Part II User's Guide

Chapter 4: *Accessing Kerberized Machines (Fermilab-Supported Methods)*

In this chapter we discuss accessing systems in the FNAL.GOV realm from UNIX, Windows and Macintosh machines using the methods recommended and supported by the Fermilab Computing Division. We cover logging in at the console, connecting over the network, and using portal mode.

Chapter 5: *Using your CRYPTOCard*

A CRYPTOCard is a calculator-style, battery-powered device used for generating a single-use password (required for access from a non-Kerberized machine). In this chapter we describe how to use and care for your CRYPTOCard.

Chapter 6: *Logging In from Off-Site*

In this chapter, we discuss what off-site users are required to do in order to access Fermilab's strengthened realm, and some of the issues they may encounter.

Chapter 7: *Accessing Kerberized Machines (Community-Supported Methods)*

In this chapter we discuss accessing systems in the FNAL.GOV realm from UNIX, Windows and Macintosh machines using programs or operating systems not supported at Fermilab.

Chapter 8: *Troubleshooting your Authentication Problems*

This chapter is intended to help users who are having trouble authenticating to Kerberos and logging in to Kerberized machines. We include information that should help you figure out what's causing your problem, and to fix it.

Chapter 9: *Using Kerberos*

This chapter provides the basic information you need in order to manage your Kerberos tickets and work in a Kerberized environment. In particular, we cover ticket options and management, and account access files. The Kerberos commands and features of Kerberized network programs are documented in Chapter 12: *Kerberos Command Descriptions* and Chapter 13: *Network Programs Available on Kerberized Machines*, respectively.

Chapter 10: *Miscellaneous Topics for the User*

In this chapter we document a variety of common operations that work differently in the Fermilab Kerberized environment.

Chapter 4: Accessing Kerberized Machines (Fermilab-Supported Methods)

In this chapter we discuss accessing systems in the FNAL.GOV realm from UNIX and Windows machines using the methods recommended and supported by the Fermilab Computing Division. We cover logging in at the console, connecting over the network, and using CRYPTOCards with portal mode.

Very important note: Any time you're about to enter your Kerberos password, first verify that you're using the host's directly-connected keyboard! On rare, necessary occasions you may transmit your password over an encrypted network connection, but this is not to be done on a regular basis. See Chapter 11: *Encrypted vs. Unencrypted Connections* for information.

4.1 Trying Out Kerberos on fnkerb.fnal.gov

As of July 2, 2001, the Computing Division has set up a mixed-mode Kerberized Linux system in the FNAL.GOV strengthened realm for use by all Fermilab employees and users, local and remote. Mixed-mode is explained below. This machine is available for users to test basic Kerberos functionality and to change passwords if no local machine is available. The machine name is fnkerb.fnal.gov, and it is accessible to everyone who has an account on fnalu.

As of October 23, all the fnalu nodes are Kerberized in mixed-mode, the same as fnkerb. For information specific to fnalu, go to the *Strong Authentication at Fermilab* page (`http://www.fnal.gov/docs/strongauth/`), and click on *FNALU in mixed-mode Kerberos phase* or see the html pages for one of the tutorials.

The fnkerb system is configured to use AFS as a login area. It is configured in mixed-mode, which means it allows the following access methods:

- non-Kerberized ssh with your AFS password (Kerberos principal not required for ssh logins)
- Portal mode via CRYPTOCard via non-Kerberized versions of **telnet**, **ftp** (see section 4.5 *Connecting from a NonKerberized Machine: Portal Mode*).

- Kerberos via Kerberized versions of **telnet**, **ftp**, **rsh**, **rlogin**, and **rcp** (see sections 4.3 *Connecting from One Kerberized Machine to Another*, 4.7 *Logging In Through WRQ® Reflection Software from Windows*, 7.2 *Logging In from a Macintosh*)

Please note the following:

- Fnkerb has a limited lifetime; it will be shutdown on or soon after December 18, 2001.
- We ask that you limit your activities on fnkerb to acquainting yourself with Kerberos. The system is not configured to support general user activities (e.g., web browsing, reading email) or physics analysis. No batch facilities, CVS respositories, etc., are available.
- All questions about Kerberos and its use on this system should be directed to *helpdesk@fnal.gov* or *kerberos-users@fnal.gov.*
- Issues with the operating system on fnkerb should be reported to *helpdesk@fnal.gov.*
- Support on fnkerb is available Monday through Friday, 9:00 to 5:00.

4.2 Logging In at the Console of a Kerberized UNIX Machine

4.2.1 Using Standard UNIX Login Program

If your desktop machine is running the standard login program, log in at the console normally, entering your standard UNIX password (note that if your machine runs AFS, your UNIX and AFS passwords may be the same). The standard login program does not accept your **kerberos** password. You need to run **`kinit`** after logging in to obtain your credentials. The credentials should then get forwarded to other strengthened machines normally. The **kerberos** login program is not installed by default with the **kerberos** product.

4.2.2 Using Kerberos Login Program

If your desktop machine is configured to use the **kerberos** login program[1], you can authenticate to Kerberos at login by entering your Kerberos password at the password prompt. You do not need to run **`kinit`** after login. (You can still login using your UNIX password, then run **`kinit`** to get Kerberos

1. Not applicable to IRIX systems, or to Linux and Solaris if using the GUI login box. The login program isn't run in these cases.

tickets, if you wish.) An advantage to using the **kerberos** login program is that it checks the `/etc/krb5.conf` file in which you or your system administrator can set defaults for Kerberized applications.

4.2.3 If you don't have a principal yet...

Note that if you have an account and a standard UNIX password on a machine (in the `passwd` file or NIS map) but no principal or Kerberos password, you can still log in at the console. (From any terminal other than the console, the Kerberized machine looks for existing Kerberos credentials, and responds in portal mode if none are found; you have no option to enter your UNIX password.) However, once logged in, you cannot make outbound connections from there since Kerberized services are unavailable to you.

You can use ssh to log into machines running mixed mode Kerberos, as described in section 4.2.4 *Machines Running Mixed Mode Kerberos*.

4.2.4 Machines Running Mixed Mode Kerberos

Machines that are Kerberized in mixed-mode allow logins via ssh for users who don't yet have a Kerberos principal. This is in addition to allowing login via Kerberized services or CRYPTOCard. Mixed-mode machines on the Fermilab site will convert to full Kerberos by the end of 2001.

The FNALU nodes are in mixed-mode until December 4, 2001, at which time they will become fully Kerberized. See `http://www.fnal.gov/docs/strongauth/misc/fnalukerberos.html` for more information on FNALU.

4.3 Connecting from One Kerberized Machine to Another

Make sure you have forwardable credentials on your desktop machine, then run the Kerberized version of the connection program you want to use (**ssh**, **slogin**, **telnet**, **rsh**, **rlogin**, **rcp**, **scp** or **ftp**) to connect and forward your credentials to the target machine. Forwarding is described in section 9.2.4 *Forwarding Tickets*. The Kerberized features of these programs are described in Chapter 13: *Network Programs Available on Kerberized Machines*.

Do not run **`kinit`** over the network to authenticate on the remote machine. As of Kerberos v1_5, **`kinit`** is equipped with a warning that appears if the userid issuing the command doesn't own the console device. It is designed to help users avoid typing their password inadvertently over the network.

Assuming your credentials get forwarded to the target machine, you should be automatically recognized and authenticated there; you should not be prompted for your Kerberos password.

A few notes:

- If the usernames on the machines differ, use the **`-l <target_host_login_name>`** option; e.g., **`ssh -l <target_host_login_name>`**.
- If ticket forwarding has been set "off" for your system, and you want to connect to a Kerberized machine with ticket forwarding turned on, use the appropriate option, e.g., **`-f`** or **`-F`** for **telnet**, **rsh**, and **rlogin (`-F` marks them reforwardable whereas `-f` does not).**
- If ticket forwarding has been set "on" for your system, and you want to connect to a Kerberized machine with ticket forwarding turned off, use the appropriate option (e.g., **`-N`** for **telnet**, **rsh**, **rlogin**, and **rcp**, or **`-k`** for Kerberized **ssh**). Forwarding is described in section 9.2.4 *Forwarding Tickets*.

Warning! If your on-site Kerberized system accepts a reusable login password over the network (even on an encrypted connection), this is a violation of the Fermilab Policy on Computing (see `http://www.fnal.gov/cd/main/cpolicy.html`).

4.4 Connecting via Kerberized SSH

Any machines that are sited at FNAL and that wish to use ssh will be required to use Kerberized ssh (available from `ftp://ftp.fnal.gov/KITS/` as ssh v1_2_27g or higher) as of the start of calendar year 2002. Non-Kerberized ssh is not permitted on these machines.

With both **kerberos** and Kerberized **ssh** installed on your machine, make sure you have a Kerberos ticket, then run the Kerberized version of the connection program you want to use (e.g., **ssh**, **slogin**, or **scp**) to connect to a remote Kerberized host. The Kerberized options for these programs are described in Chapter 13: *Network Programs Available on Kerberized Machines*. You do not get prompted for your Kerberos password during login.

Ssh encrypts the connection by default typically (check your configuration). You can always use the **`-c <cipher>`** option to ensure encryption.

4.5 Connecting from a NonKerberized Machine: Portal Mode

4.5.1 About Portal Mode

At Fermilab, strengthened machines are configured to respond in *portal mode* when requests for access come from machines outside the trusted realm[1]. In portal mode, the Kerberized machine acts as a secure gateway into the strengthened realm, requiring a single-use password for authentication. This avoids transmission of reusable clear-text passwords over a potentially unprotected network. The non-reusable authentication method for portal mode that the Computing Division currently supports is CRYPTOCard.

Once you've logged on successfully through the portal, the KDC "knows who you are", and the machine obtains your Kerberos credentials for you. You are not required to provide your Kerberos password when making further connections to other machines in the FNAL.GOV realm. If you need to reauthenticate, run the command **`new-portal-ticket`**. This provides a portal mode prompt.

4.5.2 About CRYPTOCard

Fermilab is implementing portal mode using CRYPTOCard technology. A CRYPTOCard is a calculator-style, battery-powered device used for generating a single-use password.

To read more about what a CRYPTOCard is and how it works, see Appendix : *Using your CRYPTOCard*. To request one (or to request CRYPTOCard software for Palm Pilot -- not currently available), fill out the online form *Form to Request Kerberos Principal and/or Related Items* at `http://www.fnal.gov/cd/forms/strongauth.html`. When you get your CRYPTOCard, go back to Chapter 5: *Using your CRYPTOCard* for information on how to use it and take care of it.

Two notes:

1. ...or from a trusted realm, if credentials don't grant you access to your account

- No special hardware or software is required on the nonKerberized machine for CRYPTOCard use.
- The CRYPTOCard login code assumes that the user's login name and principal match. If yours don't match, you won't be able to log in using this method.

4.5.3 Programs for Initiating CRYPTOCard Login

To log on to a machine in the FNAL.GOV realm from your nonKerberized machine, run any of the following commands:

```
% ssh <host>
% slogin <host>
% telnet <host>
% ftp <host>
```

as usual (the standard, nonKerberized version of the program, as the Kerberized version is not available on nonKerberized machines).

Two notes regarding the use of **ssh** and **slogin** with CRYPTOCard:

- The Kerberos login program supports **ssh** only when no command argument is given, i.e., when it is effectively equivalent to **slogin**. Fundamentally, the only **ssh** program supported is **slogin**.
- The Kerberized sshd on the remote host prompts for an **ssh** password before displaying the CRYPTOCard challenge. Just press Return for the **ssh** password, don't enter any characters.
- During the FNALU mixed-mode transition period, ssh and slogin to an FNALU node from a nonKerberized machine will not give CRYPTOCard prompt. You will use your AFS password to gain access.

After you issue the network command, the remote host will prompt you to provide a non-reusable password rather than your Kerberos password:

```
Press ENTER and compare this challenge to the one on your
display: [12345678]
Enter the displayed response:
```

Use your CRYPTOCard to provide this password, as described in section 5.5 *Log in Using CRYPTOCard (the First Time)*, section 5.6 *Log in Using CRYPTOCard (Subsequently)*, or briefly in section 4.5.4 *Summary of the "Normal" Login Steps with CRYPTOCard*.

Notes:

- Never type your Kerberos password over a CRYPTOCard **telnet** session! The connection is not encrypted.
- You may type your password infrequently over an encrypted CRYPTOCard **ssh/slogin** session.

• **rsh**, **rlogin**, **rcp** and **scp** are not available for portal mode.

4.5.4 Summary of the "Normal" Login Steps with CRYPTOCard

The full description of using a CRYPTOCard is given in Chapter 5: *Using your CRYPTOCard*. Assuming you've read that, this is just a reminder!

1) CRYPTOCard: **ON**, **`[PIN]`**, **ENT**, **ENT**, **ENT**

2) Host: Run **`telnet <hostname>`** or **`ftp <hostname>`** and provide your principal.

3) Host: type response, press **RETURN**

4) CRYPTOCard: **OFF**

If you want to generate another response before turning it off, just press **ENT** again three times (once to get past the `Fermilab` id, once to display the next challenge, and once to display the response).

4.5.5 Portal Mode FTP when you can't see the Challenge

If you're doing portal mode **FTP** with a client that does not show you the output text from the server (e.g., **FTP** under **emacs** or from a variety of Windows **FTP** clients), it won't display the challenge string. In this case, go ahead and use your CRYPTOcard anyway, and enter the response as your password. This works if your card is in sync with the KDC, which should generally be the case.

If you're using the WRQ® FTP client with standard (nonKerberos) security, select **VIEW > COMMAND WINDOW** to see the CRYPTOCard challenge.

If the **FTP** login is unsuccessful, you need to synchronize your card. To do so, start a **telnet** connection, and type the displayed challenge into your CRYPTOCard. Then disconnect the **telnet** session BEFORE you enter the response so that you save it for your **FTP** session! Otherwise the response will get used and you'll be out of sync again.

4.6 Logging into a UNIX Account that's not your own

If you wish to log into an account for which your login id is different from your principal name (e.g., a group account), your principal must be listed in either the `.k5login` or the `.k5users` file (**ksu** only) of the target account. See section 9.3.1 *The .k5login File*.

First log in to your own account on a Kerberized machine and obtain credentials as usual, then connect to the target account after you're authenticated. If the target account is on a different machine, simply connect to that machine using one of the Kerberized connection utilities, and use the **`-l <login_name>`** option where **`<login_name>`** is the target account name. If the account is on the same machine, use **`ksu <login_name>`**.

