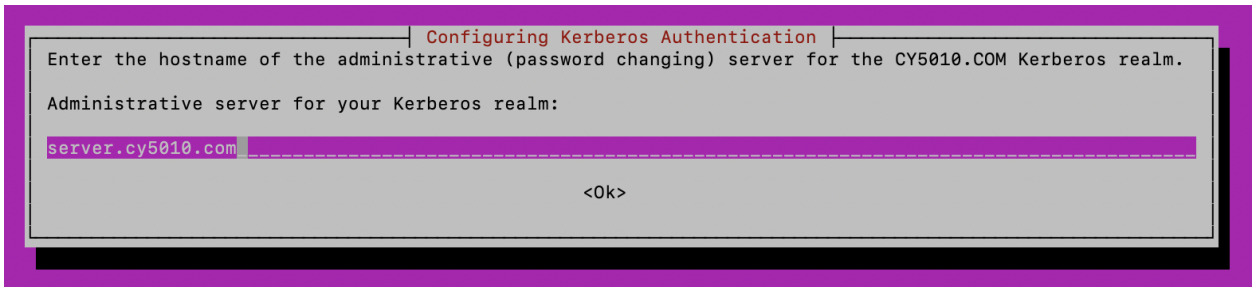
Part 1 - Kerberos Server Configuration in Ubuntu VM:

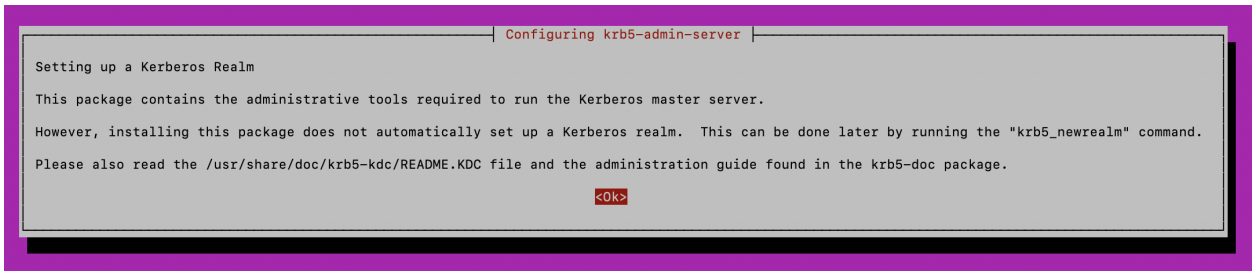
As per the setup guide we first install krb5-kdc krb5-admin-server on Ubuntu VM and assign realm, Kerberos Server for realm and Administrative Server for realm as server.cy5010.com.

```
user@ubuntu:~$ sudo apt-get install krb5-kdc krb5-admin-server
[sudo] password for user:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  krb5-config krb5-user libevent-2.1-6 libgssrpc4 libkadm5clnt-mit11 libkadm5srv-mit11 libkdb5-9
  libverto-libevent1 libverto1
Suggested packages:
  krb5-kpropd krb5-kdc-ldap krb5-doc
The following NEW packages will be installed:
  krb5-admin-server krb5-config krb5-kdc krb5-user libevent-2.1-6 libgssrpc4 libkadm5clnt-mit11
  libkadm5srv-mit11 libkdb5-9 libverto-libevent1 libverto1
0 upgraded, 11 newly installed, 0 to remove and 24 not upgraded.
Need to get 719 kB of archives.
After this operation, 2,649 kB of additional disk space will be used.
Do you want to continue? [Y/n] _
```









We then changed the hostname, pulled and ran the docker container (docker run -d sierraneu/kerberos) to use as client user requesting for tickets so as to get access using ssh authentication service.

