Acknowledgments and References

September 2006

Michael Zalokar provided updates regarding new command options in Chapters 12 and 13, and configuring OpenSSH to allow use of CRYPTOCards, and other minor corrections throughout manual. Updated HD location in Chapter 3.

December 19, 2005

Minor fixes to chapters 19 and 21, thanks to Alan Wehman.

February 11, 2005

Thanks to Rick Tesarek for alerting us to problems in Chapter 10, and to Marc Mengel for helping resolve them. Matt Crawford and Mark Kaletka provided information to update Chapter 23, and related information. Thanks also to Cele Bruce, Elizabeth Gallas and Alan Jonckheere for providing updated information for Chapters 4 and 6. Thanks also to Andy Rader, John Urish and Mark Leininger.

September 18, 2001

Thanks to Matt Crawford, Randy Reitz and Dane Skow for technical review as well as updated information. Reorganization of the manual was Dane's idea. I want to thank Cele Bruce, Nuha Elmaghrabi, Margaret Greaney and Joy Hathaway for reviews and/or contributions. Thanks to the many people mentioned below that have continued to provide information.

DRAFT Release August 1, 2001

Contributions from: Lisa Giacchetti, Randy Reitz, Mark Kaletka, Richard Partridge, Dane Skow, Chris Brew, Troy Dawson, Liz Buckley, Frank Nagy, Brian Drendel, Art Kreymer, Margaret Votava, plus of course, Matt Crawford, other usual suspects, and kerberos-users mailing list subscribers.

DRAFT Release June 8, 2001

Again, lots of info from Matt Crawford. Significant contributions also from Liz Buckley, Michael Kriss, Marc Mengel, Tim Zingelman, Frank Nagy, and from other users on the kerberos-users mailing list. Reviewers also include members of the CD-WWW group and Eileen Berman.

Releases 1.0b, February 22, 2001

A number of updates were made to the manual to reflect the changes implemented in the Strong Authentication project over the last year, and in response to issues raised on the *kerberos-pilot@fnal.gov* mailing list. Many thanks to the users who brought up their questions and ideas in this forum, and to those who lent their expertise. Glenn Cooper, Matt Crawford and Mark Kaletka provided many of the answers to the users' questions (and to mine!). Stefano Belforte, Frank Nagy, and Benn Tannenbaum provided information to me on various topics. Yolanda Valadez provided wisdom gained from her experiences escorting users into the strengthened realm. I also wish to acknowledge the authors of the CDF *Quick Start for Kerberos Users* documentation; I borrowed a couple of ideas from it! Matt Crawford very kindly took time to review the manual.

Release P1.0, February 29, 2000

Members of the Fermilab Computer Security Team as of February 2000 (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. Brenna Flaugher initially expressed the need for a document on this subject, thus getting the ball rolling. Thanks Brenna!

Additional information was collected from the following sources:

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

  `http://computing.fnal.gov/security/StrongAuth/Plan/AuthenticationSynopsis.htm`

- Kohl and Neuman (DEC) RFC 1510: The Kerberos Network Authentication Service (V5), September 1993. (If link below doesn't work, try from `http://www.isi.edu/gost/info/kerberos/documentation.html`)

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

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

  `http://www.cmf.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://hoth.stsci.edu/public/krb5/user-guide.html`

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

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

Table of Contents

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
- additional information resources
- the typeface conventions and symbols used throughout the manual
- where to send comments and questions

1. Purpose and Intended Audiences

Fermilab must demonstrate to the DOE that it implements a computer security system that exercises tight control over who uses the lab's computers and network (which are owned by the government). An analysis of the major computer security incidents at Fermilab over the past several years, 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 have a foothold for much easier attacks to gain privileged root access. In order to protect against unauthorized access to Fermilab computers, the Computing Division has implemented the Kerberos Network Authentication Service V5 to provide what is known as *strong authentication* over the network.

The manual is targeted to both administrative and end users of UNIX (all supported operating systems: SunOS, IRIX, Scientific Linux) and Windows and Macintosh systems.

2. Resources

- The Fermilab *kerberos-users@fnal.gov* mailing list archive (compiled since March 2001) is available for anyone to view at `http://listserv.fnal.gov/archives/kerberos-users.html`. Many of the issues raised on the list have been documented in this manual, but some unusual problems are discussed only there.

Subscribe to the *kerberos-users@fnal.gov* mailing list to report problems or errors that occur as you use machines that run strong authentication, and to benefit from the experience of other users. For instructions on subscribing, see `http://listserv.fnal.gov/users.asp#subscribe to list`.

- Other mailing lists include *wrq-users@fnal.gov* and *macusers@fnal.gov*.
- The MIT Kerberos site is: `http://web.mit.edu/kerberos/www/`.
- *The Moron's Guide to Kerberos*, offers some explanations in layman's terms, and is fairly short. No offense intended! It can be found at `http://www.isi.edu/gost/brian/security/kerberos.html`.
- *Kerberos A Network Authentication System* by Brian Tung, Addison-Wesley Networking Basics Series

3. 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.

4. Your Questions and Comments

Questions or comments about the *Strong Authentication at Fermilab* manual or website should be sent to *cdweb@fnal.gov*. 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

Part I Getting Started

Chapter 1: *Introduction to Strong Authentication at Fermilab*

Many of you are aware that Fermilab is in the process of implementing new methods for users to access the computers at the FNAL site. The purpose of this introduction is to summarize the plan and explain what it means for you as Fermilab computer users, system administrators, and software developers, and what you will need to do to prepare for this change.

Chapter 2: *Fermilab Computing Policy Issues*

The full text of the Fermilab Policy on Computing is maintained at `http://www.fnal.gov/cd/main/cpolicy.html`. Section 4 addresses Strong Authentication. In this chapter we summarize the important points.

Chapter 3: *Kerberos Principals and Passwords*

In this chapter we discuss choosing and obtaining a strengthened realm userid (called a *Kerberos principal*) and a Kerberos password.

Chapter 1: Introduction to Strong Authentication at Fermilab

In 2001 Fermilab implemented new methods for users to access the computers at the FNAL site. The purpose of this introduction is to summarize the method and explain what it means for you as Fermilab computer users, system administrators, and software developers.

1.1 Computing on the World Wide Web

The landscape of the computing environment has changed dramatically from the days when the Internet was primarily the domain of the academic research community. The same explosive growth in computing hardware, network connectivity, and capable software that has enabled HEP to tackle the daunting computing challenges of our field have led to a tremendous increase in the pool of participants on the Internet. There has been increasing "urbanization" of the Internet. This means, among other things, that the previous methods of access control are insufficient for today's needs.

1.2 Strong Authentication

The new access methods involve a concept known as "strong authentication".

> Strong authentication is a system of verifying the identities of networked users, clients and servers without transmitting passwords over the network. It does not require that the network be protected. Both parties in a connection must demonstrate knowledge of some "secret" to establish their identities.

The strong authentication service implemented at Fermilab is the Kerberos Network Authentication Service V5. Kerberos (throughout the manual, "Kerberos" refers to Kerberos V5) is a network authentication protocol designed to serve as a trusted third-party authentication service. It 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. In addition to establishing identity (authentication), it supports encrypted network connections, thereby providing confidentiality.

The "heart" of a Kerberos installation is the Key Distribution Center (KDC). All the computers associated with a KDC make up what's called a *strengthened realm*. At Fermilab, there are two strengthened realms: there is one for UNIX machines called FNAL.GOV, and for Windows 2000 systems there is FERMI.WIN.FNAL.GOV. The KDC's main functions include:

- Maintaining a database of users and services within its realm.
- Authenticating users by way of exchanging tickets between clients and services in the strengthened realm.

1.3 Why has Fermilab implemented strong authentication?

There have been several computer security breaches at Fermilab and other DOE facilities. Our funding agencies has required Fermilab to demonstrate that it has implemented a computer security system that exercises tight control over who uses the lab's computers and network.

In response, Fermilab has issued revisions to its Computing Policy that detail responsibilities and requirements for accessing computing resources at Fermilab. The Computing Policy is provided online at `http://www.fnal.gov/cd/main/cpolicy.pdf`. We provide the relevant information from the policy in more readable language in Chapter 2: *Fermilab Computing Policy Issues*.

This manual seeks to explain the implementation of Strong Authentication at Fermilab. Where there appear to be conflicts, the Policy prevails.

1.4 What do you need to know and do ?

Virtually all machines at Fermilab require Kerberos authentication for network access. You will need to be able to satisfy that authentication requirement in order to gain access. How you satisfy it depends on the role you play in your use of computers (e.g., user or administrator), on the OS you use, and on whether you connect from your machine over the network to other machines or not.

If you bring a machine from your university to FNAL, the machine must be Kerberized if you wish to participate in the strengthened realm. We highly recommend that you participate, as it makes your access to other FNAL machines much simpler.

For those of you at a university or other off-site location, you may include your machine in the FNAL Kerberos realm as well. Off-site machines have different requirements for doing so.

1.4.1 General User

As a general UNIX or Windows user, you should expect that the maintainer of your computer has provided the basic tools and installation necessary to configure the machine as a member of Fermilab's strengthened realm.

Your responsibilites are listed below:

General User Responsibilities	Where to find Information
(Recommended) Understand the broad outlines of Fermilab's Strong Authentication policy.	Read the entire Part I: *Getting Started*, especially Chapter 2: *Fermilab Computing Policy Issues*.
Obtain a Kerberos principal (an identifier for the realm, akin to a login name) and a Kerberos password.	See section 3.1.2 *Requesting a Principal*, or go straight to the online form at `http://computing.fnal.gov/cd/forms/actreq_form.html`.
Obtain a CRYPTOCard if necessary, learn how to use it, and care for it properly.	You can find out what a CRYPTOCard is used for and determine whether you need one by reading section 4.4 *Connecting from a NonKerberized Machine: Portal Mode*. Care and use of CRYPTOCards are described in Chapter 5: *Using your CRYPTOCard*.
Change your initial Kerberos password to an acceptable one of your choosing within 30 days of receipt.	See sections 3.2.2 *Choosing a Kerberos Password*, and 3.3 *Changing your Kerberos Password*.
Learn how to obtain your login credentials.	How to do this depends on whether you're logging in to a Kerberized machine at the console or over the network, on what software you're using, and on other factors.)
Learn how to use your login credentials without exposing them to theft.	See section(s) of Chapter 4: *Accessing Kerberized Machines (Fermilab-Supported Methods)* appropriate to your operating system(s), and Chapter 11: *Encrypted vs. Unencrypted Connections*.

General User Responsibilities	Where to find Information

And last but not least: Treat your Kerberos password as a sacred object!!

- Your Kerberos password must be known only to you.
- Make sure that you do not write it down anywhere that someone could find it.
- Do not put it in a file (encrypted or not).
- As a usual practice, type it only at the console of a system on which you authenticate.
- Only on very rare occasions, when you have no other choice, may you pass it over a network connection. The connection MUST BE ENCRYPTED. Verify that ALL connections in the chain are encrypted.
- Choose a character string different from your Kerberos password for all other passwords and other objects. (The one exception: your passwords for the FNAL.GOV and FERMI.WIN.FNAL.GOV realms may be the same.)
- If you mistakenly type your Kerberos password over an unencrypted channel, please change your password immediately!

Windows Desktop Users

Windows desktops and resources at Fermilab are for the most part in the Windows 2000 domain. The W2K domain structure supports Kerberos authentication. For information on this, see *Windows 2000 at Fermilab* at `http://computing.fnal.gov/cd/windows/w2kdoc/`.

1.4.2 System Administrator

As a system administrator (including those who administer their own machines), you need to do and understand everything the general user does, and in addition, you must understand how to setup the Kerberos tools and how to properly configure the machine for the strengthened realm. For users of the Computing Division's UPS/UPD environment, much of this has been automated. Also a number of system vendors are providing Kerberos as a standard option within their OS installation. You may use whichever tools you prefer as long as the result complies with Fermilab policy. The obligation is on you, the adminstrator, to understand your own configuration well enough to ensure compliance. The chapters in the Administrator part of the manual provide detailed instructions on many common circumstances at the lab.

1.4.3 Developer

You as an application or system developer need to understand the principles of strong authentication, and the Fermilab Computing Policy in detail. It is your responsibility to design systems and software that enhance the security of Fermilab's computing systems and to improve our ability to withstand the onslaught of attackers who would misuse our resources.

1.5 What advantages does Kerberos provide?

One big advantage is that you have *one* id, known as your Kerberos principal, and *one* password that can be used anywhere at the lab (actually two principals: name@FNAL.GOV and name@FERMI.WIN.FNAL.GOV). This simplifies life considerably. You still need authorization to use machines to which you log in (an account or an entry in an access control list), but there are no passwords that need to be locally maintained anymore.

Once you are authenticated on a system, you can move from one strengthened machine to another without having to type your password again.

And, most importantly, the computers *are more secure* from abuse by outsiders.

For more information, see Appendix A: *Implementation Details of Strong Authentication at Fermilab* and Appendix B: *About the Kerberos Network Authentication Service V5*.

1.6 What advantages does Kerberos have over other possible solutions?

In Kerberos V5, the password-checking (authentication) happens in one place, and the end systems need not store any information which can be used to try to guess a password. Further, Kerberos allows a single point of disabling an unauthorized or wayward user on all systems in the strengthened realm. This feature satisfies one of Fermilab's obligations to the DOE.

In ssh, as in standard UNIX, each end system has to store information sufficient to check the password, which is therefore also sufficient to try to guess the password. If the RSA authentication method is used, the RSA keys can give access to various accounts, and there's no way to know with certainty

who possesses which keys. In the event of a compromise of a private key, there's no mechanism for locating every host on which the corresponding public key appears.

1.7 How does Kerberos work?

Kerberos authentication operates by the exchange of tickets that allow access to all services by the user in the strengthened realm. This first sample scenario shows how it works when a user connects over the network from a Kerberized UNIX desktop to a remote Kerberized UNIX host:

1) User first logs in directly (not over the network) to a Kerberized desktop computer that is in the FNAL.GOV realm.

2) User requests authentication for the FNAL.GOV realm, and must enter his or her Kerberos password.

3) Behind the scenes: Kerberos software installed on the desktop is used to derive a key from the password. This key is used to encrypt the exchanges between the local machine and the (remote) KDC in order to achieve authentication. The password is not transmitted between the two machines.

4) When authentication is complete, user gets a "ticket" (also called a "credential") from the KDC.

5) The user can now connect over the network to other Kerberized hosts without entering his Kerberos password again. Without entering ANY password, in fact! Kerberos negotiates the authentication for each login using the ticket, all behind the scenes.

This second scenario shows how it works when the user's UNIX desktop is not Kerberized; this is where CRYPTOCards come in (see Chapter 5).

If the local desktop computer does not run Kerberos software and is not part of the FNAL.GOV realm, then the user can't authenticate locally on this computer. The user can work on the desktop with no problem, but in order to connect to remote Kerberized UNIX hosts, he or she must authenticate to FNAL.GOV first. Here's how it works:

1) The user logs into desktop computer normally, and enters his or her standard password.

2) From the desktop machine, the user opens a connection to a remote Kerberized host machine.

3) Kerberized machines in the FNAL.GOV realm are configured to require entry of a single-use password whenever they receive a login request coming from an unKerberized computer over the network. (The password gets transmitted over the network, and it could get intercepted. That's why it must be single-use only.)

How do you get a single-use password that Kerberos will recognize and honor? The FNAL.GOV realm at Fermilab is setup to use CRYPTOCards to provide these single-use passwords.

1.8 How do you obtain a Kerberos Principal?

To request a principal, use the online form at http://computing.fnal.gov/cd/forms/acctreq_form.html. But first, read more about principals in Chapter 3: *Kerberos Principals and Passwords*.

After you get a principal, you'll need to change your initial Kerberos password that comes with it. If your experiment or group doesn't have a Kerberized machine set up yet, or if you don't have an encrypted connection to a Kerberized machine, log into any of the FNALU machines to change your password and to get acquainted with Kerberos.

Chapter 2: Fermilab Computing Policy Issues

The full text of the Fermilab Policy on Computing is maintained at `http://computing.fnal.gov/cd/policy/cpolicy.pdf`. In this chapter we summarize the important points as regards Strong Authentication.

2.1 The Strong Authentication Policy in a Nutshell

Computers at Fermilab must be configured such that they require Kerberos V5 authentication for login over the network. Our working definition of *computer*, as regards strong authentication, is: "any machine to which you can log in, and on which you can run arbitrary code".

Kerberos authentication is currently **not** required for:

- uses which involve only reading public information (e.g., via the web)
- anonymous FTP
- email
- entering information into a web or database form, in most cases

All other network accesses to computers on the Fermilab site must be preceded by Kerberos V5 authentication if the access is comparable to shell or FTP service.

Compliance can be achieved in different ways:

- Run Kerberos authentication locally
- Remain unKerberized, but remove incoming network services
- (not for desktops) Remain unKerberized, but require users to gain access through a computer that either:
  - requires Kerberos authentication, or
  - is isolated from the general network and physically accessible only to individuals carrying a valid Fermilab ID card.

