Configuring Kerberos on CentOS 5

Kerberos is a ticket-oriented authentication system that was originally designed for Unix networks, but was also embraced (and extended) by Microsoft in Active Directory. I've been debugging a number of issues involving the Squid proxy server on Linux using Samba to authenticate against Active Directory, and as part of this I had to get familiar with Kerberos.

It's not trivial, so I've documented my workflow here. Hopefully it will useful to others.

The test environment consists of two virtual machines running CentOS 5, imaginatively named centos01 (krbserver) and centos02 (krbclient). For the purpose of this test, centos01 is the Kerberos server and centos02 is the client.

I followed the instructions here and broadly recommend them. These are my additional notes to clarify some parts of the install.

General notes

Make sure that you use the same time source for both client and server. I used NTP to keep the two VMs in sync. The notes do state this but it's worth stressing.

Remember how many IT problems are caused by name resolution errors! Make sure you have both the server and client registered in DNS (or have entries in /etc/hosts). If using /etc/hosts both the standalone hostname and the FQDN should be added:

```
192.168.192.26 krbserver.local.zone krbserver
192.168.192.108 krbclient.local.zone krbserver
```

Note the order of the hostname and the FQDN! This is important (see further below).

Configure the server

After installing the packages using YUM, configuring the database and ACL file, adding the first principal user and starting the three services, the server should be ready to go. Confirm this with kinit and klist. Now it's time to configure the client.

Configure the client

Install the packages using YUM and then run the kadmin command and add a new principal for the client machine. It's worth noting that this should be done using the kadmin interactive interface instead of trying to put the "addprinc" parameter on the command line. This is because the -randkey option will be interpreted by kadmin on the command line as "-r andkey" and it will try and authenticate against the "andkey" realm. So for me, the command looked like:

```
# kadmin -p julian/admin@LOCAL.ZONE
Password for julian/admin@LOCAL.ZONE: ********
kadmin: addprinc -randkey host/krbclient.local.zone
```

I assume that this is rougly analogous to adding a machine to an Active Directory domain.

Once this entry, export the principal to the workstation's /etc/krb5.keytab file.

In addition to the machine principal, I also created a normal (non-admin) a local user, julian@LOCAL.ZONE. On the client, I log in as my own non-root user ("julian") and type kinit:

```
$ kinit
Password for julian@LOCAL.ZONE: ********
```

If this succeeds, you should see the "ticket granting ticket" be assigned:

```
$ klist
Ticket cache: FILE:/tmp/krb5cc_500
Default principal: julian@LOCAL.ZONE

Valid starting Expires Service principal
07/17/09 11:14:20 07/18/09 11:14:20 krbtgt/LOCAL.ZONE@LOCAL.ZONE

Kerberos 4 ticket cache: /tmp/tkt500 klist: You have no tickets cached
```

This process shows that communication between the client and server using Kerberos is successful.

Configuring telnet (for testing)

On the server, I then enabled the krb5-telnet service in /etc/xinetd.d and started xinetd. On the client, I then ran:

```
$ /usr/kerberos/bin/telnet -a krbserver
Trying 192.168.192.26...
Connected to krbserver.local.zone (192.168.192.26).
Escape character is '^]'.
[ Kerberos V5 refuses authentication because telnetd: krb5_rd_req failed: Key
version number for principal in key table is incorrect ]
[ Kerberos V5 refuses authentication because telnetd: krb5_rd_req failed: Key
version number for principal in key table is incorrect ]
Password:
```

Problem. It was asking for a password which implied the Kerberos ticket was not being passed correctly. However, when I ran klist, it showed that the ticket for the host was passed correctly:

```
$ klist
Ticket cache: FILE:/tmp/krb5cc_500
Default principal: julian@LOCAL.ZONE

Valid starting Expires Service principal
07/17/09 10:38:20 07/18/09 10:38:20 krbtgt/LOCAL.ZONE@LOCAL.ZONE
07/17/09 10:38:31 07/18/09 10:38:20 host/krbserver.local.zone@LOCAL.ZONE
```

After running strace against the telnetd process, it appeared that the telnet server was failing when trying to read /etc/krb5.keytab. But all the documentation I had read stated that this should be run on the client and not the server. So, why does the Kerberos server need a keytab file?

Answer: The ***Kerberos server*** does *not* require a keytab file, but the ***telnet server*** does! Although they are both running on the same VM, the telnet server is itself a client to the Kerberos server. Simple when you work it out it would have semantically been easier to understand if my telnet server been different from the Kerberos server.

So I ran the kadmin command on the server and created a keytab file using the ktadd command. I restarted the Kerberos services for good measure and cleared my client and server caches using kdestroy, restarted xinetd and tried the telnet:

```
[julian@krbclient bin]$ ./telnet -a krbserver
Trying 192.168.192.26...
Connected to krbserver.local.zone (192.168.192.26).
Escape character is '^]'.
[ Kerberos V5 accepts you as ``julian@LOCAL.ZONE'' ]
Last login: Fri Jul 17 10:48:10 from krbclient
[julian@krbserver ~]$
```

Result!

Configuring SSH

The instructions state that `GSSAPIAuthentication` and `GSSAPIDelegateCredentials` need to be enabled. I did this and restarted the SSH daemon with -ddd (debug) enabled.

The first attempt at running ssh krbserver prompted for a password, but the server debug revealed the following:

```
debug1: Unspecified GSS failure. Minor code may provide more information
No principal in keytab matches desired name
```

Okay, so this is weird. Checking the output of klist showed this:

```
[julian@krbclient ~]$ ssh krbserver
julian@krbserver's password:
Connection closed by 192.168.192.26
[julian@krbclient ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_500
Default principal: julian@LOCAL.ZONE

Valid starting Expires Service principal
07/17/09 13:42:27 07/18/09 13:42:27 krbtgt/LOCAL.ZONE@LOCAL.ZONE
07/17/09 13:42:33 07/18/09 13:42:27 host/krbserver@


Kerberos 4 ticket cache: /tmp/tkt500
klist: You have no tickets cached
```

Note that krbserver@ has no realm. This turned out to be because /etc/hosts (on the client) looks like this:

```
192.168.192.108 krbclient krbclient.local.zone
192.168.192.26 krbserver krbserver.local.zone
```

Putting the hostname after the FQDN like this:

```
192.168.192.108 krbclient.local.zone krbclient
192.168.192.26 krbserver.local.zone krbserver
```

fixes the problem!

```
[julian@krbclient ~]$ ssh krbserver
Last login: Fri Jul 17 13:54:40 2009 from krbclient.local.zone
```

Klist now shows:

```
[julian@krbclient ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_500
Default principal: julian@LOCAL.ZONE

Valid starting Expires Service principal
07/17/09 13:42:27 07/18/09 13:42:27 krbtgt/LOCAL.ZONE@LOCAL.ZONE
07/17/09 13:42:33 07/18/09 13:42:27 host/krbserver@
07/17/09 13:54:38 07/18/09 13:42:27 host/krbserver.local.zone@LOCAL.ZONE


Kerberos 4 ticket cache: /tmp/tkt500
klist: You have no tickets cached
```

Summary

What you see above does not include the time spent trying things out and staring blankly at the screen. Getting Kerberos up and running is not the most trivial process and while there is some decent documentation, there are also a lot of people posting questions and asking for help when it doesn't work properly. Hopefully this will shed some light on it for others.