We also got the IP of the container to add in /etc/hosts file for local dns resolution.

```
root@server:~# hostname
server.cy5010.com
root@server:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
6b8b82ddff73        sierraneu/kerberos   "/usr/sbin/sshd -D"   5 days ago          Up 5 days           22/tcp              nifty_brahmagupta
root@server:~# docker inspect 6b8b82ddff73 | grep "IPAddress"
            "SecondaryIPAddresses": null,
            "IPAddress": "172.17.0.2",
                    "IPAddress": "172.17.0.2",
root@server:~# ping 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.060 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.073 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.047 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.043 ms
64 bytes from 172.17.0.2: icmp_seq=5 ttl=64 time=0.073 ms
^X64 bytes from 172.17.0.2: icmp_seq=6 ttl=64 time=0.046 ms
^C
--- 172.17.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5049ms
rtt min/avg/max/mdev = 0.043/0.057/0.073/0.012 ms
root@server:~#
```

After setting up /etc/hosts file we checked if the dns resolution is working fine or not by pinging to the client from VM.

```
root@server:~# hostname
server.cy5010.com
root@server:~# ping client.cy5010.com
PING client.cy5010.com (172.17.0.2) 56(84) bytes of data.
64 bytes from client.cy5010.com (172.17.0.2): icmp_seq=1 ttl=64 time=0.040 ms
64 bytes from client.cy5010.com (172.17.0.2): icmp_seq=2 ttl=64 time=0.079 ms
64 bytes from client.cy5010.com (172.17.0.2): icmp_seq=3 ttl=64 time=0.042 ms
64 bytes from client.cy5010.com (172.17.0.2): icmp_seq=4 ttl=64 time=0.047 ms
^X^C
--- client.cy5010.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.040/0.052/0.079/0.015 ms
root@server:~#
```

Took the screenshot of kdc.conf file and attached above.

Part 2 - Kerberos Database Configuration:

Then we setup the kerberos database by creating new realm, Add a user principal for our user ‘vishalmaurya’, add a host principal for your server , and generate a keytab file for the server.

```
root@server:~# sudo kadmin.local
Authenticating as principal root/admin@CY5010.COM with password.
kadmin.local:  addprinc vishalmaurya
WARNING: no policy specified for vishalmaurya@CY5010.COM; defaulting to no policy
Enter password for principal "vishalmaurya@CY5010.COM":
Re-enter password for principal "vishalmaurya@CY5010.COM":
Principal "vishalmaurya@CY5010.COM" created.
kadmin.local:  addprinc -randkey host/server.cy5010.com@CY5010.COM
```

```
kadmin.local:  ktadd -k /etc/krb5.keytab host/server.cy5010.com@CY5010.COM
Entry for principal host/server.cy5010.com@CY5010.COM with kvno 3, encryption type aes256-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/server.cy5010.com@CY5010.COM with kvno 3, encryption type arcfour-hmac added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/server.cy5010.com@CY5010.COM with kvno 3, encryption type des3-cbc-sha1 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/server.cy5010.com@CY5010.COM with kvno 3, encryption type des-cbc-crc added to keytab WRFILE:/etc/krb5.keytab.
```

After this we restarted the krb5-kdc and krb5-admin-server services and checked the status of these services if they are running.

```
root@server:~# sudo service krb5-kdc status
● krb5-kdc.service - Kerberos 5 Key Distribution Center
   Loaded: loaded (/lib/systemd/system/krb5-kdc.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2019-10-11 21:54:30 UTC; 5 days ago
  Process: 1056 ExecStart=/usr/sbin/krb5kdc -P /var/run/krb5-kdc.pid $DAEMON_ARGS (code=exited, status=0/SUCCESS)
 Main PID: 1062 (krb5kdc)
    Tasks: 1
   Memory: 2.7M
      CPU: 10ms
   CGroup: /system.slice/krb5-kdc.service
           └─1062 /usr/sbin/krb5kdc -P /var/run/krb5-kdc.pid

Oct 11 21:54:29 server.cy5010.com systemd[1]: Starting Kerberos 5 Key Distribution Center...
Oct 11 21:54:29 server.cy5010.com krb5kdc[1062]: commencing operation
Oct 11 21:54:30 server.cy5010.com systemd[1]: Started Kerberos 5 Key Distribution Center.
```

```
root@server:~# sudo service krb5-admin-server status
● krb5-admin-server.service - Kerberos 5 Admin Server
   Loaded: loaded (/lib/systemd/system/krb5-admin-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2019-10-11 21:54:29 UTC; 5 days ago
 Main PID: 1087 (kadmind)
    Tasks: 1
   Memory: 720.0K
      CPU: 11ms
   CGroup: /system.slice/krb5-admin-server.service
           └─1087 /usr/sbin/kadmind -nofork
```