Furthermore, an on-site system may not be configured to accept a reusable login password over the network.

Telnet, ssh, and other connection program daemons must not prompt for or accept a Kerberos password. To log in over the network:

- Authenticate on local desktop machine prior to remote login (and forward tickets if possible)
- From nonKerberized node, authenticate using your CRYPTOCard

Off-site computers participating in Fermilab's strengthened realm must enforce comparably secure access mechanisms, but they are not required to use Kerberos V5.

2.2 Authentication Guidelines for On-site vs. Off-site Machines

First let us distinguish between an authentication method and a transport mechanism as they pertain to on-site versus off-site machines:

- *Authentication methods* serve to identify the user; examples include: Kerberos credentials, CRYPTOCards, passwords, RSA keys, and IP addresses + "privileged ports". For on-site machines, only Kerberos credentials and CRYPTOCards are allowed as authentication methods. For off-site machines, any secure method is acceptable.
- Ssh, telnet, FTP, and so on are the network connection programs, or the *transports*, and none is forbidden per se. The restriction is imposed on the authentication methods, and the transport is restricted only in that it must support an acceptable authentication method.

The following table summarizes Fermilab's policy regarding how strong authentication may be achieved on UNIX machines in the Fermilab strengthened realm depending on whether the machine is on- or off-site:

Authentication Method	Allowed for ON-site System?	Allowed for OFF-site System?
Kerberos V5 strong authentication via kerberos product (Fermi kerberos or from other source)	yes	yes
Kerberos-based authentication via software other than Kerberos (e.g., Kerberos-based ssh)	yes	yes
CRYPTOCard challenge/response authentication	yes	yes
Clear-text reusable passwords **entered at system console**	yes	yes
Other non-reusable and/or non-clear-text password authentication over the network	no	yes

Authentication Method	Allowed for ON-site System?	Allowed for OFF-site System?
Non-Kerberos strong authentication (e.g., RSA or equivalent authentication) followed by obtaining Kerberos credentials via kinit over encrypted connection	no	no
Standard UNIX security (e.g., rhosts-based authentication)	no	no
Cleartext passwords (Kerberos or otherwise) transmitted over network	no	no

2.3 Transient Machines

Laptop machines brought in by visitors for short periods of time (e.g., a week) do not need to be registered or Kerberized. Visitors may use their host's accounts (with host's permission) at the host's responsibility, although sharing Kerberos passwords is not allowed. Local accounts that allow access only at the console will be permitted for visitors (no NIS accounts). Facilities created primarily for visitors may be granted exemptions from the requirement for Kerberos-validated users.

2.4 Obtaining an Exemption from the Policy

Exemptions from the strong authentication policy are granted on a case-by-case basis. Exemptions will be considered only for cases which involve a large effort for compliance *and* a small risk for noncompliance. If this applies to your situation, see your experiment's or your division's GCSC (General Computer Security Coordinator)[1] to request an exemption; he or she will forward your request to the Fermilab Computer Security Coordinator (FCSC). The duration of any exemption granted is determined on a case-by-case basis.

1. The GCSCs are listed on the CD Security web page `http://computing.fnal.gov/security/`.

2.5 Compliance with Policy

First, a few notes regarding good user practices:

- Fermilab's policy seeks to limit the transmission of users' Kerberos passwords over the network, even over encrypted connections. We therefore urge you to install software on your machine that allows you to authenticate to Kerberos locally, and to forward Kerberos tickets automatically to remote hosts. You are allowed to type your Kerberos password over an **encrypted** link on an emergency basis with the **`kinit`** or **`kpasswd`** commands (e.g., when initially changing your password), however as a regular practice, please authenticate locally and forward your credentials.
- Do not disclose your Kerberos password to anybody, and do not ever type it over an unencrypted connection. Try to minimize the number of times per day or per week that you need to type it for any reason.
- In short, following the usage recommendations and installation instructions provided throughout this manual will keep you in compliance with Fermilab's Computing Policy as regards Strong Authentication.

Regarding penalties for noncompliance, we quote from section 1 of the Fermilab Policy on Computing (at `http://www.fnal.gov/cd/main/cpolicy.html`):

"Hosts found to be noncompliant may be barred from obtaining Kerberos tickets from our realm. If the noncompliance is deliberate or extremely careless it may be deemed to constitute blatant disregard for computer security."

Chapter 3: Kerberos Principals and Passwords

In this chapter we discuss choosing and obtaining a strengthened realm userid (called a *Kerberos principal*) and a Kerberos password.

3.1 Your Kerberos Principal

As a user, you need to obtain a Kerberos Principal[1] (actually one for each realm, FNAL.GOV and FERMI.WIN.FNAL.GOV), in order to access machines and resources at Fermilab. A principal is essentially a username for the strengthened realm. Your principals will have the same username, and be of the form principal_name@REALM (e.g., joe@FNAL.GOV and joe@FERMI.WIN.FNAL.GOV). You must have a valid Fermilab ID.

In addition to a principal, you must have an account on each machine that you plan to use in the realm. There are significant conveniences if your principal and your account name are the same, as we discuss in section 3.1.1 *Choosing a Principal Name*.

The system administrator of a strengthened machine may require that authorized users obtain a *<username>/root* instance of their Kerberos principal in order to access sensitive accounts on the system. The root instance has tighter restrictions placed on it (see section 9.2 *Ticket Management*). If your sytem administrator tells you it's required, use the form *Request Additional Kerberos Items* at `http://computing.fnal.gov/cd/forms/extra_kerb_req_form.html`.

1. Note for sysadmins: if you have an account and a standard UNIX password (in the `passwd` file or NIS map) on a Kerberized machine, but no principal or Kerberos password, you can still log in and use non-Kerberized services. You can do this only at the console. (From any other terminal, the Kerberized system responds in portal mode, described in section 4.4 *Connecting from a NonKerberized Machine: Portal Mode*, and you have no option to enter your UNIX password.)

3.1.1 Choosing a Principal Name

The Kerberos Strong Authentication system is includes virtually all computer systems across the site. Your Kerberos principal will be used for authentication sitewide. It is to your benefit to have one login id (account name) common to all systems that you use, and for that login id to match your Kerberos principal. The Computing Division is strongly encouraging this practice for ease of use, and in fact is enforcing it for new users. Keep in mind that the principal name you choose will be your permanent ID at Fermilab. Here are guidelines for choosing the name you'll use for your Kerberos principal:

New principals must be chosen to be eight (8) or fewer characters. Please use only lowercase letters and digits 0 through 9. **Do not use any uppercase letters or any special characters**.

In Appendix C: *More about Choosing a Principal Name*, we present information for users who have pre-existing account names and/or an email address at Fermilab, and for whom the above guidelines are not straightforward to follow.

3.1.2 Requesting a Principal

Use the online *Request Form for Computing Username and Primary Accounts* at
`http://computing.fnal.gov/cd/forms/acctreq_form.html`.

3.2 About Kerberos Passwords

Once your request for a principal has been approved, you must stop by Wilson Hall, ground floor, north (the CD Help Desk) to receive your initial Kerberos password. An exception is granted for off-site visitors: you can get it over the telephone (630-840-2345); you will be asked a question to verify your identity.

You are required to change the initial password within 30 days of receipt, and once a year (actually every 400 days) thereafter.

Even if you use a CRYPTOCard exclusively, you need to change your Kerberos password as stated above in order to continue accessing machines in the FNAL.GOV realm! If your password expires, you can still change it as long as you remember what it was, but you cannot use CRYPTOCard access while it remains expired.

3.2.1 Important! Please Read!

Please treat your Kerberos password as an inviolable object. Never give your password to anybody for any reason. Doing so constitues a policy violation. 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 9.3 *Account Access by Multiple Users*.

Typing in your Kerberos password should ideally be done infrequently (i.e., no more than once each day). Do not type it in carelessly. Please authenticate locally and forward your credentials to remote systems.

Windows 2000 domain-only users: type your password **only** at the Windows login prompt.

3.2.2 Choosing a Kerberos Password

In contrast to the principal (which ideally should match your login name on each machine and your email address), your Kerberos password must be unique. That is, in order to avoid exposing your Kerberos password, it must be different from the passwords you use for any other purpose (with the single exception that you may use the same one for both strengthened realms at Fermilab).

The Fermilab Computer Security Team has imposed some restrictions on passwords in accordance with DOE guidelines. Currently, a password for the 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 for /root principals must contain a minimum of 11 characters including at least three of the five classes. Passwords the system considers "bad" will be rejected. (Passwords are checked against the "cracklib" dictionary, which will often surprise you by its thoroughness!)

Choose something that's hard to guess but that you can remember, and please make an effort to remember it!!

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 "GykoR66." for "Get your kicks on Route 66."
- an easy-to-pronounce nonsense word, like "slaRooBey" or "krangits"
- a misspelled phrase, like "2HotPeetzas!" or "ItzAGurl!!!"

---

1. These ideas were lifted from MIT's Kerberos V5 User's Guide (C) 1996, at (new link) `http://hoth.stsci.edu/public/krb5/user-guide.html`.

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.

3.3 Changing your Kerberos Password

A few notes before moving to the platform-specific instructions:

- If you forget your initial password before you get around to changing it, open a helpdesk ticket requesting a password reset. Go to `http://helpdesk.fnal.gov/`.
- Change your password on a machine that is sitting in front of you and that has Kerberos or Reflection or other Kerberos-aware program installed. Do not send your password over a network connection to a remote host!
- The Computing Division has set up a terminal at which people can change their Kerberos passwords. It is in the CD Helpdesk/Email Center in Wilson Hall, ground floor, north end. Signage is mounted on the wall above the screen with instructions.
- If you don't have an appropriate machine on which to change your password, find someone who does, and borrow his or her command prompt. (Yes, you can change it from someone else's account; just give your principal name as an argument. For Reflection, add your principal into your colleague's configuration.) Or you can install a simpler, client-only version of Kerberos on your local machine; see section 6.2 *In a Pinch: Download Client-Only Version of Kerberos*.
- If your only option is to change it on a remote host via a network connection, then before changing your password, **verify that you are using an encrypted connection!** How do you know if your connection is encrypted? See Chapter 11: *Encrypted vs. Unencrypted Connections* for some help.

3.3.1 UNIX/Linux/Cygwin

To change your password, run the **`kpasswd`** command locally on your desktop or laptop.

The **kinit** program warns you if your password is within 30 days of its expiration date, and as of **kerberos** v1_2, the **kerberos** login program includes this warning as well.

On strengthened UNIX systems running AFS, there are two **`kpasswd`** commands, one for AFS (`/usr/afsws/bin/kpasswd`) and one for Kerberos (`/usr/krb5/bin/kpasswd`). 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 along with Kerberos tickets. If you are unsure which **kpasswd** is being invoked, force the system to use the Kerberos version by running **setup kerberos** first.

```
% setup kerberos
```

Then run **kpasswd**. If borrowing someone else's account or if your principal does not match your login id, include your principal name as an argument.

```
% kpasswd [<principal_name>]
```

```
kpasswd: Changing password for aheavey@FNAL.GOV.
Old password:                 <--- type your initial password here.
kpasswd: aheavey@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.
```

If the password has expired, you'll need to get access to a machine running **kpasswd** some other way (e.g., find a friend or use a local account) to change it.

3.3.2 Windows (with WRQ® Reflection software installed)

Here we assume you are running the **WRQ® Reflection** software for **Windows** as described in Chapter 19: *Installing and Configuring WRQ® Reflection on a Windows System*.

To change your Kerberos password via the **Reflection** application, navigate to **START > PROGRAMS > REFLECTION > UTILITIES > KERBEROS MANAGER** to open the **Reflection Kerberos Manager** application. From the **TOOLS** menu select **CHANGE PASSWORD...** and change it. The password you enter does not go across the network; this is safe. Wait a few seconds for a message to appear indicating that it's been successfully changed.

If it doesn't work, try updating the Windows services file by executing `\\Pckits\WRQ\services.bat`. For Win95 or 98, you must copy it manually from `\\Pckits\WRQ\` (target directory may vary). This file is typically updated during installation of Reflection, so shouldn't normally be required at this stage.

3.3.3 Windows (with Exceed 7.0 and MIT Kerberos)

Here we assume you are running **Exceed 7** with the **MIT Kerberos** software for Windows as described in Chapter 21: *Installing MIT Kerberos on Windows, for use with Exceed 7 and FileZilla*.

Leash32 found in the **MIT Kerberos** 2.5 or later (for Windows) can be used to change the password for an MIT Kerberos principal. To change your password:

- Navigate to **START > PROGRAMS > KERBEROS UTILITIES > LEASH32**.
- On the **Leash32** window, go to the **ACTIONS** menu and select **CHANGE PASSWORD**, and follow the instructions.

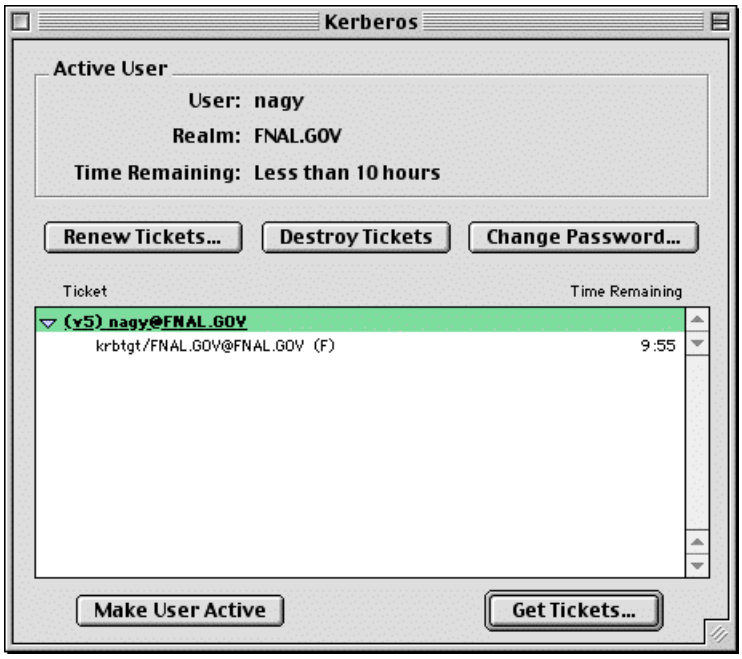
In earlier versions, the **CHANGE PASSWORD** utility in **Leash32** does not work, and **`kpasswd`** in the Command Prompt works for the AFS password. For these earlier versions, then, changing your password under this configuration requires typing your password over a network connection. Please upgrade, or try to find a machine on which you can change your password locally, instead.

3.3.4 Macintosh

Here we assume you are running the **MIT Kerberos** software for Macintosh as described in Chapter 23: *Installing and Configuring MIT Kerberos on a Macintosh System*. To change your Kerberos password on OS X, either use **`kpasswd`** at the command line as in Unix, or the Change Password button on the GUI. For OS 9 and earlier:

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

KERBEROS CONTROL STRIP module).



2) Select a username and realm and click **GET TICKETS** for which you will have to provide your current (or initial) Kerberos password.

3) Click on the ticket to highlight it, then click **CHANGE PASSWORD** and enter the old and new passwords on the pop-up screen which appears.

Part II User's Guide

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

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

Chapter 5: *Using your CRYPTOCard*

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

Chapter 6: *Logging In from Off-Site*

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

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

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

Chapter 8: *Troubleshooting your Authentication Problems*

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

Chapter 9: *Using Kerberos*

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

Chapter 10: *Miscellaneous Topics for the User*

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

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

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

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

4.1 Logging In at the Console of a Kerberized UNIX Machine

Many people connect from home using a high speed internet access through a local area router, and in many cases their ISP subjects them to NAT (Network Address Translation). We discuss NAT in section 6.5 *Network Address Translation*. In these cases you will need addressless Kerberos tickets. When obtaining your credentials with **`kinit`**, include the qualifier **`-n`**. See section 12.1 *kinit* for more information.

4.1.1 Using Standard UNIX Login Program

If your desktop machine is running the standard login program, log in at the console normally, entering your standard UNIX password (note that if your machine runs AFS, your UNIX and AFS passwords may be the same). The standard login program does not accept your **kerberos** password. You need to run **`kinit`** after logging in to obtain your credentials. The credentials should then get forwarded to other strengthened machines normally. The **kerberos** login program is not installed by default with the **kerberos** product. If you need an addressless ticket, use the `-A` option (see section 12.1 *kinit*).

4.1.2 Using Kerberos Login Program

If your desktop machine is configured to use the **kerberos** login program[1], you can authenticate to Kerberos at login by entering your Kerberos password at the password prompt. You do not need to run `kinit` after login. (You can still login using your UNIX password, then run `kinit` to get Kerberos tickets, if you wish.) An advantage to using the **kerberos** login program is that it checks the `/etc/krb5.conf` file in which you or your system administrator can set defaults for Kerberized applications.