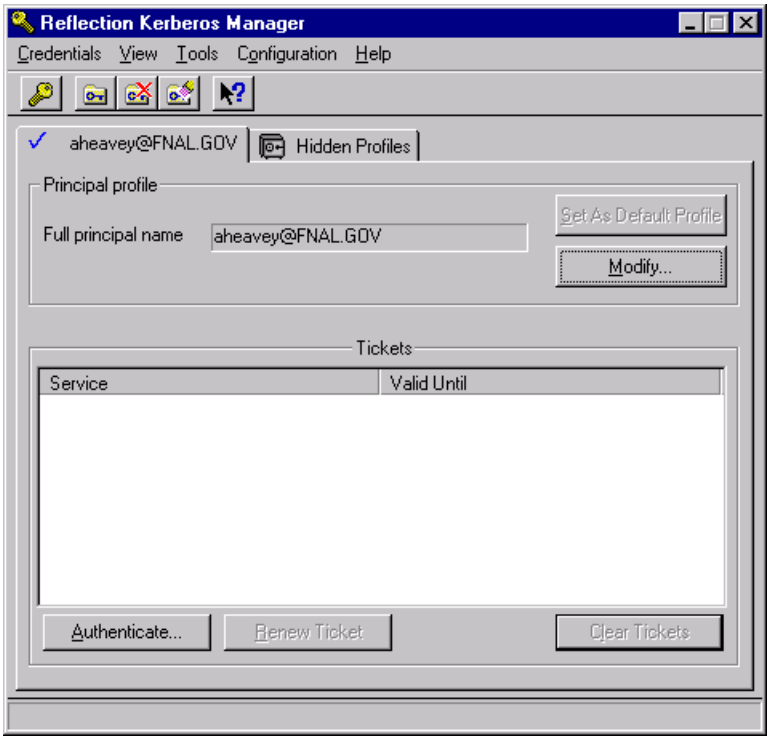
4.7 Logging In Through WRQ® Reflection Software from Windows

4.7.1 Authenticate Locally via the Kerberos Manager

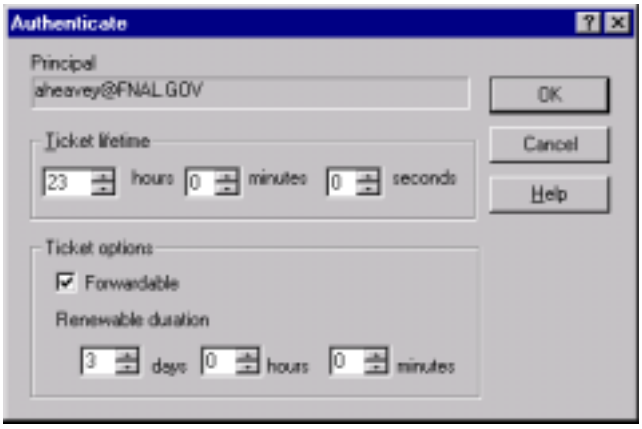
The **Reflection Kerberos Manager** program authenticates you to Kerberos and supports ticket forwarding. This means it obtains an initial Kerberos ticket for the principal on the tab chosen[1], and you, as that principal, can freely connect to Kerberized machines without needing to type your Kerberos password again.

Navigate to **START > PROGRAMS > REFLECTION > UTILITIES > KERBEROS MANAGER** to open the **Reflection Kerberos Manager** application.

1. You may have one for PILOT.FNAL.GOV and for FNAL.GOV.



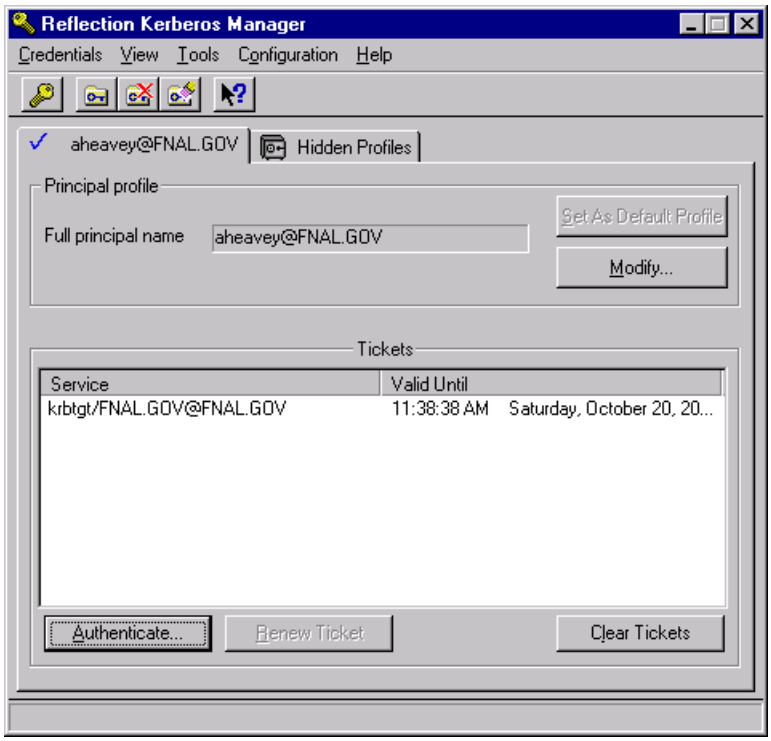
Choose your principal that corresponds to the default realm of the target machine. Click **AUTHENTICATE**.



- Verify or change **TICKET LIFETIME** (if you give a value greater than the KDC limit of 23 hours, the renewable lifetime will be set to 23 hours)
- Check **FORWARDABLE** in order to forward your ticket to target host (besides forwarding your Kerberos ticket, it's necessary in order for an AFS token to be automatically generated when you connect to a system running AFS)

- To set your ticket as renewable, enter a non-zero time for **RENEWABLE DURATION** (if you give a value greater than the KDC limit of seven days, the renewable lifetime will be set to seven days). The AFS token you get will have a lifetime equal to the Kerberos ticket's renewable duration.

Click **OK**, and provide your Kerberos password at the prompt. Back on the **KERBEROS MANAGER** window, you should see the new ticket-granting ticket (TGT) `krbtgt/FNAL.GOV@FNAL.GOV`.



If you receive an error message instead, check that the above steps were followed correctly and that you typed the right password. If you continue to receive an error message, send the exact error message text to `nightwatch@fnal.gov` together with the date and time of the error and the IP address of your system.

Once you run **Reflection Kerberos Manager** and authenticate, you do not need to keep the application active; you can exit and continue to log in to Kerberized machines. The authentication is valid for the lifetime of the ticket.

When you have finished your session and disconnected from all Kerberized machines, it's important to prevent another user at your machine from using your tickets. Bring up the application again and clear your tickets by clicking **CLEAR TICKETS** on the **REFLECTION KERBEROS MANAGER** window. You can automate this by clicking **CLEAR ALL TICKETS ON SHUTDOWN** on the **CONFIGURATION** menu.

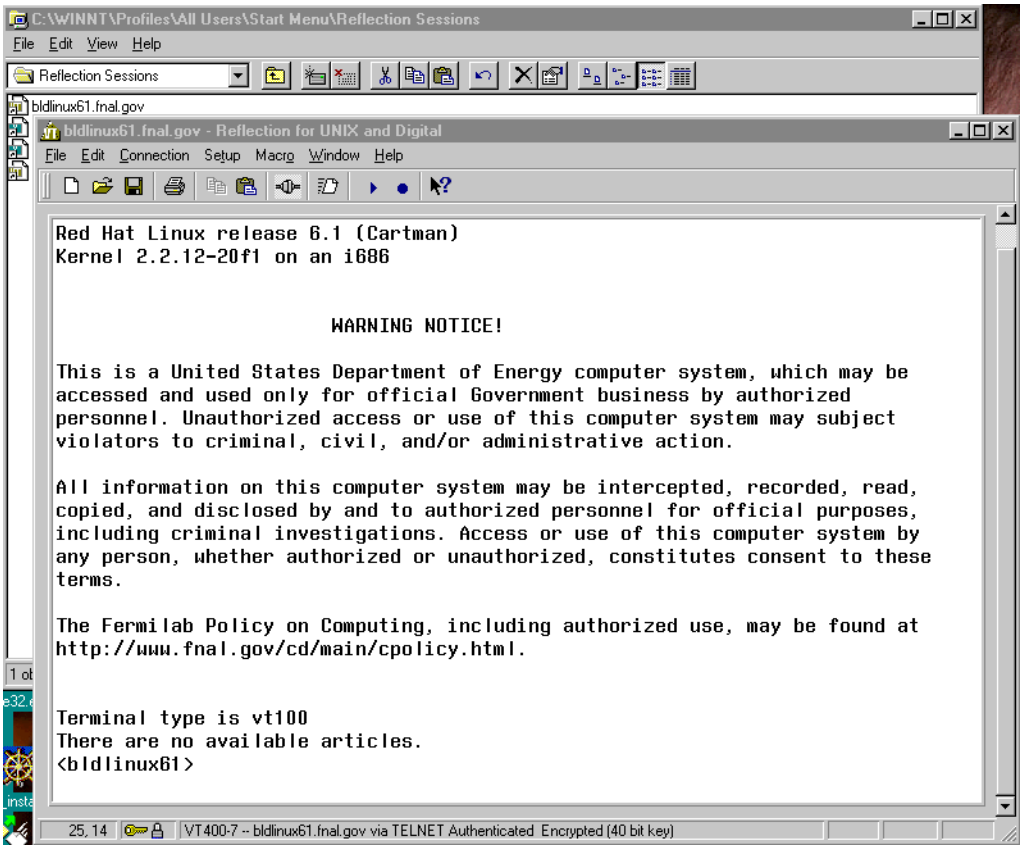
4.7.2 Run a telnet Session to Kerberized Host

To use the **WRQ® Reflection telnet** client to access machines in the strengthened realm, you first need to set (and save) a separate **telnet** configuration for each host with ticket forwarding set. The configuration procedure is outlined in section 19.6 *Configuring WRQ® Reflection telnet Connections*.

To run an Xwindows session, see section 10.1.2 *Windows NT4/98/95*.

Start the **Reflection Kerberos Manager** first to authenticate, as explained in section 4.7.1 *Authenticate Locally via the Kerberos Manager*. The easiest way to start a session is to make a short cut for your telnet configuration file, and just double-click on it. Otherwise, to start your session:

- Navigate to **START > PROGRAMS > REFLECTION > HOST - UNIX AND DIGITAL**.
- On the **REFLECTION FOR UNIX AND DIGITAL** window, select **FILE > OPEN**.
- Double click on the file in your **REFLECTION** folder corresponding to the host to which you want to connect. (If you haven't already authenticated you will be prompted to provide your Kerberos password.) It will bring up a VT window and log you in:



Assuming that you have authenticated with a forwardable ticket, and that your telnet configuration file specifies `Forward ticket`, then you have credentials on the host (including AFS token if needed).

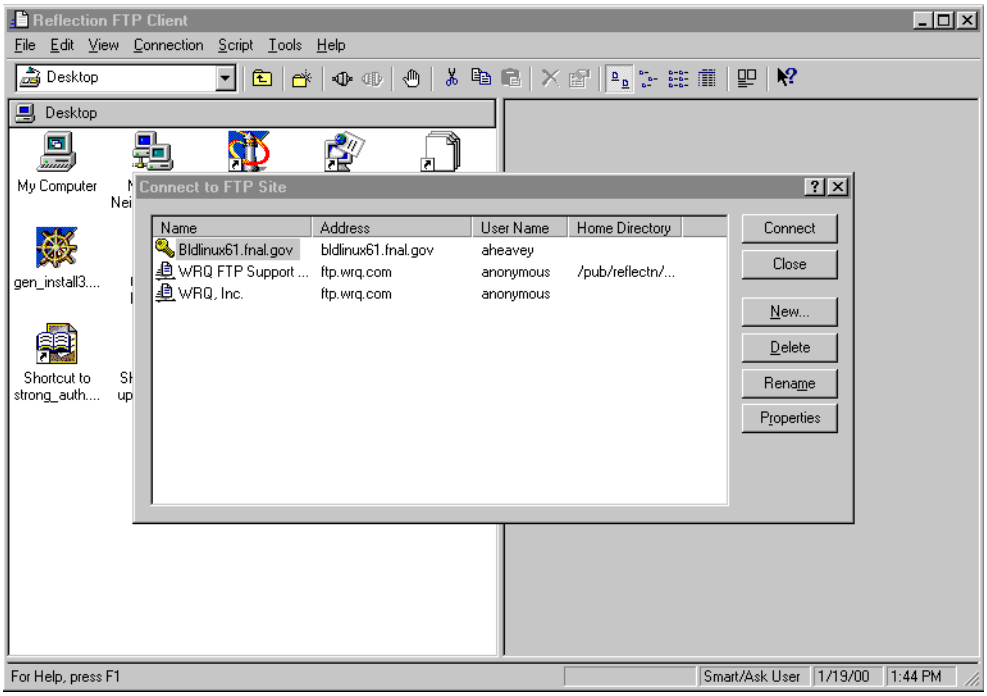
If you authenticate with the **Kerberos Manager** and get a nonforwardable ticket, and then start a telnet session with forwarding enabled, you'll get another password prompt from **WRQ®** so that it can obtain a forwardable ticket for you.

If you did not have your ticket forwarded, then to obtain credentials on the host (and to obtain an AFS token if AFS runs on the host) you will need to run **`kinit`** (see section 9.2.1 *Obtaining Tickets (Authenticating to Kerberos)*) and enter your password again after you log in. **Don't do this on a regular basis!** *Before* you enter your password, glance at the bottom of the VT window and verify that it says "Encrypted" and shows a locked lock icon (as shown on the above image). If it doesn't, *log out and verify your configuration* (under **CONNECTION>SECURITY**, check `Reflection Kerberos` and check `Encrypt data stream`)! **Always make sure the data stream is encrypted before entering your password!**

4.7.3 Run an FTP Session to Kerberized Host

Configuration of **FTP** sessions is covered in section 19.7 *Configuring WRQ® Reflection FTP Connections*. Make sure that the default realm for **REFLECTION** is set to the default realm of the target host (see number [3] in section 19.4 *Configuring WRQ® Reflection Kerberos Manager v9.0.0*).

To use the **Reflection FTP** client to access a Kerberos system: open **START > PROGRAMS > REFLECTION > FTP CLIENT**:



and double-click the file corresponding to the host you want to access.

AFS

WRQ® Reflection FTP does not forward ticket to remote host or obtain an AFS token for you on the host. This does not pose problems on non-AFS machines, but you can't get access to AFS volumes. For transferring files to AFS space, you have two options:

1) Install and use the Windows AFS client, as described in sections Chapter 20: *Installing and Configuring the Windows AFS Client* and 4.8 *Windows AFS Client for File Transfers to AFS Space.*

2) Configure the WRQ® FTP client with standard (nonKerberos) security and use a CRYPTOCard (this has also been tested with NT and Windows 2000 command line FTP, and FTP client in **FrontPage2000**).

   · Select **VIEW > COMMAND WINDOW** to see the CRYPTOCard challenge.

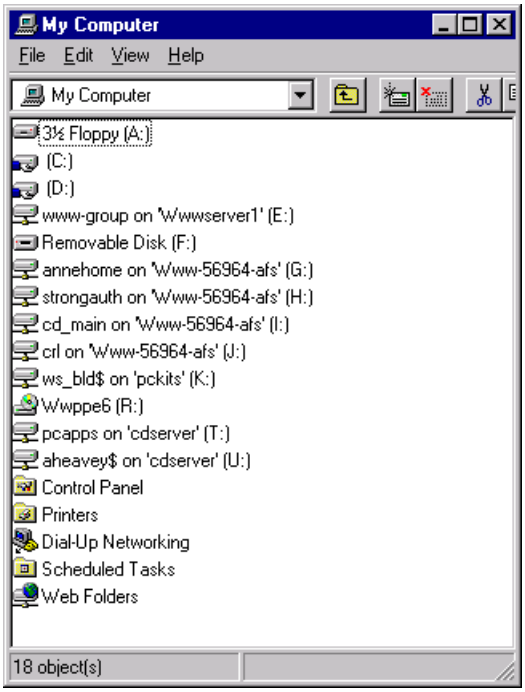
   · Connect to host, generate a response on your CRYPTOCard, and enter it at the password prompt.