Then we created a linux user on Ubuntu VM with the same name "vishalmaurya", so that we can get ssh access using that user from docker ssh-client.

```
root@server:~# adduser vishalmaurya
Adding user `vishalmaurya' ...
Adding new group `vishalmaurya' (1004) ...
Adding new user `vishalmaurya' (1004) with group `vishalmaurya' ...
Creating home directory `/home/vishalmaurya' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for vishalmaurya
Enter the new value, or press ENTER for the default
        Full Name []: Vishal Maurya
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
```

Next we update the "**/etc/ssh/sshd_config**" (enable Kerberos by updating the GSSAPI* flags in the config) on VM because we are using **ubuntu VM as our ssh auth server and docker ssh-client as a client trying to use kerberos to authenticate and use the service**.

```
root@server:~# cat /etc/ssh/sshd_config
# Package generated configuration file
# See the sshd_config(5) 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
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin prohibit-password
StrictModes yes

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

# GSSAPI options
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes

# 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
```

After this we restarted the ssh service on VM so that the changes we did are reflected while authenticating.

Part 3 - Client Setup in Docker Container:

Next we accessed the docker container and installed Kerberos client packages on the docker container.

```
root@server:~# docker ps
CONTAINER ID        IMAGE                COMMAND               CREATED             STATUS              PORTS               NAMES
6b8b82ddff73        sierraneu/kerberos   "/usr/sbin/sshd -D"   5 days ago          Up 5 days           22/tcp              nifty_brahmagupta
root@server:~# docker exec -it 6b8b82ddff73 /bin/bash
root@6b8b82ddff73:/# sudo apt-get install krb5-user krb5-config
Reading package lists... Done
Building dependency tree
```

We then updated the /etc/hosts file inside the docker container to use server as our Ubuntu VM and client as our docker container.

```
root@6b8b82ddff73:/# cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      6b8b82ddff73
10.110.206.122      server.cy5010.com
172.17.0.2  client.cy5010.com
```

To test if DNS resolution is working properly from the container we pinged the Ubuntu VM(server) from client (docker container)

```
root@server:~# docker ps
CONTAINER ID        IMAGE                COMMAND               CREATED             STATUS              PORTS               NAMES
6b8b82ddff73        sierraneu/kerberos   "/usr/sbin/sshd -D"   5 days ago          Up 5 days           22/tcp              nifty_brahmagupta
root@server:~# docker exec -it 6b8b82ddff73 /bin/bash
root@6b8b82ddff73:/# cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      6b8b82ddff73
10.110.206.122      server.cy5010.com
172.17.0.2  client.cy5010.com
root@6b8b82ddff73:/# ping server.cy5010.com
PING server.cy5010.com (10.110.206.122) 56(84) bytes of data.
64 bytes from server.cy5010.com (10.110.206.122): icmp_seq=1 ttl=64 time=0.056 ms
64 bytes from server.cy5010.com (10.110.206.122): icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from server.cy5010.com (10.110.206.122): icmp_seq=3 ttl=64 time=0.051 ms
64 bytes from server.cy5010.com (10.110.206.122): icmp_seq=4 ttl=64 time=0.089 ms
64 bytes from server.cy5010.com (10.110.206.122): icmp_seq=5 ttl=64 time=0.046 ms
^X^C
--- server.cy5010.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.046/0.058/0.089/0.017 ms
root@6b8b82ddff73:/#
```

We then verify if our krb-client config is done right by checking realms in it and if our kdc and admin server is reflected properly in /etc/krb5.conf file in docker container.

```
root@6b8b82ddff73:/# cat /etc/krb5.conf | grep -A3 -B3 "CY5010.COM"
[libdefaults]
        default_realm = CY5010.COM

# The following krb5.conf variables are only for MIT Kerberos.
        krb4_config = /etc/krb.conf
--
        fcc-mit-ticketflags = true

[realms]
        CY5010.COM = {
                kdc = server.cy5010.com
                admin_server = server.cy5010.com
        }
root@6b8b82ddff73:/#
```

Part 4 - Kerberos Authentication from Client:

Now we are done with the setup, now we will create a Ticket Granting ticket from KDC for principal user we created in Part 2. It will prompt for password as Kerberos database needs to add the key entry in its database for that user. We will use klist command to view all active tickets.

```
root@6b8b82ddff73:/# kinit vishalmaurya
Password for vishalmaurya@CY5010.COM:
root@6b8b82ddff73:/# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: vishalmaurya@CY5010.COM

Valid starting     Expires            Service principal
10/16/19 22:34:32  10/17/19 08:34:32  krbtgt/CY5010.COM@CY5010.COM
        renew until 10/17/19 22:34:28
root@6b8b82ddff73:/#
```

We can see that our ticket is active for 24 hours and can be used until then.

Screenshot of ssh -vvv <username>@server.cy5010.com:

Now we will try to use ssh authentication and check if the ssh server checks for kerberos based authentication and allows to login :

```
root@6b8b82ddff73:/# ssh -vvv vishalmaurya@server.cy5010.com
OpenSSH_7.2p2 Ubuntu-4ubuntu2.8, OpenSSL 1.0.2g  1 Mar 2016
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 19: Applying options for *
debug2: resolving "server.cy5010.com" port 22
debug2: ssh_connect_direct: needpriv 0
debug1: Connecting to server.cy5010.com [10.110.206.122] port 22.
debug1: Connection established.
debug1: permanently_set_uid: 0/0
debug1: key_load_public: No such file or directory
debug1: identity file /root/.ssh/id_rsa type -1
debug1: key_load_public: No such file or directory
debug1: identity file /root/.ssh/id_rsa-cert type -1
debug1: key_load_public: No such file or directory
debug1: identity file /root/.ssh/id_dsa type -1
debug1: key_load_public: No such file or directory
debug1: identity file /root/.ssh/id_dsa-cert type -1
debug1: key_load_public: No such file or directory
debug1: identity file /root/.ssh/id_ecdsa type -1
debug1: key_load_public: No such file or directory
debug1: identity file /root/.ssh/id_ecdsa-cert type -1
debug1: key_load_public: No such file or directory
debug1: identity file /root/.ssh/id_ed25519 type -1
debug1: key_load_public: No such file or directory
debug1: identity file /root/.ssh/id_ed25519-cert type -1
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8
debug1: Remote protocol version 2.0, remote software version OpenSSH_7.2p2 Ubuntu-4ubuntu2.8
debug1: match: OpenSSH_7.2p2 Ubuntu-4ubuntu2.8 pat OpenSSH* compat 0x04000000
debug2: fd 3 setting O_NONBLOCK
debug1: Authenticating to server.cy5010.com:22 as 'vishalmaurya'
debug3: hostkeys_foreach: reading file "/root/.ssh/known_hosts"
debug3: record_hostkey: found key type ECDSA in file /root/.ssh/known_hosts:1
debug3: load_hostkeys: loaded 1 keys from server.cy5010.com
```

```
Lab_4 –
debug1: No valid Key exchange context
debug2: we did not send a packet, disable method
debug3: authmethod_lookup gssapi-with-mic
debug3: remaining preferred: publickey,keyboard-interactive,password
debug3: authmethod_is_enabled gssapi-with-mic
debug1: Next authentication method: gssapi-with-mic
debug3: send packet: type 50
debug2: we sent a gssapi-with-mic packet, wait for reply
debug3: receive packet: type 60
debug3: send packet: type 61
debug3: receive packet: type 61
debug3: send packet: type 66
debug3: receive packet: type 52
debug1: Authentication succeeded (gssapi-with-mic).
Authenticated to server.cy5010.com ([10.110.206.122]:22).
debug1: channel 0: new [client-session]
debug3: ssh_session2_open: channel_new: 0
debug2: channel 0: send open
debug3: send packet: type 90
debug1: Requesting no-more-sessions@openssh.com
debug3: send packet: type 80
debug1: Entering interactive session.
debug1: pledge: network
debug3: receive packet: type 80
debug1: client_input_global_request: rtype hostkeys-00@openssh.com want_reply 0
debug3: receive packet: type 91
debug2: callback start
debug2: fd 3 setting TCP_NODELAY
debug3: ssh_packet_set_tos: set IP_TOS 0x10
debug2: client_session2_setup: id 0
debug2: channel 0: request pty-req confirm 1
debug3: send packet: type 98
debug1: Sending environment.
debug3: Ignored env HOSTNAME
```

```
debug2: channel 0: open confirm rwindow 0 rmax 32768
debug3: receive packet: type 99
debug2: channel_input_status_confirm: type 99 id 0
debug2: PTY allocation request accepted on channel 0
debug2: channel 0: rcvd adjust 2097152
debug3: receive packet: type 99
debug2: channel_input_status_confirm: type 99 id 0
debug2: shell request accepted on channel 0
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-159-generic x86_64)

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

55 packages can be updated.
22 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.



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.

vishalmaurya@server:~$ hostname
server.cy5010.com
vishalmaurya@server:~$
```