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

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

You can use ssh to log into machines running mixed mode Kerberos, as described in section 4.1.4 *Machines Running Mixed Mode Kerberos*.

4.1.4 Machines Running Mixed Mode Kerberos

Machines that are Kerberized in mixed-mode allow logins via ssh for users who don't yet have a Kerberos principal. This is in addition to allowing login via Kerberized services or CRYPTOCard. Mixed-mode machines are not allowed on-site.

4.2 Connecting from One Kerberized Machine to Another

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

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

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

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

A few notes:

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

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

4.3 Connecting via Kerberized SSH

Any machines that are sited at FNAL and that wish to use ssh are required to use Kerberized ssh. There are Scientific Linux distribution RPMs for Kerberized ssh, as well as the UPS version (available from `ftp://ftp.fnal.gov:8021/KITS/`). Any version 3.6 or later built against kerberos and the GSSAPI libraries should suffice. Non-Kerberized ssh is not permitted on these machines.

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

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

4.4 Connecting from a NonKerberized Machine: Portal Mode

4.4.1 About Portal Mode

If your local desktop computer does not run Kerberos software and is not configured for the FNAL.GOV realm, then you can't authenticate to FNAL.GOV locally on this computer. You can work on the desktop with no problem, but in order to connect over the network to Kerberized UNIX hosts, you must authenticate to FNAL.GOV first.

Kerberized machines in the FNAL.GOV realm are configured to require entry of a single-use password whenever they receive a login request coming from an unKerberized computer over the network. (The password gets transmitted over the network, and it could get intercepted. That's why it must be valid for only one login.) The target computer is said to respond in *portal mode* in this case. It is acting as a secure gateway into the strengthened realm.

How do you get a single-use password that Kerberos will recognize and honor? The FNAL.GOV realm at Fermilab is setup to use CRYPTOCards to provide these single-use passwords.

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

4.4.2 About CRYPTOCard

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

To read more about what a CRYPTOCard is and how it works, see Chapter 5: *Using your CRYPTOCard*. To request one, fill out the online form *Request Form for Computing Username and Primary Accounts* at `http://computing.fnal.gov/cd/forms/acctreq_form.html`. When you get your CRYPTOCard, go back to Chapter 5: *Using your CRYPTOCard* for information on how to use it and take care of it.

Two notes:

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

4.4.3 Programs for Initiating CRYPTOCard Login

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

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

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

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

- The Kerberos login program supports **ssh** only when no command argument is given, i.e., when it is effectively equivalent to **slogin**. (Fundamentally, **slogin** is the only **ssh** program supported.)
- The Kerberized sshd on the remote host prompts for an **ssh** password before displaying the CRYPTOCard challenge. Just press Return for the **ssh** password, don't enter any characters.

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

```
CryptoCard RB-1
Press ENTER and compare this challenge to the one on your
display
Challenge is [12345678], Enter the displayed response:
```

Use your CRYPTOCard to provide this password, as described in section 5.5 *Log in Using CRYPTOCard (the First Time)*, or section 5.6 *Log in Using CRYPTOCard (Subsequently)*.

Notes:

- Never type your Kerberos password over a CRYPTOCard **telnet** session! The connection is not encrypted.
- You may type your password infrequently over an encrypted CRYPTOCard **ssh/slogin** session.
- **rsh**, **rlogin**, **rcp** and **scp** are not available for portal mode.

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

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

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

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

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

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

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

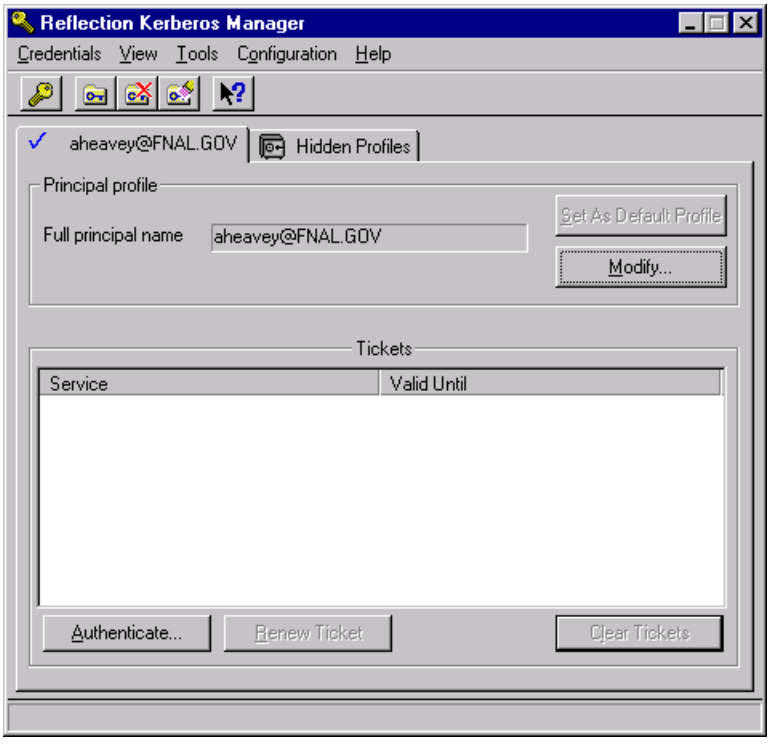
4.6 Logging In Through WRQ® Reflection Software from Windows

4.6.1 Authenticate Locally via the Kerberos Manager

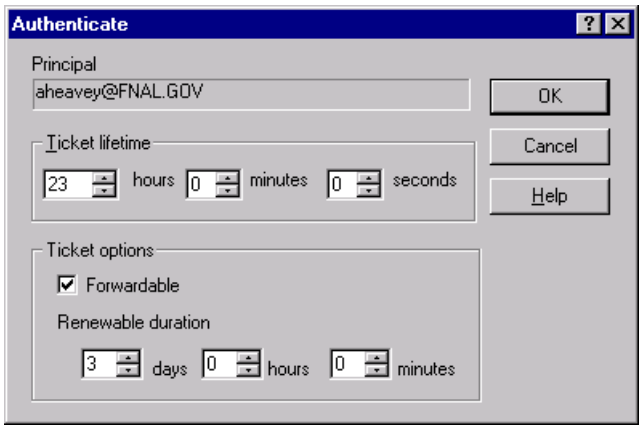
The **Reflection Kerberos Manager** program authenticates you to Kerberos and supports ticket forwarding. This means it obtains an initial Kerberos ticket for the principal on the tab chosen, and you, as that principal, can freely connect to Kerberized machines without needing to type your Kerberos password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

If you need an addressless ticket (Not sure? See section 6.5 *Network Address Translation*.), make sure your software is configured as described in section 19.3 *Configuration for Addressless Tickets*.

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

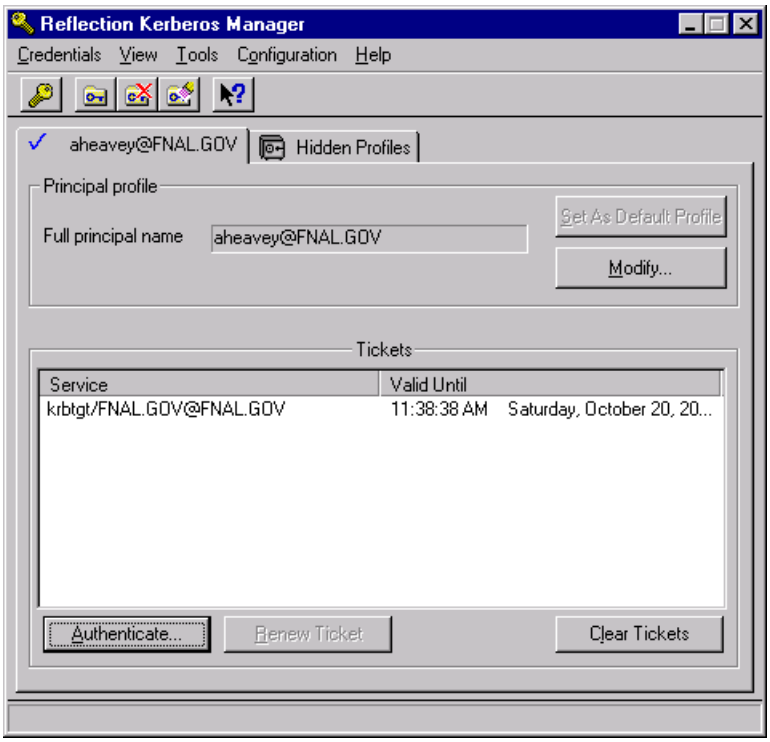


Choose your principal that corresponds to the default realm of the target machine. Click **AUTHENTICATE**.



- Verify or change **TICKET LIFETIME** (if you give a value greater than the KDC limit of 26 hours, the renewable lifetime will be set to 26 hours)
- Check **FORWARDABLE** in order to forward your ticket to target host (besides forwarding your Kerberos ticket, it's necessary in order for an AFS token to be automatically generated when you connect to a system running AFS)
- To set your ticket as renewable, enter a non-zero time for **RENEWABLE DURATION** (if you give a value greater than the KDC limit of seven days, the renewable lifetime will be set to seven days). The AFS token you get will have a lifetime equal to the Kerberos ticket's renewable duration.

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



If you want to check that the ticket is addressless, right-click on the ticket (krbtgt/FNAL.GOV@FNAL.GOV), choose Properties, and verify that the Address box is blank.

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

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

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

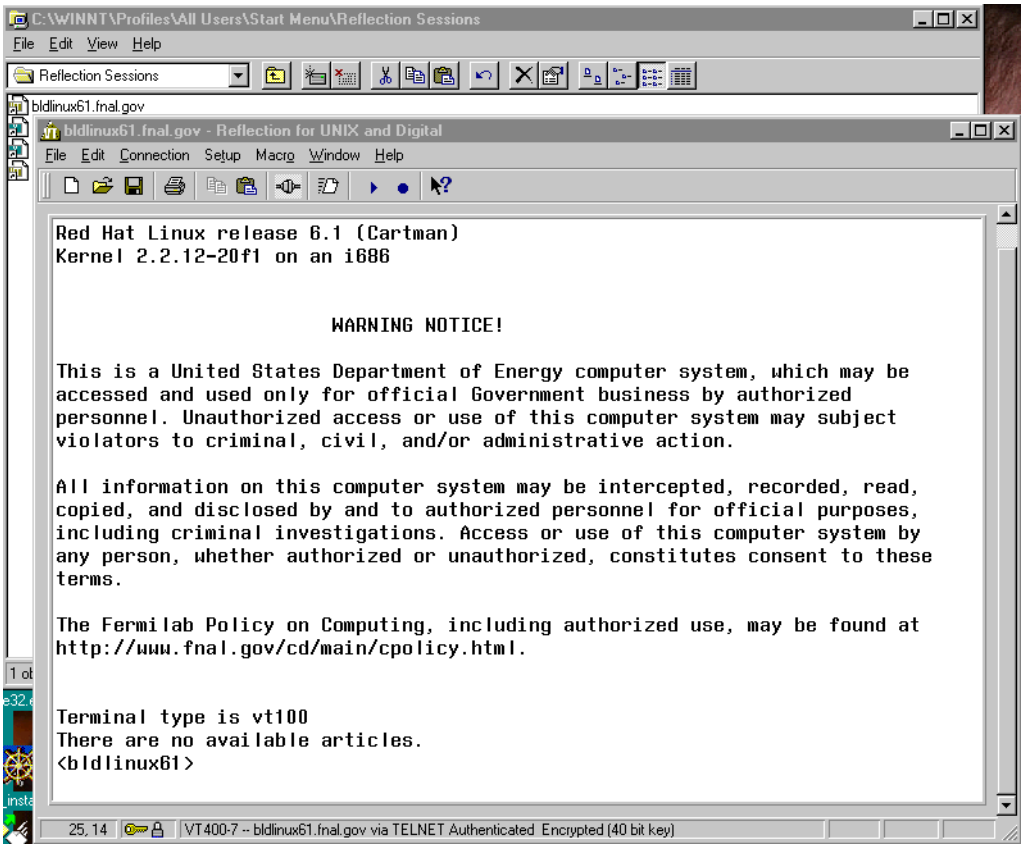
4.6.2 Run a telnet Session to Kerberized Host

To use the **WRQ® Reflection telnet** client to access machines in the strengthened realm, you first need to set (and save) a separate **telnet** configuration for each host with ticket forwarding set, as outlined in section 19.8 *Configuring WRQ® Reflection telnet Connections*.

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

Start the **Reflection Kerberos Manager** first to authenticate, as explained in section 4.6.1 *Authenticate Locally via the Kerberos Manager*. The easiest way to start a session is to make a short cut for your telnet configuration file, and just double-click on it. Otherwise, to start your session:

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



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

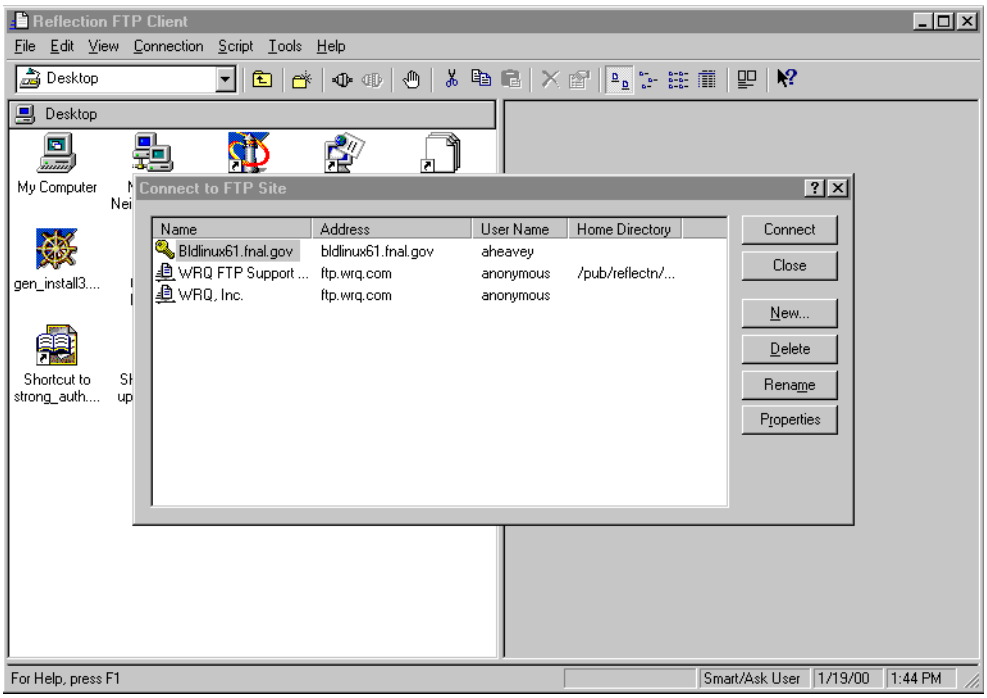
If you authenticate with the **Kerberos Manager** and get a nonforwardable ticket, and then start a telnet session with forwarding enabled, you'll get another password prompt from **WRQ®** so that it can obtain a forwardable ticket for you.

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

4.6.3 Run an FTP Session to Kerberized Host

Configuration of **FTP** sessions is covered in section 19.10 *Configuring WRQ® Reflection FTP Connections*. Make sure that the default realm for **REFLECTION** is set to the default realm of the target host (see number [3] in section 19.5 *Configuring WRQ® Reflection Kerberos Manager v12.0.*).

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



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

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

1) Install and use the Windows AFS client, as described in sections 4.7 *Windows AFS Client for File Transfers to AFS Space* and 4.7 *Windows AFS Client for File Transfers to AFS Space*.

2) Configure the WRQ® FTP client with standard (nonKerberos) security and use a CRYPTOCard (this has also been tested with NT and Windows 2000 command line FTP, and FTP client in **FrontPage2000**).
   · Select **VIEW > COMMAND WINDOW** to see the CRYPTOCard challenge.
   · Connect to host, generate a response on your CRYPTOCard, and enter it at the password prompt.

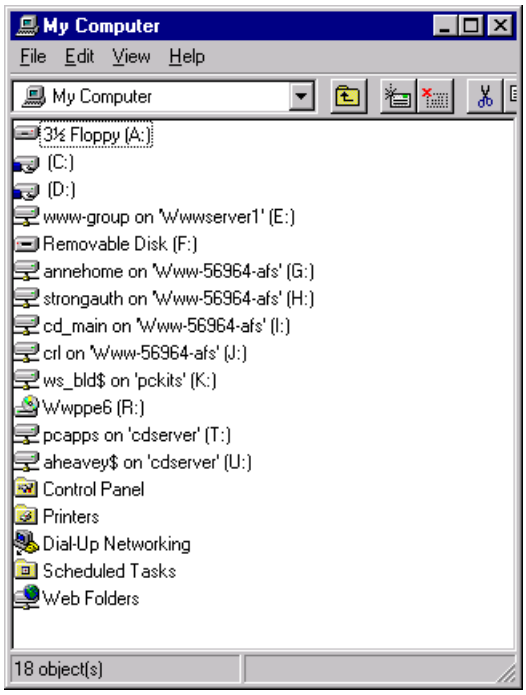
4.7 Windows AFS Client for File Transfers to AFS Space

Due to the inability of the Kerberized FTP clients for Windows, including WRQ®'s, to forward Kerberos tickets (and thus generate AFS tokens on the remote host), we recommend that you bypass FTP entirely and install the Windows AFS client for file transfers to and from AFS space. Installation and configuration is described in the document *Installing IBM AFS Client 3.6 for Windows NT/2000/XP* at `http://www-oss.fnal.gov/csi/openafs_windows/`.