4.8 Windows AFS Client for File Transfers to AFS Space

Due to the inability of the Kerberized FTP clients for Windows, including WRQ®'s, to forward Kerberos tickets (and thus generate AFS tokens on the remote host), we recommend that you bypass FTP entirely and install the Windows AFS client for file transfers to and from AFS space. Installation and configuration is described in Chapter 20: *Installing and Configuring the Windows AFS Client*.

4.8.1 How does AFS Appear on your Desktop?

The AFS client should be installed and configured such that at login the drive mapping is restored and the AFS client service restarts[1]. Your AFS drive(s) appear automatically in **MY COMPUTER**, **WINDOWS NT EXPLORER**, etc. In the image below, the drives G:, H:, I: and J:, labelled: `<description> on 'Www-56964-afs' (<drive letter>:)`, are all AFS volumes:



1. If the AFS Client Service does not start up automatically when machine is booted, click on the AFS icon on your task bar (the lock symbol; it will appear with a red X at this stage). Select the `Advanced` tab, and click **START SERVICE**. Also, if not remapped automatically at login, the AFS drive(s) must get mapped in the same way as any other drive.

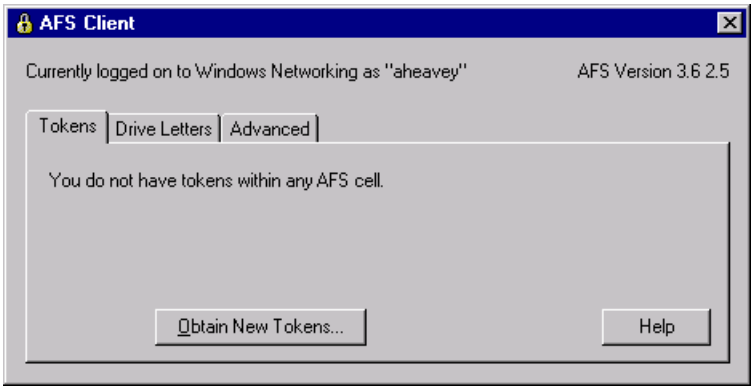
To use most AFS volumes, you must first authenticate to AFS. The exception is a public AFS volume (for which access is allowed for `system:anyuser`); this does not require a token[1].

The AFS icon in your task bar is a lock symbol. It displays a red X ( ) before you authenticate to AFS, and the X goes away ( ) after you authenticate to AFS and obtain a valid token.

4.8.2 Authenticate to AFS

1) Make sure the AFS Client Service is running.

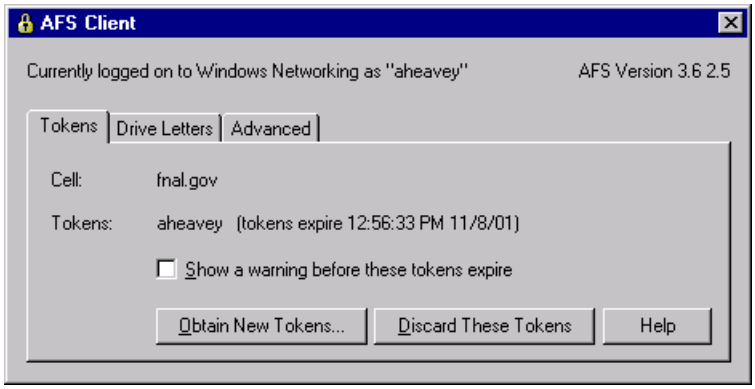
2) Authenticate to AFS space either by clicking on the AFS icon (the lock symbol with X:) on your task bar, or by navigating to **START > PROGRAMS > IBM AFS > CLIENT > AUTHENTICATION**. On the **AFS CLIENT** window, select the **TOKENS** tab. Click **OBTAIN NEW TOKENS...**



You will be prompted for your AFS password. (Currently this method does not require Kerberos authentication.)

3) The token expiration date/time then appears on the window:

1. If the AFS Client Service is not running, the AFS mapped drives display a red X and are unusable. The Xes go away when the service is restarted.



Your token is valid for six days, unless the AFS service is stopped before then. Every time you reboot, the service is halted and restarted, and thus the token is destroyed.

4) Now you're ready to copy/paste/edit files on the AFS volumes in the same manner as for other drives.

Chapter 5: Using your CRYPTOCard

Strengthened machines are configured to respond in *portal mode* when requests for access come from unKerberized machines. In portal mode the strengthened machine acts as a secure gateway into the strengthened realm, requiring a single-use password for authentication. A CRYPTOCard is a calculator-style, battery-powered device used for generating a single-use password.

To obtain a CRYPTOCard, go to the *Form to Request Kerberos Principal and/or Related Items* at `http://www.fnal.gov/cd/forms/strongauth.html`.

☞ As of March 2002, new CRYPTOCards operate a little differently from those previously sent from the vendor. When you get your CRYPTOCard, first carefully read the instruction card that comes with it.

5.1 How does your CRYPTOCard Work?

Before we issue you your CRYPTOCard, we initialize it and synchronize it with the Kerberos Key Distribution Center[1] (KDC). This process (a) associates the card with your principal, (b) sets an initial PIN on the card, and (c) creates a secret encryption key stored in both the KDC and the card.

1. The KDC is the "keymaster" of the Kerberos authentication service for all the machines in the realm. It runs on a server maintained by Fermilab's computing security team. Every principal and every initialized CRYPTOCard shares a unique encryption key with the KDC, allowing the KDC to verify the identity of each user/service request.

The KDC and the CRYPTOCard operate independently on the identical strings using the shared key, and they produce the same result. Roughly half of this resulting string is to be used as the first one-time password, the other half (plus/minus some overlapping bits) is stored for later use as the next string on which both parties will operate. And so on.

The string on which the shared key operates is called the *challenge*. The portion of the result used as the password is called the *response*. The first challenge is chosen by the KDC when you use the card.

5.2 Caring for your CRYPTOCard

You will find printed instructions with your new CRYPTOCard. Carefully read *Use and Care of your RB-1 Authentication Token*, and *Battery Replacement*.

Here we highlight a few points that we think are important:

- Your CRYPTOCard is relatively expensive; please don't lose it! Treat it as you would your house keys (if they were breakable!).
- Your CRYPTOCard looks the same as your colleague's, so make a note of the serial number printed on the back so that you can identify yours. Even though another person would need both your principal and your PIN to use your card, we recommend that you don't label your card with anything that resembles your principal. In most cases this means don't put your name on it. You can label it with a non-identifying word or sticker that you'll recognize.
- Don't drop, sit on or crush the card (don't carry it in your back pocket).
- Keep it dry and out of intense heat or cold. Don't let it go through the laundry, and don't leave it in your car in the winter or summer.
- When the display becomes dim, it's time to replace the batteries (two new CR2016, 3V lithium coin cells). CHANGE THEM ONE AT A TIME TO PREVENT DATA LOSS! Otherwise you will need to get the card reprogrammed.

5.3 Usage Notes

- We recommend using fingertips or a pencil eraser for pressing the CRYPTOCard buttons. Fingernails, pen tips and other sharp objects work less well. You don't need to remove it from the plastic cover to use it.

- When you first turn on the card, it takes a second or two to respond with a prompt.
- If you ever forget your PIN (see section 5.4) or if the card locks up (says "locked" when turned on), send email to `compdiv@fnal.gov` to arrange getting your CRYPTOCard reprogrammed. If you are on-site, you will need to come to WH8NE. If you are off-site, mention that in your email.
- Your CRYPTOCard will automatically turn itself off after 60 seconds unless it receives further input.

5.4 The First Thing to do: Reset your PIN

The CRYPTOCard comes with an initial PIN (personal code to prevent use by other individuals) that you are required to reset. The minimum length of the PIN is four digits, but it can be as long as eight. When entering your PIN, you are limited to seven consecutive wrong tries before lockout.

5.4.1 Resetting Initial PIN

Original Style Card

1) Press the **ON/OFF** button to turn on the card, enter your initial PIN and press **ENT**.

2) At the prompt `New PIN?` enter a new PIN and press **ENT**.

3) At the `Verify` prompt, enter your new PIN again and press **ENT**. The card displays a preconfigured string which you can ignore.

4) If you're not going to log on now, you can turn off the card or let it do so automatically.

New Style Card (March 2002)

1) Press **CHG PIN** (actually any of the 4 keys **PASSWORD**, **DIG SIG**, **MENU** and **CHG PIN** will work).

2) At the prompt: `PIN?` enter your initial PIN.

3) At the prompt: `New PIN?` enter a new PIN and press **ENT**.

4) At the `Verify` prompt, enter your new PIN again and press **ENT**. It displays: `Card OK`

5) If you're not going to log on now, you can turn off the card or let it do so automatically.

5.4.2 Resetting PIN (General)

Original Style Card

For subsequent PIN changes, turn the card on and enter your PIN followed by **ENT**. At the `Fermilab` prompt, press **CPIN** and proceed from step (2) for this style card, above.

New Style Card (March 2002)

For subsequent PIN changes, turn the card on using the CHG PIN button, and enter your (old) PIN followed by **ENT**. At the `New PIN?` prompt proceed from step (3) for this style card, above.

5.5 Log in Using CRYPTOCard (the First Time)

5.5.1 Original Style Card

1) Turn on your CRYPTOCard and enter your new PIN, followed by **ENT**.

2) The card is configured to display the id `Fermilab`. Press **ENT** when you see it. You'll see a preconfigured *challenge*, which you can ignore.

3) Run **ssh**, **slogin**, **telnet**, or **ftp** normally on your nonKerberized machine to the strengthened host, and enter your login id at the host prompt. The first time you use the card, the host system (in portal mode) displays the message:

```
Press CH/MAC and enter this on the keypad:
[12345678]
Enter the displayed response:
```

where `12345678` is a sample eight-digit *challenge.*

4) On your CRYPTOCard, press **CH/MAC**, then type the *challenge* displayed on the host system into your CRYPTOCard. If you mistype, press **CLR** and re-enter the *challenge*. Press **ENT** to get a *response* of eight hex digits.

5) Enter the CRYPTOCard *response* at the host system prompt (it is not case-sensitive). Press **RETURN**, and you should be logged in with Kerberos tickets.

6) Turn off your CRYPTOCard, or let it do so automatically.

5.5.2 New Style Card (March 2002)

Before the initial login, you need to synchronize the card with our KDC.

1) Run **ssh**, **slogin**, **telnet**, or **ftp** normally on your nonKerberized machine to the strengthened host, and enter your login id at the host prompt. The host system (in portal mode) displays an eight-digit *challenge*.

2) Press **MENU** to turn on your CRYPTOCard, and enter your PIN as required, followed by **ENT**.‘

3) Ignore the `Adj LCD` prompt and press **MENU** again.

4) At the prompt `ReSync`, press **ENT**.

5) At the prompt `Ready`, key the challenge displayed on your monitor into your CRYPTOCard, and press **ENT** to get a *response* of eight hex digits. (If you mistype, press **CLR** and re-enter the *challenge*. **CLR** clears one character at a time, or it will clear the whole field if held down for more than one second.)

6) The *response* (password) associated with that challenge now displays on the CRYPTOCard.

7) Enter the CRYPTOCard response at the host system prompt (it is not case-sensitive). Press **RETURN**, and you should be logged in with Kerberos tickets.

5.6 Log in Using CRYPTOCard (Subsequently)

5.6.1 Original Style Card

1) Turn on your CRYPTOCard and enter your PIN, followed by **ENT**. (You are limited to seven consecutive wrong-PIN tries before lockout.)

2) The card is configured to display the id `Fermilab`. Press **ENT** when you see it. The CRYPTOCard displays a *challenge*.

3) Run **ssh**, **slogin**, **telnet**, or **ftp** normally on your nonKerberized machine to the strengthened host, and enter your userid at the host prompt. The host system (in portal mode) displays the message:

```
Press ENTER and compare this challenge to the
one on your display: [12345678]
Enter the displayed response:
```

where `12345678` is a sample eight-digit *challenge*.

4) Compare the *challenge* on the host to the one on the CRYPTOCard:

a) If the *challenges* are the same, press **ENT** again on the CRYPTOCard to get the *response*. (In this case the KDC and your CRYPTOCard are synchronized. As long as they remain in sync, the CRYPTOCard will generate the right *response*.)

b) If the *challenges* are different (you may see all zeroes), press **CH/MAC** on the CRYPTOCard and enter the *challenge* displayed on the host system into the card. (This resynchronizes the CRYPTOCard.) Then press **ENT** to get the *response*.

5) Enter the *response* at the host system prompt. Press **RETURN** and you should be logged in with tickets.

6) Turn off your CRYPTOCard, or let it do so automatically.

5.6.2 New Style Card (March 2002)

There are two ways to use the CRYPTOCard to log in, one using the **PASSWORD** key and the other using **DIG SIG**.

PASSWORD

IN THIS MODE, THE CRYPTOCARD DOES NOT DISPLAY THE CHALLENGE!

1) Run **ssh**, **slogin**, **telnet**, or **ftp** normally on your nonKerberized machine to the strengthened host, and enter your userid at the host prompt. The host system (in portal mode) displays the message:

```
Press ENTER and compare this challenge to the
one on your display: [12345678]
Enter the displayed response:
```

where `12345678` is a sample eight-digit *challenge*.

2) Press **PASSWORD** to turn the CRYPTOCard on

3) At the `PIN?` prompt, enter your PIN followed by **ENT**.

4) The card is configured to display the id `Fermilab`. Press **ENT** when you see it.

5) The card now displays the response, not the challenge! If the card is synchronized with the KDC, this response will work. If not, using **DIG SIG** (below) will work, but before ever using **PASSWORD** again, you'll have to resynchronize your card.

6) Enter the response at the host system prompt. Press **RETURN** and you should be logged in with tickets.

DIG SIG

This method works even if your card has gotten out of sync (assuming that initial synchronization has been done), but it does not resynchronize your card for future logins. A drawback to this method is that you have to key the challenge into your CRYPTOCard each time.

1) Run **ssh**, **slogin**, **telnet**, or **ftp** normally on your nonKerberized machine to the strengthened host, and enter your userid at the host prompt. The host system (in portal mode) displays the message:

```
Press ENTER and compare this challenge to the
one on your display: [12345678]
Enter the displayed response:
```

where `12345678` is a sample eight-digit *challenge*.

2) Press **DIG SIG** to turn the CRYPTOCard on

3) At the `PIN?` prompt, enter your PIN followed by **ENT**.

4) At the `Ready` prompt, enter the challenge (displayed on your monitor) into the CRYPTOCard, and press **ENT**. (If you mistype, press **CLR** and re-enter the challenge. **CLR** clears one character at a time, or it will clear the whole field if held down for more than one second.)