And voila!! We got in!

Screenshot of the /var/log/auth.log file showing Kerberos authentication:

```
root@server:~# tail -fn30 /var/log/auth.log
Oct 16 22:33:22 server useradd[5187]: new user: name=vishalmaurya, UID=1004, GID=1004, home=/home/vishalmaurya, shell=/bin/bash
Oct 16 22:33:34 server passwd[5194]: pam_unix(passwd:chauthtok): password changed for vishalmaurya
Oct 16 22:33:45 server chfn[5195]: changed user 'vishalmaurya' information
Oct 16 22:34:12 server krb5kdc[1062]: AS_REQ (6 etypes {18 17 16 23 25 26}) 172.17.0.2: NEEDED_PREAUTH: vishalmaurya@CY5010.COM for krbtgt/CY5010.COM@CY5010.C
Oct 16 22:34:17 server krb5kdc[1062]: preauth (encrypted_timestamp) verify failure: Decrypt integrity check failed
Oct 16 22:34:17 server krb5kdc[1062]: AS_REQ (6 etypes {18 17 16 23 25 26}) 172.17.0.2: PREAUTH_FAILED: vishalmaurya@CY5010.COM for krbtgt/CY5010.COM@CY5010.C
Oct 16 22:34:28 server krb5kdc[1062]: AS_REQ (6 etypes {18 17 16 23 25 26}) 172.17.0.2: NEEDED_PREAUTH: vishalmaurya@CY5010.COM for krbtgt/CY5010.COM@CY5010.C
Oct 16 22:34:32 server krb5kdc[1062]: AS_REQ (6 etypes {18 17 16 23 25 26}) 172.17.0.2: ISSUE: authtime 1571265272, etypes {rep=18 tkt=18 ses=18}, vishalmaury
Oct 16 22:35:36 server krb5kdc[1062]: TGS_REQ (6 etypes {18 17 16 23 25 26}) 172.17.0.2: ISSUE: authtime 1571265272, etypes {rep=18 tkt=18 ses=18}, vishalmaur
0.COM
Oct 16 22:35:36 server sshd[5272]: Authorized to vishalmaurya, krb5 principal vishalmaurya@CY5010.COM (krb5_kuserok)
Oct 16 22:35:36 server sshd[5272]: Accepted gssapi-with-mic for vishalmaurya from 172.17.0.2 port 49742 ssh2
Oct 16 22:35:36 server sshd[5272]: pam_unix(sshd:session): session opened for user vishalmaurya by (uid=0)
Oct 16 22:35:36 server systemd: pam_unix(systemd-user:session): session opened for user vishalmaurya by (uid=0)
Oct 16 22:35:36 server systemd-logind[1072]: New session 30 of user vishalmaurya.
Oct 16 22:37:26 server sshd[5314]: Received disconnect from 172.17.0.2 port 49742:11: disconnected by user
Oct 16 22:37:26 server sshd[5314]: Disconnected from 172.17.0.2 port 49742
Oct 16 22:37:26 server sshd[5272]: pam_unix(sshd:session): session closed for user vishalmaurya
Oct 16 22:37:26 server systemd-logind[1072]: Removed session 30.
Oct 16 22:37:31 server sudo: pam_unix(sudo:session): session closed for user root
Oct 16 22:37:32 server sshd[4904]: Received disconnect from 10.0.2.2 port 63415:11: disconnected by user
Oct 16 22:37:32 server sshd[4904]: Disconnected from 10.0.2.2 port 63415
Oct 16 22:37:32 server sshd[4865]: pam_unix(sshd:session): session closed for user vagrant
Oct 16 22:37:32 server systemd-logind[1072]: Removed session 29.
Oct 16 22:37:32 server systemd: pam_unix(systemd-user:session): session closed for user vagrant
Oct 16 22:48:45 server sshd[5747]: Accepted publickey for vagrant from 10.0.2.2 port 63510 ssh2: RSA SHA256:O+aRW0tcUyXA23q1V5dcA/r9Der57WPn0999sIs4wu0
Oct 16 22:48:45 server sshd[5747]: pam_unix(sshd:session): session opened for user vagrant by (uid=0)
Oct 16 22:48:45 server systemd: pam_unix(systemd-user:session): session opened for user vagrant by (uid=0)
Oct 16 22:48:45 server systemd-logind[1072]: New session 31 of user vagrant.
Oct 16 22:48:47 server sudo:  vagrant : TTY=pts/0 ; PWD=/home/vagrant ; USER=root ; COMMAND=/bin/bash
Oct 16 22:48:47 server sudo: pam_unix(sudo:session): session opened for user root by vagrant(uid=0)
```