4.7.1 How does AFS Appear on your Desktop?

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

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



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

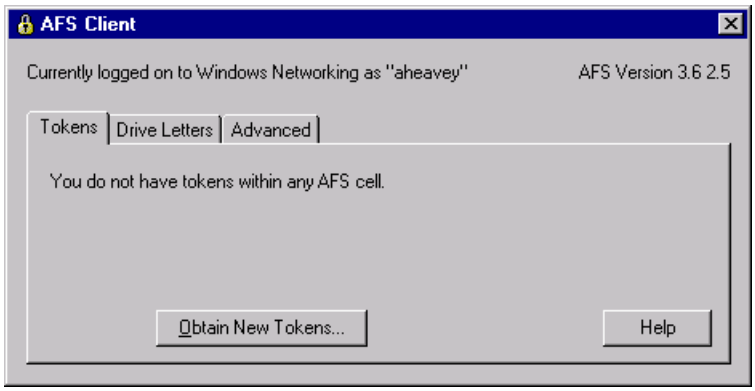
The AFS icon in your task bar is a lock symbol. It displays a red X ( ) before you authenticate to AFS, and the X goes away ( ) after you authenticate to AFS and obtain a valid token.

4.7.2 Authenticate to AFS

1) Make sure the AFS Client Service is running.

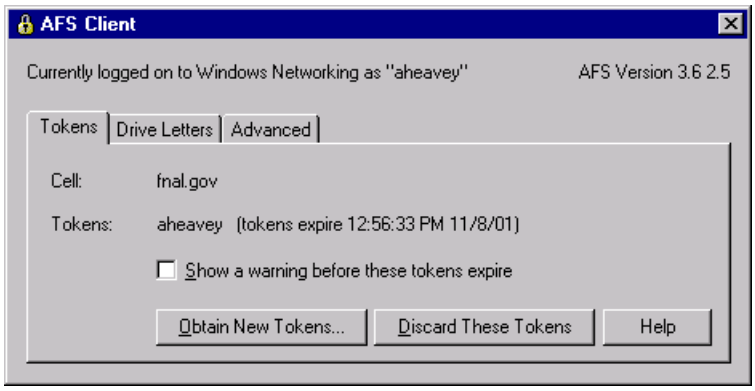
2) Authenticate to AFS space either by clicking on the AFS icon (the lock symbol with X:) on your task bar, or by navigating to **START > PROGRAMS > IBM AFS > CLIENT > AUTHENTICATION**. On the **AFS CLIENT** window, select the **TOKENS** tab. Click **OBTAIN NEW TOKENS...**

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



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

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



Your token is valid for six days, unless the AFS service is stopped before then. Every time you reboot, the service is halted and restarted, and thus the token is destroyed. 'Now you're ready to copy/paste/edit files on the AFS volumes in the same manner as for other drives.

Chapter 5: Using your CRYPTOCard

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

To obtain a CRYPTOCard, go to the *Request Form for Computing Username and Primary Accounts* at `http://computing.fnal.gov/cd/forms/acctreq_form.html`.

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

5.1 How does your CRYPTOCard Work?

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

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

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

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

5.2 Caring for your CRYPTOCard

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

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

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

5.3 Usage Notes

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

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

5.4 The First Thing to do: Reset your PIN

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

5.4.1 Resetting Initial PIN

Original Style Card

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

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

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

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

New Style Card (March 2002)

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

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

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

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

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

automatically.

5.4.2 Resetting PIN (General)

Original Style Card

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

New Style Card (March 2002)

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

5.5 Log in Using CRYPTOCard (the First Time)

5.5.1 Original Style Card

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

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

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

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

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

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

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

Kerberos tickets.

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

5.5.2 New Style Card (March 2002)

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

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

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

3) Ignore the `Adj LCD` or `Contrast` prompt (the latter appears on cards issued after November 2002) and press **MENU** again.

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

5) At the prompt `Ready` (for cards issued Nov. ‘02 or later, you see a flashing cursor instead), key the challenge displayed on your monitor into your CRYPTOCard, and press **ENT** to get a *response* of eight hex digits. (If you mistype, press **CLR** and re-enter the *challenge*. **CLR** clears one character at a time, or it will clear the whole field if held down for more than one second.)

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

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

5.6 Log in Using CRYPTOCard (Subsequently)

5.6.1 Original Style Card

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

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

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

```
CryptoCard RB-1
Press ENTER and compare this challenge to the one on your
display
Challenge is [12345678], Enter the displayed response:
```

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

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

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

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

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

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

5.6.2 New Style Card (March 2002)

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

PASSWORD

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

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

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

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

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

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

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

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

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

DIG SIG

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

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

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

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

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

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

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

5) The card now displays the response.

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

5.7 Reauthenticate using your CRYPTOCard

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

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

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

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

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

5.8 Resync your CRYPTOCard

5.8.1 Original Style Card

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

5.8.2 New Style Card (March 2002)

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

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

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

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

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

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

Chapter 6: Logging In from Off-Site

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

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

6.1 Description of Choices for Off-Site Machines

The choices for off-site machines include:

1) Install the Kerberos client (and optionally the Kerberized ssh client) software on your machines and sign up to be part of the FNAL.GOV strengthened realm. This means you can authenticate to Kerberos locally and connect to Fermilab computers using the Kerberized version of a network connection program. This is the preferred method. (Kerberos-lite is available, too; see section 6.2 *In a Pinch: Download Client-Only Version of Kerberos*).

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

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

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

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

If people need to log in from your site to change their passwords, there must be at least one local machine on which there is software which will allow it to be done locally (best) or over an encrypted connection (second best).

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

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

FNAL-Kerberos-clientonly versions have been created for RedHat Linux 7.1 and compatible systems, and for Windows 2000 (other Windows systems have not been tested but may work). Look for the software in the FermiTools area of Fermilab's FTP server:
`ftp://ftp.fnal.gov:8021/pub/fnal-kerberos-clientonly/current/`.
Instructions on how to setup and uninstall the software are included in the product.

For the distribution for Windows, it seems the DISPLAY variable needs to be set on the Windows machine before invoking ssh in order to trigger X forwarding (the value of DISPLAY doesn't seem to matter).

6.3 Obtaining CRYPTOCards

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

If you lose your CRYPTOCard or it becomes unusable for any reason, please open a helpdesk ticket (`http://helpdesk.fnal.gov/` or email *helpdesk@fnal.gov*) to request a new one. Then ask the person designated for your group or experiment to pick it up and mail it to you. Currently we do not have a way of restoring your access more quickly. By the end of 2001, we expect to have a mechanism in place whereby we can fax you a one-time password.

6.4 Exporting CRYPTOCards

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

6.5 Network Address Translation

There is an issue concerning users who maintain a small network of computers at home and whose ISP subjects them to NAT (Network Address Translation). NAT creates a firewall of sorts in which one computer (or the router itself) sits on your assigned IP address and routes traffic to a number of machines inside your house (wireless or not), all at the same time, using that one IP address.

When you authenticate, normally your IP address is part of that authentication. But that would be your *local* IP address, the one the machine knows, not the one that the outside world knows you by. Authentication won't work in this case. You can get an addressless ticket that doesn't have this problem.

A remote process (e.g., X Client) must be able to send its messages back to the correct machine through the NAT. The two simplest ways to do this are:

1) Use Fermilab's VPN (Virtual Private Network) to tunnel through the NAT. This gives you a Fermilab address (...fnal.gov) for Fermilab machines, but to the rest of the world, your address is still the one your ISP gave you. You must use VPN for tasks such as connecting to Windows disk servers on site, changing your Windows password, etc.

2) Tunnel through the NAT using ssh.

Kerberos 5 has the ability to natively generate addressless tickets, and Fermilab has built the Kerberos binaries with this functionality enabled. So you can use **`kinit -A`** instead of plain **`kinit`** to obtain a Kerberos ticket not bound to a particular IP address, which can then be passed through your firewall. In this case, the problem described above just doesn't arise.

Secondly, the Kerberos 4 compatibility libraries used to build the new Kerberos 5-based Kerberos Kits have been modified such that they do not check the IP addresses on Kerberos 4 tickets. This means that all the new server binaries (klogind, telnetd, etc.) also don't check IP address of Kerberos 4 tickets anymore, and therefore should work with clients behind NAT.

6.5.1 Windows

Install a version of **WRQ®** Reflection that supports OpenSSH connections and creation of addressless tickets (as with the **`kinit -A`** option). Versions 11 and following will work (for everything but FTP). To support remote processes (e.g., X Client), OpenSSH connections should be configured with the "Kerberos key exchange" box checked (an option under "Advanced" button on the WRQ Reflection X Manager - Connection template). The resulting communications tunnel through the NAT transparently.

6.5.2 Linux

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

6.5.3 Macintosh

For Macintosh OS 9 and earlier: To enable **BetterTelnet** to work for a Kerberized Macintosh in a NAT environment, you must add the following line to the `libdefaults` section of the `Kerberos Preferences` file (Note that this reduces the security of your Kerberos credentials.):

```
noaddresses = true
```

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

For Mac OS X and later: Add to the [libdefaults] section:

```
noaddresses = TRUE
```

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

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

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

7.1 Logging In Through Kerberized Exceed 7 Software from Windows

7.1.1 Telnet Connections

You should create one secure telnet profile for each Kerberized host you wish to access, according to the instructions in section 21.5 *Configuring the Exceed 7 Telnet Application*. To authenticate:

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

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

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

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

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

To connect:

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

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

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

7.1.2 FTP Connections

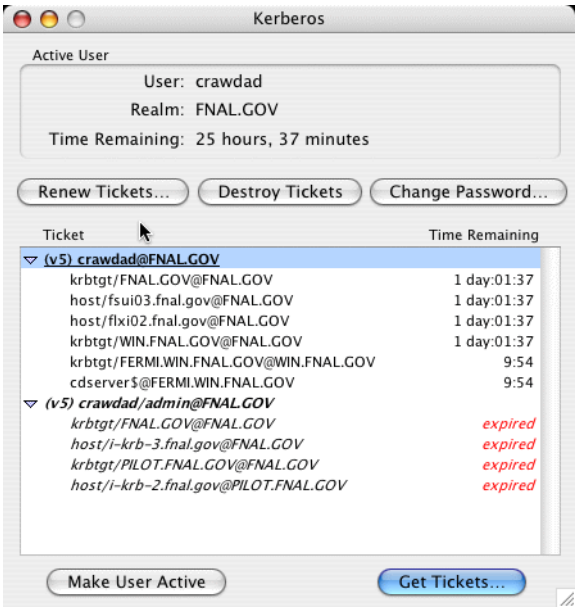
Exceed 7 does not provide a Kerberized FTP client. Furthermore, you cannot connect using your CRYPTOCard (as you may for WRQ® FTP, described in section 4.6.3 *Run an FTP Session to Kerberized Host*), since the Exceed 7 FTP client stores your password, and doesn't let you enter it each time you connect. Choose a different product! Suggestions: WRQ®, FileZilla, AFS Windows Client (for remote hosts using AFS).

7.2 Logging In from a Macintosh

Here we assume you are running the **MIT Kerberos** OS X 10 software for Macintosh as described in Chapter 23: *Installing and Configuring MIT Kerberos on a Macintosh System*.

There are two ways to authenticate to Kerberos on the Macintosh:

- Open a terminal window and use the command line **`kinit`** as you would on a Unix system. If you are logged into the machine under a username that's the same as your principal, just run **`kinit`** from your home directory, and Kerberos will pick the right principal for you. Otherwise you'll have to give your principal in the command: **`kinit [principal]`**.
- If you've installed the "Extras", go to the /Applications/Utilities folder and select Kerberos. You have to tell the Kerberos GUI what your Kerberos principal is. Click "Get Tickets".



You should see a ticket appear. Now you can invoke your **telnet** or **ssh** client and connect to one or more strengthened hosts without having to provide your password again. You have to tell telnet or ssh the name of the remote account you want to log in to, unless it's the same as the local account name (regardless of what your Kerberos principal is).

Chapter 8: Troubleshooting your Authentication Problems

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

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

- In many cases, when authentication fails, one of four things is likely to be wrong:
  - (1) your password,
  - (2) the date/time on your system (see section 14.1.7 *Synchronize your Machine with Time Server* for UNIX, 19.4 *Time Synchronization* for Windows, or 23.1.5 *Time Synchronization* for Mac OS X),
  - (3) the local host name in the `/etc/hosts` file (see section 17.3 *The /etc/hosts File*), or
  - (4) your CRYPTOCard is not configured for the target realm. The error message doesn't necessarily help you determine the problem: "`Preauthentication failed ...`", or "`Cannot establish a session with Kerberos administrative server...`" If this is the problem, bring your card to WH8NE to have it reprogrammed.

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

- When using the Kerberized versions of **telnet**, **rlogin**, or **rsh** (see Chapter 13: *Network Programs Available on Kerberized Machines*) to connect to another machine in the strengthened realm, some users have had to use the **`-l <login_name>`** option even when the login names on both systems match. (Don't ask why.) You definitely need to use this option if the login names don't match.
- "KDC policy rejects request" or "KDC can't fulfill requested option" usually means either you're requesting a forwardable ticket for a /root or /admin instance of your principal (not allowed), or you're trying to forward a ticket that's not forwardable, or renew one that's not renewable.
- "Key version number for principal in key table is incorrect" means either the keytab has changed since the service ticket was obtained (to solve, run **`kinit -R`** or **`kinit`**), or the service key (for host principal) in the KDC was changed after the keytab file was created (to solve, recreate keytab file on host, see section 17.10 *Installing Service Host Keys*).
- "Cannot contact any KDC for requested realm." Caused by firewall blocking KDC request or reply, or DNS failure.
- "Server not found in Kerberos database" Possible causes include: local hosts file or NIS map giving wrong name for host (check `/etc/hosts` file and make sure the full official host name appears first, not a nickname; see section 17.3 *The /etc/hosts File*), or a bad or missing `[domain_realm]` mapping in `/etc/krb5.conf`. It was also a bug in Fermi Kerberos v1_2; to solve, upgrade.
- "aklog: Couldn't get fnal.gov AFS tickets:, aklog: unknown RPC error (-1765328352) while getting AFS tickets". You may have failed to get fresh tickets from your screensaver unlock. A fresh **`kinit`** should clear this right up.
- Syslog message: Principal <principalname>@FNAL.GOV ... for local user <user> failed krb5_kuserok. `krb5_kuserok` is a function in the kerberos library. It is accessed by krshd, and fails for these reasons:
  - · requested user has no account on target system
  - · `krb5_unparse_name` fails
  - · can't open `~user/.k5login`
  - · `~user/.k5login` not owned by user or root
  - · principal doesn't match any line in `.k5login` (try **`od -c ~user/.k5login`** to look for any "invisible" characters in this file).
- If Kerberos functions are very slow on a client host, check its Kerberos logs for long intervals between "NEEDED_PREAUTH" and "ISSUE" and see if there are few or no repeats of the same request to different KDCs. If so, the client host's first-configured DNS server may be slow or dead.

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

SSH Problems

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

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

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

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

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

Chapter 9: Using Kerberos

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

9.1 Ticket Properties and Options

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

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

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

9.1.1 Default Ticket Flags and Lifetimes

At Fermilab, the maximum ticket lifetime is set to 26 hours, and the default ticket lifetime as set on individual systems is constrained to be this value or less. The default flags and lifetimes of tickets obtained on a UNIX machine by login and **`kinit`** are set by entries in that machine's `/etc/krb5.conf`. (For other operating systems, the default values are typically set via a more user-friendly interface.) The maximum renewable ticket lifetime is seven days. We discuss the `krb5.conf` file in Chapter 16: *The Kerberos Configuration File: krb5.conf*.