5) The card now displays the response.

6) Enter the response at the host system prompt. Press **RETURN** and you should be logged in with tickets.

5.7 Reauthenticate using your CRYPTOCard

To remain logged in and reauthenticate safely, issue the command:

```
% new-portal-ticket
```

This provides a portal mode prompt, and allows you to use your CRYPTOCard as in section 5.6 *Log in Using CRYPTOCard (Subsequently)* to get new tickets. E.g.,:

```
Press ENTER and compare this challenge to the one on your
display: [12345678]
Enter the displayed response: <enter response>
18960 Terminated
Connection closed by foreign host.
```

Don't be dismayed by the messages that appear! The **`new-portal-ticket`** command works by opening a telnet connection to "localhost" and letting the user answer the portal challenge. There's a sleep command going on to keep the telnet connection from closing too soon, and `Terminated` comes when that sleep is no longer needed and is killed by the script. `Connection closed...` comes when that telnet session is over.

5.8 Resync your CRYPTOCard

5.8.1 Original Style Card

Commence the login procedure as outlined in 5.6 *Log in Using CRYPTOCard (Subsequently)*. If the *challenges* are different, press **CH/MAC** on the CRYPTOCard and enter the *challenge* displayed on the host system into the card. (This resynchronizes the CRYPTOCard.) Then press **ENT** to get the *response*.

5.8.2 New Style Card (March 2002)

1) Run **ssh**, **slogin**, **telnet**, or **ftp** normally on your nonKerberized machine to the strengthened host, and enter your login id at the host prompt. The host system (in portal mode) displays an eight-digit *challenge*.

2) Press **MENU** to turn on your CRYPTOCard, and enter your PIN as required, followed by **ENT**.

3) Ignore the `Adj LCD` prompt and press **MENU** again.

4) At the prompt `ReSync`, press **ENT**.

5) At the prompt `Ready`, key the challenge displayed on your monitor into your CRYPTOCard, and press **ENT**. (If you mistype, press **CLR** and re-enter the *challenge*. **CLR** clears one character at a time, or it will clear the whole field if held down for more than one second.)

Your card is now resynchronized and the correct *response* now displays on the CRYPTOCard. You can complete your login at this point by typing the response at the host system prompt, followed by **RETURN**.

Chapter 6: Logging In from Off-Site

In this chapter, we discuss what off-site users are required to do in order to access Fermilab's strengthened realm, and some of the issues they may encounter.

Due to practical considerations, namely the fact that off-site machines at universities may be shared by many people, some of whom do not access Fermilab at all, off-site users are not required to install a Kerberos 5 server. Off-site machines participating in Fermilab's strengthened realm have a choice of authentication methods, including ssh with passwords, public/private keys, host-based keys or Kerberos. Access to a system on-site at Fermilab requires Kerberos credentials or a CRYPTOCard.

6.1 Description of Choices for Off-Site Machines

The choices for off-site machines include:

1) Install the Kerberos client (and optionally the Kerberized ssh client) software on your machines and sign up to be part of the FNAL.GOV strengthened realm. This means you can authenticate to Kerberos locally and connect to Fermilab computers using the Kerberized version of a network connection program. This is the preferred method.

2) Leave your machines unstrengthened and always log in to Fermilab using your CRYPTOCard (see Chapter 5: *Using your CRYPTOCard*). Note that if you choose to do this, we recommend that you use ssh as the transport program in order to ensure encryption. You must NEVER type in your password if you are on an unencrypted channel! There is no way to perform any Kerberos command that requires a password while logged in using an X-terminal. And please, as much as possible, refrain from performing operations that involve typing your Kerberos password over the network.

3) Your site may have its own version of strong authentication which may be acceptable to Fermilab and then you could become a trusted realm.

4) In addition, a stripped-down kerberos product exists for emergency off-site use, e.g., for people who've misplaced their CRYPTOCard. It is called **FNAL-kerberos-clientonly** and is described in section 6.2 *In a Pinch: Download Client-Only Version of Kerberos*. This product is intended for temporary use. People using the same machine repeatedly will likely find a full Kerberos installation more useful and convenient.

The Cryptography Publishing Project is making MIT Kerberos V5 release 1.2.1 available for export without restriction (software for Macintosh excepted); see `http://www.crypto-publish.org/`.

If people need to log in from your site to change their passwords, there must be at least one local machine on which there is software which will allow it to be done locally (best) or over an encrypted connection (second best). The fnkerb.fnal.gov system, described in section 4.1 *Trying Out Kerberos on fnkerb.fnal.gov*, is a Kerberized host available for changing passwords. It is accessible to anyone with an account on FNALU.

6.2 In a Pinch: Download Client-Only Version of Kerberos

New as of June 2002:

FNAL-Kerberos-clientonly is a stripped-down version of Fermi Kerberos containing only the client applications and supporting files needed to connect to an FNAL Kerberized machine from a remote location. It is intended for temporary use by off-site users who have neither a CRYPTOCard nor a machine with a Kerberos installation available. **FNAL-Kerberos-clientonly** is publicly-available, it is provided in tar format, it can be downloaded via a web browser and installed in any user directory, and it does not require root/administrator privileges to operate.

FNAL-Kerberos-clientonly versions have been created for RedHat Linux 7.1 and compatible systems, and for Windows 2000 (other Windows systems have not been tested but may work). Look for the software in the FermiTools area of Fermilab's FTP server: `ftp://ftp.fnal.gov/pub/fnal-kerberos-clientonly/current/`. Instructions on how to setup and uninstall the software are included in the product.

6.3 Obtaining CRYPTOCards

All users, on-site and off-site, can request a CRYPTOCard using the *Form to Request Kerberos Principal and/or Related Items* at `http://www.fnal.gov/cd/forms/strongauth.html`. If you visit Fermilab occasionally, come by WH8NE to pick it up when it's ready. For those experimenters or other users who will not be visiting Fermilab, CRYPTOCards can be mailed. Each group or experiment should have a person designated to mail CRYPTOCards; contact the appropriate person to request mailing.

If you lose your CRYPTOCard or it becomes unusable for any reason, please email *compdiv@fnal.gov* or call Yolanda Valadez at 630-840-8118 to request a new one. Then ask the person designated for your group or experiment to pick it up from her and mail it to you. Currently we do not have a way of restoring your access more quickly. By the end of 2001, we expect to have a mechanism in place whereby we can fax you a one-time password.

6.4 Exporting CRYPTOCards

For users outside the U.S., you can carry a CRYPTOCard back to your home or institution with no customs problems since the cards are for authentication, not encryption. They can be mailed outside the U.S., too.

6.5 Network Address Translation

There is an issue concerning users who maintain a small network of computers at home and whose ISP subjects them to NAT (Network Address Translation). Typically, the user dials up with a NAT box or a Linux host configured to do NAT for the house network, and receives one address from his or her ISP. This address may be static or dynamic. In either case, NAT can make it difficult or impossible to authenticate over the network to the FNAL.GOV realm.

There are a couple of solutions, one of which is to keep your home machines unKerberized, and use a CRYPTOCard. If you want to Kerberize your home machines, we would first recommend that you change ISPs to one that eschews NAT. Barring that, you may be able to work around NAT:

- *if* your home machines (Linux or Macintosh) have **kerberos** installed,

- *if* there is a single fixed "public" IP address associated with your machines' real IP addresses, and the outside world sees this public IP address as the source of packets that come from your machines,
- and *if* you can determine this fixed IP address.

To be able to authenticate, you'd need to include this public IP address in your local `/etc/krb5.conf` file under the `[libdefaults]` section as: `proxy_gateway = <fixed.IP.address>`. If the address is dynamic, this solution will rapidly become annoying, no doubt.

We recommend that you use ssh to connect to the lab. Kerberized ssh is of course best, but any ssh with CRYPTOCard works too (only one CRYPTOCard use per remote host, not per window). Because ssh includes an automatic tunnel for X sessions, most users will find this more convenient than telnet/rlogin connection methods.

6.5.1 Windows

If you've installed **WRQ®** on your Windows system(s), you will not be able to authenticate if your ISP uses NAT. Remove this software from your system(s) and use a CRYPTOCard. The vendor is aware of this problem, and may address it in future releases of the software.

In the meantime, you can use a combination of an ssh client (e.g., F-secure) with Exceed or the Reflection X Manager (but make sure your site is not behind a firewall in addition to NAT).

6.5.2 Linux

If you install Linux, configure your machine such that its hostname is equivalent to the external hostname your ISP uses, then install a Kerberos client. (If you're not sure how to configure, send an email to *kerberos-users@fnal.gov*, or check the archives.)

6.5.3 Macintosh

To enable **BetterTelnet** to work for a Kerberized Macintosh in a NAT environment, you must add the following line to the `libdefaults` section of the `Kerberos Preferences` file (Note that this reduces the security of your Kerberos credentials.):

```
noaddresses = true
```

Forwardable tickets to do not work. Opening a connection with **BetterTelnet** results in a dialog box from the Kerberos5 Telnet Plugin about the forwarded credentials being refused due to bad address. Clicking **OK** will result in the telnet connection opening as expected, otherwise.

Chapter 7: Accessing Kerberized Machines (Community-Supported Methods)

In this chapter we discuss accessing systems in the FNAL.GOV realm from UNIX, Windows and Macintosh machines using programs or operating systems not supported at Fermilab.

Very important note: Any time you're about to enter your Kerberos password, first verify that you're using the host's directly-connected keyboard or using an encrypted connection! Otherwise you risk exposing your password. See Chapter 11: *Encrypted vs. Unencrypted Connections* for information.

7.1 Logging In Through Kerberized Exceed 7 Software from Windows

7.1.1 Telnet Connections

You should create one secure telnet profile for each Kerberized host you wish to access, according to the instructions in section 22.5 *Configuring the Exceed 7 Telnet Application*. To authenticate:

- using the **Leash32** utility, navigate to **START** > **PROGRAMS** > **KERBEROS UTILITIES** > **LEASH32**. Select **GET TICKET** on the **ACTION** menu.

  You will be required to enter your Kerberos password. Ignore the CRYPTOCard prompt that may follow (press **CANCEL**). You ticket will appear in the **Leash32** window. Click on the Windows Explorer-style plus signs (+) to get details.

- using the command prompt, type **`kinit -5`** to request a ticket.

  You will be required to enter your Kerberos password. Ignore the CRYPTOCard prompt that may follow (just press **ENTER**). To verify the ticket and its flags, either bring up the **Leash32** window, or type **`klist -f`** at the command prompt.

  You can request a renewable ticket at the command prompt by using the **`-r`** option (see section 9.2.5 *Renewing Tickets*). Your AFS token will have a lifetime equal to the renewable lifetime of the Kerberos ticket.

To connect:

1) Start the Exceed 7 telnet program. Navigate to **START** > **PROGRAMS** > **HUMMINGBIRD CONNECTIVITY V7.0** > **HOSTEXPLORER** > **TELNET**.

2) On the **OPEN SESSION** window, with the desired telnet profile selected, the target host name or IP address should appear in the Host Name window. To connect, click on the **CONNECT** button. If you've preauthenticated, you should get right in without having to provide your Kerberos password.

3) The **LEASH32** window should now show your host connection in addition to the kerberos ticket.

7.1.2 FTP Connections

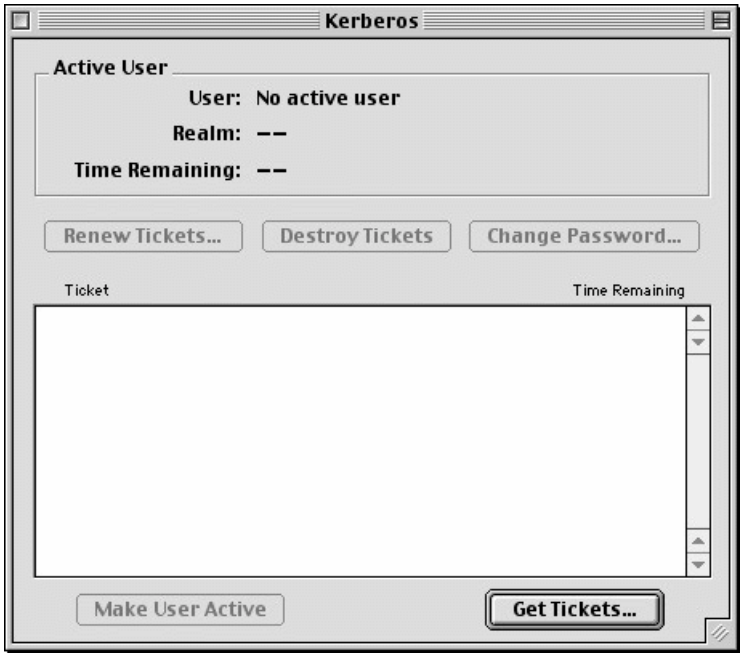
Exceed 7 does not provide a Kerberized FTP client. Furthermore, you cannot connect using your CRYPTOCard (as you may for WRQ® FTP, described in section 4.7.3 *Run an FTP Session to Kerberized Host*), since the Exceed 7 FTP client stores your password, and doesn't let you enter it each time you connect. Choose a different product! Suggestions: WRQ®, FileZilla, AFS Windows Client (for remote hosts using AFS).

7.2 Logging In from a Macintosh

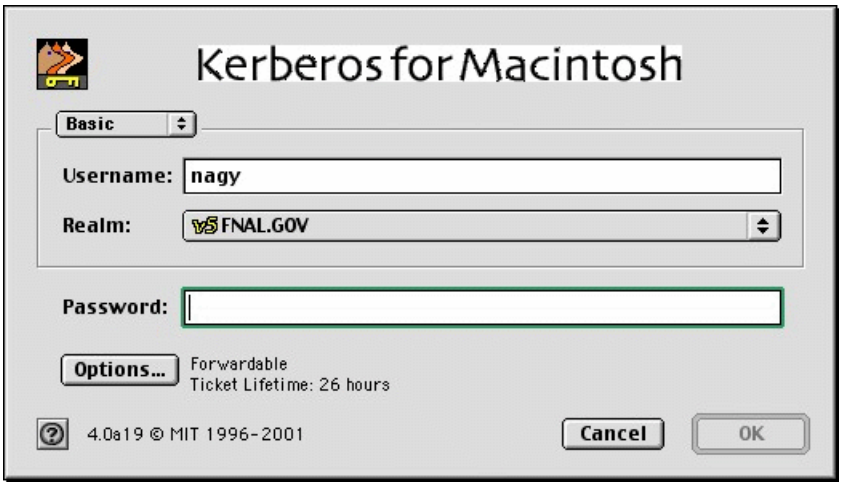
Here we assume you are running the **MIT Kerberos** v4.0 software for Macintosh as described in Chapter 24: *Installing and Configuring MIT Kerberos on a Macintosh System*.