Meaning of the logs :

The first three lines related to kerberos in logs is :

Oct 16 22:34:12 server krb5kdc[1062]: AS_REQ (6 etypes {18 17 16 23 25 26}) 172.17.0.2: NEEDED_PREAUTH: vishalmaurya@CY5010.COM for krbtgt/CY5010.COM@CY5010.COM, Additional pre-authentication required
Oct 16 22:34:17 server krb5kdc[1062]: preauth (encrypted_timestamp) verify failure: Decrypt integrity check failed
Oct 16 22:34:17 server krb5kdc[1062]: AS_REQ (6 etypes {18 17 16 23 25 26}) 172.17.0.2: PREAUTH_FAILED: vishalmaurya@CY5010.COM for krbtgt/CY5010.COM@CY5010.COM, Decrypt integrity check failed

Which was pushed in the log because I had entered the wrong password initially in the kinit step to see if the server is notified if someone tries to login with wrong password and request a TGT, which was captured beautifully by the audit log.

After this the log shows the transaction that happened because of the ssh we tried :

Oct 16 22:34:28 server krb5kdc[1062]: AS_REQ (6 etypes {18 17 16 23 25 26}) 172.17.0.2: NEEDED_PREAUTH: vishalmaurya@CY5010.COM for krbtgt/CY5010.COM@CY5010.COM, Additional pre-authentication required

This line means that the client needs authorization (AS_REQ : - Authentication service request) to use the ssh resource.

Oct 16 22:34:32 server krb5kdc[1062]: AS_REQ (6 etypes {18 17 16 23 25 26}) 172.17.0.2: ISSUE: authtime 1571265272, etypes {rep=18 tkt=18 ses=18}, vishalmaurya@CY5010.COM for krbtgt/CY5010.COM@CY5010.COM

Next client realises that it has a TGT already it will use this TGT to raise a ticket for using the service by the help of a AS_REQ, adding the timestamp and verifier as SSH-service running on server.cy5010.com.

Client receives an equivalent AS_REP for this request from server with the ticket to use the SSH service, this AS_REP is encrypted with the key of SSH-AUTH-Server.

Oct 16 22:35:36 server krb5kdc[1062]: TGS_REQ (6 etypes {18 17 16 23 25 26}) 172.17.0.2: ISSUE: authtime 1571265272, etypes {rep=18 tkt=18 ses=18}, vishalmaurya@CY5010.COM for host/server.cy5010.com@CY5010.COM

Next client uses this ticket to get access to the SSH-service running on server.cy5010.com with the TGS_REQ request.

The client receives an equivalent TGS_REP from the ssh authentication service running on server and the client is authenticated to use ssh successfully.

References :

[1] https://kb.iu.edu/d/aumh
[2]https://web.mit.edu/kerberos/krb5-1.5/krb5-1.5.4/doc/krb5-install/What-is-Kerberos-and-How-Does-it-Work_003f.html
[3]https://searchwindowsserver.techtarget.com/feature/Five-steps-to-using-the-Kerberos-protocol