9.1.2 Credential Caches

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

9.1.3 Tickets for Root Instance of Kerberos Principal

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

---

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

9.2 Ticket Management

9.2.1 Obtaining Tickets (Authenticating to Kerberos)

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

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

To authenticate:

UNIX desktop with Kerberos software and Kerberos login program	Log in, and provide your Kerberos password. See section 4.1 *Logging In at the Console of a Kerberized UNIX Machine*.
UNIX desktop with Kerberos and standard UNIX login program	Log in with your UNIX password, then run **`kinit`**. See section 4.1 *Logging In at the Console of a Kerberized UNIX Machine*. Also see 12.1 *kinit*.
Windows desktop with WRQ®	Navigate to **START > PROGRAMS > REFLECTION > UTILITIES > KERBEROS MANAGER** to open the **Reflection Kerberos Manager** application. With your principal tab selected, click **AUTHENTICATE**. Provide your Kerberos password as prompted (and click **FORWARDABLE**). See section 4.6 *Logging In Through WRQ® Reflection Software from Windows*.
Windows desktop with Leash32 and Kerberos	Using the Leash32 utility, navigate to **START > PROGRAMS > KERBEROS UTILITIES > LEASH32** . Select **GET TICKET** on the **ACTION** menu. Provide your Kerberos password as prompted. See section 21.4 *Getting a Ticket* .
Macintosh desktop with Kerberos	For OS X, see section 23.1.4 *Authenticate to Kerberos*. For OS 9 and earlier: Invoke the **KERBEROS CONTROL PANEL** (from **CONTROL PANELS** under the Apple menu, from the **KERBEROS MENU** in the menu bar, or from the **KERBEROS CONTROL STRIP** module). Click **GET TICKETS**. Enter your Kerberos password on the pop-up screen. See section 7.2 *Logging In from a Macintosh*.
Remote UNIX host (from desktop with no Kerberos software installed)	Start an ssh (or telnet or FTP) session to a Kerberized host, use your CRYPTOCard to generate a password, and log into the remote host using that one-time password. See section 4.4 *Connecting from a NonKerberized Machine: Portal Mode*.

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

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

9.2.2 Viewing Tickets

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

To view tickets:

UNIX desktop with Kerberos software	Run the **`klist`** command (**`-f`** option recommended to show ticket flags). See section 12.2 *klist*.
Windows desktop with WRQ®	Navigate to **START > PROGRAMS > REFLECTION > UTILITIES > KERBEROS MANAGER** to open the **Reflection Kerberos Manager** application. Ticket should be visible on this window. Right-click on ticket to see ticket properties. See section 4.6 *Logging In Through WRQ® Reflection Software from Windows*.
Windows desktop with Leash32.	Using the Leash32 utility, navigate to **START > PROGRAMS > KERBEROS UTILITIES > LEASH32** . Ticket should be visible on this window. See section 21.4 *Getting a Ticket*.
Macintosh desktop with Kerberos	For OS X, use the Unix method, or click on the ticket in the GUI. For OS 9 and earlier: Invoke the **KERBEROS CONTROL PANEL** (from **CONTROL PANELS** under the Apple menu, from the **KERBEROS MENU** in the menu bar, or from the **KERBEROS CONTROL STRIP** module). Ticket should be visible on this window. See section 7.2 *Logging In from a Macintosh*.
Remote Kerberized UNIX host	Run the **`klist`** command (**`-f`** option recommended to show ticket flags). See section 12.2 *klist*.

About the klist Command

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

```
% klist -f
```

This produces output of the form:

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

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

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

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

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

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

9.2.3 Destroying Tickets

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

To destroy tickets:

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

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

Windows desktop with WRQ®	Navigate to **START > PROGRAMS > REFLECTION > UTILITIES > KERBEROS MANAGER** to open the **Reflection Kerberos Manager** application. Tickets should be visible on this window. Click on **CLEAR TICKETS**. To automate the clearing of tickets, you can click **CLEAR ALL TICKETS ON SHUTDOWN** from the **CONFIGURATION** menu.
Windows desktop with Leash32	Using the Leash32 utility, navigate to **START > PROGRAMS > KERBEROS UTILITIES > LEASH32** . Ticket should be visible on this window. Click on **DESTROY TICKET(S)**. To automate the clearing of tickets, you can click **DESTROY TICKETS/TOKENS ON EXIT** from the **OPTIONS** menu to clear tickets when you exit Leash32.
Macintosh desktop with Kerberos	For OS X, use the Unix method, or click Destroy Tickets on the GUI. For OS 9 and earlier: Invoke the **KERBEROS CONTROL PANEL** (from **CONTROL PANELS** under the Apple menu, from the **KERBEROS MENU** in the menu bar, or from the **KERBEROS CONTROL STRIP** module). Ticket should be visible on this window. Click on **DESTROY TICKETS**.
Remote Kerberized UNIX host	Run the **`kdestroy`** command.

Destroying Tickets Selectively

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

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

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

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

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

9.2.4 Forwarding Tickets

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

1) you must first obtain a forwardable ticket,

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

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

UNIX desktop with Kerberos software and Kerberos login program	To obtain a forwardable ticket, the `/etc/krb5.conf` must show `forwardable = true` for `login`) under `[appdefaults]`). If not, check for `forwardable = true` for `kinit`. If this is true, run **`kinit`**. If false, run **`kinit -f`**. To forward your forwardable ticket to a remote UNIX host, use a Kerberized connection program with ticket forwarding on[a] (e.g., **`telnet -F`**).
UNIX desktop with Kerberos and standard UNIX login program	To obtain a forwardable ticket, check for `forwardable = true` for `kinit` in `/etc/krb5.conf`. If true, run **`kinit`**. If false, run **`kinit -f`**. To forward your forwardable ticket to a remote UNIX host, use a Kerberized connection program with ticket forwarding on (e.g., **`telnet -F`**). (See footnote a.)
Windows desktop with WRQ®	To obtain a forwardable ticket, click **FORWARDABLE** when you authenticate. See section 4.6 *Logging In Through WRQ® Reflection Software from Windows*. To forward your forwardable ticket to a remote telnet session, verify that the telnet configuration file you're using specifies **FORWARD TICKET** on the **SECURITY PROPERTIES** window. See section 19.8 *Configuring WRQ® Reflection telnet Connections*. Note: WRQ®'s FTP client doesn't support forwarding tickets. This only poses a problem for remote hosts running AFS since you don't get your AFS token upon connection. See section 4.6.3 *Run an FTP Session to Kerberized Host*.

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

Windows desktop with Leash32, MIT Kerberos and Exceed 7	To obtain a forwardable ticket, make sure your configuration specifies `Forwardable` under **TICKET OPTIONS** as described in section 21.3 *Configuring Kerberos using Leash32*. To forward your ticket to a telnet session, verify that the telnet configuration file you're using specifies `Forwarding` under **KERBEROS 5 OPTIONS**. See section 21.5 *Configuring the Exceed 7 Telnet Application*. Then run the telnet client. Note: The Exceed 7 FTP client cannot be Kerberized; try FileZilla FTP.
Macintosh desktop with Kerberos	For OS X, use the Unix method, or (GUI method not documented yet). For OS 9 and earlier: To obtain a forwardable ticket, edit your Preferences and check **FORWARDABLE TICKETS ALWAYS**. To forward the ticket via a BetterTelnet connection, check **KERBEROS FORWARDING** when you're configuring the Security portion of Favorites for that application.
Remote Kerberized Host via Portal Mode	When you obtain your ticket upon CRYPTOCard login to a remote host, the ticket's properties are determined by the `/etc/krb5.conf` file on the host. Run **`klist -f`** to see if the `F` flag shows up indicating a forwardable ticket. If it doesn't, and if you used ssh to connect thus providing an encrypted connection, then you can run **`kinit -f`** to get one, BUT ONLY RARELY! To forward your forwardable ticket to a remote UNIX host, use a Kerberized connection program with ticket forwarding on.

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

Descriptions of the forwarding option (and other Kerberos functions) added to the connection programs in the Kerberos V5 package can be found in Chapter 13: *Network Programs Available on Kerberized Machines* and at `http://hoth.stsci.edu/public/krb5/user-guide.html#SEC16`.

Tickets and IP Addresses: How forwarding works

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

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

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

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

A Note about AFS tokens and Forwarding

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

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

A Word about Ticket Caches and Forwarding

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

Example (UNIX)

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

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

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

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

9.2.5 Renewing Tickets

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

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

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

UNIX desktop with Kerberos software	To request a renewable ticket, use **`kinit -r <renewable_lifetime>`**. This requires password entry, therefore it must only be performed at the keyboard of a strengthened machine or (infrequently) over an encrypted connection. To renew the ticket, use **`kinit -R`** before the ticket expires. **`kinit -R`** does not require password entry.
Windows desktop with WRQ®	To request a renewable ticket, navigate to **START > PROGRAMS > REFLECTION > UTILITIES > KERBEROS MANAGER** to open the **Reflection Kerberos Manager** application. With your principal tab selected, click **AUTHENTICATE**. Provide a non-zero value for **RENEWABLE DURATION**. See section 4.6 *Logging In Through WRQ® Reflection Software from Windows*. To renew the ticket, again open the **Reflection Kerberos Manager** application. With your principal tab selected, click
Windows desktop with Leash32 and Kerberos	To request a renewable ticket, use the command prompt, and type **`kinit -r <renewable_lifetime>`**, as for UNIX. To renew the ticket, use the **`kinit -R`** option before the ticket expires. **`kinit -R`** does not require password entry.
Macintosh desktop with Kerberos	It appears that tickets obtained via the Macintosh Kerberos software are renewable by default (although the "R" flag does not appear). For OS X, use the Renew Tickets button on the GUI, or use the Unix method. For OS 9 and earlier: to renew a ticket, invoke the **KERBEROS CONTROL PANEL** (from **CONTROL PANELS** under the Apple menu, from the **KERBEROS MENU** in the menu bar, or from the **KERBEROS CONTROL STRIP** module). Click **RENEW TICKETS...**.
Remote Kerberized host via Portal Mode	Run the command **`new-portal-ticket`** and use your CRYPTOCard.

Example

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

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

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

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

```
% kinit -R
```

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

9.2.6 Update Tickets on Remote Terminal Sessions

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

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

k5push

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

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

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

k5push options

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

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

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

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

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

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

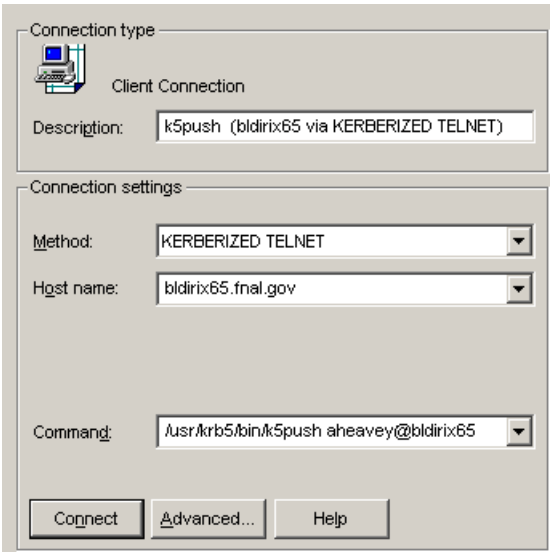
```
<host>.fnal.gov
<host>.<domain>
<account>@<host>.<domain>
```

Using k5push from Windows with WRQ®

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

To update tickets on a single remote session:

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

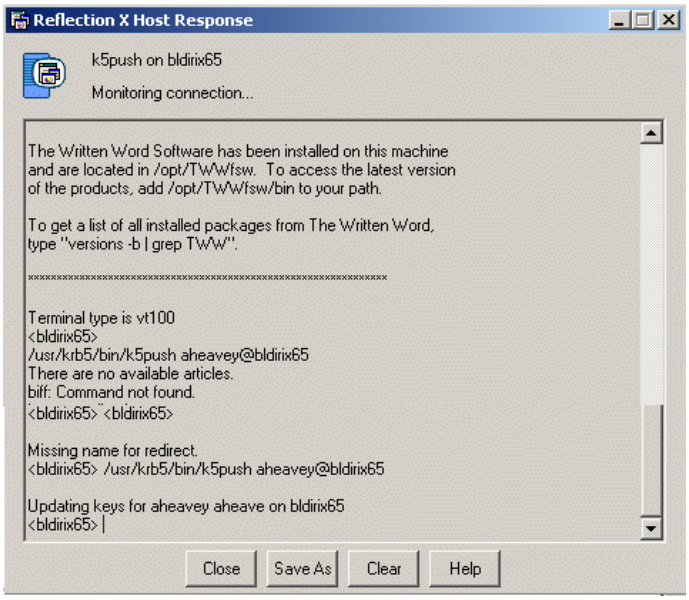


- Still on the **X Client Manager** window, click **ADVANCED...**

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

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

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



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

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

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

9.3 Account Access by Multiple Users

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

9.3.1 The .k5login File

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

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

This file overrides all other rules for granting login access!

Do you need a .k5login file?

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

Sample .k5login

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

9.3.2 About Group Accounts

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

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

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

1) Authenticate to Kerberos under your own account.

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

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

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

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

9.3.3 The .k5users File

If you want to give restricted super user access to your account to another principal (access method limited to **ksu**; see section 12.5 *Kerberized su (ksu)*), you can create a `.k5users` file. The `.k5users` file is similar to the `.k5login` file, except that each principal is optionally followed by a list of commands which restricts the principal to those commands, and the file is only consulted by the **ksu** command.

Here is a sample `.k5users` file:

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

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

Two bombs:

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

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

9.4 Using Root Instance of your Principal

9.4.1 What is a Root Instance of a Principal?

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

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

9.4.2 How do You Use your /root Principal?

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

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

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

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

where:

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

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

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

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

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

9.4.3 How Should You NOT Use It?

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

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

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

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

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

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

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

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

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

Chapter 10: Miscellaneous Topics for the User

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

10.1 Running Xwindows

10.1.1 UNIX

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

```
% rsh -n -f -x <remote_host>.fnal.gov -l <username> \
  xauth add `xauth list $HOSTNAME:0`
```

(Those are backquotes around **`xauth list $DISPLAY`**.) Executing a command like this can be made more convenient. You can create an alias or shell script that sends over your **`xauth`** magic cookie (or performs an **`xhost +<remote_node>`** locally, if you use **`xhost`**, but it's considerably less safe -- someone on that host could get access to your screen and keyboard). Run it before starting the connection program. Change the script to mode 755. Here is some sample content for such a script, which we call `kxtelnet` (it's been tested between Linux, Solaris and IRIX machines):

```
#!/bin/sh

if [ $# -gt 2 -o $# -lt 1 ]; then
  echo "usage: kxtelnet RemoteHostName [RemoteUserName]" 1>&2
  exit 1
fi

host=$1
user=${2:-$USER}
```

```
case "$DISPLAY" in
  :*) disp=`hostname`$DISPLAY;;
  *)  disp=$DISPLAY;;
esac

/usr/krb5/bin/rsh -n -x -l $user $host \
    /bin/sh -c \
      "'PATH=/usr/X11R6/bin:/usr/openwin/bin:/usr/bin/X11:\$PATH; \
        export PATH; \
        xauth add `xauth list $disp`; \
        xauth list $disp'"

exec /usr/krb5/bin/telnet -x -l $user $host
```

(In the 9th line, `:*) disp=`hostname`$DISPLAY;;`, those are single backquotes around `hostname`. Same for `xauth list $disp` in 3rd to last line.) Instead of a script, you can set up an alias for a command like the following, and run it each time you restart Xwindows, before connecting to the remote host:

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

works on some machines and not others, while

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

seems to work on those machines where ssh doesn't appear to work. (Often the failure is due to **`xauth`** not being in the default path.) Run this manually rather than with **`startx`** so that you can still get into Xwindows if for some reason this fails.

10.1.2 Windows NT4/98/95

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

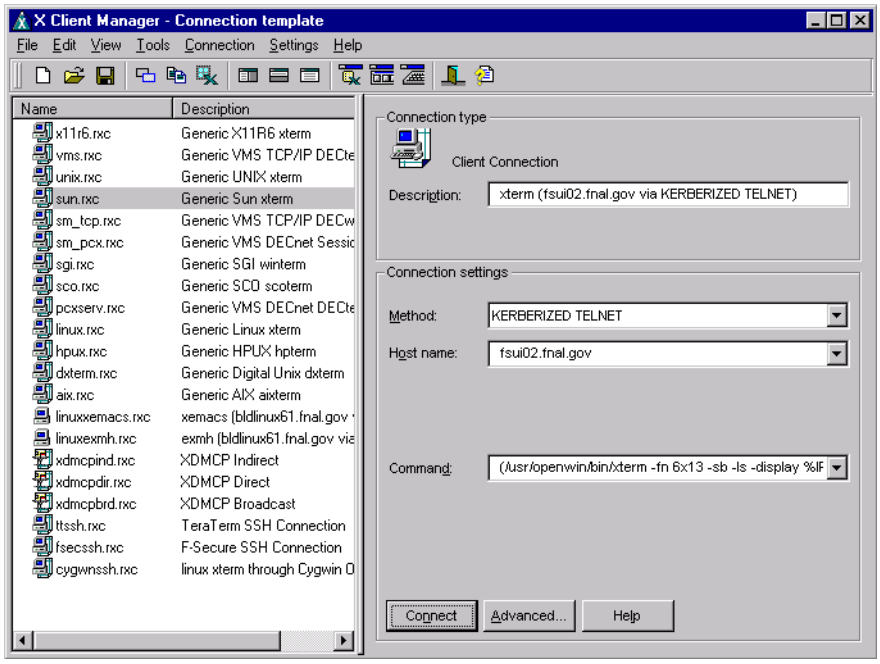
In the kerberos-users@fnal.gov mailing list archive, you can find a message containing couple of handy scripts for connecting to nodes using WRQ® Reflection X. Search for "handy scripts" and you'll find the right message![1] The first script shows how to use your own kerberos principal to log in to your

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

own account on a remote node. The second script shows how to use the your */root* principal (see section 9.4 *Using Root Instance of your Principal*) to log in to a different account, where forwarded/renewable tickets aren't allowed.

