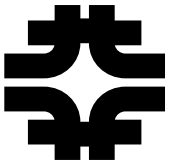


Fermilab

GG0019

Strong Authentication at Fermilab (Pilot Phase)

Release P1.0

February 29, 2000

Computing Division
Fermi National Accelerator Laboratory

Compiled by Anne Heavey

ABSTRACT

In order to protect against unauthorized access to Fermilab computers we are implementing the Kerberos Network Authentication Service V5 to provide *strong authentication* over the network. This manual is currently intended to document the strong authentication system as implemented in the pilot phase. The manual is targeted to both administrative and end users of UNIX and Windows® systems.

- It describes how to "strengthen" a UNIX machine (all supported flavors: Linux, SunOS, IRIX, OSF1).
- It describes how to configure a PC to connect to strengthened UNIX machines.
- It provides an introduction to the concepts, goals, features and terms involved in our strong authentication project, and describes how to access strengthened machines and use the Kerberos features and commands.

Revision Record

February 29, 2000 Original Pilot Release P1.0

Acknowledgments and References

Members of the Fermilab Computer Security Team (Matt Crawford, Mark Kaletka and Lauri Loebel Carpenter) provided and reviewed much of the information for this manual. Other contributors/reviewers include Liz Buckley-Geer, Mike Lindgren and Mike Stolz.

Additional information was collected from the following sources:

- Crawford and Kaletka (Fermilab), *Computer Security Strong Authentication Project: Synopsis*, July 1999.

  `http://www-dcd.fnal.gov/computersecurity/strongauth/plan/AuthenticationPlan.htm`

- Kohl and Neuman (DEC) *RFC 1510: The Kerberos Network Authentication Service (V5)*, September 1993.

  `ftp://ftp.isi.edu/in-notes/rfc1510.txt`

- Kerberos Frequently Asked Questions (U.S. Naval Research Laboratory), September 1999 update.

  `http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html`

- *Kerberos V5 UNIX User's Guide, Release: 1.0*, Document Edition: 1.0, Massachusetts Institute of Technology (frequently updated).

  `http://www-dcd.fnal.gov/computersecurity/StrongAuth/UserDocs/user-guide_toc.html`

- *Kerberos: The Network Authentication Protocol*, Massachusetts Institute of Technology, October 1999.

  `http://web.mit.edu/kerberos/www/`

Table of Contents

Chapter 1: About this Manual

This chapter provides an introduction to the *Strong Authentication at Fermilab* manual. In particular you will find:

- the purpose and intended audience
- a summary of the contents of all the chapters
- where to find the manual on-line or obtain a hardcopy
- how updates to the manual are handled
- the typeface conventions and symbols used throughout the manual
- where to send comments and questions

1.1 Purpose and Intended Audiences

An analysis of the major computer security incidents at Fermilab over the past year, as well as the general sense of security incidents prior to that, shows that a common root cause of these incidents is the compromise of user passwords by their transmission in clear text over the network. Once intercepted, passwords can be re-used to gain unauthorized access to the destination system. Further, with user access to a compromised system, hackers can fairly easily gain privileged root access. In order to protect against unauthorized access to Fermilab computers we are implementing the Kerberos Network Authentication Service V5 to provide what is known as *strong authentication* over the network.

This manual is currently intended to document the strong authentication system as implemented in the pilot phase. As development on the system continues and it gets rolled out, we plan to update the documentation accordingly. The manual is targeted to both administrative and end users of UNIX (all supported operating systems: SunOS, IRIX, RedHat Linux, OSF1) and Windows® systems:

- Pulling information from a variety of sources, chapters 2 and 3 provide a brief but we hope(!) adequate introduction for the novice user to the concepts, goals, features and terms involved in our strong authentication project.
- Chapter 4 is intended for UNIX system administrators. It describes how to “strengthen” a UNIX machine.
- In Chapter 5 we describe for NT (and other Windows®) users how to configure a PC to connect to strengthened UNIX machines.
- Chapters 6, 7, and 8 are intended for all users who need to log on to strengthened machines. We explain how to access the machines, how to use the new features, and how to accomplish miscellaneous tasks that work a little differently in a strengthened environment.

• Chapters 9 and 10 are also intended for all users. They are command references.

1.2 Summary of Chapters and Appendices

Chapter 2: *Overview of Strong Authentication at Fermilab*

In this chapter we discuss the concept of strong authentication and the features and environment as implemented at Fermilab.

Chapter 3: *About the Kerberos Network Authentication Service V5*

In this chapter we provide an introduction to the Kerberos Network Authentication Service V5, discuss the important terms and components, and describe the authentication process.

Chapter 4: *UNIX Administrator's Guide*

In this chapter we provide instructions for Kerberizing a UNIX machine and discuss some system administration issues. The information is valid for all supported flavors of UNIX, namely: RedHat Linux, SunOS, IRIX and OSF1. The instructions assume that **UPS/UPD** is running on the machine.

Chapter 5: *Configuring a Windows® System to Access Kerberized Nodes*

In this chapter we describe how to install and configure the **WRQ Reflection®** software on your **Windows®** system (NT4, 95 or 98) in order to access Kerberized machines and encrypt your data transmissions.

Chapter 6: *Accessing the PILOT.FNAL.GOV Realm*

In this chapter we discuss accessing machines in the PILOT.FNAL.GOV realm from UNIX machines and PCs, directly and/or via applications such as **telnet** and **FTP** and/or through portal mode.

Chapter 7: *Using Kerberos*

This chapter provides all the basic information you need in order to manage your Kerberos tickets and work in a Kerberized environment. In particular, we cover passwords, ticket options and management, and account access files.

Chapter 8: *Special Topics for the User*

In this chapter we document a variety of common operations that work differently in our Kerberized environment. As the pilot project progresses this chapter will grow!

Chapter 9: *Kerberos Command Descriptions*

In this chapter we list the native Kerberos commands, and provide a brief description and option list with descriptions adapted from the man pages. Programs that Kerberos provides for ticket and password management include **kinit**, **klist**, **kpasswd** and **kdestroy**.

Chapter 10: *Kerberized Network Programs*

In this chapter we document the Kerberized features of several network programs.

1.3 Availability

Copies of *Strong Authentication at Fermilab* (document number GG0019) can be obtained from the following sources:

On-line	`http://www.fnal.gov/docs/strongauth/`
	Under **Documentation Search** on the Computing Division home page (`http://www.fnal.gov/cd/`), search using any of the following keywords: strong authentication, authentication, security, computer security, kerberos, network, network connections, UNIX, Windows, NT.
Paper Copies	Wilson Hall, 8th floor, NE
	Or print your own copy from the on-line PostScript file under `http://www.fnal.gov/docs/strongauth/ps/`

1.4 Updates

Pending subsequent releases of this manual, updates will be maintained on the Web at `http://www.fnal.gov/docs/strongauth/misc/updates.html`.

1.5 Notational Conventions

The following notational conventions are used in this document:

bold	Used for product and program names (e.g., **telnet**).
italic	Used to emphasize a word or concept in the text. Also used to indicate logon ids and node names.
`typewriter`	Used for filenames, pathnames, contents of files, output of commands.
`typewriter-bold`	Used to indicate commands and prompts.
<**CTRL**-*char*>	Indicates a control character. To enter a control character, hold down the control key (labeled Ctrl, usually) while pressing the key specified by *char.*
[ ]	In command formats, indicates **optional** command arguments and options.
%	Prompt for C shell family commands (**%** is also used throughout this document when a command works for both shell families).
$	Prompt for Bourne shell family commands; also standard UNIX prefix for environment variables (e.g., *$VAR* means "the value to which *VAR* is set").

< >	In commands, paths and environment variables, indicates strings for which the user must make context-specific substitutions.

All command examples are followed by an implicit carriage return key. The following symbols are used throughout the text to draw your attention to specific items:

A “bomb”; this is used to indicate a potential pitfall.

This symbol is intended to draw your attention to a particularly important piece of information.

This symbol indicates information for AFS systems.

1.6 Your Comments are Welcome!

The *Strong Authentication at Fermilab* manual may contain some errors, however we endeavor to minimize the error count! We encourage all the readers of this document to report back to us:

- errors or inconsistencies that we have overlooked
- any parts of the manual that are confusing or unhelpful -- please offer *constructive* suggestions!
- other topics to include (keeping in mind the purpose of the manual)
- information that other users might find helpful

Send your comments via email to *cdlibrary@fnal.gov.*

Chapter 2: Overview of Strong Authentication at Fermilab

In this chapter we discuss the concept of strong authentication and the features and environment as implemented at Fermilab.

2.1 What is "Strong Authentication"?

A succinct definition of strong authentication was given by Tardo and Alagappan[1]:

> "Techniques that permit entities to provide evidence that they know a particular secret without revealing the secret."

In more practical terms, it is a system of verifying workstation user and network server identities on an unprotected network in which the parties must demonstrate knowledge of a "secret". Typically the verification is done via a trusted third-party authentication service using conventional cryptography. Strong authentication avoids relying on authentication by the host operating system or basing trust on host addresses. It does not require that the network be safe from eavesdropping, or from injection of hostile packets or alteration/deletion of packets[2].

2.2 Goals of Strong Authentication at Fermilab

A primary goal of our strong authentication implementation is to essentially eliminate (so far as is practical) the transmission of clear text reusable passwords over the network and their storage on local systems. Secondary, but important, goals for us include:

- Providing a single sign-on environment for users
- Integrating existing accounts

1. J.J. Tardo and K. Alagappan, "SPX: Global Authentication Using Public Key Certificates." In *Proc IEEE Symp. Research in Security and Privacy.* IEEE CS Press, 1991.
2. The Kerberos authentication process can fail if too many packets are altered or deleted (i.e., all of them in one or both directions, until the client gives up).

- Centralizing account maintenance
- Enforcing password policies such as length, quality and lifetime consistently

2.3 The Authentication Model Implemented at Fermilab

The strong authentication service implemented at Fermilab is the Kerberos Network Authentication Service V5. We describe many of its features in Chapter 3: *About the Kerberos Network Authentication Service V5*. In this section we describe the model more generally. The model employed divides the computing environment into three *realms*:

The strengthened realm

The strengthened realm consists of all systems (whether on- or off-site) that require strong authentication for access from the network. On a strengthened system, all traditional means of access that use weak authentication, such as **telnet**, **rlogin**, **FTP**, and so on, are replaced with strengthened versions of these programs. Means of access over the network that do not expose passwords are allowed. Weak authentication (standard security) is allowed for local access only, i.e., via the console or locally attached display. Machines that are configured for **ssh**-only access are not part of the strengthened realm.

For the pilot phase of our strong authentication project, the strengthened realm at Fermilab is called PILOT.FNAL.GOV. Once the system is rolled out we plan to change the realm name to FNAL.GOV.

The trusted realm

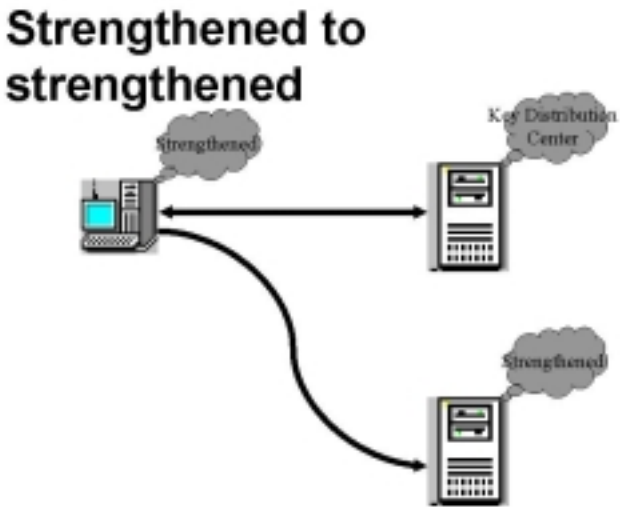
Other sites which implement strong authentication, and which meet certain criteria, may be recognized by the strengthened realm at Fermilab as a “trusted” realm. Trusted realms provide levels of security and authentication equivalent to our own. Trust relations (cross-authentication) between the trusted realm and the strengthened realm allow access without further authentication (i.e., the authentication takes place only when user accesses either realm individually).

The untrusted realm

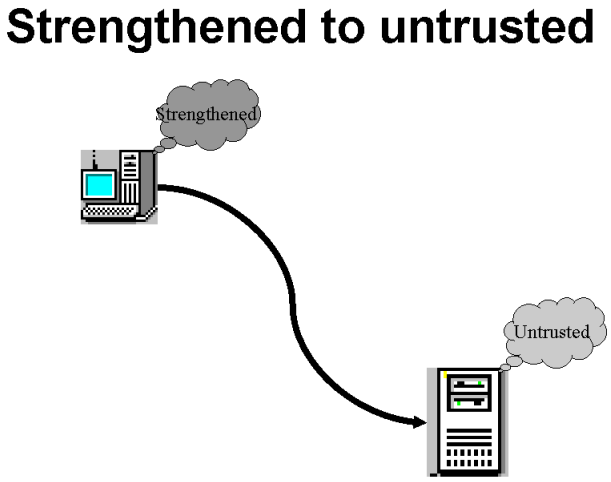
The untrusted realm consists of those systems that do not require strong authentication and that permit weak authentication and traditional means of access. These systems typically expose clear-text passwords on the network.

The figures below illustrate the relationships between these realms. (The Key Distribution Center, or KDC, shown on these figures is described in Chapter 3: *About the Kerberos Network Authentication Service V5*.)

Direct connections between machines in the strengthened realm are allowed (the Key Distribution Center is involved in providing credentials to the client’s machine which can be passed along to access the other strengthened machine):



Direct connections *from* the strengthened *to* the untrusted realm are allowed:



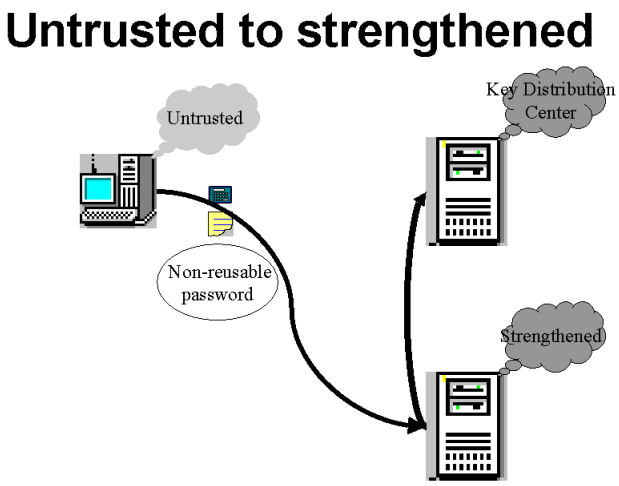
The reverse, that is, direct connections from the untrusted to the strengthened realm, are not; a secure gateway called a *portal* must be employed.