7.2.1 Authenticate via Kerberos Control Panel

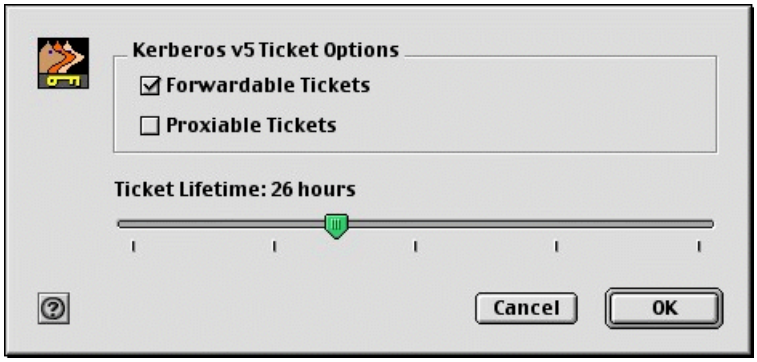
- Invoke the **Kerberos Control Panel** (from **CONTROL PANELS** under the Apple menu, from the **KERBEROS MENU** in the menu bar, or from the **KERBEROS CONTROL STRIP** module).



• Click **GET TICKETS**.

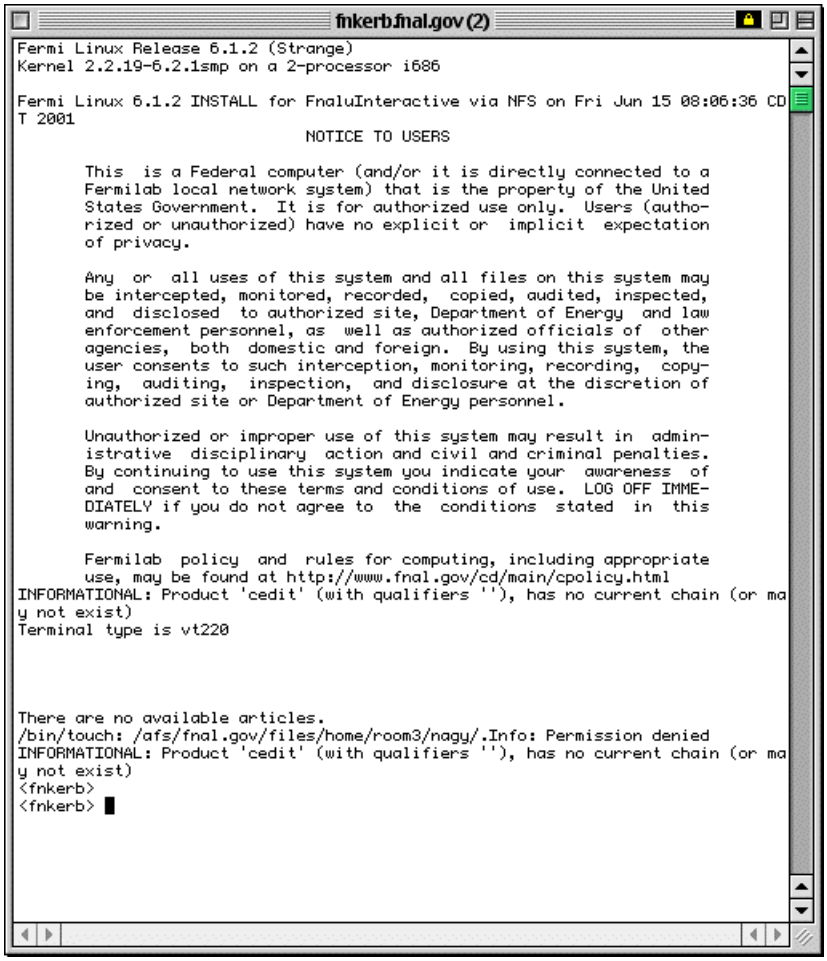


• Select the right username and realm. Click **OPTIONS...** to specify the ticket options; you should generally choose forwardable.



• Click **OK**. Then enter your Kerberos password on the pop-up screen.

You should see a ticket appear. Now you can invoke your **telnet** product (**BetterTelnet** or **NiftyTelnet**) and connect to one or more strengthened hosts without having to provide your password again.



7.2.2 Authenticate at Login

Invoke **BetterTelnet** or **NiftyTelnet** and connect to a strengthened host. You will be prompted for your Kerberos password, and then authenticated once you have provided it.

Chapter 8: Troubleshooting your Authentication Problems

This chapter is intended to help users who are having trouble authenticating to Kerberos and logging in to Kerberized machines. We include information that should help you figure out what's causing your problem, and to fix it.

If you don't find the solution to your problem here, send mail to *kerberos-users@fnal.gov* requesting help in diagnosing the failure. Please include: principal name, date, time and IP address from which authentication failed, in addition to the error message and other error-related information.

- In many cases, when authentication fails, one of four things is likely to be wrong:
  - (1) your password,
  - (2) the date/time on your system (see section 14.1.7 *Synchronize your Machine with Time Server* for UNIX, 19.3 *Time Synchronization* for Windows, or 24.1.4 *Installation Instructions* for Macintosh),
  - (3) the local host name in the `/etc/hosts` file (see section 16.3 *The /etc/hosts File*), or
  - (4) your CRYPTOCard is not configured for the target realm. The error message doesn't necessarily help you determine the problem: "`Preauthentication failed ...`", or "`Cannot establish a session with Kerberos administrative server...`" If this is the problem, bring your card to WH8NE to have it reprogrammed.

  For **WRQ** connections, click **HELP** for possible causes. It's usually a realm mismatch, a wrong password, or a system clock error.
- "Incorrect net address" usually refers to NAT (see section 6.5 *Network Address Translation*) or a multiple-IP address host. For UNIX, edit the `[libdefaults]` in `/etc/krb5.conf`: add `proxy_gateway=<your fixed IP address>`. For **WRQ**, there is no solution other than to change ISP or **WRQ** software. For Macintosh, edit the `[libdefaults]` in the Kerberos Preferences file: add `noaddresses=true`.
- YP problem: The error "`do_ypcall: clnt_call: RPC: Timed out`" typically indicates a local problem on your system or site network. Your machine is likely using YP (NIS) for host name-to-address resolution and you have a transient problem with your YP server(s).

- When using the Kerberized versions of **telnet**, **rlogin**, or **rsh** (see Chapter 13: *Network Programs Available on Kerberized Machines*) to connect to another machine in the strengthened realm, some users have had to use the **`-l <login_name>`** option even when the login names on both systems match. (Don't ask why.) You definitely need to use this option if the login names don't match.
- "KDC policy rejects request" or "KDC can't fulfill requested option" usually means either you're requesting a forwardable ticket for a /root or /admin instance of your principal (not allowed), or you're trying to forward a ticket that's not forwardable, or renew one that's not renewable.
- "Key version number for principal in key table is incorrect" means either the keytab has changed since the service ticket was obtained (to solve, run **`kinit -R`** or **`kinit`**), or the service key (for host principal) in the KDC was changed after the keytab file was created (to solve, recreate keytab file on host, see section 16.10 *Installing Service Host Keys*).
- "Cannot contact any KDC for requested realm." Caused by firewall blocking KDC request or reply, or DNS failure.
- "Server not found in Kerberos database" Possible causes include: local hosts file or NIS map giving wrong name for host (check `/etc/hosts` file and make sure the full official host name appears first, not a nickname; see section 16.3 *The /etc/hosts File*), or a bad or missing `[domain_realm]` mapping in `/etc/krb5.conf`. It was also a bug in Fermi Kerberos v1_2; to solve, upgrade.
- "aklog: Couldn't get fnal.gov AFS tickets:, aklog: unknown RPC error (-1765328352) while getting AFS tickets". You may have failed to get fresh tickets from your screensaver unlock. A fresh **`kinit`** should clear this right up.
- Syslog message: Principal <principalname>@FNAL.GOV ... for local user <user> failed krb5_kuserok. `krb5_kuserok` is a function in the kerberos library. It is accessed by krshd, and fails for these reasons:
  - requested user has no account on target system
  - `krb5_unparse_name` fails
  - can't open `~user/.k5login`
  - `~user/.k5login` not owned by user or root
  - principal doesn't match any line in `.k5login` (try **`od -c ~user/.k5login`** to look for any "invisible" characters in this file).
- If Kerberos functions are very slow on a client host, check its Kerberos logs for long intervals between "NEEDED_PREAUTH" and "ISSUE" and see if there are few or no repeats of the same request to different KDCs. If so, the client host's first-configured DNS server may be slow or dead.

To resolve this, check the DNS server list (`/etc/resolv.conf` on UNIX-like systems, Network Control Panel on Windows) and test each one, moving dead servers down in the list or removing them.

SSH Problems

- Make sure the instance of the **ssh** product you're using matches the OS version of your target UNIX machine.
- When you use the Kerberos-aware ssh or scp client (v1_2_27f) to connect to a node that's running a non-Kerberos-aware sshd, the client ignores a `.shost` file on the remote node. It tries Kerberos, that of course fails, then it prompts for a password. Supplying the password works. (This is an unavoidable side-effect.)
- Some users of Kerberized **ssh** v1_2_27 have encountered a harmless but misleading message upon authentication:

  ```
  aklog: can't get afs configuration
  (afsconf_Open(/usr/vice/etc))
  ```

  To get rid of this message, add `AFSRunAklog no` to `/etc/sshd_config` and restart **sshd**.
- Logins from Kerberized **ssh** clients to unstrengthened **ssh** servers can fail. This does not happen with the Fermi **ssh**. You can work around this by explicitly using the **`-l <login_name>`** option even if the login names on both systems match. (Again, don't ask why.)
- If you get prompted for a password when you login from a machine with Kerberized ssh, and you already have valid tickets, check to make sure the following line is in the `[domain_realm]` section of your `/etc/krb5.conf` file:

  ```
  .fnal.gov = FNAL.GOV
  ```

  Kerberized ssh token-passing won't work without it, nor will FTP.

Chapter 9: Using Kerberos

This chapter provides the information you need in order to manage your Kerberos tickets and work in a Kerberized environment. In particular, we cover ticket options and management, account access files and /root principal tickets. The Kerberos commands and features of Kerberized network programs are documented in Chapter 12: *Kerberos Command Descriptions* and Chapter 13: *Network Programs Available on Kerberized Machines*, respectively.

9.1 Ticket Properties and Options

Kerberos uses encrypted records called *tickets* to authenticate to Kerberized services (the terms *tickets* and *credentials* are used interchangeably). Tickets reside in a file called a ticket cache or credentials cache. Generally the only ticket you need to know about is the ticket-granting-ticket (TGT), which you obtain upon authentication to Kerberos. Kerberos tickets can be forwardable, renewable, post-dated and/or proxiable. The Kerberized versions of network programs generally provide options to exploit these features (see Chapter 13: *Network Programs Available on Kerberized Machines*).

Forwardable	A forwardable ticket can be "passed on" to a remote host, thereby allowing the user to connect to the host without further authentication. Generally only the TGT is set forwardable, since it can be used to obtain other needed tickets.
Renewable	A renewable ticket can have its lifetime extended, by action of the user, beyond the initial lifetime, up to an established limit (seven days at Fermilab).
Post-dated	A post-dated ticket becomes valid at a specified time in the future.
Proxiable	A proxiable ticket is like a forwardable ticket, except that the new ticket with the new address list is not allowed to be a TGT, it must be for some other service.

Our Kerberos implementation is integrated with AFS. This means that if your machine is part of the strengthened realm and it runs AFS, then when you obtain Kerberos credentials (or forward them to an AFS system), you also automatically get an AFS token. The other operations described in this chapter (e.g., listing, destroying tickets) also run on both the Kerberos tickets and the AFS token. The lifetime of the AFS token is set to the renewable lifetime of the Kerberos TGT.[1] (Note that if you're editing a file when the AFS token expires, it will suddenly become write-protected!)

9.1.1 Default Ticket Flags and Lifetimes

At Fermilab, the maximum ticket lifetime is set to 26 hours, and the default ticket lifetime as set on individual systems is constrained to be this value or less. The default flags and lifetimes of tickets obtained on a UNIX machine by login and **`kinit`** are set by entries in that machine's `/etc/krb5.conf`. (For other operating systems, the default values are typically set via a more user-friendly interface.) The maximum renewable ticket lifetime is seven days. We discuss the `krb5.conf` file in Chapter 17: *The Kerberos Configuration File: krb5.conf*.

9.1.2 Ticket Caches

A *ticket cache* is a file containing your tickets and session keys. Each window on your desktop that is running a remote session has a separate ticket cache[2], with a separate expiration. The variable $KRB5CCNAME points to the credentials cache in use on each host.[3] Note that forwarded tickets and tickets obtained via **`kinit`** are stored in different caches.

9.1.3 Tickets for Root Instance of Kerberos Principal

The system administrator of a strengthened machine may require that authorized users obtain a *<username>/root* instance of their Kerberos principal in order to access the root account (and/or other sensitive accounts) on the machine. This is described in section 9.4.1 *What is a Root Instance of a Principal?* The */root* instance has the properties of disallowing forwardable tickets and having a shorter default ticket lifetime.

---

1. Because AFS uses the Kerberos V4 ticket format, which squeezes the ticket lifetime into a small field, the expiration time of the AFS token may not *exactly* coincide with the end of the Kerberos ticket's renewable lifetime.
2. In some cases, there may be more than one per window.
3. Tickets generated by kinit end up in /tmp/krb5cc_[UID], forwarded tickets go to /tmp/krb5cc_p[PID], and hardware token tickets go into /tmp/krb5cc_[ttyname].

9.2 Ticket Management

9.2.1 Obtaining Tickets (Authenticating to Kerberos)

The way to authenticate depends on your operating system and software. Upon authentication you get a Kerberos ticket-granting-ticket (TGT). As you access Kerberized services in the strengthened realm, the tickets needed for the services are granted automatically. As regular practice, authenticate locally and forward tickets to remote machines.

As of Kerberos v1_5, the **kinit** program is equipped with a warning that appears if the userid issuing the command doesn't own the console device. It is designed to help users avoid typing their password inadvertently over then network.

To authenticate:

UNIX desktop with Kerberos software and Kerberos login program	Log in, and provide your Kerberos password. See section 4.2 *Logging In at the Console of a Kerberized UNIX Machine*.
UNIX desktop with Kerberos and standard UNIX login program	Log in with your UNIX password, then run **`kinit`**. See section 4.2 *Logging In at the Console of a Kerberized UNIX Machine*. Also see 12.1 *kinit*.
Windows desktop with WRQ®	Navigate to **START > PROGRAMS > REFLECTION > UTILITIES > KERBEROS MANAGER** to open the **Reflection Kerberos Manager** application. With your principal tab selected, click **AUTHENTICATE**. Provide your Kerberos password as prompted (and click **FORWARDABLE**). See section 4.7 *Logging In Through WRQ® Reflection Software from Windows*.
Windows desktop with Leash32 and Kerberos	Using the Leash32 utility, navigate to **START > PROGRAMS > KERBEROS UTILITIES > LEASH32** . Select **GET TICKET** on the **ACTION** menu. Provide your Kerberos password as prompted. See section 22.4 *Getting a Ticket* .
Macintosh desktop with Kerberos	Invoke the **KERBEROS CONTROL PANEL** (from **CONTROL PANELS** under the Apple menu, from the **KERBEROS MENU** in the menu bar, or from the **KERBEROS CONTROL STRIP** module). Click **GET TICKETS**. Enter your Kerberos password on the pop-up screen. See section 7.2 *Logging In from a Macintosh*.
Remote UNIX host (from desktop with no Kerberos software installed)	Start an ssh (or telnet or FTP) session to a Kerberized host, use your CRYPTOCard to generate a password, and log into the remote host using that one-time password. See section 4.5 *Connecting from a NonKerberized Machine: Portal Mode*.