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

Run a telnet Session



The best thing to do at this point is to minimize the above window, start a **telnet** session, and run X applications from there as described in section 4.6 *Logging In Through WRQ® Reflection Software from Windows*. Once you're connected, verify that your $DISPLAY is set correctly on the remote host (at Fermilab, this should already be set for you in your UNIX login files; if it's not, check these files).

Connect Directly from X Client Manager

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

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

Run a telnet Session with Automatic X Application Startup

For applications that you run often, you might find it useful to configure a **telnet** connection that includes an automatic X application startup. This is described in section 19.9 *Configure X Connection to Host*. Once you have your host-specific, application-specific configurations created and saved, they should appear in the **REFLECTIONS SESSIONS** folder. To invoke, double click on the file corresponding to the host/application you want. The system will log you in and start the application in your X window manager.

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

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

10.1.3 Macintosh

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

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

10.2.1 Cutting and Pasting

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

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

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

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

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

10.2.2 Using Matrix through X Windows Interface

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

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

10.3 Automated Processes

10.3.1 Specific-User Processes (cron Jobs)

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

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

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

Note that default cron ticket lifetime is picked up from the `[ftpd]` in `/etc/krb5.conf`. This is important to know if you want to increase the time limit for a cron job ticket.

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

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

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

```
% setup perl
% setup kcroninit
% kcroninit
```

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

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

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

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

```
% setup kcroninit
% kcrondestroy
```

10.3.2 Processes Running as root

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

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

Then run **`kinit`**:

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

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

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

10.3.3 Non-root, Non-Specific-User Processes

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

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

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

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

Setting Up the Task

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

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

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

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

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

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

```
% /usr/krb5/sbin/kadmin -p <user>/<group>/<jobname>
   Enter password:  <-- type in your password
kadmin:  addprinc -randkey
  <jobname>/<group>/<hostname>.<domain>
kadmin:  ktadd -k /path/to/keytab/file
  <jobname>/<group>/<hostname>.<domain>
kadmin:  exit
```

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

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

Running the Task

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

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

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

10.4 Sending Data from Unstrengthened to Strengthened Machines

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

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

10.5 CVS

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

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

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

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

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

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

allows only CVS commands).

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

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

Part III User's Reference Manual

Chapter 11: *Encrypted vs. Unencrypted Connections*

In this chapter, we provide guidance on determining whether your connection is encrypted, and ensuring that you open an encrypted connection.

Chapter 12: *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 13: *Network Programs Available on Kerberized Machines*

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

Chapter 11: Encrypted vs. Unencrypted Connections

In this chapter, we provide guidance on determining whether your connection is encrypted, and ensuring that you open an encrypted connection, as needed.

To comply with Fermilab policy, you only strictly need an encrypted network connection when you type your Kerberos password. And to further comply with policy, you should type your Kerberos password over the network extremely rarely, if at all!

11.1 How do you know if your connection is encrypted?

When you're connecting to a strengthened machine over the network, it's very important to know if your connection is encrypted. If it is, you can reasonably safely run Kerberos commands that require input of your Kerberos password (see note above). If your connection is not encrypted, you must not type your Kerberos password, since it would be transmitted in the clear. Notice that there are lots of bombs in this chapter!

If you have a chain of multiple connections (e.g., machine1 to machine2, machine2 to machine3, and so on), and if only one connection is unencrypted, then your connection as a whole is **unencrypted**. Do not type your Kerberos password in this case!

Now, we'll discuss the individual connections ...

11.1.1 Connecting from Kerberized UNIX/Linux Desk-

tops

SSH

If you connect via Kerberized ssh, verify your ssh client configuration to make sure it initiates encrypted sessions. This will vary depending on the ssh client. If you're not sure, use the command with the **`-c`** option as follows:

```
% ssh -c 3des <host>
```

or other argument to `-c` (except for **`none`**).

Other Kerberized Connection Program (e.g., telnet)

Your connection is encrypted if you are connected via one of the Kerberized programs with the "encryption on" flag set. The program generally tells you. For example, for telnet, you can tell if the default is set for encryption by typing the escape character (default is **CTRL-]**, it can be reset with **`-e`** flag), and entering **`status`**. Encryption information should be listed.

For any Kerberized connection program, you can always check the default setting in the `[appdefaults]` section of your `/etc/krb5.conf` file. Look for `encrypt=true` for the program you're using. If encryption is not on by default, use the encryption flag, e.g.,:

```
% rsh -x <host>
```

```
% telnet -x <host>
```

where **`rsh`** and **`telnet`** reside in `/usr/krb5/bin`. Reference Chapter 13: *Network Programs Available on Kerberized Machines* for command syntax.

If you connected in a different way, or if you're not sure, then **assume that the connection is not safe**, log out and log in again as shown here.

11.1.2 Connecting over a CRYPTOCard ssh Session

Verify your ssh client configuration to make sure it initiates encrypted sessions. This will vary depending on the ssh client. If you're not sure, use the command with the **`-c`** option as follows:

```
% ssh -c 3des <host>
```

or other argument to `-c` (except for **`none`**).

11.1.3 Connecting over a CRYPTOCard telnet Session

CRYPTOCard telnet connections are **unencrypted**, and it's **never safe** to issue your Kerberos password. See section 11.2 *If it's unencrypted, what do I do when I need to reauthenticate?*.

11.1.4 Connecting over a CRYPTOCard ftp Session

CRYPTOCard ftp connections are **unencrypted**.

11.1.5 Connecting from an X Terminal

The connection from an X terminal to a host is **never encrypted**, so you must **never** issue your Kerberos password from an X terminal, no matter how secure the connections are beyond that point.

11.1.6 Connecting from a PC Running Windows

Helpful hint: look for the locked lock symbol in your session window to ensure the connection is encrypted!

With ssh

This will vary depending on the ssh client. Verify your client configuration to make sure it initiates encrypted sessions.

With WRQ® telnet client

WRQ® Reflection Security Components v8.0.0 supports ticket forwarding to the remote host, so you shouldn't need to run any commands on the remote system that require password entry. Therefore you may not need an encrypted connection (see section Chapter 19: *Configuring WRQ® Reflection telnet Connections*). If you need to type your password on the remote host for any reason, then you do need an encrypted connection. Make sure that the **WRQ® Reflection** telnet client is configured as described in section 19.8 *Configuring WRQ® Reflection telnet Connections*.:

If you've installed **WRQ® Reflection X**, you can opt to connect to a host directly from the **X CLIENT MANAGER** window, *but* it **does not provide encrypted connections**. If you will need credentials on the host, go through a normal **telnet** connection. **Do not `kinit` from an X window!**

With MIT Kerberos and Exceed 7.0 telnet client

Exceed 7.0 supports ticket forwarding to the remote host, so you shouldn't need to run any commands on the remote system that require password entry. Therefore you may not need an encrypted connection. If you need to type your password on the remote host for any reason, then you do need an encrypted connection. To enable encryption, configure your Kerberized Exceed 7.0 telnet connections as described in section 21.5 *Configuring the Exceed 7 Telnet Application*.

11.1.7 Macintosh: MIT Kerberos and BetterTelnet

In section 23.5 *Configuring Telnet* pay attention to item (3). To summarize:

Invoke **BetterTelnet**. On the **FAVORITES** menu, choose **EDIT FAVORITES**. On the pop-up screen, click **NEW** to create a new configuration or edit an existing one. Change to the **SECURITY** tab, check `Kerberos authentication` and `Kerberos encryption`. Click **OK** to save the configuration.

11.2 If it's unencrypted, what do I do when I need to reauthenticate?

One option for updating tickets on remote sessions is to use the `k5push` script documented in section 9.2.6 *Update Tickets on Remote Terminal Sessions*.

For portal mode connections, a script is provided with the Fermi **kerberos** product as of version v1_2, that safely reauthenticates you on a Kerberized host using your CRYPTOCard over an unencrypted connection. The process exploits the portal mode feature that telnet with a CRYPTOCard always gets you a new key. The script is found at `/usr/krb5/bin/new-portal-ticket` (the script content is provided at the end of this section). Here's how it works:

From your X terminal or unstrengthened machine, you run telnet to a Kerberized machine and use your CRYPTOCard to authenticate. You get logged in on, say, `/dev/ttyp3`, with Kerberos tickets cached in `/tmp/krb5cc_ttyp3`. Now your ticket expires. Still logged into the Kerberized node, you log into the same machine again but using the nonKerberized telnet:

```
% /usr/bin/telnet localhost
```

The machine responds in portal mode, you use your CRYPTOCard, and you're now logged in on `/dev/ttyp4`, for example, with a new `/tmp/krb5cc_ttyp4` file that has a new cached Kerberos ticket.

So now you have two Kerberos cache files, and you're logged into the machine twice. One cache file (`/tmp/krb5cc_ttyp3`) has an old expired ticket in it, and the other (`/tmp/krb5cc_ttpy4`) has a fresh, new, usable ticket.

Next, copy your fresh `/tmp/krb5cc_ttyp4` file onto `/tmp/krb5cc_ttyp3` (both cache files live on the destination machine, so you're doing a safe, local file copy), run **`kdestroy`** (which removes `/tmp/krb5cc_ttyp4`), and log out once. Now you're back at `ttyp3`, with a fresh new kerberos ticket in `/tmp/krb5cc_ttyp3`, and you can continue doing whatever you were doing when your ticket expired.

Script Contents

```
#!/bin/sh

# get uid
eval `id | sed -e 's/(.*//'`

# figure ticket cache
if [ "x$KRB5CCNAME" = x ]
then
    krb5file=/tmp/krb5cc_$uid
else
    krb5file=`echo $KRB5CCNAME | sed -e sxFILE:xx`
fi

(
   read line
   echo $line
   sleep 1000 &
   pid=$!
   echo 'rkrb5file=`echo $KRB5CCNAME | sed -e sxFILE:xx`'
   echo "cp \$rkrb5file $krb5file"
   echo "kdestroy"
   echo "echo xyzzy $pid xyzzy"
   echo "exit"
   wait $pid
) | (
    /usr/krb5/bin/telnet localhost
) |
    while read line
    do
        set : $line
        case $2 in
        Press)
    printf "$line\n"
    printf "Enter the displayed response: "
    ;;
        xyzzy)
    kill $3
    ;;
        esac
    done
```

Chapter 12: 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**.

12.1 kinit

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

12.1.1 Syntax

```
% kinit [-A] [-c cache_name] [-f] [-F] [-g [-h] | G]\
 [-k [-t keytab_file]] [-l lifetime] [-p] [-P]\
 [-r renewable_life] [-R] [-s start_time] [-S service_name]\
 [-v] [-V] [-4] [-5] [principal]
```

12.1.2 Option Descriptions

`-A` requests addressless ticket (used to obtain a Kerberos ticket not bound to a particular IP address, which can then be passed through a NAT-created "firewall"; see section 6.5 *Network Address Translation*).

`-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 by system. 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 existing contents of the cache are destroyed by **kinit**.

-f requests forwardable tickets

-F requests nonforwardable tickets

-g runs **aklog** after obtaining tickets; if you choose this, you may also choose **-h**.

-G does not run **aklog** after obtaining tickets

-h does AFS aklog setpag (goes with **-g**)

-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 (except to *root*); users cannot read it). Keytab files are generally used for service principals. They are also used for **cron** jobs (see section 10.3.1 *Specific-User Processes (cron Jobs)*).

-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 (26 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**.

-p requests proxiable tickets

-P requests nonproxiable 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.

-s <start_time>

requests a postdated ticket, which can be validated (by action of the 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**.

-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.

-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.

-V displays verbose output

-4 gets Kerberos v4 tickets only; by default get v5 only

-5 gets Kerberos v5 tickets only (default)

12.1.3 Examples

Default

Typically you can run the **kinit** command without options. This gets you a 26-hour ticket with the flags FIA set by default (Forwardable, Initial, Preauthenticated; flags are viewable using **klist -f**, see section 12.2 *klist*), plus an AFS token if AFS is running on the machine.

Get Ticket with Specified Lifetime

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

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

Get Renewable Ticket

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 sets the AFS token lifetime to four days):

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

Then, before the lifetime of 26 hours has passed, and before four days expire (you can renew a ticket multiple times within its renewable lifetime, but not after it has expired), renew the ticket using the **-R** option:

```
% kinit -R
```

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

Get Postdated Ticket

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
```

Get Ticket based on Key

The following command requests a TGT for the principal project/group/host.fnal.gov, for the duration 30 minutes, with authentication done on the basis of a key previously stored in the keytab file

`/usr/tmp/project.keytab` (this command would normally be included in a **cron** job file, not run interactively; see section 10.3.1 *Specific-User Processes (cron Jobs)*):

```
% kinit -l 30m -k -t /usr/tmp/project.keytab \
  project/group/host.fnal.gov
```

If you have an automatic process running as *root*, it is simplest to consider that the host on which the job runs is the party responsible for the accesses it initiates, and have it use the `/etc/krb5.keytab` to obtain credentials as `host/<hostname>.<domain>`:

```
% kinit -l 30m -k host/<hostname>.<domain>
```

12.2 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.

12.2.1 Syntax

```
% klist [-e] [[-c] [[-f] [-s] [-a [-r]] [<cache_name>] ]]\
  [-k [-t] [-K] [<keytab_name>]] [-4] [-5]
```

12.2.2 Option/Argument Descriptions

`-a`	displays the address list. Requires **`-c`**. Invalid with **`-k`**.
`-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.
`<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>`**. Requires **`-c`**. Invalid with **`-k`**.

-e displays the encryption types of the session key and the ticket for each credential in the credential cache (by default), or each key in the keytab file (if **-k** is specified).

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

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

Requires **-c**. 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**.

-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**.

-R does not reverse-resolve[1]. Requires **-a** (and thus **-c**). 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. Requires **-c**. Invalid with **-k**.

-t displays the time entry timestamps for each keytab entry in the keytab file. Invalid with **-c**.

1. To reverse-resolve involves getting a message (i.e. kerberos ticket or email) with an originating IP address. The recieving machine then would check the IP address with the nameserver (DNS). The -R option skips this test/check.

-4	lists only Kerberos v4 tickets/keys (default lists both 4 and 5)
-5	lists only Kerberos v5 tickets/keys (default lists both 4 and 5)

12.2.3 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@FNAL.GOV

   Valid starting     Expires            Service principal
   02/11/00       12:45:33              02/12/00       01:45:33
   krbtgt/FNAL.GOV@FNAL.GOV
           Flags: FIA
   02/11/00 12:45:33  02/12/00 01:45:33  afs/fnal.gov@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 10.3.1 *Specific-User Processes (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@FNAL.GOV
```

12.3 kpasswd

The **kpasswd** command is used to change a Kerberos principal's password. You can change a principal's password from any account on a machine in the realm. **kpasswd** prompts for the current Kerberos password, and if supplied correctly, the user is then prompted twice for the new password, and the password is changed. **kpasswd** works even if the old password has expired.

In the 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. For *root*, the password must contain at least 13 characters of at least three classes. The character classes are: lower case, upper case, numbers, punctuation, and all other characters.

12.3.1 Syntax

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

12.3.2 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.

12.4 kdestroy

The **kdestroy** utility destroys the user's active Kerberos credentials (tickets) by writing zeros to the specified credentials cache that contains them, and then deleting the cache. If the credentials cache is not specified, the default credentials cache specified by $KRB5CCNAME is destroyed.

12.4.1 Syntax

```
% kdestroy [-q] [-c cache_name] [-4] [-5]
```

12.4.2 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 the $KRB5CCNAME environment variable is set, its value is used to name the default cache. At Fermilab, this variable is typically set to **FILE:/tmp/krb5cc_<some_string>**.

-4 destroys only Kerberos v4 tickets/keys (default destroys both 4 and 5)

-5 destroys only Kerberos v5 tickets/keys (default destroys both 4 and 5)

12.5 Kerberized su (ksu)

Be aware that you need to have a host principal in order to use ksu. See section 14.1.6 *Do you Need to Allow Incoming Kerberos Connections?* about host principals.