Portal mode

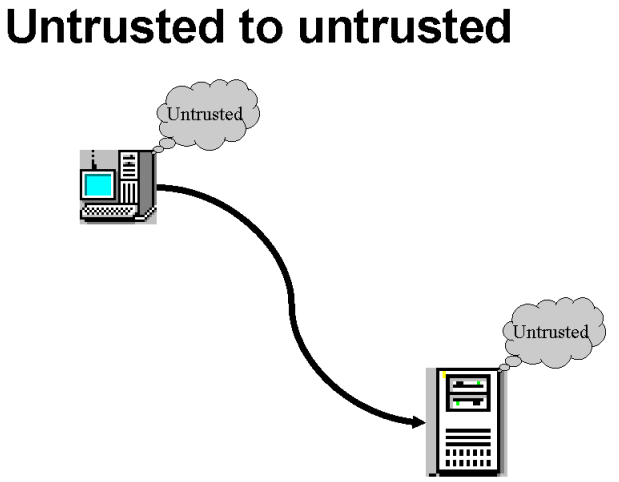
At Fermilab we are not using separate machines to act as portals; instead, the strengthened machines are configured to respond in *portal mode* when requests for access are made from the untrusted realm. To access a strengthened machine from the untrusted realm, a user must log in using a standard communication application (e.g., **telnet** or **FTP**[1]) and supply a

1. As of this writing, portal-mode **FTP** is not implemented. To initiate file transfer between strengthened and untrusted hosts, you must first log into the strengthened host, e.g., via portal-mode **telnet**.

non-reusable password. Different programs exist for generating non-reusable passwords, and at Fermilab we support CryptoCard® and (in the future) OTP, an S/Key One-Time Password method. No special hardware or software is required on the untrusted system other than physical possession of a card or a list of one-time passwords by the user.



For connections between untrusted machines, strong authentication is not involved. The standard network programs are used in the normal way:



2.4 Features of Strong Authentication at Fermilab

The strong authentication model implemented at Fermilab:

- improves authentication and access control
- is adaptable to new computer security threats and changes in system security requirements and to new styles of computing
- is integrated with AFS (I.e., if your machine is part of the strengthened realm and it runs AFS, then when you log on and get Kerberos authentication, you also automatically get an AFS token.)
- is robust and stable
- can be readily deployed to collaborating universities and laboratories, including those outside the United States
- accommodates all the supported UNIX operating systems, as well as Windows NT[1]
- is capable of establishing trust relationships with other institutions where similar strong authentication systems are in place, allowing each user to have a single identity (userid) encompassing multiple institutions
- provides meaningful improvements in security and authentication for the Run II experiments, and is being incorporated into the software infrastructure currently under development for Run II
- provides access for users and systems from outside the strengthened realm via the portal function, without the installation of special hardware or software on the users' desktops (this allows access via systems that do not or can not have strong authentication directly installed, e.g., a public terminal, a "dumb" X terminal, or a PDA)

1. We hope to eventually support for Macintosh and Windows 95/98 systems, too. Certain systems, such as embedded systems or specialized on-line systems may not be capable of participating directly in strong authentication. These systems may be accommodated by alternate access.

Chapter 3: About the Kerberos Network Authentication Service V5

In this chapter we provide an introduction to the Kerberos Network Authentication Service V5, discuss the important terms and components, and describe the authentication process.

3.1 Introduction to Kerberos V5

Kerberos V5 is a network authentication protocol designed to serve as a trusted third-party authentication service. It is a single-sign-on system, meaning that a user only has to type his password once, and the Kerberos V5 programs do the authenticating (and, optionally, encrypting) for the user.

Kerberos was developed at MIT in 1987 and has matured into a stable product with widespread operating system and application support. Microsoft is basing the authentication in Windows® 2000 on Kerberos V5. Kerberos continues to see active development, with new releases occurring approximately twice per year. Kerberos V4 is already in use at Fermilab as part of AFS, and both Kerberos V4 and V5 are widely used at other laboratories and universities.

Kerberos verifies the identity of a user or a network service (users and services are collectively called *principals*) on an unprotected network using conventional cryptography in the form of a shared secret key. The shared secret key technology allows a client and server (e.g., a principal and a strengthened machine) to mutually establish their identity across an insecure network connection without exposing passwords. They can also encrypt all of their communications to assure privacy and data integrity.

A machine on which Kerberos has been installed and which enforces the Kerberos authentication is referred to as a *strengthened* or *Kerberized* machine. Kerberos has been built into each of a suite of network programs, including **telnet**, **FTP**, **rsh**, **rcp**, and **rlogin**. It can be built into other programs as well. The Kerberized version of a program is also referred to as *strengthened* or *Kerberized*, and requires individual authentication for use.

3.2 Keys, Tickets and the KDC

Kerberos authentication is implemented primarily via a service called the *key distribution center* (*KDC*).[1] The KDC shares a permanent secret key with each principal (user and service).[2] Most KDC implementations store the principals in a database; therefore the term "Kerberos database" is sometimes applied to the KDC. The KDC implements the Authentication Service (AS) and the Ticket-Granting Service (TGS) for all the machines in the realm. To understand what these do, you first need to know what session keys, tickets and credentials are:

Session Key — A session key is similar to a principal's long-lived secret key, but differs in these ways: it is generated at random by the KDC to be shared between two principals (usually a user and a service), and its validity is limited to the lifetime of an accompanying ticket. The session key is used in place of the permanent key to authenticate the two principals to each other, possibly multiple times during the ticket lifetime. Its purpose is to limit exposure of the permanent key, i.e., the user's (encrypted) password, over the network. If encryption or integrity protection of bulk data is required, another key is negotiated by the two principals, called a *subkey* or a *sub-session key.*

Ticket — Kerberos uses encrypted records called *tickets* to authenticate to Kerberized services[3]. Tickets generally contain the session key, the user and service ids and the host's IP address. All the information is encrypted with the service's permanent key, known only to the service and the KDC. A ticket is accompanied by an extra copy of the session key encrypted under the user's key. The ability of both user and service to correctly decrypt the relevant parts of the ticket establishes knowledge of the correct keys and therefore establishes authentication for the service.

Credential — The combination of the initial ticket and the session key is called a *credential*.

The Authentication Service (AS) issues secret session keys and tickets (credentials) based on a user password or encryption key. It can issue both Ticket-Granting Tickets (TGTs) and individual service tickets. A TGT is a ticket that authenticates a user process to the Ticket-Granting Service (TGS) portion of the KDC. The Ticket-Granting Service (transparently) issues tickets to clients for individual Kerberized services.

---

1. A Kerberos strengthened realm has one primary KDC, and may have one or more secondary KDCs. We refer to them here collectively as "the KDC". Authentication is still possible if the primary KDC is not reachable, but certain administrative tasks are not (e.g., changing passwords, creating new principals).
2. For a user, this shared secret key is a hash of the user's password; for a service, the key is a random bit string.
3. Technically, both a ticket and a record called an *authenticator* are required. An authenticator is generated and sent by the user process any time a ticket gets used. It contains, among other things, a timestamp and optionally a sequence number, all encrypted with the session key in the ticket. This proves to the service that the client knows the session key, and hence is the legitimate holder of the ticket, and that this is not an adversary's replay of a previously used ticket/authenticator.

3.3 The Login Authentication Process

When a user principal attempts to log in at the keyboard of a strengthened machine, short “behind-the-scenes” messages are exchanged between the principal and the AS portion of the KDC. The password is not actually transmitted, but rather is used as a key to encrypt and decrypt the exchanged messages. The ability to correctly decrypt messages from the KDC demonstrates knowledge of the key and authenticates the identity of the principal.

Once the principal’s identity is established, the KDC supplies a set of initial credentials to the principal. The credentials consist of a secret session key and an initial ticket. The session key is encrypted inside the initial ticket. The initial ticket is by default a ticket-granting ticket (TGT).

There are some cases in which you might have to renew or re-obtain credentials after you're already logged in, but generally you're ready to access any strengthened realm host by means of your initial login credentials.

When you initiate a Kerberized connection over the network, your client application obtains a service ticket for the destination (or re-uses a valid one from a credential cache) and presents it, together with an authenticator it constructs fresh for each access, to the target host. The application can optionally forward a TGT to the target host, enabling access from that host to others.

Kerberized hosts at Fermilab running AFS are configured to obtain AFS tokens automatically at login via the **aklog** program. This program authenticates to a cell or directory in AFS.

Chapter 4: UNIX Administrator's Guide

In this chapter we provide instructions for Kerberizing a UNIX machine and discuss some system administration issues. The information is valid for all supported flavors of UNIX, namely: RedHat Linux, SunOS, IRIX and OSF1. The instructions assume that **UPS/UPD**[1] is running on the machine.

For systems on which **UPS/UPD** is not installed: You can download the MIT Kerberos product in a variety of formats from the Web (e.g., RPM for RedHat Linux). Two caveats:

- The Computing Division only supports **UPS/UPD** installations.
- While an off-the-shelf Kerberos V5 will interoperate with Fermilab's system, it won't have some of our locally-added or configured features.

4.1 Before You Install Kerberos

The way to Kerberize a UNIX machine is to install the Fermilab **kerberos** product on it, available from *fnkits.fnal.gov*[2]. This product is preconfigured and should require no further actions beyond the installation instructions given in section 4.2 *Kerberos Installation Instructions*. However, there are a few things to do in preparation for installing it.

4.1.1 Obtain a Kerberos Principal

First of all, if you don't already have a Kerberos principal, you'll need to get one (plus an initial password) just to have access to the PILOT.FNAL.GOV realm. Since all users need a principal, we provide information on how to obtain one in the 6.1 *Obtaining a Principal* section of Chapter 6: *Accessing the PILOT.FNAL.GOV Realm*.

4.1.2 Create an Account that Matches your Principal

You must have an account on the machine that matches your user principal. If you don't, you will be locked out of your machine after it's Kerberized!

---

1. For information on **UPS/UPD**, see `http://www.fnal.gov/docs/products/ups`.
2. Installing products from *fnkits* is described in Part II of the **UPS/UPD** documentation, available from `http://www.fnal.gov/docs/products/ups`.

4.1.3 Obtain Host and FTP Passwords

Contact your Computing Division liaison to request host-specific principals (plus initial passwords) for the machine on which you plan to install **kerberos**. The liaison will need to provide to the Kerberos administrators the full hostname of the node and arrange a means of getting you the initial passwords securely. The principal names are of the form `host/<full.node.name>` and `ftp/<full.node.name>` (e.g., `mynode/mynode.fnal.gov` and `ftp/mynode.fnal.gov`).

4.1.4 Synchronize your Machine with Time Server

When using Kerberos, the client and server must be time-synchronized with each other. Kerberos is configured to allow a discrepancy of about five minutes. **xntp** is a product that you can install on your machine to maintain the system time in agreement with Internet standard time servers. It is available from *fnkits* for some platforms.

If your system runs AFS, don't install **xntp** or any other synchronizing software; AFS does its own synchronization.

4.2 Kerberos Installation Instructions

For UNIX machines (all supported flavors), you need to install the product **kerberos** on the machine that you want to place in the PILOT.FNAL.GOV strengthened realm. The product is available from *fnkits*. The following instructions for installing this product are copied from the `README.INSTALL` file for **kerberos** v0_4[1]:

```
When installing kerberos, many different configuration
 changes must also be applied to the target systems.  Please
 read the following to determine how to configure kerberos
 for your situation.  NOTE, kerberos must be properly
 installed on each individual node.


 1)  OBTAIN PASSWORDS for "host/<full.node.name>" and
     "ftp/<full.node.name> principals (first-time only).
      The FIRST time you install kerberos on a node, you will need to contact
      the Kerberos administrators to obtain an initial "host" service password
      and an initial "ftp" service password for that node.  You will need to
      provide the full hostname of the node, and arrange a means of securely
      obtaining the initial password.

 2)  DETERMINE WHICH SITUATION APPLIES TO YOU:

      a) Install kerberos in "fully strengthened" mode (which will
          disable all non-kerberized means of access to this node)
         SEE section 2a, "FULLY STRENGTHENED MODE".
  OR
```

1. The "v0_4" is a Fermilab version designation. For the foreseeable future, all Fermilab versions will refer to the **Kerberos V5** product.

```
     b) Install kerberos in a mixed mode (which will enable
         kerberized services, will NOT disable ssh access,
         but WILL DISABLE all other non-kerberized means of
         access to this node)
        SEE section 2b, "STENGTHENED MODE WITH SSH"

       (if neither of these apply to you, please read the
       file README.INSTALL.DETAILS which describes *all* of the gory
       installation options; recommended ONLY for experts).

2a) FULLY STRENGTHENED MODE: enable kerberized access to this
     node, and disable all non-kerberized means of accessing the
     node.

     As 'root' you should issue the command

       $ ups install kerberos [vN_M]

     on each node you wish to fully strengthen.  You need to
     include the specific version number of the kerberos release
     (vN_M) if it has not been declared as 'current' on your local node.

     The "ups install" command will perform all steps necessary for a
     fully-strengthened configuration.

2b) STRENGTHENED MODE WITH SSH: enable kerberized access to this
    node, do NOT disable any existing ssh access to the node, but
    disable all other non-kerberized means of accessing the
    node.

    As 'root' you should issue the command
     $ ups install-keep-ssh kerberos [vN_M]

    on each node you wish to configure.  You need to include the
    specific version number of the kerberos release (vN_M) if it
    has not been declared as 'current' on your local node.

    The "ups install-keep-ssh" command will perform all steps necessary
    to enable kerberos on this node.  It will not disable any existing
    ssh access that is allowed to the node, but it will disable all
    other non-kerberized means of accessing the node.

3) CLEANUP OF OLD VERSIONS OF KERBEROS.
    If you installed kerberos v0_1 or v0_2, you will need to
    clean out the files which had been copied to /usr/local
    (and are now copied to /usr/krb5).  Use the command

       $ ups clean kerberos [vN_M]

    You will need to enter the version [vN_M] if this version
    of kerberos is not declared 'current' on your local node.

================================================================
```

4.3 Kerberos Installation Examples

These will be provided soon.

4.4 System Administration Issues

4.4.1 The /etc/hosts File

In the `/etc/hosts` file, make sure that the localhost names include `.fnal.gov`.