When you're logging in as *root* you have to make sure you have tickets as some principal known to the KDC in order to access Kerberos network services. Whether you logged in as yourself and ran **ksu** to *root*, or logged in as *<yourprincipal>/root* over the network, you have credentials for the principal under which you previously authenticated.

If you have a laptop that you move from one network to another, then you will have to reobtain your credentials when you move to a new network because the IP address changes. Similarly, if you use DHCP, every time your IP address changes you need to get new credentials.

9.2.2 Viewing Tickets

The way to view your tickets depends on your operating system and software. Valid and expired tickets alike will be displayed.

To view tickets:

UNIX desktop with Kerberos software	Run the **`klist`** command (**`-f`** option recommended to show ticket flags). See section 12.2 *klist*.
Windows desktop with WRQ®	Navigate to **START > PROGRAMS > REFLECTION > UTILITIES > KERBEROS MANAGER** to open the **Reflection Kerberos Manager** application. Ticket should be visible on this window. Right-click on ticket to see ticket properties. See section 4.7 *Logging In Through WRQ® Reflection Software from Windows*.
Windows desktop with Leash32.	Using the Leash32 utility, navigate to **START > PROGRAMS > KERBEROS UTILITIES > LEASH32** . Ticket should be visible on this window. See section 22.4 *Getting a Ticket*.
Macintosh desktop with Kerberos	Invoke the **KERBEROS CONTROL PANEL** (from **CONTROL PANELS** under the Apple menu, from the **KERBEROS MENU** in the menu bar, or from the **KERBEROS CONTROL STRIP** module). Ticket should be visible on this window. See section 7.2 *Logging In from a Macintosh*.
Remote Kerberized UNIX host	Run the **`klist`** command (**`-f`** option recommended to show ticket flags). See section 12.2 *klist*.

About the klist Command

The command **`klist`** displays your tickets (the **`-f`** option displays the flags set for the tickets), e.g.,:

```
% klist -f
```

This produces output of the form:

```
Ticket cache: /tmp/krb5cc_6302
Default principal: aheavey@FNAL.GOV
```

```
Valid starting     Expires            Service principal
12/08/99       11:29:47              12/09/99       00:29:47
krbtgt/FNAL.GOV@FNAL.GOV
        Flags: FIA
12/08/99 11:29:48  12/09/99 00:29:47  afs/fnal.gov@FNAL.GOV
        Flags: FA
```

- The first listed ticket is a Kerberos TGT (`krbtgt`) for the service principal `krbtgt/FNAL.GOV@FNAL.GOV`[1]. Underneath it the flags are listed. This ticket has flags set for “forwardable”, “initial”, and “preauthenticated”.
- The second listed ticket indicates that AFS is running on this machine and that an AFS token has also been granted; this is again followed by a list of the flags associated with the ticket.

If you have no tickets you will see output like this:

```
klist: No credentials cache file found (ticket cache /tmp/krb5cc_6302)
```

Several options are available for **`klist`**, as listed in section 12.2 *klist* and in the man pages.

9.2.3 Destroying Tickets

Tickets can outlive an interactive session and they can be stolen. They are just encrypted records in a file. Therefore it’s a good idea to explicitly destroy your tickets when you log out. Similarly, if you are going to be away from your machine but don’t want to log out, it is safest to either destroy your tickets, or use a screensaver that locks the keyboard.

To destroy tickets:

UNIX desktop with Kerberos software	Run the **`kdestroy`** command. This destroys all the tickets in the cache to which $KRB5CCNAME points. To automate this, add the command **`kdestroy`** to your `.logout` file. See section 12.4 *kdestroy* or the man pages for a description of **`kdestroy`**. If you’re sharing a credentials cache among several login sessions (by setting the $KRB5CCNAME variable), issuing the **`kdestroy`** command on any of the sessions destroys the tickets for all of them.

1. See *principal* in the *Glossary* for an explanation of the syntax.

Windows desktop with WRQ®	Navigate to **START > PROGRAMS > REFLECTION > UTILITIES > KERBEROS MANAGER** to open the **Reflection Kerberos Manager** application. Tickets should be visible on this window. Click on **CLEAR TICKETS**. To automate the clearing of tickets, you can click **CLEAR ALL TICKETS ON SHUTDOWN** from the **CONFIGURATION** menu.
Windows desktop with Leash32	Using the Leash32 utility, navigate to **START > PROGRAMS > KERBEROS UTILITIES > LEASH32** . Ticket should be visible on this window. Click on **DESTROY TICKET(S)**. To automate the clearing of tickets, you can click **DESTROY TICKETS/TOKENS ON EXIT** from the **OPTIONS** menu to clear tickets when you exit Leash32.
Macintosh desktop with Kerberos	Invoke the **KERBEROS CONTROL PANEL** (from **CONTROL PANELS** under the Apple menu, from the **KERBEROS MENU** in the menu bar, or from the **KERBEROS CONTROL STRIP** module). Ticket should be visible on this window. Click on **DESTROY TICKETS**.
Remote Kerberized UNIX host	Run the **`kdestroy`** command.

Destroying Tickets Selectively

If you have several tickets in your cache and you run **`kdestroy`**, you'll destroy them all. But say you want to destroy only one or some of them. If your TGT is renewable, running **`kinit -R`** will discard all but the TGT, which gets renewed. If your tickets are forwardable, you can forward the TGT alone to your own machine by **`rsh`** or other program, and then overwrite your existing cache, e.g.,:

```
% rsh -F 'hostname' cp \$KRB5CCNAME $KRB5CCNAME
```

(Backquotes around *hostname*) If the KRB5CCNAME value has `FILE:` on the front of it (true of the recent kerberos releases), the preceding command will fail; in this case, try:

```
% rsh -F 'hostname' cp ''echo $KRB5CCNAME | sed -e sxFILE:xx''\
  'echo $KRB5CCNAME | sed -e sxFILE:xx'
```

(All the quotes are backquotes.) To do anything more specific you'd have to write a program with the credentials cache API (which is beyond the scope of this document).

9.2.4 Forwarding Tickets

You can use your current, valid credentials on your desktop to get valid credentials on another machine by forwarding them.[1] You should forward tickets if you plan to use Kerberized services on the remote host (e.g., if you plan to connect from there to another remote Kerberized machine) and/or if you need an AFS token. To forward tickets, there are two steps:

1) you must first obtain a forwardable ticket,

2) and then make sure the "forward" option is used by your connection program.

The way to do this of course depends on your OS and software:

UNIX desktop with Kerberos software and Kerberos login program	To obtain a forwardable ticket, the `/etc/krb5.conf` must show `forwardable = true` for `login`) under `[appdefaults]`). If not, check for `forwardable = true` for `kinit`. If this is true, run **`kinit`**. If false, run **`kinit -f`**. To forward your forwardable ticket to a remote UNIX host, use a Kerberized connection program with ticket forwarding on[a] (e.g., **`telnet -F`**).
UNIX desktop with Kerberos and standard UNIX login program	To obtain a forwardable ticket, check for `forwardable = true` for `kinit` in `/etc/krb5.conf`. If true, run **`kinit`**. If false, run **`kinit -f`**. To forward your forwardable ticket to a remote UNIX host, use a Kerberized connection program with ticket forwarding on (e.g., **`telnet -F`**). (See footnote a.)
Windows desktop with WRQ®	To obtain a forwardable ticket, click **FORWARDABLE** when you authenticate. See section 4.7 *Logging In Through WRQ® Reflection Software from Windows.* To forward your forwardable ticket to a remote telnet session, verify that the telnet configuration file you're using specifies **FORWARD TICKET** on the **SECURITY PROPERTIES** window. See section 19.6 *Configuring WRQ® Reflection telnet Connections.* Note: WRQ®'s FTP client doesn't support forwarding tickets. This only poses a problem for remote hosts running AFS since you don't get your AFS token upon connection. See section 4.7.3 *Run an FTP Session to Kerberized Host.*

1. The KDC administrator has the option of disallowing forwardable tickets on a per-site or per-principal basis.

Windows desktop with Leash32, MIT Kerberos and Exceed 7	To obtain a forwardable ticket, make sure your configuration specifies `Forwardable` under **TICKET OPTIONS** as described in section 22.3 *Configuring Kerberos using Leash32*. To forward your ticket to a telnet session, verify that the telnet configuration file you're using specifies `Forwarding` under **KERBEROS 5 OPTIONS**. See section 22.5 *Configuring the Exceed 7 Telnet Application*. Then run the telnet client. Note: The Exceed 7 FTP client cannot be Kerberized; try FileZilla FTP.
Macintosh desktop with Kerberos	To obtain a forwardable ticket, edit your Preferences and check **FORWARDABLE TICKETS ALWAYS**. To forward the ticket via a BetterTelnet connection, check **KERBEROS FORWARDING** when you're configuring the Security portion of Favorites for that application.
Remote Kerberized Host via Portal Mode	When you obtain your ticket upon CRYPTOCard login to a remote host, the ticket's properties are determined by the `/etc/krb5.conf` file on the host. Run **`klist -f`** to see if the `F` flag shows up indicating a forwardable ticket. If it doesn't, and if you used ssh to connect thus providing an encrypted connection, then you can run **`kinit -f`** to get one, BUT ONLY RARELY! To forward your forwardable ticket to a remote UNIX host, use a Kerberized connection program with ticket forwarding on.

a. Check for `forward = true` in `[appdefaults]` section of `/etc/krb5.conf` for your program of choice (ssh has its own configuration). If false, use the program's command line option for ticket forwarding; hese are documented in Chapter 13: *Network Programs Available on Kerberized Machines*.

Descriptions of the forwarding option (and other Kerberos functions) added to the connection programs in the Kerberos V5 package can be found in Chapter 13: *Network Programs Available on Kerberized Machines* and at `http://www-dcd.fnal.gov/computersecurity/StrongAuth/UserDocs/user-guide.html#SEC16`.

Tickets and IP Addresses: How forwarding works

A ticket normally includes a list of IP addresses from which it may be used. A forwardable ticket may be presented to the KDC to obtain a ticket with a different address list, which can then can be forwarded to another host and used from there.

The IP address (or list of IP addresses) of the client is encoded inside of every Kerberos ticket. This information is used by application servers and the KDC to verify the address of the client. By default, then, a ticket that was acquired

on one host cannot be used on another. This is where forwarding comes in. A forwardable ticket (usually a TGT) can be used to request a new ticket, but with a different IP address.

The new IP addresses to be included in a forwarded ticket are determined from the DNS entry for the target hostname. If that host turns out to have other IP addresses which are not listed under that name, the forwarded ticket may or may not be usable, depending on how that host routes packets to the KDC or to the other nodes you try to access.

A Note about AFS tokens and Forwarding

Telnet, **rsh** and **rcp** and **ftp** work without strictly requiring that credentials be forwarded. These programs always present a service-specific credential to get access, but don't necessarily forward it to the remote system.

- For **telnet**, you typically want to forward your credential (and automatically obtain an AFS token as needed), in order to avoid running `kinit` over the network. But if you don't plan to make any further connections from the remote host, and AFS is not running, forwarding is not strictly necessary.
- For **rsh** you'd only need to forward if the remote process you're invoking might need to make a further network access, or access files in an AFS file system.
- For **rcp** and **ftp** only the AFS case would lead you to want to forward credentials.

A Word about Ticket Caches and Forwarding

Forwarding actually involves asking the KDC to rewrite the ticket to be valid from the remote machine instead of from your desktop. In the case of telnet, the telnetd on the remote host receives the forwarded ticket, creates a credential cache file in `/tmp` and puts its name into the variable $KRB5CCNAME. The shell spawned by telnetd inherits this variable, so any kerberos client programs you run in that shell will use the forwarded ticket in that cache. If you then start an xterm process, it and the shell (or other process) it spawns inherit this environment variable and therefore know where to find your ticket. When the shell process created by telnetd exits, telnetd destroys the credential cache it created -- unless the host's `/etc/krb5.conf` tells telnetd "retain_ccache = true". As a user, you have no control over that setting.

Example (UNIX)

You will automatically obtain a forwardable ticket if under `[appdefaults]` in `/etc/krb5.conf` you see `forward=true` set for `kinit` or `login`, depending on how you got your ticket. You can always run **`klist -f`** and look for the `F` flag in the output if you're not sure:

```
12/08/99 11:29:47  12/09/99 00:29:47  krbtgt/FNAL.GOV@FNAL.GOV
        Flags: FIA
```

If you need to replace your ticket with a forwardable one, run **`kinit -f`**.

Now, to forward this ticket to a remote host via telnet, first check under `[appdefaults]` in `/etc/krb5.conf` to see if `forward=true` is set for telnet. If so, just run **`telnet <host>`**. If not, run **`telnet -f <host>`** or **`telnet -F <host>`**. With **`-f`**, the forwarded ticket on the remote host is not set as reforwardable, and thus you can't forward it from that host to another. With **`-F`**, the forwarded ticket is marked as reforwardable from that host.

9.2.5 Renewing Tickets

In order to support both long interactive sessions and batch jobs, tickets can be issued as *renewable*[1], and given a *renewable lifetime*. This lifetime must be less than or equal to the maximum allowable renewable lifetime, which is set to seven days at Fermilab. A renewable ticket still has the normal lifespan (up to 26 hours), but before it expires it can be renewed as long as its renewable life has not expired. Once the ticket expires, new connections cannot be opened, but existing connections are not terminated. The lifetime of the AFS token that you get is equal to the Kerberos ticket's renewable lifetime.

1. If the `/etc/krb5.conf` file on the machine sets `renewable=true` and `default_lifetime=<value greater than 26 hours>`, the user will get a renewable ticket by default when they first log in. The Fermilab template for this file does not set `renewable=true`, but the system administrator can change this.

Make sure you read about **k5push** in section 9.2.6 *Update Tickets on Remote Terminal Sessions*, which renews tickets on multiple remote sessions simultaneously. For a local session, how you go about requesting a renewable ticket and renewing it depend upon your OS and software:

UNIX desktop with Kerberos software	To request a renewable ticket, use **`kinit -r <renewable_lifetime>`**. This requires password entry, therefore it must only be performed at the keyboard of a strengthened machine or (infrequently) over an encrypted connection. To renew the ticket, use **`kinit -R`** before the ticket expires. **`kinit -R`** does not require password entry.
Windows desktop with WRQ®	To request a renewable ticket, navigate to **START > PROGRAMS > REFLECTION > UTILITIES > KERBEROS MANAGER** to open the **Reflection Kerberos Manager** application. With your principal tab selected, click **AUTHENTICATE**. Provide a non-zero value for **RENEWABLE DURATION**. See section 4.7 *Logging In Through WRQ® Reflection Software from Windows.* To renew the ticket, again open the **Reflection Kerberos Manager** application. With your principal tab selected, click
Windows desktop with Leash32 and Kerberos	To request a renewable ticket, use the command prompt, and type **`kinit -r <renewable_lifetime>`**, as for UNIX. To renew the ticket, use the **`kinit -R`** option before the ticket expires. **`kinit -R`** does not require password entry.
Macintosh desktop with Kerberos	It appears that tickets obtained via the Macintosh Kerberos software are renewable by default (although the "R" flag does not appear). To renew a ticket, invoke the **KERBEROS CONTROL PANEL** (from **CONTROL PANELS** under the Apple menu, from the **KERBEROS MENU** in the menu bar, or from the **KERBEROS CONTROL STRIP** module). Click **RENEW TICKETS...**.
Remote Kerberized host via Portal Mode	Run the command **`new-portal-ticket`** and use your CRYPTOCard.