12.5.1 Syntax

The following discussion 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 ...>]]
```

12.5.2 Description

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.

12.5.3 Option Descriptions

More complete option descriptions are available at the **ksu** man page.

-n target_principal_name; if **ksu** is invoked without **-n**, a default principal name is assigned

-c source_cache_name; if **-c** option is not used then the name is obtained from KRB5CCNAME environment variable.

-C target_cache_name

-k Do not delete the target cache upon termination of the target shell or a command (**-e <command>**).

-D turn on debug mode.

Ticket granting ticket options:

-l <lifetime> option specifies the lifetime to be requested for the ticket; if this option is not specified, the default ticket lifetime (configured by each site) is used instead.

-r <time> option specifies that the RENEWABLE option should be requested for the ticket, and specifies the desired total lifetime of the ticket.

-p specifies that the PROXIABLE option should be requested for the ticket.

-f option specifies that the FORWARDABLE option should be requested for the ticket.

-z restrict the copy of tickets from the source cache to the target cache to only the tickets where client = the target principal name. Use the **-n** option if you want the tickets for other then the default principal. Note that the (lower case) **-z** option is mutually exclusive with **-C** and (upper case) **-Z** options.

-Z Don't copy any tickets from the source cache to the target cache. Just create a fresh target cache, where the default principal name of the cache is initialized to the target principal name. Note that (upper case) **-Z** option is mutually exclusive with **-C** and (lower case) **-z** options.

-q suppress the printing of status messages.

-e command [args ...] **ksu** proceeds exactly the same as if it was invoked without the **-e** option, except instead of executing the target shell, **ksu** executes the specified command.

-a args specify arguments to be passed to the target shell. All options intended for **ksu** must precede **-a**.

12.6 kvno

The **kvno** command acquires a service ticket for the specified Kerberos principals and prints out the key version numbers of each. It uses the variables KRB5CCNAM (location of the credential cache) and KRBTKFILE (location of the v4 ticket file).

12.6.1 Syntax

```
% kvno [-q] [-h] [-4 | [-c <ccache>] [-e <etype>]] \
  <service1> <service2> ...
```

12.6.2 Option Descriptions

-c <ccache> specifies the name of a credentials cache to use (if not the default). Invalid with **-4**.

-e <etype> specifies the enctype[1] which will be requested for the session key of all the services named on the command line. This is useful in certain backward compatibility situations. Invalid with **-4**.

-q suppresses printing

-h prints a usage (help) statement and exits

-4 specifies that Kerberos version 4 tickets should be acquired and described. This option is only available if Kerberos 4 support was enabled at compilation time.

1. From the Mozilla development center web page: DOM:form.enctype gets/sets the content type of the FORM element.

Chapter 13: Network Programs Available on Kerberized Machines

In this chapter we document the Kerberized features of the network connection programs that are usable with Kerberos v5.

13.1 Introduction

The Kerberos V5 network programs are versions of existing UNIX network programs with the Kerberos features added. We call these versions "Kerberized". They include **telnet**, **rsh**, **rlogin**, **FTP**, and **rcp** which come with the installation of a Kerberos 5 client, and **ssh**, **slogin** and **scp** which come with a Kerberized **ssh** client. These programs have the original features of the corresponding non-Kerberized 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 network program is configured and whether the target machine is Kerberized, you may be prompted for either your login id or password, both, or neither.

You can check the defaults set for the (non-ssh) programs in the `[appdefaults]` section of the `/etc/krb5.conf` file. For **ssh** configuration, see the **ssh** man pages. 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).

In this chapter we list only the command syntax and the Kerberos-added features for these programs.

13.2 Kerberized telnet

Communicate with another host using the TELNET protocol. Use with a host argument to open a connection to that host.

```
% telnet [-8] [-E] [-F] [-K] [-L] [-N] [-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:

-a	attempts automatic login using your tickets. **telnet** will assume you want the same login id on the remote host unless you explicitly specify another (using **-l**).
-f	forwards a copy of your existing tickets to the remote host, but does not mark them as reforwardable from there. Use of this option overrides any forwarding defaults specified in your machine's configuration files.
-F	forwards a copy of your existing tickets to the remote host, and marks them as re-forwardable from there. Use of this option overrides any forwarding defaults specified in your machine's configuration files.
-k <REALM>	requests tickets 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; i.e., specifies "no auto-login".
-N	turns off ticket forwarding to the remote system. Use of this option overrides any forwarding defaults specified in your machine's configuration files.
-x	(encrypt) turns on encryption. Use of this option overrides any encryption defaults specified in your machine's configuration files.
-X <atype>	disable **atype** type of authentication

Example:

Log in to the remote Kerberized machine fsgi03.fnal.gov, assume your username is different on this machine (**-l qsmith**). Forward tickets and mark them as reforwardable from the target machine (**-F**):

```
% telnet -F -l qsmith fsgi03.fnal.gov
```

13.3 Kerberized rsh

Connect to a specified host, and execute a specified command on that host.

```
% rsh <host> [-l <login_name>] [-n] [-d] [-k <REALM>] [-K]\
 [-f | -F] [-N] [-PN | -PO] [-x] [-X] <command>
```

If **<command>** is left off, **rsh** runs **rlogin**.

The following are the Kerberos options:

-d	turns on socket debugging (via `setsockopt(2)`) on the TCP sockets used for communication with the remote host.
-f	forwards a copy of your existing tickets to the remote host, but does not mark them as reforwardable from there. Use of this option overrides any forwarding defaults specified in your machine's configuration files.
-F	forwards a copy of your existing tickets to the remote host, and marks them as re-forwardable from there. Use of this option overrides any forwarding defaults specified in your machine's configuration files.
-k <REALM>	requests tickets in the specified realm, which may be different from the one the system would use by default.
-K	turns off TCP keepalives (via `setsockopt(2)`) on the TCP socket used for stdin and stdout.
-n	This is not a Kerberos option, but we include it with a usage note. As in non-Kerberized rsh, **-n** redirects input from the special device `/dev/null`. If you put a command **rsh <host> <command>** in the background with **&**, it will stop because only a foreground process can access the tty for input. If you make it **rsh -n <host> <command>**, the **rsh** command does not have the tty open for input at all, so it does not get stopped.
-N	turns off ticket forwarding to the remote system. Use of this option overrides any forwarding defaults specified in your machine's configuration files.
-PN or **-PO**	Explicitly requests New or Old version of the Kerberos "rcmd" protocol. The new protocol avoids many security problems found in the old one, but is not

interoperable with older servers. (An "input/output error" and a closed connection is the most likely result of attempting this combination.) If neither option is specified, some simple heuristics are used to guess which to try.

-x (encrypt) turns ON encryption for the session

Use of this option overrides any encryption defaults specified in your machine's configuration files.

-X turns OFF encryption of the session.

Use of this option overrides any encryption defaults specified in your machine's configuration files.

Example:

Run the command **date** on the remote Kerberized machine fsui03.fnal.gov, and assume your username is different on it (**-l qsmith**). The command doesn't require Kerberos tickets in order to run, nor does it require encryption (**-X** turns it off):

```
% rsh fsgi03.fnal.gov -l qsmith -X date
```

13.4 Kerberized rlogin

Log into a remote host. Kerberos authentication is used in place of the rhosts mechanism to determine if user is authorized to use remote account.

```
% rlogin <rhost> [-e<c>] [-8] [-c] [ -a] [-f] [-F] [-N] \
[-t <termtype>] [-n] [-7] [-noflow] [-d] [-k <REALM>] [-x] \
[-X] [-L] [-PN|-PO] [-4] [-l <username>]
```

The following are the Kerberos options:

-f forwards a copy of your existing tickets to the remote host, but does not mark them as reforwardable from there.

Use of this option overrides any forwarding defaults specified in your machine's configuration files.

-F forwards a copy of your existing tickets to the remote host, and marks them as re-forwardable from there.

Use of this option overrides any forwarding defaults specified in your machine's configuration files.

-k <REALM>	requests tickets in the specified realm, which may be different from the one the system would use by default.
-N	turns off ticket forwarding to the remote system. Use of this option overrides any forwarding defaults specified in your machine's configuration files.
-PN or **-PO**	Explicitly requests New or Old version of the Kerberos "rcmd" protocol. The new protocol avoids many security problems found in the old one, but is not interoperable with older servers. (An "input/output error" and a closed connection is the most likely result of attempting this combination.) If neither option is specified, some simple heuristics are used to guess which to try.
-x	(encrypt) turns ON encryption for the session Use of this option overrides any encryption defaults specified in your machine's configuration files.
-X	turns OFF encryption of the session. Use of this option overrides any encryption defaults specified in your machine's configuration files.
-4	Uses Kerberos v4; default is v5.

Example:

Log into the remote Kerberized machine fsui03.fnal.gov, assume your username is different on it (**-l qsmith**), forward a reforwardable copy of the local Kerberos credentials (**-F**):

```
% rlogin fsgi03.fnal.gov -l qsmith -F
```

13.5 Kerberized FTP

Transfer files to and from a remote host. FTP prompts the user for a command. Type **help** to see a list of commands.

```
% ftp [-v] [-d] [-i] [-n] [-g] [-k <REALM>] [-f] [-x] [-u] [-t]\
 [<host>]
```

The following are the Kerberos options:

-f	requests that your tickets be forwarded to the remote host. (This is necessary if the remote host runs AFS.)

-k <REALM> Ignore this option of the ftp client. It has nothing to do with Kerberos v5. It does work for telnet and the r-commands.

-n no auto-login attempt at initial connection, but still does Kerberos authentication

protect <level> (issued at the **ftp>** prompt) sets the protection level. The level **clear** is "no protection"; **safe** ensures data integrity, and **private** encrypts the data and ensures data integrity.

-u restrains **FTP** from attempting auto-authentication; also disables auto-login.

Note: If your local and remote login names don't match, you can enter your login name for the remote system at the prompt that you get after you issue the **ftp** command.

Examples:

Transfer files from a remote nonKerberized machine www.xyz.org, and assume your username is different on it:

```
% ftp www.xyz.org
```

```
Connected to xyz.org.
220 ...
500 AUTH not understood.
KERBEROS_V4 rejected as an authentication type
Name (www.xyz.org:aheavey): anneh
331 Password required for anneh.
Password:
230 User anneh logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
-rw-rw-r--   1 batavia  site23       1700 Jan 25 10:52 header_1.GIF
...
ftp> get header_1.GIF
local: header_1.GIF remote: header_1.GIF
200 PORT command successful.
150 Opening BINARY mode data connection for header_1.GIF (1700 bytes).
226 Transfer complete.
1700 bytes received in 0.016 seconds (1e+02 Kbytes/s)
ftp> bye
221 Goodbye.
```

Transfer files from a remote Kerberized machine abc.minos-soudan.org that runs AFS (you must forward credentials, **-f**). Assume your username is different on each machine. Set the protection to "private" in order to encrypt the data and ensure data integrity:

```
% ftp -f abc.minos-soudan.org
```

```
Connected to abc.minos-soudan.org.
...
```

```
220 abc.minos-soudan.org FTP server (Version 5.60) ready.
334 Using authentication type GSSAPI; ADAT must follow
GSSAPI accepted as authentication type
GSSAPI authentication succeeded
Name (abc.minos-soudan.org:aheavey): crluser
232 GSSAPI user aheavey@FNAL.GOV is authorized as crluser
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> private
200 Data channel protection level set to private.
ftp> get lmnop.qrs
...
ftp> bye
221 Goodbye.
```

13.6 Kerberized rcp

Copy files between machines. Each file or directory argument is either a remote file name of the form `remote_host:path` or a local file name/path.

```
% rcp [-p] [-x] [-X] [-k <REALM>] [-D <port>] [-n] [-N] \
 [-c <cache>] [-C <config>] [-PN|-PO] <file1> <file2>
```

or

```
% rcp [-p] [-x] [-X] [-k <REALM>] [-r] [-D <port>] [-F] [-N]\
 [-c <cache>] [-C <config>] [-PN|-PO]  <file> ... <directory>
```

The following are the Kerberos options:

-c <cache>	uses credentials file **<cache>** instead of default
-F	forwards credentials to remote system (This is needed if the other end runs AFS.)
-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.
-N	turns off ticket forwarding to the remote system. Use of this option overrides any forwarding specified in your machine's configuration files.
-PN or **-PO**	Explicitly requests New or Old version of the Kerberos "rcmd" protocol. The new protocol avoids many security problems found in the old one, but is not interoperable with older servers. (An "input/output error" and a closed connection is the most likely result

of attempting this combination.) If neither option is specified, some simple heuristics are used to guess which to try.

-r If any of the source files are directories, copy each subtree rooted at that name; in this case the destination must be a directory.

-x (encrypt) turns on encryption.

-X turns off encryption of the session.

Use of this option overrides any encryption specified in your machine's configuration files.

Examples:

Copy the local files `def.histo` and `ghi.histo` to your home directory on the remote machine jkl.myuniv.edu. Assume the remote machine does not run AFS. Your username is the same on both:

```
% rcp def.histo ghi.histo jkl.myuniv.edu:
```

Copy the local directory `histo` and all subdirectories to your home directory on the remote machine jkl.myuniv.edu. Assume the remote machine does not run AFS. Your username is the same on both:

```
% rcp /path/to/histo jkl.myuniv.edu:
```

Copy all the files from the directory `/path/to/mno` on the remote node pqr.myuniv.edu into the local directory `~stu/vwx` (quote the first argument to prevent filename expansion from occurring on the local machine):

```
% rcp “pqr.myuniv.edu:/path/to/mno/*” ~stu/vwx
```

13.7 Kerberized ssh and slogin

The **ssh** and **slogin** commands are intended to replace **rsh** and **rlogin** (see sections 13.3 *Kerberized rsh* and 13.4 *Kerberized rlogin*) and to provide secure encrypted connections between two untrusted hosts over an insecure network. If the **<command>** argument is left off, **ssh** runs **slogin**.

```
% ssh [-a] [-c idea|blowfish|des|3des|arcfour|none]     \
  [-e <escape_char>] [-i <identity_file>] [-l <login_name>]\
  [-n] [-k] [-V] [-o <option>] [-p <port>] [-q] [-P] [-t] [-v]\
  [-x] [-C] [-g] [-L <port>:<host>:<hostport>] [-R \
  <port>:<host>:<hostport>] <hostname> [<command>]
```

-c Specifies cipher for encrypting connection; not needed if specified in configuration file

-k Disables forwarding of the kerberos tickets. This may also be specified on a per-host basis in the configuration file.

Any Kerberos options would be used within **-o <ssh-options>**.

Examples:

From your local machine, log into the remote node fsgi03.fnal.gov on which your (different) username is qsmith. Respond **yes** if asked if you want to continue:

```
% slogin fsgi03.fnal.gov -l qsmith

   Host key not found from the list of known hosts.
   Are you sure you want to continue connecting (yes/no)?
```

From your local machine, run the **date** command on the remote node fsgi03.fnal.gov, but don't start a session:

```
% ssh fsgi03.fnal.gov -l qsmith date
```

13.8 Kerberized scp

Copy files between hosts on a network, using ssh for data transfer.

```
% scp [-a] [-A] [-q] [-Q] [-p] [-r] [-v] [-B] [-C] [-L] [-1] \
  [-S <path_to_ssh>] [-o <ssh-options>][-P <port>]    \
  [-c idea|blowfish|des|3des|arcfour|none]  [-i <identity>]\
  [[user@host1:]filename1... [user@host2:]filename2
```

Any Kerberos options would be used within **-o <ssh-options>**.

Example:

Log into a Kerberized machine at Fermilab, and pull files from a remote machine, mynode.myuniv.edu. On the remote node the username is qsmith, and on the local node, it's quentins. The user wants to pull a file from mynode.myuniv.edu to his local Fermilab machine:

```
% scp qsmith@mynode.myuniv.edu:/home/qsmith/muonrun47.histo \
  ~quentins/geant4/work/muonhistos
```

Part IV System Administrator's Guide "A": Recommended and Supported Implementations

Chapter 14: *Installing Fermi Kerberos on a UNIX (non-Linux) System*

In this chapter we provide instructions for installing the Fermilab **kerberos** product on a UNIX machine (Linux is treated separately in Chapter 15: *Installing Fermi Kerberos on a Linux System*) and for installing Kerberized **ssh**, as the combination works very well. These products are available from *fnkits.fnal.gov*. We describe how to install them using **UPS/UPD**. The information is valid for all supported flavors of UNIX, namely: SunOS, IRIX and OSF1.

Chapter 15: *Installing Fermi Kerberos on a Linux System*

In this chapter we provide instructions for installing the Fermilab **kerberos** product and Kerberized **ssh** on a RedHat Linux machine. These products are available as **UPS** products from *fnkits.fnal.gov*, and in **RPM** format.

Chapter 17: *Kerberized UNIX System Administration Issues*

In this chapter we discuss some UNIX system administration issues related to the installation of Kerberos software.

Chapter 16: *The Kerberos Configuration File: krb5.conf*

In this chapter we describe the Kerberos configuration file `krb5.conf`.