4.4.2 Portal Mode Configuration

A UNIX host running **kerberos v0_4** or later will perform the portal function by default when accessed from the untrusted realm unless this mode is specifically disabled. In the `inetd.conf` file (which resides in either `/etc` or `/etc/inet`) you should find a line similar to:

```
telnet stream tcp nowait root /usr/kerb5/sbin/telnetd telnetd -aP valid
```

The **`P`** flag in the **`telnetd -aP valid`** portion enables portal mode. To disable this mode, remove the **`P`** flag.

4.4.3 Providing Root Access

To allow someone access to the *root* account on a Kerberized UNIX machine, add the person's principal to the `.k5login` file in the root account's home directory (`/root/` for Linux, `/` for the other supported flavors). This file is described in section 7.4 *Account Access for Multiple Principals*. Assuming the user's TGT has "forwarding" set (see section 7.2 *Ticket Options*; forwarding should be set by default) the user then can log in to the machine under his own principal and use **`ksu`** (instead of **`su`**) to run as *root* (see section 9.3 *Kerberized su (ksu)*).

Chapter 5: Configuring a Windows® System to Access Kerberized Nodes

In this chapter we describe how to install and configure the **WRQ Reflection®** software on your **Windows®** system (NT4, 95 or 98) in order to access Kerberized machines and encrypt your data transmissions.

5.1 Getting Ready

5.1.1 Obtain a Kerberos Principal

First, verify that you have administrator privileges on the PC. Next, you need to obtain a Kerberos principal and initial password for the PILOT.FNAL.GOV realm. A principal is essentially a realm userid. Since all users need a principal, we provide information on how to obtain one in the 6.1 *Obtaining a Principal* section of Chapter 6: *Accessing the PILOT.FNAL.GOV Realm*.

5.1.2 About the Software You Need to Install

For PCs running Windows NT4, 95 or 98, you need to install two **WRQ Reflection**® software products, **Reflection Signature**® which runs Kerberos on your PC, and **Reflection Suite for X**® which is a terminal emulation package similar to **Hummingbird eXceed**®, but with Kerberos authentication added. You need a license for the **WRQ Reflection**® software; contact your group's PC administrator or your local NT server administrator to request one.

Installing the following recommended components of these products will consume at least 110 MB of disk space:

From **Signature**:

- **Kerberos Manager**
- **TimeSync**
- **WRQ Event Viewer**

From **Suite for X** (subcomponents listed in section 5.4 *Installing Reflection Suite for X*®):

- **Reflection X** (partial)
- **UNIX/VMS Host Access** (partial)
- **Network Applications** (partial)
- **Utilities**

It is possible to run both the **Hummingbird** and **WRQ Reflection** products, however the PC Support group may not support combined installs. If you plan to use **eXceed** for X terminal emulation, you can get by with a minimal **Reflection** product installation consisting of:

- **Kerberos Manager**
- **TimeSync**

from **Signature**, and:

- **UNIX/VMS Host Access** (partial)
- **Network Applications** (partial)

from **Suite for X**.

Note that after installing this software you will still log into your PC the same way as before (e.g., using your FNAL NT domain userid and password). You will need to provide your principal and Kerberos password only when you run the **Kerberos Manager** or attempt to connect to a Kerberized node over the net from your PC.

You can configure the **Reflection** software to access nonKerberized nodes, however it doesn't provide access to **ssh**-only nodes. For that you'll need to install **ssh** software on your PC. See the *Ssh Programs for Windows*® page for suggestions (`http://www.fnal.gov/www/docs/StrongAuth/html/sshprograms.html`).

5.2 Installing Reflection Signature®

1) Contact your group's PC administrator or your local NT server administrator to request a license for the **WRQ Reflection**® software.

2) Log on to your PC as `Administrator`.

3) Browse from **NETWORK NEIGHBORHOOD** to `\\Pckits\WRQ\ReflectionSignature` and run the `Install.exe` you find there.

4) On the initial setup screen, choose `Reflection Signature (using single-user Setup)`.

5) Click through the usual license agreement and welcome screens.

6) On the **REFLECTION FOLDER** screen, leave the default `C:\Program Files\Reflection` and click **NEXT >**.

7) On the **USER FOLDER** screen, leave the default `C:\Program Files\Reflection\User` and click **NEXT >**.

8) When prompted to select components, uncheck `Admit One` from the default set. `Kerberos`, `TimeSync` and `WRQ Event Viewer` should by default be checked (check them if they're not). `Deployment Manager` should remain unchecked. Click **NEXT >**.

   If doing a minimal install, check only `Kerberos` and `TimeSync`.

9) On the **SHORTCUT FOLDER OPTIONS** screen, leave the default `Reflection`, and *don't* check `Do not create shortcuts`. Click **NEXT >**.

10) On the **CREDENTIALS FOLDER** screen, uncheck `Use my Windows NT home folder (shown below)` to select `C:\Program Files\Reflection\User`. Click **NEXT >**.

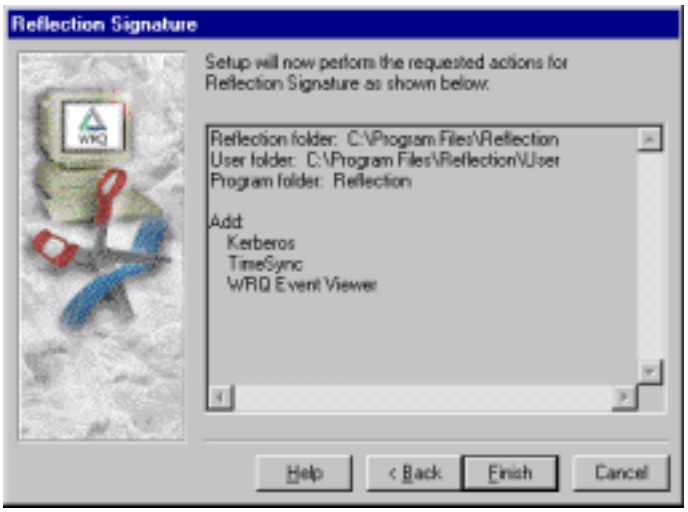
11) On **CHOOSE THE DEFAULT REALM**, enter the name of the pilot project Kerberos realm `PILOT.FNAL.GOV` (no quotes, all upper case) and click **NEXT >**.

12) On **SPECIFY THE KERBEROS KEY DISTRIBUTION CENTER**, enter the host name of the KDC as `i-krb-1.fnal.gov` (no quotes, all lower case) and click **NEXT >**.

13) On **CHOOSE DEFAULT PRINCIPAL**, enter your principal (user) name in lower case without the "@PILOT.FNAL.GOV" part, and click **NEXT >**. The default shown is your NT userid, *which may not be your principal name*.

14) In the **CONFIGURE REALM SECURITY** dialog, uncheck `Require administrator privileges` and click **NEXT >**.

15) A window pops up (see below) that shows your installation parameters and says `Setup will now perform the requested actions for Reflection Signature as shown below`. Everything should look OK, so click **FINISH >** and prepare to wait a little while.



16) When the installation process finishes, click **RESTART WINDOWS**[1] and then continue with the configuration steps, below.

1. If you have placed several items in the **STARTUP** folder in your user profile, logon can be slowed down considerably. To log on quickly and suppress all these applications, press down the Shift key while you're logging on.

5.3 Configuring Reflection Signature®

1) (Note: Kerberos requires a reasonably accurate time sync, within about five minutes. If you already run time synchronizing software on your Windows system, you can skip this step[1].) Navigate to **START > PROGRAMS > REFLECTION > TIMESYNC** to open the **Reflection TimeSync** application. Make sure the *Synchronize* tab is selected, and enter the IP addresses of the default primary and secondary time servers. (You can use the Fermilab core router 131.225.8.200 as primary and 131.225.17.200 as secondary.) Check `NTP` for both.

   At the top of the window, check `Automatically synchronize time:` and check `Once at system startup`. Click `Synchronize Now` to set the system clock and check the time server setting. Click **OK**.

2) Navigate to **START > PROGRAMS > REFLECTION > UTILITIES > KERBEROS MANAGER** to open the **Reflection Kerberos Manager** application. Pull down the **CONFIGURATION > CONFIGURE REALMS** menu, make sure the *Configuration* tab is selected. Highlight the `PILOT.FNAL.GOV` realm and click **PROPERTIES**.

   Click the *Realm Defaults* tab and change the **PRE-AUTHENTICATION** from `None` to `Encrypted timestamp`. Click **OK**.

   Select the *Machine Defaults* tab and change the **DEFAULT TICKET LIFETIME** to 13 hours. Click **OK**.

   Back on the **REFLECTION KERBEROS MANAGER** window, check that your full principal name is filled in (it should be by default). To go ahead and authenticate, click **AUTHENTICATE** and enter your Kerberos password when prompted. You should see a ticket-granting ticket `krbtgt/PILOT.FNAL.GOV@PILOT.FNAL.GOV`. If you receive an error message instead, check that the above steps were followed correctly and you typed the right password. If you continue to receive and error message, send the exact error message text to *dcd_computer_security_team@fnal.gov.*

3) You may want to create a shortcut for the **Reflection Kerberos Manager** application in your **PROGRAMS > STARTUP** folder to start the application automatically each time you log into Windows.

4) Proceed with the installation of **Reflection Suite for X**, below.

5.4 Installing Reflection Suite for X®

1) Make sure you have a license for the **WRQ Reflection**® software.

2) Browse from **NETWORK NEIGHBORHOOD** to `\\Pckits\WRQ\ReflectionX` and run the `Install.exe` you find there.

3) On the initial setup screen, choose `Reflection Suite for X (using single-user Setup)`.

---

1. The NT servers and some workstations (including ESH and FESS) use the **timeserv** program that comes with the **NT Resource Kit**. The servers are configured to look at the gateway given in the IP request.

4) Click through the usual license agreement and welcome screens.

5) On the **REFLECTION FOLDER** screen, leave the default `C:\Program Files\Reflection` and click **NEXT >**.

6) On the **USER FOLDER** screen, leave the default `C:\Program Files\Reflection\User` and click **NEXT >**.

7) When selecting components to install, the components included in the typical configuration are selected by default (click the **TYPICAL** box to make sure you're starting at this configuration). From this default set, uncheck `IBM Host Access` and `NFS`. *Don't* be tempted to install the NFS client!

Click the small box just to the left of `Network Applications`, then on the window that pops up, uncheck `Servers` and click **OK** to return to the main component selection box. Click **NEXT >**.

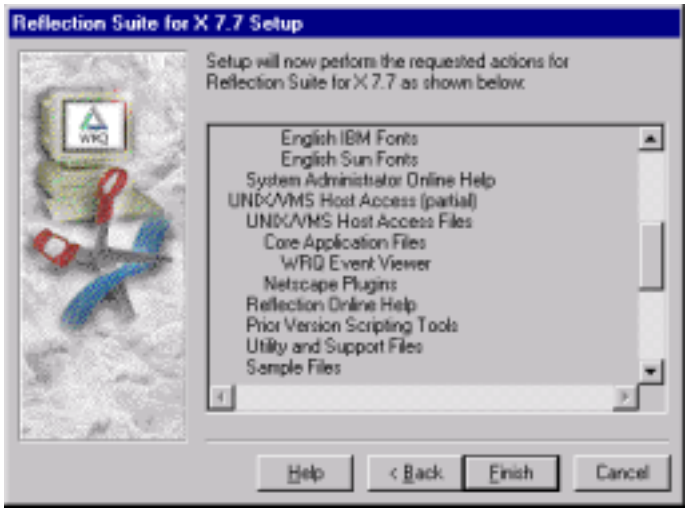
If doing the minimal install, make sure that only `UNIX/VMS Host Access` and `Network Applications > FTP Client` are checked.

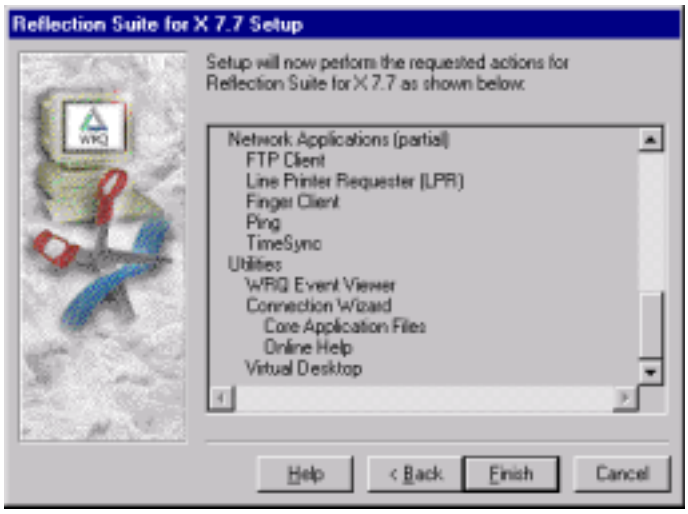
8) On the **SHORTCUT FOLDER OPTIONS** screen, leave the default `Reflection` and *don't* check `Do not create shortcuts`. Click **NEXT >**.

9) On **DEFAULT APPLICATION SETTINGS**, `telnet` should be checked. Leave it checked. Click **NEXT >**.

10) On the password dialog box, uncheck the default `Save Passwords` and click **NEXT >**. (Saving passwords isn't a good idea.)

11) A window pops up (see below) that shows your installation parameters and says `Setup will now perform the requested actions for Reflection Suite for X 7.7 as shown below`. To show you all the information to expect, we include a series of three images which display the three contiguous portions of the list (due to the number of items, there is no overlap):





12) Assuming all the information is correct, click **FINISH** and prepare to wait while the system runs the install and copies fonts. When the installation process finishes, click **RESTART WINDOWS**[1], and then continue with the configuration steps, below.

1. If you have placed several items in the **STARTUP** folder in your user profile, logon can be slowed down considerably. To log on quickly and suppress all these applications, press down the Shift key while you're logging on.

5.5 Configuring Reflection Suite for X®

1) When Windows restarts, invoke the **Reflection X Client Manager**. You will be prompted to run the **Reflection X Performance Tuner**. Click **YES** to run these tests to optimize performance before using the X client manager.

2) The **Reflection X Client Manager** next prompts you to **SELECT XDMCP HOST**. Click **CANCEL** if you don't use XDMCP (X Display Manager Control Protocol) to start clients.

3) When prompted to run the "Client Wizard", click **CANCEL** (the wizard does not set up Kerberos connections, so you need to set them up without the wizard). When you click **CANCEL**, the **Reflection X Client Manager** comes up automatically.

This or another X client manager, e.g., **eXceed**, must be running before any X client connections can be opened. Close it for now.