Example

Request a renewable ticket with a maximum renewable lifetime of four days using the **`-r`** option:

```
% kinit -r 4d

   Password for aheavey@FNAL.GOV:   <--- type your password here.
```

Then, before the default lifetime of 26 hours has passed (you cannot renew an expired ticket), and before four days expire, renew the ticket using the **`-R`** option:

```
% kinit -R
```

The ticket will remain active an additional 26 hours or until its original four day term expires, whichever comes first.

9.2.6 Update Tickets on Remote Terminal Sessions

What do you do when you have connections open to remote machines, and your tickets on these machines expire? Well, you most certainly *don't* run **`kinit`** over the network! And it turns out you don't have to exit and restart each session, either:

- You can push your valid tickets from your local machine to these remote machines via a script called `k5push`.
- From Windows (using **WRQ® Reflection**) you need to connect to a remote UNIX host first and run `k5push` from there, as we'll show you.
- For a session authenticated using a CRYPTOCard, use **`new-portal-ticket`**, as described in section 5.7 *Reauthenticate using your CRYPTOCard*.

k5push

Authenticate to Kerberos locally first before using `k5push`. The `k5push` script connects to an open session on a remote UNIX system using Kerberized **`rsh`**, and updates the remote ticket cache file in `/tmp` with the new tickets from your desktop machine. `k5push` does not create a ticket cache; one must already exist on the remote node. To run the script, type this command at your local session prompt:

```
% k5push <host1> [ <host2> <host3>...]
```

The script makes quite a few checks to make sure that the ticket file is really one of yours, and belongs to a running session. The `k5push` script is included in the Fermi Kerberos product as of v1_5. It is also available from `http://www.fnal.gov/docs/strongauth/misc/k5push_script.txt`.

k5push options

1. You can run this to an account with a different name:

```
% k5push <username>@<host> [ [<username>@]<host2> ...]
```

but be aware that if the target account is a shared account, you might update other users' ticket files with your tickets.

2. You can keep a list of systems to update in a text file, and run:

```
% k5push -f <file>
```

to update them simultaneously. (From UNIX, this file must be local; from Windows, this file must be on the UNIX host to which you connect.) The text file must list hosts and/or accounts on hosts each on a separate line, e.g.,:

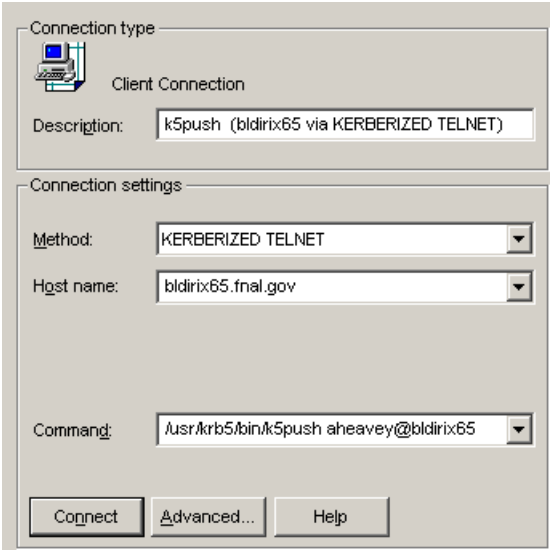
```
<host>.fnal.gov
<host>.<domain>
<account>@<host>.<domain>
```

Using k5push from Windows with WRQ®

As usual, use the **WRQ® Reflection Host - UNIX and Digital** program to run your remote VT100 sessions. Use the **WRQ® Reflection X Client Manager** to run the **`k5push`** command on a remote UNIX host session. If you use the **`-f`** option with a file, the file must exist on this UNIX host.

To update tickets on a single remote session:

- verify that the **k5push** program exists on the remote UNIX host, and find its path (usually `/usr/krb5/bin`)
- invoke the **WRQ® Reflection X Client Manager**, if it's not already running
- on the right half of the **X Client Manager** window (as displayed in the default "Split Window Vertically" view), do the following:
  - add a description (e.g., `k5push (hostname via KERBERIZED TELNET)`)
  - select `KERBERIZED TELNET` as the connection method
  - enter the remote host name
  - enter the command **`/path/to/k5push [username@]host`**

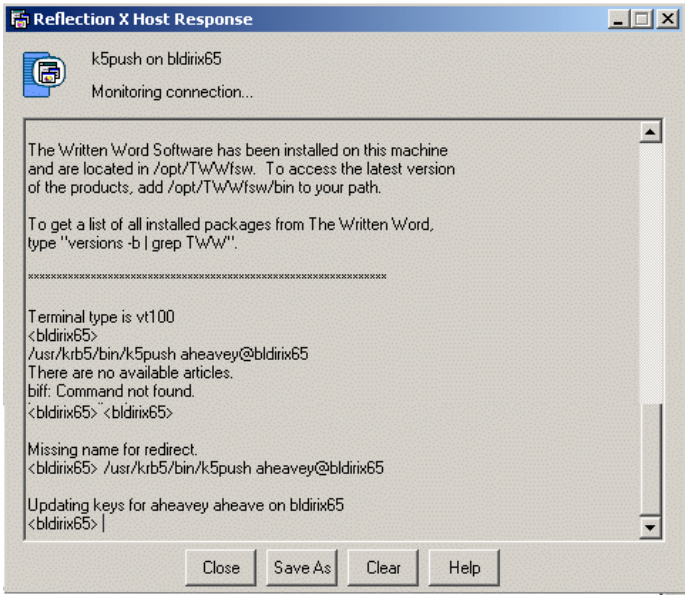


- Still on the **X Client Manager** window, click **ADVANCED...**

- On **ADVANCED CLIENT CONNECTION SETTINGS**, make sure the prompt symbol you need is shown; also check **NEVER CLOSE CLIENT STARTER CONNECTION** (and if shown, check **HOST RESPONSE**).
- Click **CONFIGURE KERBEROS**.
- On **SECURITY PROPERTIES**, verify that **REFLECTION KERBEROS**, **MUTUAL AUTHENCATION**, and **FORWARD TICKET** are checked. Verify that principal, realm and User ID are correct. Click **OK**.
- Back on **ADVANCED CLIENT CONNECTION SETTINGS**, click **OK**.

• Back on the **X CLIENT MANAGER** window, if necessary, open **CONNECTION > HOST RESPONSE** to monitor the process. Click **CONNECT**.

In the **HOST RESPONSE** window, you should see a session open to the remote host and see that it runs the **`k5push`** command as you entered it:



If you have multiple remote sessions and want to update credentials on all of them simultaneously:

- first choose one of your remote VT100 sessions as the “primary”
- on the primary host, verify that the **k5push** program exists, and find its path (usually `/usr/krb5/bin`)
- create a file on the primary host with all the necessary hostnames, as described above
- invoke the **WRQ® Reflection X Client Manager**, if it’s not already running
- fill in the right half of the **X Client Manager** window as described above, including the **ADVANCED...** options. Replace the command with:

```
% /path/to/k5push -f filename
```

9.3 Account Access by Multiple Users

Kerberos provides a way to grant account login access to multiple users, each with his/her own principal. There must be a `.k5login` file in the account's home directory and the principals must obtain credentials before logging into the shared account.

9.3.1 The .k5login File

The `.k5login` file is a text file that may exist in an account's home directory on a UNIX machine. It contains a list of the principals who have permission log into the account. Authenticated principals that are listed in the file can log in and use the account without limitations. A `.k5login` file is valid only on the individual strengthened host on which it resides.

Make sure that all principals that require login access are listed in it, **including your own FNAL.GOV principal**! Each principal must be on a separate line, with no trailing blanks.

This file overrides all other rules for granting login access!

Do you need a .k5login file?

As long as the only principal to log into your account is your own FNAL.GOV principal, and your principal matches your login id, you don't need a `.k5login` file. If other principals need login access to the account, and/or if your login id doesn't match your principal, you need one. And it must include your own principal!

Sample .k5login

```
xsmith@FNAL.GOV
qjones@FNAL.GOV
jenniferp@FNAL.GOV
jpedersen@MYUNIV.EDU
```

9.3.2 About Group Accounts

Sharing of any Kerberos password is a violation of Fermilab policy. Therefore, a multiple-user account must have a `.k5login` file in its home directory containing an entry for each user that needs to log into the account. The account may have but does not need a corresponding principal.

AFS ACLs should be set up so that everyone in the group can read (and write, if necessary) the files with his/her own AFS login and token. (This avoids the problem of running **klog** with a group AFS password.)

Users log in to the multiple-user account as follows:

1) Authenticate to Kerberos under your own account.

2) Log in to the multiple-user account, by identifying it on the connection program command line, and forward the ticket, e.g.,

```
% telnet -f -l <group-account-name> <host>.
```

3) Assuming tickets are automatically forwarded, you're now logged on under the account name, but your Kerberos ticket and AFS token are associated with your principal name.

4) Run **klog** to get an AFS token for the group account. If AFS is installed, you need to set the ACLs for file permissions for each principal.

9.3.3 The .k5users File

If you want to give restricted super user access to your account to another principal (access method limited to **ksu**; see section 13.7 *Kerberized su (ksu)*), you can create a `.k5users` file. The `.k5users` file is similar to the `.k5login` file, except that each principal is optionally followed by a list of commands which restricts the principal to those commands, and the file is only consulted by the **ksu** command.

Here is a sample `.k5users` file:

```
firstuser@MYUNIV.EDU /bin/ls /usr/bin/more
seconduser@MYUNIV.EDU /bin/ls
jenniferp@FNAL.GOV
jpedersen@MYUNIV.EDU
```

This restricts the first and second listed principals to the shown commands, and prohibits `jenniferp@FNAL.GOV` and `jpedersen@MYUNIV.EDU` from executing any command.

Two bombs:

- Be aware that arbitrary flags and arguments may be given to the listed commands by the authorized **ksu** user.
- If you list a principal more than once in this file, only the first entry is used.

If AFS is installed, you need to set the ACLs for file permissions for each principal.

9.4 Using Root Instance of your Principal

9.4.1 What is a Root Instance of a Principal?

A Kerberos principal has three parts and is of the form `primary/instance@REALM`. For a user, the instance portion is generally null, and the principal is of the form `primary@REALM`. If the instance is not null, the instance portion gives information that qualifies the primary, and is generally used to describe the intended use of the corresponding credentials. The root instance of a principal is also called a */root* principal. The word *root* in *<username>/root@FNAL.GOV* need not have anything to do with the UNIX *root* account, although that is presumed to be one of the most common uses. All */root* principals are created with the DISALLOW_FORWARDABLE flag set so that tickets are always unforwardable. The tickets also have a shorter default lifetime.

A root instance of your principal is only useful if your system administrator wants to make use of its restrictive ticket properties to protect sensitive accounts. Typically these accounts are set up with a `.k5login` file containing only */root* principals. Your system administrator should inform you if you need to obtain a */root* principal.

9.4.2 How do You Use your /root Principal?

To connect to such an account via a network connection from your desktop, you need to first **`kinit`** on your local machine as *<user>/root* (we use *me/root* as an example) and specify "nonforwardable ticket" with the **`-F`** flag[1]:

```
% kinit -F me/root[@FNAL.GOV]
```

Now, connect to the sensitive account on the remote host using all of the options shown here:

```
% telnet -x -N -l <sensitive_account_name> <remote_host>
```

where:

- **`-x`** encrypts the connection (generally a good idea)
- **`-N`** tells the program not to forward tickets (you'll get an error if you fail to include this)

1. If the Kerberos configuration file (`/etc/krb5.conf`) specifies forwarding "on" and you leave off the **`-F`**, you'll get an error.

- **`-l <sensitive_account_name>`** logs you in directly to the named account. If you didn't include this, the remote host would try to log you into an account with the same name as your local UNIX username.

Note that once you're logged in remotely, you have no tickets. You cannot use any Kerberized services from here to connect to other accounts or machines.

If the sensitive account is in AFS space, or if you require read/write access to nonpublic AFS areas from that account, you need to authenticate the *machine* to AFS. Contact your AFS administrator for assistance.

9.4.3 How Should You NOT Use It?

There are some limitations associated with the use of */root* principals for access to privileged accounts, and that is why their use is not mandatory.

- You can't use **`ksu`** (or other Kerberized client) on a remote machine under your */root* principal because you don't have tickets on that machine.
- Never type your */root* principal password over the network except on rare, necessary occasions; always authenticate on your desktop machine.

- Do not use your */root* principal with unencrypted CRYPTOCard connections, and rarely if at all with encrypted CRYPTOCard connections. Remote authentication for the */root* principal would require transmission of the password.

9.4.4 How do you Maintain Credentials for your Normal Principal while Using the /root Principal?

To maintain tickets on your desktop machine for both instances of your principal, you must keep the ticket caches separate. First authenticate under your normal principal, e.g.,:

```
% kinit [me[@FNAL.GOV]]
```

This gets you a ticket cache in the default area. You may find it useful to pick one of your local xterm windows to use for your */root* principal (maybe give it a special title bar or color) and set a separate ticket cache file there. In that window, reset the environment variable KRB5CCNAME to a location for the */root* principal ticket cache, then authenticate under your */root* principal to get (nonforwardable) tickets for this instance without overwriting the ones you got as "yourself":

```
% setenv KRB5CCNAME /tmp/krb5cc_me_root_$$
% kinit -F me/root[@FNAL.GOV]
```

When you request a Kerberized service, Kerberos will look at the credential cache to which KRB5CCNAME points, and assume that the principal holding this cache is the requestor. Reset this variable to the other cache as necessary.

Chapter 10: Miscellaneous Topics for the User

In this chapter we document a variety of common operations that work differently in the Fermilab Kerberized environment.

