Advanced Excercise 10

Due: Saturday, November 26, 10:00PM (Hard Deadline)

Submission Instructions

To receive credit for this assignment you will need to stop by someone's office hours and demo your local AFS access.

You are welcome to do this assignment either on your native operating system on your machine or inside the class VM, whichever you are more comfortable with. I have done this on Linux and OS X. I have not personally tried this on Windows, however I believe it should work.

The Andrew File System

One thing you've hopefully noticed is that when you log on to any CAEN machine (or ssh to any CAEN login server) all of your files are magically there. That's because your files are stored using AFS, a network file system. When you save files on a CAEN machine, they don't actually save to the local hard disk, instead they go out over the Internet to a machine in a server room somewhere.[1] File reads similarly go over the network to grab the data, so everything stays in sync automatically.[2] Note that while this is similar to Dropbox, it is slightly different as well since Dropbox stores and edits local copies on disk and then syncs them in the background. That wouldn't work at the university scale, imagine if every CAEN machine had to have enough disk space to hold every file for every student! This means that AFS will only work if you are online.

Since AFS is a networked file system, you can also set up your local machine to join the network. This way, whenever you change anything on your machine, it'll automatically update on CAEN and vice-versa. You could run your text editor on your machine and your compiler on CAEN if you like. Along the way, you'll also be able to set things up so that you don't need to type your password every time that you ssh into CAEN.

Kerberos

To get things working, there are two key things you need to get set up, the first is **Kerberos**. Kerberos is a tool for authentication. You've been using it every time you log into CAEN (or Cosign[3]). With Kerberos, you must periodically (every 24 hours in umich's setup) enter your password to get a new *ticket*. Once you have this ticket on your machine, however, you can use it to log into any Michigan service without ever typing your password again. AFS uses this ticket to gain access to the files that you own.

Big Hint: You will need a configuration file (`/etc/krb5.conf`). Don't try to write this yourself! Instead login to a CAEN machine and grab a copy from that machine.

To test if you have Kerberos working, you can add this to your `.ssh/config` file on your local machine:

```
Host login.engin.umich.edu
GSSAPIAuthentication yes
GSSAPIDelegateCredentials yes
GSSAPITrustDNS yes
```

If you have a valid Kerberos ticket, when you run `ssh login.engin.umich.edu` you should not be prompted for a password, you'll just log straight in!

[1] Last I checked, for folks with uniqnames starting a-k, this was the MITC Data Center on Oakbrook off of State. I don't recall where the rest of the alphabet is stored. I think it was in the Southfield data center at one point.

[2] This glosses over some details about caching that's done for performance, but the basic idea is there.

[3] You can actually set up your web browser to not need your password either. It can also use the same Kerberos ticket. I know this is do-able in Firefox, it may be do-able in other browsers as well.

AFS

The next thing you'll need to install and set up is AFS. The same hint applies here where the best way to find the correct AFS configuration is look at it on a CAEN machine and then copy it to your local machine.

Notice that once you install AFS you will have *list* permissions on much of the directory hierarchy. This will let you explore a little, such as seeing that files exist, but not their contents. For example, while holding credentials for `ppannuto`, I try to look at Matt's files, I can see that he has a `test.sh` file in his home directory, but not what's in it:

```
$ ls /afs/umich.edu/user/m/t/mterwil
ls: cannot access /afs/umich.edu/user/m/t/mterwil/Private: Permission denied
ls: cannot access /afs/umich.edu/user/m/t/mterwil/MPrint: Permission denied
ls: cannot access /afs/umich.edu/user/m/t/mterwil/Desktop: Permission denied
ls: cannot access /afs/umich.edu/user/m/t/mterwil/Downloads: Permission denied
ls: cannot access /afs/umich.edu/user/m/t/mterwil/bin: Permission denied
ls: cannot access /afs/umich.edu/user/m/t/mterwil/Documents: Permission denied
ls: cannot access /afs/umich.edu/user/m/t/mterwil/Music: Permission denied
ls: cannot access /afs/umich.edu/user/m/t/mterwil/Pictures: Permission denied
ls: cannot access /afs/umich.edu/user/m/t/mterwil/Videos: Permission denied
ls: cannot access /afs/umich.edu/user/m/t/mterwil/test.sh: Permission denied
bin Desktop Documents Downloads MPrint Music Pictures Private Public Shared test.sh Videos
$ cat /afs/umich.edu/user/m/t/mterwil/test.sh
cat: /afs/umich.edu/user/m/t/mterwil/test.sh: Permission denied
```

If you find yourself getting a similar set of permission denied issues in your own home directory, it means you are connected to AFS correctly, but you haven't connected Kerberos and AFS correctly.

The Assignment

Get AFS installed and working on your local machine. You should have read/write access to your own files.

The writeup in this assignment gives you some intuition for what's going on and the steps you'll have to take, but it is not a step-by-step guide – there are plenty of those out there (I even found a few umich-specific ones for the Kerberos half, not that there's anything special you have to do to set things up to work here, but they use umich as their example).