4) Log off and log back on with your normal userid (as opposed to `Administrator`). You should find **REFLECTION FTP NEIGHBORHOOD** on your desktop.

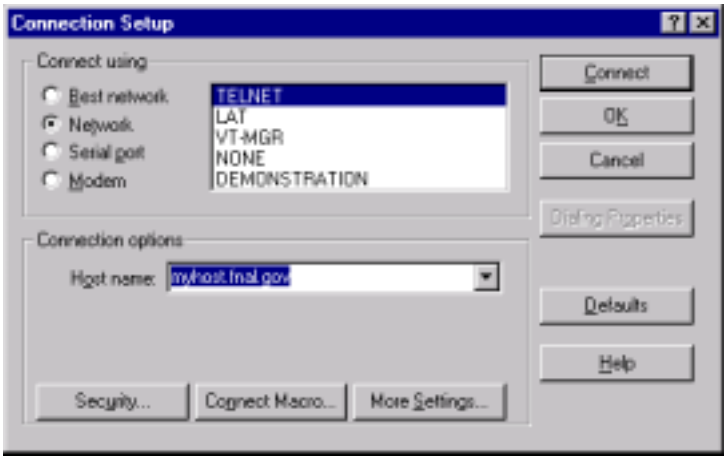
5) You may want to create a shortcut for the **Reflection X Client Manager®** application in your **PROGRAMS > STARTUP** folder to start the application automatically each time you log into Windows. If so, we recommend that you specify "Run: Minimized" in the shortcut properties.

5.6 Configuring Reflection telnet Connections

5.6.1 Connect to Kerberized Host

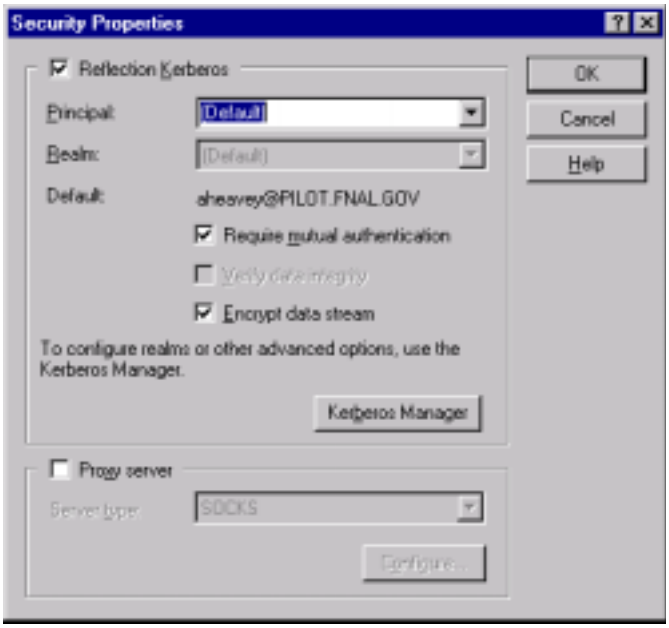
1) To configure the **Reflection telnet** client to access a remote Kerberos system, first open **START > PROGRAMS > REFLECTION > HOST - UNIX AND DIGITAL**.

2) To configure your connection, start from the **UNTITLED - REFLECTION FOR UNIX AND DIGITAL** window. Pull down the **CONNECTION > CONNECTION SETUP...** menu, click the **NETWORK** radio button in the **CONNECT USING** area, and make sure `TELNET` is highlighted in the scroll box.



3) Fill in the **HOST NAME** of the Kerberos system.

4) Click **SECURITY**, check `Reflection Kerberos` and check `Encrypt data stream`.



This is very important! **Reflections** does not forward credentials from your PC to the host, therefore to access Kerberized resources on the host you will need to **`kinit`** (see section 7.3.1 *Obtaining Tickets*) and enter your password once you're on the host. **Always make sure the data stream is encrypted before entering your password!**

`Require mutual authentication` should already be checked; leave it checked. Click **OK** to return to the **CONNECTION SETUP** window.

5) If you want to connect immediately, click **CONNECT**. (If you haven't already run Kerberos Manager to obtain a ticket-granting ticket, you'll be prompted for your Kerberos password and then logged in. In either case, the **Reflection** software does not forward your credentials to the host system.)

6) Optional: From the **UNTITLED - REFLECTION FOR UNIX AND DIGITAL** window you can go to the **SETUP** menu and choose to configure a number of nonessential but useful features in the areas of terminal emulation, keyboard mapping, mouse mapping, display, and so on.

If you will be logging onto several different hosts, it is particularly useful to set each Window Title to the node name. (To do this, in the **SETUP > DISPLAY... > OPTIONS** dialog box, click on the **?** (upper right corner, as usual), then on **WINDOW TITLE > DETAILS** for instructions.)

7) Run **FILE > SAVE AS** to save the host-specific settings in a file that you name. The system prompts you to save the file in the **REFLECTIONS > USER** folder. It creates the **REFLECTIONS SESSIONS** folder in your **START** menu and asks if you want to create a shortcut on your desktop.

5.6.2 Connect to nonKerberized Host

For connections allowing weak (standard) authentication, you don't need to worry about the **Kerberos Manager** since credentials aren't an issue. To configure a standard **telnet** connection, follow the same steps as in section 5.6.1 *Connect to Kerberized Host*, but make sure the host name is a nonKerberized node, and eliminate step (4) which sets the Kerberos security.

5.6.3 Create a Template Configuration

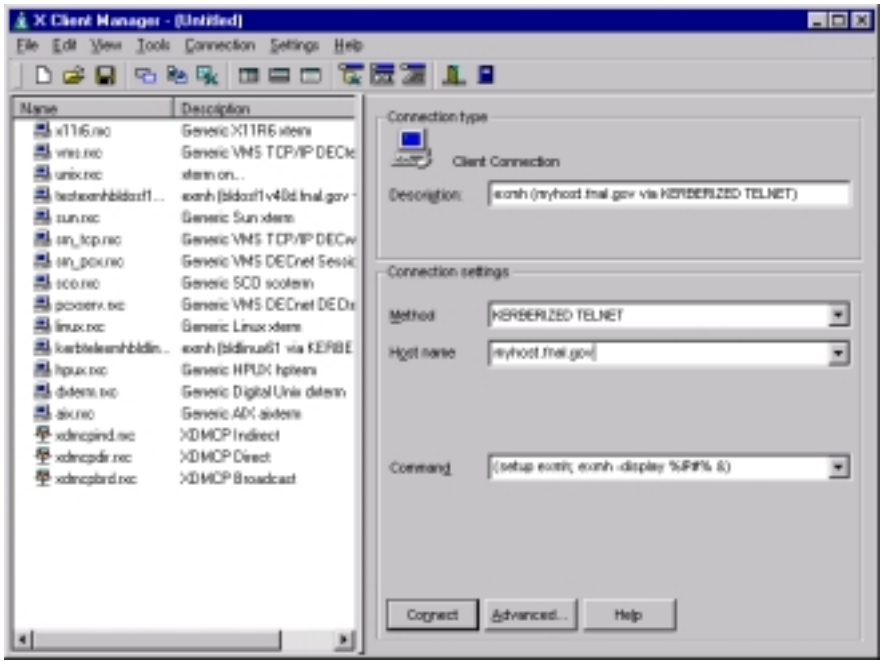
To create a template configuration, first create and save a model configuration for any Kerberized or nonKerberized host, as appropriate, as described in section 5.6.1 *Connect to Kerberized Host*. Pull up that configuration, log on to the host, and exit out. Select **CONNECTION > CONNECTION SETUP...**. Remove the host name from the configuration and save it as a template file (choose an appropriate filename). To use the template to create a host-specific configuration, bring up the template (it should appear in your **REFLECTIONS SESSIONS** folder), add the desired host name, and save it as a different file with a host-specific name.

5.7 Configuring telnet Connection to Host with Application Startup

Here we describe how to connect to a host using the **Reflection** software and start a generic X application. (This procedure is somewhat dependent on the target OS.) Be aware that this method provides unencrypted connections only, so use this only for applications that don't require Kerberos authentication.

1) Use **START > PROGRAMS > REFLECTION > REFLECTION X** to start the **Reflection X Client Manager** if it isn't already running. If the X client manager is running minimized, click its icon in the system tray to restore its window.

2) Use **FILE > NEW** to open the **NEW CONNECTION** dialog, and select `Client Connection` and click **OK**; *or* highlight an existing connection in the left pane of the **X CLIENT MANAGER** window to use as a template.



3) In the **X CLIENT MANAGER** window, under **CONNECTION SETTINGS** pull down **METHOD**, and scroll down and select `KERBERIZED TELNET`.

4) Enter the **HOST NAME** or select it from the pull down list. (The pull down list is generated from the replies to the **XDMCP** broadcast plus any systems you have used recently.)

5) Enter the following **COMMAND** for execution on the remote host:

```
(setup <Xprogram>; <Xprogram> -display %IP#% &)
```

where **`<Xprogram>`** is some X application, for example **`exmh`** or **`xemacs`**. The special string **`IP#`** substitutes the IP address and display number of the local display (i.e., the PC). Make sure that your UNIX login files don't reset this variable to a different display. Other special strings are documented in the **Reflection X** help file under "Command Line Macro Syntax".

6) Click the **CONNECT** button to establish the connection and run the remote command. (If you haven't already run Kerberos Manager to obtain a ticket-granting ticket, you'll be prompted for your Kerberos password. It's OK to enter it at this stage.)

7) Choose **FILE > SAVE** or **FILE > SAVE AS...** to permanently save the settings.

Other remote client commands and variations are left as an exercise for the reader(!).

Troubleshooting

- To debug the dialog between the **X Client Manager** and the remote host, select **CONNECTION > HOST RESPONSE** before clicking the **CONNECT** button.
- The remote host's prompt character(s) must be recognized by the **X Client Manager** for the connection script to work correctly. Add the correct character(s) if they're not already in the list(s) by selecting **ADVANCED...**.

There is extensive on-line help for other problems or applications.

5.8 Configuring Reflection FTP Connections

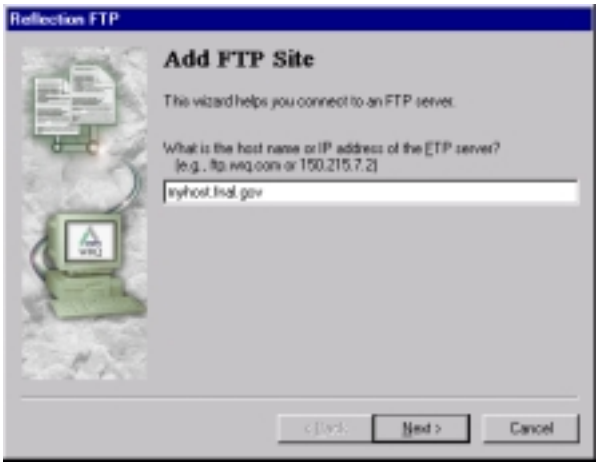
5.8.1 Connect to Kerberized Host

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

2) Open **START > PROGRAMS > REFLECTION > FTP CLIENT** or open the **REFLECTION FTP NEIGHBORHOOD** on your desktop.

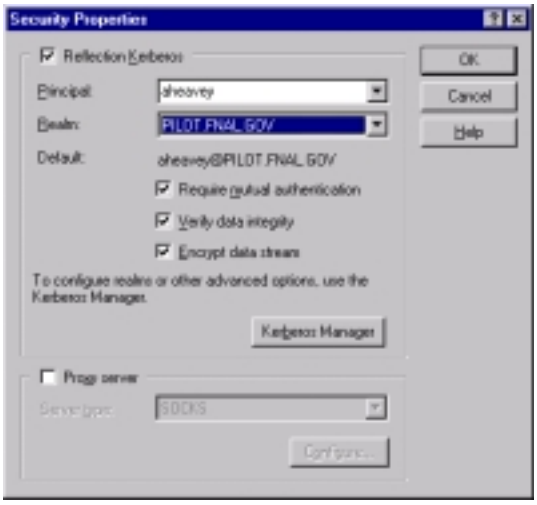
3) If going through the **START** menu, select **CONNECTION > CONNECT...** Click **NEW** in the **CONNECT TO FTP SITE** dialog box. If going through **REFLECTION FTP NEIGHBORHOOD**, double-click on `Add Site`.

On the **ADD FTP SITE** screen, fill in the name of the Kerberized host and click **NEXT >**.



4) In the **LOGIN INFORMATION** box, click the **USER** radio button and click **ADVANCED...** to get to the **<HOST> PROPERTIES** screen.

5) Click **SECURITY** and check `Reflection Kerberos`. Enter your principal (userid portion only), and choose `PILOT.FNAL.GOV` for the realm. Check `Encrypt Data Stream`. (`Require mutual authentication` and `Verify data integrity` are already checked; leave them checked)



6) Click **OK** twice to return to the **LOGIN INFORMATION** screen. Click **NEXT >**.

7) In the **FTP USER LOGIN** screen, your username should be filled in. Don't check `Save my password`. Click **NEXT >**.

8) On the **CONNECT** screen, verify the name of the FTP host, choose whether you want to connect immediately, then click **FINISH**.

5.8.2 Connect to nonKerberized Host

For connections allowing weak (standard) authentication, you don't need to worry about the **Kerberos Manager** since credentials aren't an issue. To configure a standard FTP connection, follow the same steps as in section 5.8.1 *Connect to Kerberized Host*, but make sure the host name is a nonKerberized node, and don't bother with **ADVANCED...** in step (4). Instead, click **NEXT >** and continue from step (7).

5.8.3 Edit an FTP Setup

1) Open **START > PROGRAMS > REFLECTION > FTP CLIENT** or open the **REFLECTION FTP NEIGHBORHOOD** on your desktop.

2) If going through the **START** menu, select **CONNECTION > CONNECT...**. In the **CONNECT TO FTP SITE** dialog box, select the configuration you want to edit and click **PROPERTIES**. If going through **REFLECTION FTP NEIGHBORHOOD**, select the configuration you want to edit and click **FILE > PROPERTIES**.

Chapter 6: Accessing the PILOT.FNAL.GOV Realm

In this chapter we discuss accessing a machine in the PILOT.FNAL.GOV realm from UNIX machines and PCs, directly and/or via applications such as **telnet** and **FTP** and/or through portal mode.

Very important note: Any time you're about to enter your Kerberos password, first verify that you're using an encrypted connection or using the host's directly-connected keyboard! We'll show you how to do this in the different sections that follow.