10.1 Running Xwindows

10.1.1 UNIX

Typically, a process on a remote kerberized host isn't automatically given access to your local X display (as it is when you use **ssh**). There are a few solutions to this. One is to use the kerberized **ssh** v1_2_27g which is now available from the KITS repository. Another is to use **kerberos** and give access with **`xauth`**, e.g.,:

```
% rsh -n -f -x <remote_host>.fnal.gov -l <username> \
  xauth add ‘xauth list $DISPLAY‘
```

(Those are backquotes around **`xauth list $DISPLAY`**.) Executing a command like this can be made more convenient. You can create an alias or shell script that sends over your **`xauth`** magic cookie (or performs an **`xhost +<remote_node>`** locally, if you use **`xhost`**, but it's considerably less safe -- someone on that host could get access to your screen and keyboard). Run it before starting the connection program (**telnet** will forward the $DISPLAY, other utilities will not). Change the script to mode 755. Here is some sample content for such a script, which we call `kxtelnet` (this script is untested, but it's similar to a script that's known to work!):

```
#!/bin/sh
if [ $# != 2 ]; then
  echo "usage: kxtelnet RemoteHostName RemoteUserName" 1>&2
  exit 1
fi
case "$DISPLAY" in
  :*) disp=‘hostname‘$DISPLAY;;
  *)  disp=$DISPLAY;;
esac
/usr/krb5/bin/rsh -n -x -l $2 $1 xauth add ‘xauth list $disp‘
exec /usr/krb5/bin/telnet -x -l $2 $1
```

(In the 7th line, `:*) disp=‘hostname‘$DISPLAY;;`, those are single backquotes around `hostname`. Same for `xauth list $disp` in 2nd to last line.) Instead of a script, you can set up an alias for a command like the following, and run it each time you restart Xwindows, before connecting to the remote host:

```
% xauth nlist <localnode>:0 | ssh <remotehost> xauth nmerge -
```

or

```
% xauth nlist <localnode>:0 | rsh -f -x <remotehost> \
  xauth nmerge -
```

Run this manually rather than with **`startx`** so that you can still get into Xwindows if for some reason this fails.

10.1.2 Windows NT4/98/95

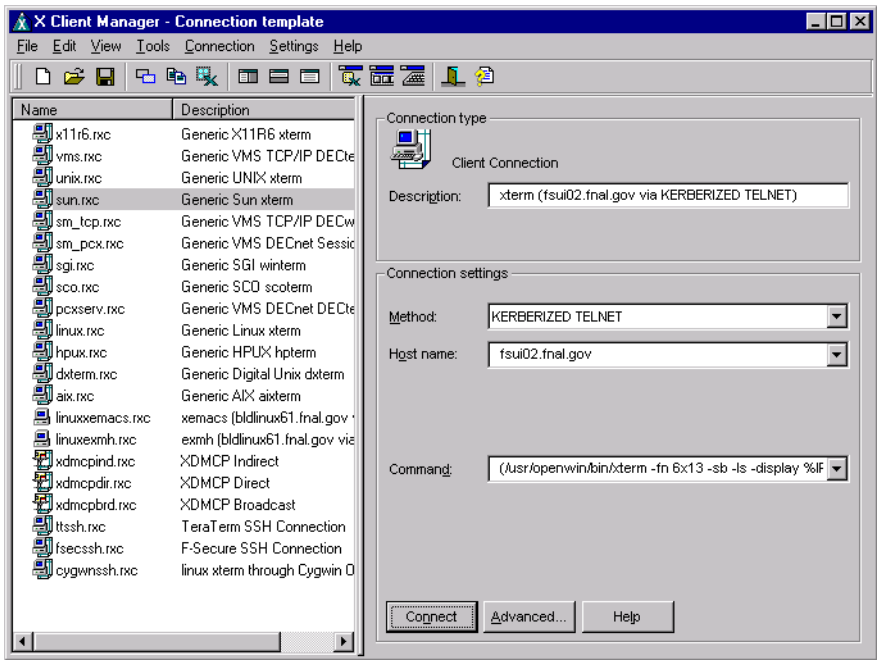
If you plan to run any X applications, you’ll need an X window manager. The **Reflection X Client Manager** (or other X manager, e.g., the **Hummingbird eXceed** window manager) must be running before any X client connections can be opened. You may want to place a shortcut to your X manager in **PROGRAMS > STARTUP** so that it starts automatically when you log into your PC. (And if so, it’s a good idea to specify “Run: Minimized" in the shortcut properties.) We document only the **Reflection X Client Manager**.

In the kerberos-users@fnal.gov mailing list archive, you can find a message containing couple of handy scripts for connecting to nodes using WRQ® Reflection X. Search for “handy scripts” and you’ll find the right message![1] The first script shows how to use your own kerberos principal to log in to your own account on a remote node. The second script shows how to use the your */root* principal (see section 9.4 *Using Root Instance of your Principal*) to log in to a different account, where forwarded/renewable tickets aren’t allowed.

To start **Reflection X** manually, navigate to **START > PROGRAMS > REFLECTION > REFLECTION X**. Click on it and the following screen comes up:

1. Message reference: Item #: 001654, Date: 01/11/03, Time: 09:14, Subject: “handy scripts for connecting to nodes using WRQ Reflection X”.

Run a telnet Session



The best thing to do at this point is to minimize the above window, start a **telnet** session, and run X applications from there as described in section 5.6.2 *Run a telnet Session to Kerberized Host.* Once you're connected, verify that your $DISPLAY is set correctly on the remote host (at Fermilab, this should already be set for you in your UNIX login files; if it's not, check these files).

Connect Directly from X Client Manager

You can opt to connect to a host directly from the **X CLIENT MANAGER** window, *but* it does not provide encrypted connections. **Do not `kinit` from an X window!**

To connect using this window, choose a connection option from the left pane and customize it as desired, or create (and optionally save) a new connection configuration. In the right-hand pane, select `KERBERIZED TELNET` as the **METHOD** (if you leave it as just `TELNET`, the remote host will respond in portal mode). Then click **CONNECT**.

Run a telnet Session with Automatic X Application Startup

For applications that you run often, you might find it useful to configure a **telnet** connection that includes an automatic X application startup. This is described in section 19.6.4 *Connect to Host with X Application Startup*. Once you have your host-specific, application-specific configurations created and

saved, they should appear in the **REFLECTIONS SESSIONS** folder. To invoke, double click on the file corresponding to the host/application you want. The system will log you in and start the application in your X window manager.

If you let the **WRQ®** X client starter close the initial telnet connection after the X client starts, your remote credential cache will be destroyed. You should either copy your credential cache to another file, or check the box that keeps the initial telnet open.

Procedures for other Windows X window manager products are not documented here.

10.1.3 Macintosh

We are not recommending any particular X client for Macintosh, and the process of bringing up X windows will depend on the software used. Suggested web sites for getting information are `http://www.ncsu.edu/mac/sma/sma.html` and `http://web.mit.edu/cggriffi/www/mackerberos`/.

10.2 Usage Notes for PC's with WRQ® Reflection Installed

10.2.1 Cutting and Pasting

To cut and paste between a VT terminal window and your Windows applications using the default mouse mapping[1]:

1) Select the information in the X terminal window using the left mouse button.

2) Click the right mouse button to pop up a menu. Select **CUT** or **COPY**.

3) Click in your local application where you want to paste.

4) Click the right mouse button to pop up a menu. Select **PASTE**.

10.2.2 Using Matrix through X Windows Interface

If you use the Computing Division's **Matrix** product through the X windows interface (i.e., the software is not locally installed on your NT machine), then you must change a couple of items in the configuration. Open the **X Client Manager** (**START > PROGRAMS > REFLECTION > REFLECTION X**) and go to **SETTINGS**:

- Select **COLOR...**. In the **COLOR SETTINGS** area, change **DEFAULT VISUAL TYPE** to `PseudoColor Emulation`. Click **OK**.
- Select **FONTS...** and under **OPTIONS**, check `Allow font scaling`. Click **OK**.

10.3 Automated Processes

10.3.1 Specific-User Processes (cron Jobs)

The **kcroninit** product is provided for setting up **cron** jobs in a Kerberized environment. It gets installed automatically as part of the **kerberos** product, and as of **kerberos** v0_6, it works without **systools**. **kcroninit** creates the necessary **cron** principal and keytab file so that **cron** jobs may be authenticated

1. You can reconfigure the mouse mapping. Navigate to the **UNTITLED - REFLECTION FOR UNIX AND DIGITAL** window and find it on the **SETUP** menu.

under the user's principal. **kcroninit** can be used on each node where **cron** jobs need to be authenticated, either for AFS tokens or for remote access to other Kerberos systems.

For no discernible reason, many systems have been found to have permission 701 on `/var/adm`, which stops **kcroninit** from working for any user in the group to which that directory belongs. Make sure this directory is set to mode 711 or 755 before trying **kcroninit**. A later version will fix this problem automatically when encountered.

To configure a **cron** job, follow this procedure:

1) First, create the **cron** principal and keytab file. You will need to enter your Kerberos principal and password, so **you must be on a secure channel**. (The **kcroninit** program will create the new principal `<user>/cron/<host>.<domain>@FNAL.GOV` for the current user, host and domain, and will write the corresponding keytab file.) Run the commands:

```
% setup perl

% setup kcroninit

% kcroninit
```

2) Use the **`kcron`** command to initiate the **cron** jobs in an authenticated manner. Note that you will need to specify the full path to **`kcron`**, since this is not normally in your $PATH at the start of a **cron** job. In the following sample crontab entry, the command **`/home/files/myjob -ak`** is authenticated as `<user>/cron/<host>.<domain>@<REALM>` (This is sufficient if authentication is needed only for access to the user's AFS files):

```
0 2 * * 0,4 /usr/krb5/bin/kcron /home/files/myjob -ak
```

3) For access to remote systems, the `.k5login` file on the remote end must allow access to `<user>/cron/<host>.<domain>@FNAL.GOV`. If you're just creating this file, don't forget to add your `<user>@FNAL.GOV` principal, too.

To destroy the principal and keytab file (and prevent authenticated **cron** jobs from running under your account on this node), run:

```
% setup kcroninit

% kcrondestroy
```

10.3.2 Processes Running as root

If you're setting up an automated process such as a **cron** job, you have to arrange for it to get credentials when it runs. If the process is running as *root*, it is simplest, both conceptually and practically, to consider that the host on which the job runs is the party responsible for the accesses it initiates, and to have it use the `/etc/krb5.keytab` to obtain credentials as `host/<hostname>.<domain>`. To do so, first set the variable KRB5CCNAME, e.g.,:

```
% KRB5CCNAME=FILE:/tmp/krb5cc_root_$$
```

Then run **`kinit`**:

```
% /usr/krb5/bin/kinit -k host/<hostname>.<domain>
```

When you're done, get rid of the tickets:

```
% /usr/krb5/bin/kdestroy
```

10.3.3 Non-root, Non-Specific-User Processes

Here is a scheme that works for jobs that run neither as *root* nor as a specific user. This scheme provides AFS access.

First, contact the Computing Division's KDC administrator via *nightwatch@fnal.gov* and describe the job that you want to set up. Some discussion may be required to determine if this method is appropriate for your needs. If you agree to go ahead, the KDC admin will need the following information:

- a name for your job (`<jobname>`)
- the name of your division, section, or experiment (`<group>`)
- the hostnames that will need to initiate Kerberos-authenticated network connections for the job (`<hostname>.<domain>`), or ...

  the names and principals of one to three people in your group who will be the Kerberos "sub-administrators" for the job

Setting Up the Task

The KDC administrator creates a principal `<user>/<group>/<jobname>` for each Kerberos "sub-administrator", gives each one a password, and describes any extra steps that need to be taken. These principals have the authority to create, delete and change passwords for all the principals matching the pattern `<jobname>/<group>/*`.

If you're working on a farm cluster, there are certain tools available that other random systems don't have. The KDC administrator can create the special farm principal names such that when a job starts on the farm, it will have both

a Kerberos TGT as `<jobname>/<group>/farm` and an AFS token as AFS user `<jobname>`. In this case, the KDC and farm administrators take care of everything; the rest of the instructions do not apply.

The Kerberos "sub-administrators" in your group will need to:

- create a principal `<jobname>/<group>/<hostname>.<domain>` (`@REALM` is implicit) for every host which may initiate a `<jobname>` network activity.
- create a keytab file on each host containing an encryption key for the `<jobname>/<group>/<hostname>.<domain>` principal and put it in a file somewhere such that only the right UNIX id(s) on that host have access to it.

In order to create the principals and keytab files, do the following as the `<jobname>` user on each host (the **`kadmin`** commands should be issued on a single line):

```
% /usr/krb5/sbin/kadmin -p <user>/<group>/<jobname>

   Enter password:  <-- type in your password

kadmin:  addprinc -randkey
  <jobname>/<group>/<hostname>.<domain>

kadmin:  ktadd -k /path/to/keytab/file
  <jobname>/<group>/<hostname>.<domain>

kadmin:  exit
```

Then, in the home directories of the accounts which will be the targets of **`<jobname>`** activity, list all the initiator principals in a `.k5login` file as usual, e.g.,:

```
<jobname>/<group>/host1.fnal.gov@FNAL.GOV
<jobname>/<group>/host2.fnal.gov@FNAL.GOV
<jobname>/<group>/host3.fnal.gov@FNAL.GOV
```

Running the Task

In all your scripts, include a **`kinit`** command as follows:

```
% kinit -k -t /path/to/keytab/file
  <jobname>/<group>/<hostname>.<domain>
```

This must occur in the script before the script initiates a network access. (If the hostname is properly set to the full domain name, you could just use **`'hostname'`** in the last argument.) If you need access to AFS but your host's `/etc/krb5.conf` file does not specify `krb5_run_aklog = true` as an `[appdefault]` for **`kinit`**, then add an explicit **`-a`** flag to **`kinit`**, or run **`aklog`** as a separate step.

10.4 Sending Data from Unstrengthened to Strengthened Machines

Sending data from the strengthened realm to an unstrengthened machine is straightforward via **FTP** or an r-command. Portal mode **FTP** is available to handle sending data from an unstrengthened machine to a strengthened one.

If the strengthened target machine has a properly configured anonymous `incoming` **FTP** directory, an outside process (which can be running on an unstrengthened machine) can deposit data into it. If the target machine is *not* configured properly, the outside process can send an unauthenticated signal, e.g., an email or some other connection that signals “look for data now”, and the strengthened target machine can initiate a pull.

10.5 CVS

Different groups may implement CVS differently under Kerberos at Fermilab. Here we discuss the Computing Division’s recommended configuration which is used for its repository CDCVS. This configuration is also used by SDSS and the CD/D0/CDF Run II Code Management Working Group.

cvsh v1_4 supports Kerberized access to **CVS** repositories. **CVS** uses the *cvsuser* account. On the server side, **cvsh** must be made the default shell for the *cvsuser* account. Users must be added to that account’s `.k5login` or `.k5users` file. On the client side, users can access the CVS repository via ssh (authorized key access allowed), Kerberized rsh[1], or pserver. So if you have been accessing a repository via (nonKerberized) rsh, you’ll need to convert.

This configuration and converting to it is documented at `http://cdcvs.fnal.gov/connecting.html` and `http://www.fnal.gov/docs/products/cvs/cvs_ssh.html`. CDF users can reference `http://www-cdf.fnal.gov/offline/code_management/Dist/doc/cvsaccess.txt`.

To run a nonKerberized CVS client on a Kerberized machine, you can run two sshds:

1) the first runs on a separate IP address, allows RSA authentication, and allows only *cvsuser* to log in (*cvsuser* uses a restricted shell which

1. If you’re using Kerberos rsh as the transport, and if your /etc/krb5.conf [appdefaults] says "forward = true" for rsh (or for all apps), then you have to have a forwardable ticket or create a wrapper script that does "rsh -N".

allows only CVS commands).

2) the second sshd runs on the usual IP address (but it is specified) and allows anyone to log in with Kerbreros authentication.

The CVSROOT is advertised using the IP address from item 1.