6.1 Obtaining a Principal

In order to access a machine in the PILOT.FNAL.GOV realm, you need to have a special logon id for the realm, called a *principal*, and an associated Kerberos password. A principal is essentially a realm userid. If you will require access from a nonKerberized machine, also request a CryptoCard® when you get your principal. CryptoCard®s are discussed in section 6.3 *Connecting from a NonKerberized UNIX Machine*.

We plan to have a web-based form available soon for requesting principals. Keep an eye on the updates page!

> Currently, CDF users will be given a Kerberos principal automatically when they apply for an account on the CDF central SGI analysis machine *fcdfsgi2.fnal.gov* (using the **CDF UNIX Account Request Form** at `http://www-cdf.fnal.gov/computing/unixaccountrequest.html`). CDF users who want a *only* a Kerberos principal can send email to *cdfsys@fnal.gov.*
>
> Other users involved in the pilot project may request a principal during this phase by contacting their experiment's Computing Division liaison.

In order of preference, your Kerberos principal should match (1) your FNAL email userid, (2) your FNALU account userid, and/or (3) your FNAL NT domain account userid. In addition to the principal, you must have an account on each machine in the realm that you plan to use. The account name should match your principal.

Your principal will come with an initial Kerberos password which you will be required to change after you log on. We defer discussion of passwords to section 7.1.1 *Choosing a Kerberos Password*, but point out here that in contrast to the principal, your Kerberos password should be unique.

6.2 Logging On at a Kerberized UNIX Machine

When you're at a Kerberized UNIX desktop in the PILOT.FNAL.GOV strengthened realm, you log in normally, entering your Kerberos password. You will automatically obtain your credentials, including an initial ticket-granting ticket (TGT).

If authentication fails, one of two things is likely to be wrong: your password or the date/time on your system. The clocks on the client and server machines must be within five minutes of each other. If you determine that the problem might be the time synchronization, see section 4.1.4 *Synchronize your Machine with Time Server*.

6.3 Connecting from a NonKerberized UNIX Machine

In order to avoid transmission of reusable clear-text passwords over an unprotected network, connecting directly to a machine in the PILOT.FNAL.GOV realm from a nonKerberized machine involves the entry of a single-use password. The machines in the PILOT.FNAL.GOV realm are configured to behave in *portal mode* when access is attempted from outside the strengthened realm or a trusted realm. In portal mode, the machine acts as a secure gateway between the strengthened and untrusted realms, requiring single-use passwords for authentication. The non-reusable authentication method that the Computing Division currently supports is CryptoCard®. We plan to offer the option of using a One Time Password (OTP) in the future. These methods are described in sections 6.3.1 *CryptoCard®* and 6.3.2 *One Time Password (OTP) (not yet supported)*. If you require access from a nonKerberized machine, request a CryptoCard® from your Computing Division liaison.

To log on to a machine in the PILOT.FNAL.GOV realm from your nonKerberized machine, run **`telnet <host>`** (the standard, nonstrengthened version of **telnet**, as the Kerberized version is not available on nonstrengthened machines) as usual.[1] It will prompt you to provide a non-reusable password rather than your Kerberos password.

Once you've logged on successfully using non-reusable authentication, the KDC "knows who you are", and the machine obtains your Kerberos credentials for you. From a given strengthened machine, you should not be required to provide your Kerberos password when accessing other machines in the PILOT.FNAL.GOV. **In particular, never enter your Kerberos password when using an unencrypted connection!**

1. In addition to **telnet**, we plan to make **FTP** available soon.

6.3.1 CryptoCard®

Currently, if you will require access from a nonKerberized machine, you need to request a CryptoCard® from your Computing Division liaison. A CryptoCard® is a calculator-style, battery-powered token that must be initialized and synchronized with the KDC prior to issue.

It comes with an initial PIN that you are required to change. You need to keep your CryptoCard® with you whenever you might need to access a machine in the strengthened realm from a nonKerberized machine.

No special hardware or software is required on the nonKerberized machine. Every time you login from an untrusted machine, the KDC generates an eight-digit string called a *challenge*. The CryptoCard® encrypts the challenge with the secret key shared by itself and the KDC in order to generate a *response*. The response is a single-use eight-digit hex password. The only information the card stores is the shared encryption key, the PIN, and the response from the previous challenge.

If you ever forget your PIN or if the card locks up, send email to *dcd-security@fnal.gov* to arrange getting your CryptoCard® reprogrammed.

Using CryptoCard® (the First Time)

To set your PIN:

When you first obtain your CryptoCard®, you are required to change the initial PIN.

1) Turn on the card (using the **ON/OFF** button), enter your initial PIN and press **ENT**.

2) At the prompt `New PIN?` enter a new PIN and press **ENT**. The minimum length is four digits, but it can be as long as eight.

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

4) If you're not going to log on now, turn off the 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) above.

To log on:

1) Run **telnet** normally on your nonKerberized machine to the strengthened host, and enter your userid 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.

2) Turn on your CryptoCard® and enter your new PIN, followed by **ENT**. (You are limited to seven wrong-PIN tries before lockout.)

3) 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.

4) 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** and see a response of eight hex digits.

5) Enter the CryptoCard® response at the host system prompt (it is case-insensitive). Press **RETURN** and you should be logged in with tickets.

6) Turn off your CryptoCard®, or it will do so automatically in 60 seconds. 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).

Using CryptoCard® (Normally)

1) Run **telnet** 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 8-digit challenge.

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

3) The card is configured to display the id `Fermilab`. Press **ENT** again.

4) The CryptoCard® displays a challenge. Compare the challenge on the host to the one on the card:

   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 it will do so automatically in 60 seconds. 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).

Summary of the "normal" logon steps:

1) CryptoCard®: **ON**, **`[PIN]`**, **ENT**, **ENT**, **ENT**

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

3) CryptoCard®: **OFF**

6.3.2 One Time Password (OTP) (not yet supported)

This system is not yet implemented, and many of the details of its implementation are not yet determined. We expect it will work roughly like this:

When you get your account and request access via OTP, the KDC will generate a numbered list of passwords for you. The KDC maintains the list internally, and you will need to keep a physical copy with you whenever you plan to access the Kerberized system from a nonKerberized machine. To gain access:

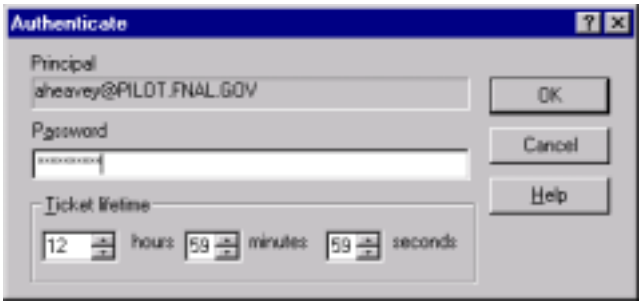
- Run **telnet** normally from the "untrusted" machine to the host.
- Enter your userid at the host "portal-mode" prompt. The KDC chooses a number corresponding to one of the passwords in the list (it chooses them in order, from the highest number down), and the host system communicates the number to you.
- The host then displays a prompt at which you enter the corresponding password from the list.

6.4 Logging On Through Reflection® Software from a PC

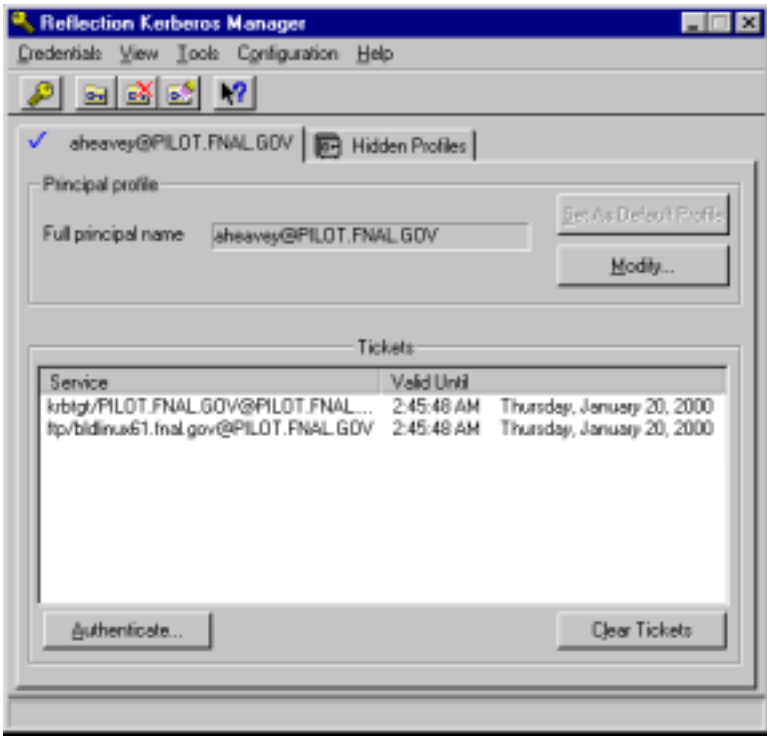
6.4.1 Start the Kerberos Manager (Optional)

The **Reflection Kerberos Manager®** program preauthenticates you to the strengthened realm. When you start it, you give it your Kerberos password, and it then obtains (nonforwardable) initial tickets for you and allows you to freely access Kerberized hosts in the PILOT.FNAL.GOV. You can start this application ahead of time if you like (if you don't, you will be required to provide your Kerberos password the first time you attempt access to a host). You may even want to create a shortcut for the **Reflection Kerberos Manager®** application in your **PROGRAMS > STARTUP** folder to start the application automatically each time you log into your PC.

To start the **Kerberos Manager**, navigate to **START > PROGRAMS > REFLECTION > UTILITIES > KERBEROS MANAGER** and double click. In the **KERBEROS MANAGER** window, check that your full principal name is filled in (it should be by default). You may also see old tickets displayed. Click **AUTHENTICATE** and enter your Kerberos password when prompted:



Click **OK**. Back on the Kerberos Manager window, you should see (at least) the new ticket-granting ticket (TGT) `krbtgt/PILOT.FNAL.GOV@PILOT.FNAL.GOV` (You may see a service ticket as well, as shown below for **FTP**):



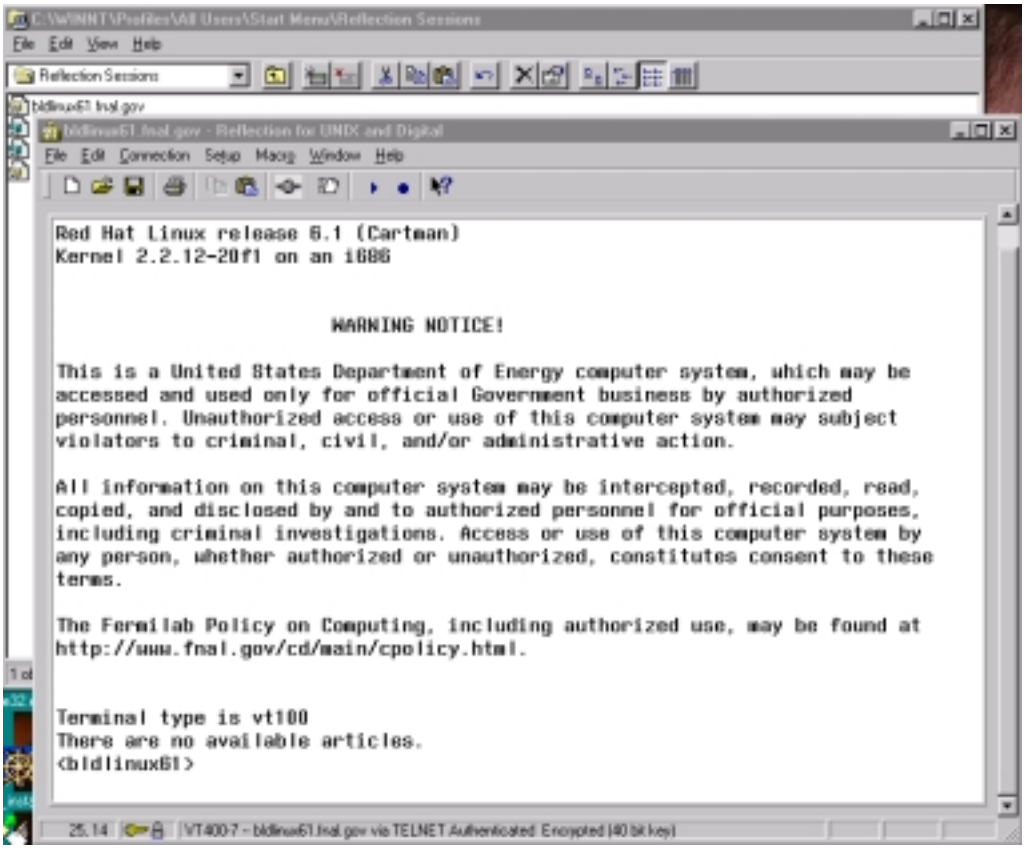
If you receive an error message instead, check that the above steps were followed correctly and that you typed the right password. If you continue to receive an error message, send the exact error message text to *dcd_computer_security_team@fnal.gov.*

6.4.2 Run a telnet Session to Kerberized Host

To use the **WRQ Reflection® telnet** client to access machines in the strengthened realm, you need to set (and save) a separate **telnet** configuration for each host. This procedure is outlined in section 5.6 *Configuring Reflection telnet Connections*. Each saved configuration is maintained in the **REFLECTIONS SESSIONS** folder on your desktop, its default filename corresponding to the host name.

You can choose to start the **Reflection Kerberos Manager®** first to obtain your credentials, as explained in section 6.4.1 *Start the Kerberos Manager (Optional)*. Unless you plan to run X applications, you don't need to start the X window manager.

Double click on the file in your **REFLECTIONS SESSIONS** folder corresponding to the host to which you want to connect. It will bring up a VT window and log you in:

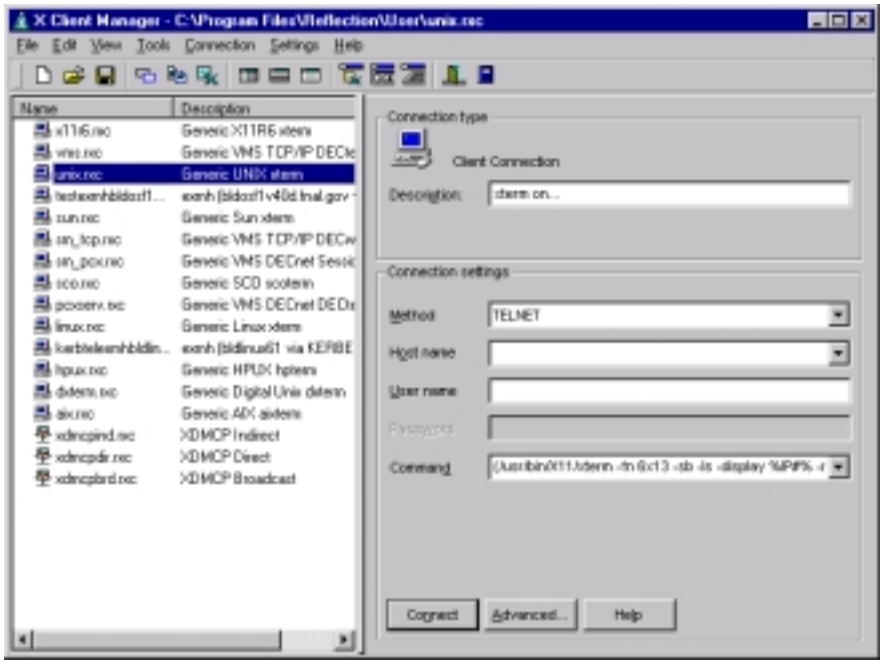


Very important note: The **Reflections** software does not forward credentials from your PC to the host, therefore you must `kinit` (see section 7.3.1 *Obtaining Tickets*) and enter your password once you're on the host if you plan to access any Kerberized resources. *Before* you enter your password, glance at the bottom of the VT window and verify that it says "Encrypted" (as shown on the above image). If it doesn't, *log out and recheck your configuration* (under **CONNECTION>SECURITY**, check `Reflection Kerberos` and check `Encrypt data stream`)!

6.4.3 Display UNIX Host X Applications on your PC

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.)

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



The best thing to do at this point is to minimize this window and start a **telnet** session as described in section 6.4.2 *Run a telnet Session to Kerberized Host*. Once you're connected, verify that your $DISPLAY is set correctly on the UNIX host (at Fermilab, this should already be set for you in your UNIX login files; if it's not, check these files). Run **`kinit`** if necessary (after verifying that your connection is encrypted), then run your X applications from this VT window.

You can opt to connect to a host directly from the **X CLIENT MANAGER** window, *but* it does not provide encrypted connections. Remember that the **Reflection** software doesn't forward credentials to the host. If you will need credentials on the host, go through a normal **telnet** connection. **Do not `kinit` from an X window!**

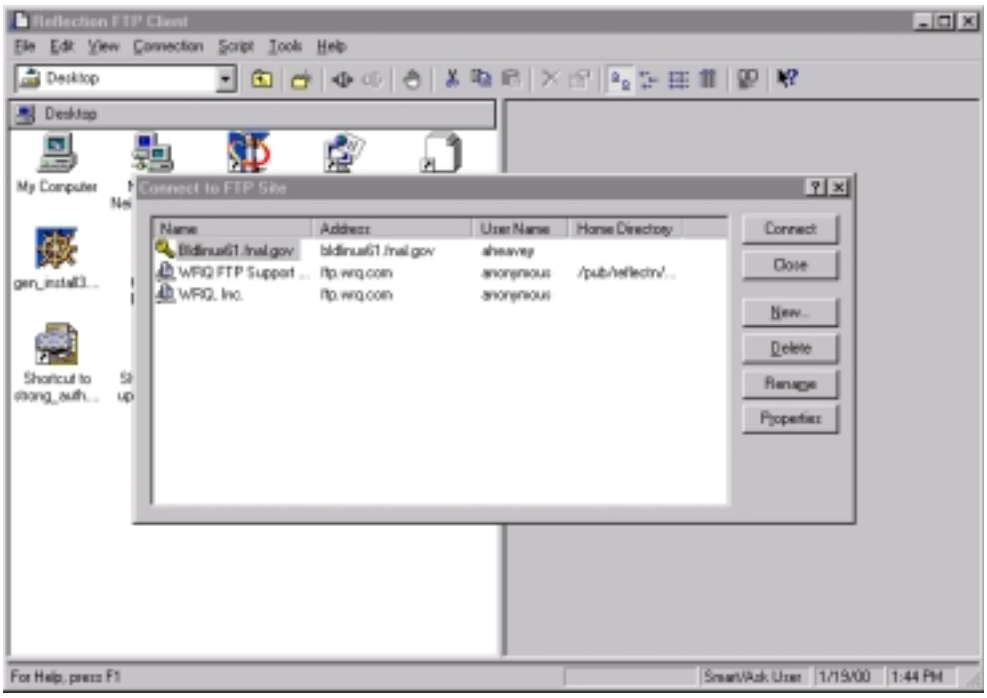
To connect using this window, choose a connection option from the left pane and customize it as desired, or create (and optionally save) a new connection configuration. Select `KERBERIZED TELNET as` the **METHOD** (if you leave it as just `TELNET`, it will require use of a CryptoCard®). Then click **CONNECT**.

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 5.7 *Configuring telnet Connection to Host with 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.

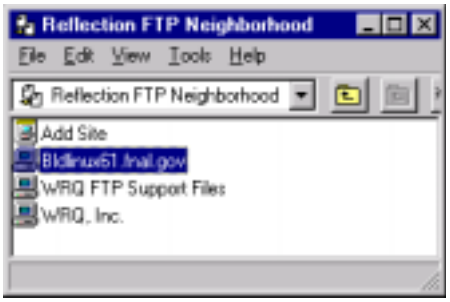
Procedures for other X window manager products are not documented here.

6.4.4 Run an FTP Session to Kerberized Host

Configuration of **FTP** sessions is covered in section 5.8 *Configuring Reflection FTP Connections*. There are two ways to use the **Reflection FTP** client to access a Kerberos system: (1) open **START >PROGRAMS > REFLECTION > FTP CLIENT**:



or (2) go straight to **REFLECTION FTP NEIGHBORHOOD** on your desktop:



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

6.5 Logging On from Home

Users who require access to Fermilab strengthened machines from home can acquire a CryptoCard® (see section 6.3.1 *CryptoCard*®) or install Kerberos clients locally.

NonKerberized	If you choose to keep your home PC unstrengthened, then you will be able to connect to an on-site strengthened machine normally using **telnet** or other network utility, and the strengthened machine will respond in portal mode. Use your CryptoCard® to generate a password.
Kerberized	If you have installed **kerberos** (Linux) or **Reflection®** (Windows®), you can preauthenticate (using **`kinit`** or the **Kerberos Manager**, respectively)[1]. Then you can connect to an on-site strengthened machine normally using **telnet** or other network utility without typing your password each time and without using a CryptoCard®.

1. During installation of either of these products, your machine obtains information from the KDC which permits it to preauthenticate you; no live connection to the KDC is needed for preauthentication.

Chapter 7: Using Kerberos

This chapter provides all the basic information you need in order to manage your Kerberos tickets and work in a Kerberized environment. In particular, we cover passwords, ticket options and management, and account access files. The Kerberos commands and features of Kerberized network programs are documented in Chapter 9: *Command Descriptions* and Chapter 10: *Kerberized Network Programs*.

7.1 Your Kerberos Password

When you get an account on the PILOT.FNAL.GOV strengthened realm, you are given an initial Kerberos password that you are required to change. The Fermilab Computer Security Team has imposed some restrictions on passwords in accordance with DOE guidelines. Currently, a password for the PILOT.FNAL.GOV strengthened realm is required to contain a minimum of ten characters from at least two of the following five classes: lowercase letters, uppercase letters, numbers, punctuation, and all other characters. Passwords the system considers “bad” will be rejected. (Passwords are checked against the “cracklib” dictionary, which will often surprise you by its thoroughness!)

7.1.1 Choosing a Kerberos Password

Need some ideas for thinking up a good password?[1] Remember, a good password is one you can remember, but that no one else can easily guess. Examples of passwords that would be good *if they weren’t listed in this manual* include:

- some initials, like "GykoR-66." for "Get your kicks on Route 66."
- an easy-to-pronounce nonsense word, like "slaRooBey" or "krang-its"
- a misspelled phrase, like "2HotPeetzas!" or "ItzAGurl!!!"

Note: Don't actually use any of the above passwords. They're only meant to show you how to make up a good password. Passwords that appear in a manual are the first ones intruders will try.

Another note: Choose a unique password for the PILOT.FNAL.GOV strengthened realm. In other words, don’t use this same password for other (nonstrengthened) systems; it risks exposing your Kerberos password.

---

1. These ideas were lifted from MIT’s Kerberos V5 User’s Guide (C) 1996, local copy stored at `http://www-dcd.fnal.gov/computersecurity/Strong-Auth/UserDocs/user-guide.html`.

Yet another note: Never give your password to anybody for any reason. If you really need to give someone access to your account (this practice is discouraged, by the way), add the person's principal to your `.k5login` or `.k5users` file as described in section 7.4 *Account Access for Multiple Principals*.

7.1.2 Changing your Kerberos Password

Before changing your password, verify that you're using an encrypted connection or are using the host's directly-connected keyboard!

To change your password, run the **`kpasswd`** command, e.g.,:

```
% kpasswd
kpasswd: Changing password for aheavey@PILOT.FNAL.GOV.
Old password:                 <--- type your initial password here.
kpasswd: aheavey@PILOT.FNAL.GOV's password is controlled by the policy default,
which
requires a minimum of 10 characters from at least 2 classes (the five classes
are lowercase, uppercase, numbers, punctuation, and all other characters).
New password:                 <--- type your new password here.
New password (again):         <--- type your new password here for confirmation.
Kerberos password changed.
```

If you choose a password that is too short, you will see this error message:

```
kpasswd: New password is too short.
Please choose a password which is at least 10 characters long.
```

If it's long enough but you haven't met the multiple-class requirement, you'll see:

```
kpasswd: New password does not have enough character classes.
The character classes are:
        - lower-case letters,
        - upper-case letters,
        - digits,
        - punctuation, and
        - all other characters (e.g., control characters).
Please choose a password with at least 2 character classes.
```

On strengthened UNIX systems running AFS, there are two **`kpasswd`** commands, one for AFS and one for Kerberos. Your $PATH should be set such that the Kerberos **`kpasswd`** comes first. Kerberos is implemented at Fermilab such that your AFS tokens will be obtained automatically. If you are unsure which **`kpasswd`** is being invoked, force the system to use the Kerberos version by running **`setup kerberos`** first.

7.2 Ticket Options

Kerberos uses encrypted records called *tickets* to authenticate to Kerberized services. Kerberos tickets can be forwardable, renewable, post-dated and/or proxiable. The strengthened versions of network programs generally provide options to enable these features (see section 9.2 *Kerberized Network Programs*).

Forwardable — 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. Generally one forwards only the TGT, since that can be used to obtain other needed tickets.
Renewable	A renewable ticket can have its lifetime extended, by action of the user, beyond the default 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 may not be a TGT, it must be for some other service.

7.3 Ticket Management

The PILOT.FNAL.GOV strengthened realm uses the Kerberos V5 login program which issues you a Kerberos ticket-granting ticket (TGT) by default when you log in. As you access services in the strengthened realm, the tickets needed for the services will be granted automatically.

Generally the only ticket you need to worry about is the TGT. The default lifetime of TGTs and other tickets is set to 13 hours. In order to support both long interactive sessions and batch jobs, a TGT can be renewed (before expiration) up to the maximum renewable lifetime of seven days. Once the TGT expires, new connections cannot be opened, but existing connections are not terminated.

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 log on and get Kerberos credentials, you also automatically get an AFS token. The other operations described in this section (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 maximum renewable lifetime of the Kerberos TGT.[1]

7.3.1 Obtaining Tickets

If you need to obtain a service ticket or a TGT yourself (e.g., when tickets expire, when using **Reflection** software from a PC, when you need to use a different userid, or if you want different parameters), use the **`kinit`** command:

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

Several flags are available for **`kinit`**, as listed in section 9.1.1 *kinit* and in the man pages, which allow you to specify ticket lifetime (**`-l <lifetime>`**), and/or ticket start time (**`-s <start_time>`**), request renewable tickets (**`-r <renewable_life`**), specify a particular service (**`-S <service_name>`**), and other options.

1. However, 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.

Before running **kinit**, first verify that you're using an encrypted connection or are using the host's directly-connected keyboard! Exceptions:

- **kinit -s <start_time>** requests a ticket with a start time in the future (called a post-dated ticket). In order to validate it, you will need to run **kinit -v** after the specified start time. **kinit -v** does not require password entry, therefore it can be issued over any connection.
- **kinit -r <renewable_life>** requests a renewable ticket. In order to renew it, you will need to run **kinit -R** before it expires. **kinit -R** does not require password entry, therefore it can be issued over any connection.

7.3.2 Viewing Tickets

Use the command **klist** to check 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@PILOT.FNAL.GOV

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

- The first listed ticket is a Kerberos TGT (`krbtgt`) for the service principal `krbtgt/PILOT.FNAL.GOV@PILOT.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 9.1.2 *klist* and in the man pages.

7.3.3 Destroying Tickets

Tickets can outlive an interactive session and they can be stolen. Therefore it's a good idea to explicitly destroy your tickets when you log out. The command **kdestroy** destroys all your tickets. To automate this, add the command **kdestroy** to your `.logout` file.

Also, if you are going to be away from your machine and are concerned about an intruder using your permissions, it is safest to either destroy your tickets, or use a screensaver that locks the screen.

See section 9.1.4 *kdestroy* or the man pages for a description of **kdestroy**.

---

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

7.3.4 Forwarding Tickets

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. However, tickets can be set as forwardable (by action of the user). A forwardable ticket (usually a TGT) can be used to request a new ticket, but with a different IP address. Thus, a user can use his current credentials to get valid credentials on another machine.[1]

The strengthened versions of programs such as **rsh**, **rcp**, **telnet**, and **rlogin** support forwarding copies of your tickets to the remote host. See section 9.2 *Kerberized Network Programs.*

7.3.5 Renewing Tickets

Tickets can be requested as *renewable*.[2] They can be given a renewable lifetime less than or equal to the maximum allowable lifetime (seven days). A renewable ticket still has the normal default lifespan of 13 hours, but before it expires it can be renewed as long as its renewable life has not expired.

Request a Renewable Ticket

Renewable tickets are requested using the **`kinit`** command with the **`-r`** option (this sets the `R` flag on the ticket for `Renewable`). **`kinit -r`** requires password entry, therefore it should only be performed at the keyboard of a strengthened machine or over an encrypted connection. If your tickets expire and you're using an unencrypted connection to the strengthened node, **don't run `kinit -r`!** Log out and log back in so as not to expose your password; you will get new tickets.

Renew a Ticket

To renew a renewable ticket, use the **`kinit`** command with the **`-R`** option. **`kinit -R`** does not require password entry, therefore it can be issued over any connection.

Example

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

```
% kinit -r 4d
    Password for aheavey@PILOT.FNAL.GOV:                    <--- type your password here.
```

Then, before the default lifetime of 13 hours has passed (you cannot renew an expired ticket), and before four days expire, renew the ticket using the **`-R`** option:

```
% kinit -R
```

---

1. The KDC administrator has the option of disallowing forwardable tickets on a per-site or per-principal basis.
2. If the `/etc/krb5.conf` file on the machine sets `renewable=true` and `default_lifetime=<value greater than 13 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.

The ticket will remain active an additional 13 hours or until its original four days expires, whichever comes first.

7.4 Account Access for Multiple Principals

There are situations in which it is useful to allow access to an account on a strengthened machine by more than one principal. A shared account is one example. Another is when a user has accounts with different userids on machines from which he or she needs to access the strengthened realm. Access to an account for principals other than the one whose userid matches the principal primary/instance can be granted through Kerberos, without giving the password to the other users. It can be done using either the `.k5login` or the `.k5users` file.

7.4.1 The .k5login File

To add principals to your account, first create a file called `.k5login` in the home directory. Add to this file the Kerberos principal for each userid to which you wish to grant access. Each principal must be on a separate line. Here is a sample `.k5login` file:

```
jenniferp@PILOT.FNAL.GOV
jpedersen@MYUNIV.EDU
```

This file would allow the listed userids to use the strengthened account's user ID without limitations, provided that they had Kerberos tickets in their respective trusted realms. If one or more principals other than your default will be using your account to log in to multiple Kerberized hosts across a network, you need create a `.k5login` file on each of these hosts, and include the necessary principal(s) in each one.

Be aware that if you have a `.k5login` file, all principals that require access must be listed in it, including your local PILOT.FNAL.GOV id (e.g., `jenniferp@PILOT.FNAL.GOV`). If you do not have a `.k5login` file, your default principal is the only one that can be used.

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

7.4.2 The .k5users File

If you want to give restricted access to your account to another principal (access method limited to **`ksu`**; see section 9.3 *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:

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

This allows `jenniferp@PILOT.FNAL.GOV` and `jpedersen@MYUNIV.EDU` to execute any command, but restricts the other listed principals to the shown commands.

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

Chapter 8: Special Topics for the User

In this chapter we document a variety of common operations that work differently in our Kerberized environment. As the pilot project progresses this chapter will grow!

8.1 Usage Notes for PC's with Reflection® Installed

8.1.1 Cutting and Pasting

To cut and paste between a VT terminal window and your PC 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**.

8.1.2 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**.

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

8.2 Configuring cron Jobs

A **cron** job must be able to (a) get a ticket and (b) run unattended. This is a little tricky. An initial Kerberos ticket (the ticket-granting-ticket, or TGT) for a Kerberos principal can be obtained in two ways:

1) based on a password, which is hashed into an encryption key which matches the principal's key as known by the KDC

2) based on a key which has been stashed in a *keytab* file (This key, of course, must also match the key the KDC has on file for the principal.)

Since a **cron** job must be able to run unattended, the first method is ruled out, as we don't want the user to save a copy of the Kerberos password on-line. Two considerations preclude saving the key derived from the user's password keytab file for the second method:

- First, saving that key in a file would be just as insecure as saving the password itself.
- Second, the process of creating a keytab file causes the randomization of the key. This not only invalidates any previous keytab which may have been created, but makes it essentially impossible for the user to ever use a password again to authenticate as that principal.

The solution that has been chosen at Fermilab for granting tickets to **cron** jobs involves allowing the user to create a separate principal solely for authenticating **cron** jobs. These principals may be deleted at will and they can be given the minimum authorization necessary for the user's needs. Here we provide instructions for creating a "**cron** principal".

Note: We plan to create a new command that will create or delete a user's **cron** keytab file in a protected directory which will be separate for each host. In the meantime, we discourage the running of **cron** jobs, in the interest of protecting the keytab files.

8.2.1 Create a cron Principal and a Keytab File

First setup **kerberos**, then invoke the Kerberos administration program, **`kadmin`** (In these instructions, we use the sample Kerberos user principal *user1@PILOT.FNAL.GOV* and define the instance *user1/cron@PILOT.FNAL.GOV* for **cron** jobs.):

```
% setup kerberos
```

```
% kadmin -p user1@PILOT.FNAL.GOV
```

`kadmin` requests your password then responds with the **`kadmin:`** prompt. Use the **`addprinc`** command to add a new principal for your **cron** jobs:

```
kadmin: addprinc -randkey user1/cron
```

You should see the response:

```
Principal "user1/cron@PILOT.FNAL.GOV" created.
```

Next create a keytab file in a local writable storage area, e.g., `/usr/tmp/`, using the **`ktadd`** function:

```
kadmin: ktadd -k /usr/tmp/user1-cron.keytab user1/cron
```

You should see the response:

```
Entry for principal user1/cron with kvno 9, encryption type DES-CBC-CRC added to keytab
WRFILE:/usr/tmp/user1-cron.keytab.
```

Then exit out:

```
kadmin: exit
```

8.2.2 Configure the cron Principal

You need to grant *user1/cron* access to the required host/account resources. This involves creating a `.k5login` file (see section 7.4 *Account Access for Multiple Principals*) if you don't already have one, and adding the necessary principals. Log into the desired host (shown here as *somehost.fnal.gov*) with the appropriate userid (usually your default). Then run, for example:

```
% echo user1@PILOT.FNAL.GOV >[>] .k5login
% echo user1/cron@PILOT.FNAL.GOV >> .k5login
    % logout
```

Notes:

1) If you have no `.k5login` file at all, the system grants access only to your default principal.

2) If a `.k5login` file exists, it must contain an entry for each principal that requires access, including your default and this new **cron** principal.

8.2.3 Create a cron Job

Now set up the **cron** job that needs network access. In order to be authenticated, the **cron** job *must* contain a **`kinit`** command, e.g.,:

```
kinit -k -t <your-keytab-file> user1/cron
```

Once this **`kinit`** command is run, only *user1/cron* is authenticated. Under normal circumstances you cannot be authenticated as two separate users at the same time[1]. Here is a sample file `rcp-cron-job` which runs an **`rcp`** command:

```
#!/bin/sh
PATH=/usr/krb5/bin:$PATH # just making sure ...
export PATH
# Get a 1/2 hour ticket for user1/cron ...
kinit -l 30m -k -t /usr/tmp/user1-cron.keytab user1/cron
# The following is a sample command to run as the cron job:
rcp /master/files/* someuser@somehost.fnal.gov:/duplicate/files
kdestroy
EOF
```

A crontab entry to run this job every night might look like:

```
0 0 * * * /path/to/rcp-cron-job
```

1. It is possible to be authenticated as multiple users, but it is rather complicated and requires use of different ticket caches. We do not document how to do this.

Chapter 9: Kerberos Command Descriptions

In this chapter we list the native Kerberos commands, and provide a brief description and option list with descriptions adapted from the man pages. Programs that Kerberos provides for ticket and password management include **kinit**, **klist**, **kpasswd** and **kdestroy**.

The man pages, and therefore the information in this section, are not complete (as of February 2000). We believe we have documented all the important options at this point. Updates will be made to the man pages in the coming weeks.

9.1 kinit

`kinit` obtains and caches a ticket (a ticket-granting ticket, by default) for the default or specified principal.

9.1.1 Syntax

```
% kinit [-l <lifetime>] [-s <start_time>] [-v] [-p] [-f] \
  [-k [-t <keytab_file>]] [-r <renewable_life>] [-R]    \
  [-c <cache_name>] [-S <service_name>] [<principal>]
```

9.1.2 Option Descriptions

`-l <lifetime>` requests a ticket with the lifetime **`<lifetime>`**. The value for **`<lifetime>`** must be a number followed immediately by a delimiter indicating the unit of time, as follows:

`<n>s` (seconds)

`<n>m` (minutes)

`<n>h` (hours)

`<n>d` (days)

For example: **`kinit -l 90m`**. You cannot mix units; e.g., a value of "**`-l 1h30m`**" will result in an error.

If the **-l** option is not specified, the default ticket lifetime (13 hours, at Fermilab) is used. This option is only useful for specifying a ticket lifetime shorter than the default; to extend the lifetime beyond this limit you must renew the ticket; see **-r** and **-R**.

-s <start_time> requests a postdated ticket, which can be validated (by action of user) any time after **<start_time>**. Its lifetime starts when it gets validated. Format for the date and time can be any of the following:

yyyymmddhhmmss

yyyy.mm.dd.hh.mm.ss

yymmddhhmmss

yy.mm.dd.hh.mm.ss

yymmddhhmm

hhmmss

hhmm

hh:mm:ss

hh:mm

Postdated tickets are issued with the "invalid" flag set, and need to be validated before use; see **-v**.

-v requests that the post-dated ticket in the cache (with the "invalid" flag set) be passed to the KDC for validation. If the start time has passed, the cache is replaced with the validated ticket.

-p requests proxiable tickets

-f requests forwardable tickets

-r <renewable_life>

requests renewable tickets, with a maximum lifetime of **<renewable_life>**. If given a value longer than the preconfigured seven day limit, it will be set to seven days. **<renewable_life>** uses the same format as the **<lifetime>** associated with the **-l** option, with the same delimiters.

-R requests renewal of the renewable ticket. Renewal must take place before the ticket's lifetime expires. An expired ticket cannot be renewed, even if the ticket is still within its renewable life.

-k [-t <keytab_file>]

requests a host ticket, obtained from a key in the local host's keytab file. The name and location of the keytab file should be specified with the **-t <keytab_file>** option; otherwise the default name and location will be used (the default `/etc/krb5.keytab` is not useful here; users cannot read it). Keytab files are generally used for service principals. They are also used for **cron** jobs (see section 8.2 *Configuring cron Jobs*).

-c <cache_name> uses **<cache_name>** as the credentials (ticket) cache name and location; if this option is not used, the default cache name and location are used.

The default credentials cache may vary between systems. If the KRB5CCNAME environment variable is set, its value is used to name the default ticket cache. At Fermilab, this variable is typically set to **FILE:/tmp/krb5cc_<some_string>**. Any pre-existing contents of the cache are destroyed by **kinit**.

-S <service_name> specifies a particular service name to use when getting initial tickets. If this option is not used, you get a ticket-granting-ticket by default.

9.1.3 Examples

Typically you can run the **kinit** command without options. This gets you a 13-hour TGT with the flags `FIA` set by default (`Forwardable`, `Initial`, `Preauthenticated`; flags are viewable using **klist -f**, see section 9.2 *klist*), plus an AFS token if AFS is running on the machine:

```
% kinit
```

Request a ticket valid for three hours using the **-l** option:

```
% kinit -l 3h
```

Using the **-r** option, request a renewable ticket with a maximum renewable lifetime of four days (this sets the `R` flag on the ticket for `Renewable`), and the default lifetime (13 hours):

```
% kinit -r 4d
```

Then, before the lifetime of 13 hours has passed (you cannot renew an expired ticket), and before four days expire (you can renew a ticket multiple times within its renewable lifetime), renew the ticket using the **-R** option:

```
% kinit -R
```

The ticket will remain active an additional 13 hours or until its original four days expires, whichever comes first.

Next, request a postdated ticket (using the **-s** option), with a lifetime of six hours (the lifetime starts at validation time):

```
% kinit -s 12:25 -l 6h
```

Until it gets validated, the invalid ticket has the flags `FdiIA` set by default, where `d` is `PostDated` and `i` is `Invalid`. Validate it after the start time has passed (using the **-v** option):

```
% kinit -v
```

The following command requests a TGT for the principal `user1/cron`, for the duration 30 minutes, with authentication done on the basis of a key previously stored in the keytab file `/usr/tmp/user1-cron.keytab` (this command would normally be included in a **cron** job file, not run interactively; see section 8.2 *Configuring cron Jobs*):

```
kinit -l 30m -k -t /usr/tmp/user1-cron.keytab user1/cron
```

9.2 klist

klist lists the Kerberos principal and Kerberos tickets held in a credentials cache (the default), or lists the keys held in a keytab file.

9.2.1 Syntax

```
% klist [-e] [[-c] [-f] [-s] [<cache_name>]] \
  [-k [-t] [-K] [<keytab_name>]]
```

9.2.2 Option/Argument Descriptions

-e displays the encryption types of the session key and the ticket for each credential in the credential cache, or each key in the keytab file.

-c lists tickets held in a credentials cache (as opposed to keys in a keytab file). Invalid with **-k**. This is the default if neither **-c** nor **-k** is specified.

-f shows the flags present in the credentials, using the following abbreviations:

F	Forwardable
f	forwarded
P	Proxiable
p	proxy
D	postDateable
d	postdated
R	Renewable
I	Initial
i	invalid

Invalid with **-k**.

-s causes **klist** to run silently (produce no output), while still setting the exit status according to whether it finds the credentials cache. The exit status is "0" if **klist** finds a credentials cache, and "1" if it does not. Invalid with **-k**.

<cache_name> specifies the credentials cache. If **<cache_name>** is not specified, **klist** will display the credentials in the default credentials cache (unless instructed to operate on a keytab file). If the KRB5CCNAME environment variable is set, its value is used to name the default ticket cache. At Fermilab, this variable is typically set to **FILE:/tmp/krb5cc_<some_string>**. Invalid with **-k**.

-k	lists keys held in a keytab file (as opposed to tickets in a credentials cache). Keytab files are generally used for service principals. Invalid with **-c**.
-t	displays the time entry timestamps for each keytab entry in the keytab file. Invalid with **-c**.
-K	displays the value of the encryption key in each keytab entry in the keytab file. Invalid with **-c**.
<keytab_name>	specifies the keytab file. If **<keytab_name>** is not specified, **klist** will display the keys in the default keytab file (unless instructed to operate on a credentials cache). Invalid with **-c**.

9.2.3 Examples

Most frequently this command is issued with the **-f** option to indicate the flags set on each ticket:

```
% klist -f
```

```
Ticket cache: /tmp/krb5cc_ttyp0
Default principal: aheavey@PILOT.FNAL.GOV

Valid starting     Expires            Service principal
02/11/00 12:45:33  02/12/00 01:45:33  krbtgt/PILOT.FNAL.GOV@PILOT.FNAL.GOV
        Flags: FIA
02/11/00 12:45:33  02/12/00 01:45:33  afs/fnal.gov@PILOT.FNAL.GOV
        Flags: FA
```

To list the keys in a keytab file (for example a keytab file created for use with a **cron** job, see section 8.2 *Configuring cron Jobs*), use the **-k** and **-t <filename>** options:

```
% klist -k -t /usr/tmp/user1.keytab
```

```
Keytab name: FILE:/usr/tmp/user1.keytab
KVNO Timestamp         Principal
---- ----------------- --------------------------------------------------------
   9 02/15/00 10:34:28 user1/cron@PILOT.FNAL.GOV
```

9.3 kpasswd

The **kpasswd** command is used to change a Kerberos principal's password. **kpasswd** prompts for the current Kerberos password, which is used to obtain a changepw ticket from the KDC. If **kpasswd** successfully obtains the changepw ticket, the user is prompted twice for the new password, and the password is changed.

In the PILOT.FNAL.GOV realm, a policy is in effect that specifies the length and minimum number of character classes required in the new password. The password must be at least ten characters long and contain at least two character classes. The character classes are: lower case, upper case, numbers, punctuation, and all other characters.

9.3.1 Syntax

```
% kpasswd [<principal>]
```

9.3.2 Argument Description

<principal>	Change the password for the Kerberos principal **<principal>**. If not given, the principal is derived from the identity of the user invoking the **kpasswd** command.

9.4 kdestroy

The **kdestroy** utility destroys the user's active Kerberos authorization tickets by writing zeros to the specified credentials cache that contains them. If the credentials cache is not specified, the default credentials cache is destroyed.

9.4.1 Syntax

```
% kdestroy [-q] [-c cache_name]
```

9.4.2 Option Descriptions

-q	Runs quietly. Normally **kdestroy** beeps if it fails to destroy the user's tickets. The **-q** flag suppresses this behavior.
-c <cache_name>	Uses **<cache_name>** as the credentials (ticket) cache name and location; if this option is not used, the default cache name and location are used. If thSe KRB5CCNAME environment variable is set, its value is used to name the default ticket cache. At Fermilab, this variable is typically set to **FILE:/tmp/krb5cc_<some_string>**.

Chapter 10: Kerberized Network Programs

In this chapter we document the Kerberized features of several network programs.

10.1 Introduction

The Kerberos V5 network programs are versions of existing UNIX network programs with the Kerberos features added. These programs include **rlogin**, **telnet**, **FTP**, **rsh**, and **rcp**. These programs have all of the original features of the corresponding non-Kerberos programs, plus additional features that transparently use your Kerberos tickets for negotiating authentication and optional encryption with the remote host. In most cases, all you'll notice is that you no longer have to type your password, because Kerberos has already proven your identity. Be aware that depending on how the program is configured, and whether the target machine is Kerberized, you may be prompted for either userid or password, both, or neither.

You can check the defaults set for these programs in the `[appdefaults]` section of the `/etc/krb5.conf` file. These defaults can be overridden via command line options (and in the cases of **telnet** and **FTP** when invoked without a hostname argument, via commands inside the program).

Note: The Fermilab template `etc/krb5.conf` file sets ticket forwarding and session encryption on for **telnet**, **rlogin** and **rsh**, and off for **rcp**.

Here we list only the command syntax and the Kerberos-added features for these programs.

10.2 telnet

```
% telnet [-8] [-E] [-F] [-K] [-L] [-S <tos>] [-X <authtype>] \
  [-a] [-c] [-d] [-e <escapechar>] [-f] [-k <realm>] [-l     \
  <user>] [-n <tracefile>] [-r] [-x] [<host> [<port>]]
```

The following are the Kerberos options:

-f	(forward) forwards a copy of your tickets to the remote host. If left off, (noforward) turns off forwarding of tickets to the remote host. (This option overrides any forwarding specified in your machine's configuration files.)
-N	turns off ticket forwarding. (This option overrides any forwarding specified in your machine's configuration files.)
-F	(forwardable) forwards a copy of your tickets to the remote host, and marks them re-forwardable from the remote host. If left off, (noforwardable) makes any forwarded tickets nonforwardable. (This option overrides any forwardability specified in your machine's configuration files.)
-k <realm>	requests tickets for the remote host in the specified realm, which may be different from the one the system would use by default.
-K	uses your tickets to authenticate to the remote host, but does not log you in.
-a	attempts automatic login using your tickets. **telnet** will assume the same userid unless you explicitly specify another (using **-l**).
-x	(encrypt) turns on encryption. If left off, (noencrypt) turns off encryption.

10.3 rlogin

```
% rlogin <rhost> [-e<c>] [-8] [-c] [ -a] [-f] [-F]     \
 [-t <termtype>] [-n] [-7] [-d] [-k <realm>] [-x] [-L] \
 [-l <username>]
```

The following are the Kerberos options:

-f	(forward) forwards a copy of your tickets to the remote host. If left off, (noforward) turns off forwarding of tickets to the remote host. (This option overrides any forwarding specified in your machine's configuration files.)
-F	(forwardable) forwards a copy of your tickets to the remote host, and marks them re-forwardable from the remote host. If left off, (noforwardable) makes any forwarded tickets nonforwardable. (This option overrides any forwardability specified in your machine's configuration files.)
-k <realm>	requests tickets for the remote host in the specified realm, which may be different from the one the system would use by default.
-x	(encrypt) turns on encryption. If left off, (noencrypt) turns off encryption.
-X	turns off encryption of the session. (This option overrides any forwarding specified in your machine's configuration files.)

10.4 FTP

```
% ftp [-v] [-d] [-i] [-n] [-g] [-k <realm>] [<host>] [-forward]
```

The following are the Kerberos options:

-k <realm>	requests tickets for the remote host in the specified realm, which may be different from the one the system would use by default.
-forward	requests that your tickets be forwarded to the remote host. The **-forward** argument must be the last argument on the command line.
protect level	(issued at the **ftp>** prompt) sets the protection level. The level **clear** is no protection; **safe** ensures data integrity by verifying the checksum, and **private** encrypts the data. Encryption also ensures data integrity.

10.5 rsh

```
% rsh <host> [-l <username>] [-n] [-d] [-k <realm>] [-f | -F] \
  [-x] <command>
```

The following are the Kerberos options:

-f	(forward) forwards a copy of your tickets to the remote host. If left off, (noforward) turns off forwarding of tickets to the remote host. (This option overrides any forwarding specified in your machine's configuration files.)
-F	(forwardable) forwards a copy of your tickets to the remote host, and marks them re-forwardable from the remote host. If left off, (noforwardable) makes any forwarded tickets nonforwardable. (This option overrides any forwardability specified in your machine's configuration files.)
-k <realm>	requests tickets for the remote host in the specified realm, which may be different from the one the system would use by default.
-x	(encrypt) turns on encryption. If left off, (noencrypt) turns off encryption.
-X	turns off encryption of the session. (This option overrides any forwarding specified in your machine's configuration files.)

10.6 rcp

```
% rcp [-p] [-x] [-k <realm>] [-D <port>] [-n] <file1> <file2>
```

or

```
% rcp [-p] [-x] [-k <realm>] [-r] [-D <port>] [-n] <file> ... \
  <directory>
```

The following are the Kerberos options:

-k <realm>	requests tickets for the remote host in the specified realm, which may be different from the one the system would use by default.
-x	(encrypt) turns on encryption. If left off, (noencrypt) turns off encryption.
-X	turns off encryption of the session. (This option overrides any forwarding specified in your machine's configuration files.)

10.7 Kerberized su (ksu)

The discussion here is adapted from the **ksu** man pages. See them for more information, in particular for option descriptions. The command syntax is:

```
% ksu [<target_user>] [-n <target_principal_name>] [-c       \
  <source_cache_name>] [-C <target_cache_name>] [-k] [-D] [-r \
  <time>] [-pf] [-l <lifetime>] [-zZ] [-q] [-e <command>     \
  [<args ...>]] [-a [<args ...>]]
```

The Kerberos V5 **ksu** program is a Kerberized version of the **su** program that has two missions: one is to securely change the real and effective user ID to that of the target user, the other is to create a new security context.

To fulfill the first mission, **ksu** operates in two phases: authentication and authorization. Resolving the target principal name is the first step in authentication. If the source user is *root* or the target user is the source user, no authentication or authorization takes place. In all other cases, **ksu** looks for an appropriate Kerberos ticket in the source cache. If no ticket is in the cache, then depending on how **ksu** was compiled, the user may be prompted for a Kerberos password.

Make sure you are logged in using an encrypted connection before typing your password!

Upon successful authentication, **ksu** checks whether the target principal is authorized to access the target account. In the target user's home directory, authorization is based on whether appropriate entries exist in either `.k5login` or `.k5users`, or by name-mapping rules if neither file exists.

ksu can be used to create a new security context for the target program. The target program inherits a set of credentials from the source user. By default, this set includes all of the credentials in the source cache plus any additional credentials obtained during authentication. The source user is able to limit the credentials in this set.

Glossary

AFS

A distributed file service (formerly known as the Andrew File System). It is installed on many systems at Fermilab, including FNALU. On strengthened systems, it is integrated with Kerberos.

authentication

The process of verifying the claimed identity of a principal.

authentication service (AS)

The portion of the KDC that issues tickets and secret session keys based on a user password or encryption key. The AS can issue both ticket-granting tickets (TGTs) and service tickets.

authenticator

A record containing information that can be shown to have been recently generated using the session key known only by the client and KDC.

authorization

The process of determining whether a client may use a service, which objects the client is allowed to access, and the type of access allowed for each.

challenge

(used with CryptoCard® as non-reusable authentication; see *CryptoCard*®, also see *response*) Every time you login from an untrusted machine, the KDC generates an eight-digit string called a *challenge*. The CryptoCard® encrypts the challenge with the secret key shared by itself and the KDC in order to generate a non-reusable password, called a *response*.

client

An entity that can obtain a ticket. This entity is usually either a user or a host principal.

credentials

A ticket (usually a TGT) plus the secret session key needed to successfully use that ticket in an authentication exchange. Obtaining credentials from the KDC is tantamount to being authenticated on a strengthened machine.

cross-authentication

This implies trust relations between two strengthened realms (see *trust relations*). Cross-authentication implies the freedom to access systems in either realm if authentication has been established in one of them.

CryptoCard®

An authentication technology that provides tokens via calculator-style one-time-password DES cards. At Fermilab, CryptoCard®s are issued upon demand to users who require access to the PILOT.FNAL.GOV realm from untrusted machines. The cards are synchronized with the KDC prior to issue. (See *challenge* and *response*; see also *portal mode*)

host

A computer that can be accessed over a network.

KDC (Key Distribution Center)

The service which implements Kerberos authentication via the Authentication Service (AS) and the Ticket Granting Service (TGS). The KDC has a copy of every password associated with every principal. Most KDC implementations store the principals in a database, so you may hear the term *Kerberos database* applied to the KDC.

Kerberized application

A software application that requires Kerberos authentication for use.

Kerberized machine

A machine on which the Kerberos product has been installed and which requires strong authentication.

Kerberos

In Greek mythology, the three-headed dog that guards the entrance to the underworld. In the computing world, Kerberos is a network security package that was developed at MIT.

Kerberos client

Any entity that gets a service ticket for a Kerberos service. A client is typically a user, but any principal can be a client.

Kerberos password

A password used to obtain authentication on a Kerberized system.

Kerberos server

This generally refers to the Key Distribution Center (KDC).

key

A string used to encrypt passwords and other data.

keytab file

A keytab file is used by the KDC to store keys.

portal

A secure gateway between the untrusted and strengthened realms requiring non-reusable passwords. At Fermilab we do not have separate machines acting as portals; instead, Kerberized hosts are configured to respond in *portal mode* to requests for access from untrusted machines (see *portal mode*).

portal mode

When a request for access comes from an untrusted machine, Kerberized hosts at Fermilab respond in *portal mode* and thus require entry of a non-reusable password for authentication. (See *portal*.)

principal

A uniquely named client or server instance that participates in a network communication. It is essentially a string that names a specific entity to which a set of credentials may be assigned. For a user, it can be thought of as a realm userid. It has three parts and is of the form `primary/instance@REALM`. Very often the instance portion is null, and the principal is of the form `primary@REALM`. The parts are defined as:

primary

The first part of a Kerberos principal. In the case of a user, it is the userid. In the case of a service, it is the name of the service.

instance

The second part of a Kerberos principal, preceded by a slash (/). It gives information that qualifies the primary. The instance may be null. In the case of a user, the instance is often used to describe the intended use of the corresponding credentials. In the case of a host, the instance is the fully qualified hostname.

realm

The logical network served by a single Kerberos database and a set of Key Distribution Centers. By convention, realm names are generally all uppercase letters, to differentiate the realm from the internet domain.

response

(used with CryptoCard® as non-reusable authentication; see *CryptoCard®*, also see *challenge*) A response is a single-use eight-digit hex password generated by a CryptoCard® as a result of encrypting a challenge.

secret key

A long-term encryption key shared by a principal and the KDC, used to encrypt/decrypt the session key included with the TGT on the initial authentication. In the case of a human user's principal, the secret key is derived from his or her Kerberos password.

server

A particular principal which provides a resource to network clients.

service

Any program or computer you access over a network.

session key

A temporary encryption key used between two principals, with a lifetime limited to the duration of an accompanying ticket.

strengthened application

A software application that requires strong authentication for use.

strengthened machine

A machine on which the Kerberos (or other authentication service) product has been installed and which requires strong authentication.

strengthened realm

The set of all systems (whether on- or off-site) that require strong authentication for access from the network.

strong authentication

A system of verifying workstation user and network server identities on an unprotected network that eliminates the transmission of clear text reusable passwords over the network and their storage on local systems. Typically the authentication is done via a trusted third-party authentication service using conventional cryptography.

ticket

A set of electronic credentials that verifies the identity of a client for a particular service.

TGS (Ticket-Granting Service)

The portion of the KDC that issues tickets to clients for specific services. The user process communicates with the TGS via a ticket-granting ticket (TGT).

TGT (Ticket-Granting Ticket)

A special Kerberos ticket that permits the client to obtain additional Kerberos tickets transparently.

trusted realm

Sites which implement strong authentication, and which meet certain criteria, may be recognized as “trusted” realms. Trusted realms provide levels of security and authentication equivalent to our own.

trust relations

This refers to relations between two strengthened realms. Trust relations imply the freedom to access systems in either realm if authentication has been established in one of them (see *cross-authentication*).

untrusted realm

The set of all systems that do not require strong authentication, but which permit traditional means of access.

XDMCP

(X Display Manager Control Protocol) provides a mechanism for an X terminal to request a session from a remote host. (See `http://www.con.wesleyan.edu/~triemer/network/xdmcp/xdmcp_upd.html`)

Index

Files

A

C

D

E

F

K

L

M

N

O

P

R

S

T

U

V

W

X