Chapter 18: *Additional UNIX Sysadmin Information for Off-Site Installations*

In this chapter, we discuss some miscellaneous issues that sysadmins of off-site Kerberos installations should be aware of. Also see Chapter 6: *Logging In from Off-Site*.

Chapter 19: *Installing and Configuring WRQ® Reflection on a Windows System*

In this chapter we describe how to install and configure the **WRQ® Reflection** software on your Windows system (Win 2k, NT4, 95, or 98) in order to access Kerberized machines and optionally encrypt your data transmissions.

Chapter 14: Installing Fermi Kerberos on a UNIX (non-Linux) System

In this chapter we provide instructions for installing the Fermilab **kerberos** product on a UNIX machine (RH Linux is treated separately in Chapter 15: *Installing Fermi Kerberos on a Linux System*[1]) and for installing Kerberized **ssh**, as the combination works very well. These products are available from *fnkits.fnal.gov*. We describe how to install them using **UPS/UPD**[2]. The information is valid for all supported flavors of UNIX, namely: SunOS, IRIX and OSF1.

14.1 Before You Install Kerberos

14.1.1 Obtain a Kerberos Principal

Strictly speaking, you don't need a Kerberos principal to just install the software. It will be difficult to judge your results without one, however. You'll need to get a principal (plus an initial password) to have access to the FNAL.GOV realm. See section 3.1 *Your Kerberos Principal* for information. Use the online *Request Form for Computing Username and Primary Accounts* at `http://computing.fnal.gov/cd/forms/acctreq_form.html`.

14.1.2 Create an Account that Matches your Principal

We strongly recommend that you create an account/login name on the machine that matches the "primary" (the username part) of your user principal. See section C.2 *If your Principal and Login Name do not Match* in Appendix C:

1. The information is also valid for Fermi RedHat Linux, but more options are available for Linux.
2. For documentation on **UPS/UPD**, see `http://computing.fnal.gov/docs/products/ups`. Installing products from *fnkits* is described in Part II of the **UPS/UPD** documentation.

More about Choosing a Principal Name. Note that even if your login name and principal don't match you can still log into your machine at the console after it's Kerberized, as long as your UNIX password is there.

14.1.3 Understand your Installation Options

If you don't wish to maintain the **UPS/UPD** software on your machine, we recommend that you install it temporarily in order to install **ssh** and **kerberos**, and then remove it. Instructions for a temporary **UPS/UPD** install are online at `http://www.fnal.gov/docs/products/ups/ReferenceManual/misc/TemporaryInstall.html`.

If you choose not to use **UPS/UPD**, it will be difficult to install the Fermilab **kerberos** product (unless you install via RPM on RH Linux, discussed in Chapter 15: *Installing Fermi Kerberos on a Linux System*). Instead you can download the MIT Kerberos product in a variety of formats from the Web and install it. See Chapter 20: *Installing Kerberos on a non-Fermi-Supported Linux System*.

14.1.4 Install UPS/UPD (Recommended)

If **UPS/UPD** is not already installed on your machine, go ahead and install it (for instructions, see Part III of the *UPS, UPD and UPP v4 Complete Guide and Reference Manual* at `http://www.fnal.gov/docs/products/ups/ReferenceManual/parts.html#partIII`). If your node is not in the fnal.gov domain, make sure that you first register your node for product distribution using the form at `http://computing.fnal.gov/cd/forms/upd_registration.html`.

14.1.5 Install Kerberized SSH (Recommended)

Using Kerberized **ssh** with the **kerberos** product in fully strengthened mode smoothes out several operations that can cause extra work in a non-ssh installation. Most importantly, ssh can be configured to always provide encrypted connections. Also, you get X11 connection forwarding so that you don't have to set the $DISPLAY variable, and the X11 connections are encrypted.

As of version 1_2_27, the first Kerberized ssh version, the ssh product components no longer reside in the `/usr/local` directory tree. The newer versions get installed in the `/usr/krb5` directory tree, which should be local to individual machines.

If you have ssh-afs installed from a previous version of ssh, you must remove it in order for the Kerberos, ssh and AFS to work together properly. The new Kerberized versions of ssh know how to work with AFS.

If you've already installed kerberos and want to add Kerberized ssh via UPS/UPD, make sure you run **`ups install-sshd kerberos`** after installing ssh, for reasons discussed below. (The Kerberized ssh RPM can be installed either before or after kerberos.)

Why Install SSH First?

Make sure you install ssh BEFORE you install kerberos (and install the latter in fully-strengthened mode). The UPS kerberos installation checks for the sshd configuration file and, if it exists, makes the appropriate modifications to turn off the authentication methods that shouldn't be allowed, i.e., password and RSA hosts.

The ssh installation, on the other hand, only checks whether an sshd configuration already exists. If so, it simply keeps it, and if not, it creates a default one (a more permissive one). So, if you install ssh after kerberos, you end up with a too-permissive-for-kerberos ssh configuration. This can be fixed by running **`ups install-sshd kerberos`** which invokes the part of the kerberos install script which modifies the sshd configuration.

To Install SSH using UPD

1) First, log in as the appropriate user for product installation (usually *products* or *root*).

2) We recommend that you stop sshd prior to the installation (as *root*):

```
% /etc/rc.d/init.d/sshd stop
```

3) Setup **UPD** by running the command:

```
% setup upd
```

4) Next run the **`upd install`** command to retrieve **ssh** from the product server, and set it as "current" in the database:

```
% upd install ssh [v<N_M>] -G -c
```

5) Log out, if necessary, and log in now as *root* (or **`su`** to *root*).

6) Run the following configuration command (on each individual machine, if installing on a cluster):

```
% ups InstallAsRoot ssh
```

7) Note: The ssh installation sets the values of `RhostsRSAAuthentication`, `RSAAuthentication` and `PasswordAuthentication` in `/etc/sshd_config` to "yes".

They must be set to “no”! (`KerberosOrLocalPasswd` must also be “no”.) If you proceed to install kerberos, these values will get set properly. If kerberos is already installed, you must either set the values to “no” by hand, or you can do it by running:

```
% ups install-sshd kerberos
```

8) Restart sshd (as *root*):

```
% /etc/rc.d/init.d/sshd start
```

9) Verify your $PATH is pointing to the right ssh (in case you had an older version of ssh running previously). To test, run the command: **`which ssh`**. It should return `/usr/krb5/bin/ssh`. If not, go into `/usr/bin` and reset the ssh link to `/usr/krb5/bin/ssh`.

Documentation on **ssh** is provided under `http://www.fnal.gov/docs/products/ssh/`.

14.1.6 Do you Need to Allow Incoming Kerberos Connections?

If you plan to log in to your machine over the network and/or offer services, your machine must allow incoming Kerberos connections (including portal mode connections). In this case, you must get a service principal for the host, and one for **FTP** if that is an offered service. These service principal names are of the form `host/<full.node.name>` and `ftp/<full.node.name>` (e.g., `host/mynode.fnal.gov` and `ftp/mynode.fnal.gov`, or for off-site nodes, something like `host/mynode.myuniv.edu` and `ftp/mynode.myuniv.edu`, according to your institution’s domain). We also recommend that you get a fixed IP address.

If you need host and ftp principals, first register yourself in the database of system administrators. Go to *System Administrator Registration* at `http://miscomp.fnal.gov/sysadmindb/` to register.

Before installing **kerberos** on a machine the first time, request the host-specific service principals (plus initial passwords) for that machine, using the form at `http://computing.fnal.gov/cd/forms/extra_kerb_req_form.html`. You will need to provide the full hostname of the machine. Later, you will need to install the hostkeys that you receive; see section 17.10 *Installing Service Host Keys*.

Notes:

- For a machine with two or more active (static) IP addresses or multiple node names, see section 17.13 *Multiple IP Addresses or Node Names*.

- If you are reinstalling **kerberos** on a machine, you should keep the same host and **FTP** principals. If the `krb5.keytab` is not lost, there's nothing you have to do for these principals. If it is lost, contact *compdiv@fnal.gov* to get passwords reset on the principals.

If you don't intend to allow incoming connections, don't request these service principals, and just answer "no" when asked if you have the passwords for them during installation of the **kerberos** product. You can request and install them at a later date, if needed. To do so, log on as *root* and run the command:

```
% ups install-hostkeys kerberos
```

and provide the passwords as prompted.

14.1.7 Synchronize your Machine with Time Server

When using Kerberos, the client and server must be time-synchronized with each other, each in its local time zone. A wrong system clock is the single most common authentication problem (it typically appears as a "preauthentication failed" message). Use the command **`date -u`** to check the date/time that really counts. Kerberos is configured to allow a discrepancy of 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[1].

If your system runs AFS, don't install **xntp** or any other synchronizing software; AFS does its own time synchronization. But beware: AFS doesn't set the hardware clock, so, for example, when daylight savings time starts or ends, your clock may be an hour off. Choose ONLY ONE of the following solutions:

- start **xntp**, let it sync the clock, then turn it off
- see if the **afsd** has a **`-nosettime`** option; if so, set it and run **xntp** to handle the timekeeping instead
- (Linux) make sure the date is correct, then run **`/sbin/hwclock --systohc`** to change the hardware clock to match the system clock (or edit your `crontab` to run the above command at some frequency; e.g., to sync it up once a month, add the line `33 3 3 * * /sbin/hwclock --systohc`)

1. If your node is not in the fnal.gov domain, make sure that you first register your node for product distribution via fnkits using the form at `http://comput-ing.fnal.gov/cd/forms/upd_registration.html`.

14.1.8 Determine Kerberos Access Mode(s)

Before installing you must first determine whether you want **kerberos** configured in fully strengthened mode, in mixed mode (Kerberos plus **ssh**), or in a customized mode.

Fully Strengthened Mode (Kerberos Only)

This mode enables only Kerberized access to the node. This includes Kerberized ssh. It disables *all* non-Kerberized means of accessing the node. This is the mode on-site Kerberized systems are obliged to choose beginning Jan. 1, 2002.

Mixed Mode (Kerberos plus SSH)

This mode enables Kerberized access to this node, does not disable any existing non-Kerberized **ssh** access to the node, but disables *all other* non-Kerberized means of accessing the node. This mode is incompatible with Kerberized **ssh**.

For ON-SITE SYSTEMS, this mode is not in compliance with the Computing Policy, and thus is NOT ALLOWED as of January 1, 2002.

Other

If neither of these configurations applies, read the file `README.INSTALL.DETAILS` which describes all of the possible installation options in detail.

This is recommended only for experts.

14.1.9 Choose Login Program

Secondly, you can choose to use the standard UNIX login program or to install the Kerberos login program[1]. As of September 2001 the installation of Fermi **kerberos** automatically replaces the system login program with the Kerberized version. The Kerberos login program is required for CRYPTOCard support.

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

14.2 Installing Fermi Kerberos using UPS/UPD

The Fermilab **kerberos** product is preconfigured and in general should require no further actions beyond the installation instructions found here (this information has been taken from its `README.INSTALL` file). **kerberos** must be properly installed on each individual node. For more information, or to do a custom install, see the the various README files that come with the product.

1) First, log in as the appropriate user for product installation (usually *products* or *root*).

2) Setup **UPD** by running the command:

```
% setup upd
```

3) Next run the **`upd install`** command to retrieve **kerberos** from the product server, and set it as "current" in the database[1]:

```
% upd install kerberos [v<N_M>] -G -c
```

4) Log out, if necessary, and log in now as *root* (or **`su`** to *root*).

5) Choose the configuration option appropriate to your situation (as described in section 14.1.8 *Determine Kerberos Access Mode(s)*) and issue the corresponding **`ups install`** command (note the **`ups`** in place of the **`upd`**) to complete the installation of the **kerberos** product. You need to include the version (**`v<N_M>`**) only if **kerberos** has not been declared as "current". (See section 17.1 *Alterations Made to your System when Fermi Kerberos is Installed* for information on what changes this portion of the installation makes to your system.)

   a) For fully strengthened mode (required for on-site systems):
   `ups install kerberos [v<N_M>]`

   b) For mixed mode (allowed for off-site systems):
   `ups install-keep-ssh kerberos [v<N_M>]`

   c) For any other configuration, refer to the file
   `README.INSTALL.DETAILS` (recommended for experts only)

6) If you wish to override the standard UNIX login program on the machine with a Kerberized login program as discussed in section 14.1.9 *Choose Login Program*, issue the command:

1. Running the **`upd install`** command just puts the kerberos files in the products area. At this point you can run **`setup kerberos`** and you can get Kerberized network connections or do password maintenance. You cannot yet do any thing requiring a host key.

```
% ups install-login kerberos [v<N_M>]
```

where **v<N_M>** is not needed if the **kerberos** product is chained to "current".

7) If you had installed **kerberos** v0_1 or v0_2 and are now reinstalling **kerberos** on the node, you need to clean out the files which had been copied to `/usr/local` (and are now copied to `/usr/krb5`). To so so, run the command:

```
% ups clean kerberos [v<N_M>]
```

where **v<N_M>** refers to the newly installed version, and is not needed if the new version is chained to "current".

Also, if you are reinstalling, keep the same host and FTP service principals to reuse the identity of the machine.

Chapter 15: Installing Fermi Kerberos on a Linux System

In this chapter we discuss installing the Fermilab **kerberos** product and Kerberized **ssh** on a RedHat Linux or Fermi RedHat Linux machine via **RPM**[1], and point you to installation instructions. These products are also available as **UPS** products from *fnkits.fnal.gov*.

For your reference, the Fermilab Linux pages are online at `http://www-oss.fnal.gov/projects/fermilinux/`. Be aware that FermiLinux comes with kerberized network services turned off.

15.1 Before You Install Kerberos

15.1.1 Choose your Installation Method

Both the RPM and UPS/UPD installation frameworks are available for Kerberos on Linux machines. Both methods perform the installation of all the Fermilab Kerberos tools and configuration settings and satisfy the Fermilab policy requirements, but RPM is the recommended method.

The RPM install leaves the systems in a PAM-aware configuration such that more of the normal RedHat tools function as expected. We recommend this installation for people using either the stock RedHat or the FRHL configuration. The major advantages of this method are seen to be:

1) the potential for automatic updates via the AutoRPM service

2) the closer alignment with stock RH product management tools

3) increased ease of use for non-FNAL/non-UPS/UPD configurations

We recommend the UPS/UPD method only for people running servers in the UPS framework.

15.1.2 Differences between the UPS/UPD and RPM Ker-

1. We describe installation for fully-strengthened mode only; due to details of the PAM configuration, the mixed-mode installation would violate Fermilab policy.

beros Products

Configuration

The UPS product configuration uses a perl script, the RPMs use bash scripts. For the most part all the RPM install scripts immitate what is done during the UPS product install. The perl product in UPS can sometimes interfere with the perl native to RHL and cause problems.

The UPS Kerberos product is designed to be installed interactively, whereas an RPM is designed to be installed without any interaction, except for the `makehostkeys` script which must be run manually after everything else is installed. The `makehostkeys` script creates the `/etc/krb5.keytab` file, which allows Kerberized logins to a machine.

The `/etc/krb.conf` configuration file for RPM currently differs from that for the UPD/UPS Kerberos product, in order to work with the PAMs.

Login Program

The RPM Kerberos login program (`krb5-fermi-login`) authenticates users at the login prompt with their Kerberos passwords, but may prohibit users from starting X windows. We plan to eventually replace this program with PAM modules. For now, we recommend that you install this login program and try it out with the rest of the RPM Kerberos package. If you have any problems starting X windows, just remove the login RPM.

15.1.3 Follow Same Pre-install Steps as for UNIX

Obtain a Kerberos principal	See section 3.1.2 *Requesting a Principal*.
Create an account on the machine that matches your principal	See section 14.1.2 *Create an Account that Matches your Principal*.
Determine if you need to allow incoming Kerberos connections and/or FTP access. If so get a fixed IP and obtain host and service principals and passwords.	See section 14.1.6 *Do you Need to Allow Incoming Kerberos Connections?*.
Synchronize your machine with a time server	See section 14.1.7 *Synchronize your Machine with Time Server*.

15.1.4 Create a Local Account

For individuals who administer their own desktops, we recommend that you create two accounts: one that matches your principal and from which you will authenticate to Kerberos (listed in section 15.1.3 *Follow Same Pre-install Steps as for UNIX*), and a local account for which the username does not match your principal and which will not be used for Kerberos-related activity. The local account is really just a convenience so that you can always access your machine, even if the network is down or you are not able to access the Kerberos servers. For a local account, its password must adhere to the following three conditions:

- the password hash must be stored locally (no NIS, LDAP, etc.),
- the password cannot be used for network access (restrict to `securetty`)
- the password cannot contain your Kerberos password and cannot be not similar to it

15.1.5 PAM and Passwords for Desktop Environment Applications

A number of applications on Linux (e.g., screensaver, graphical login, console login) use local authentication checks via the PAM libraries.

There is no easy way to use kerberos with PAM in the 6.x enviroment. Thus the passwords that you use in your desktop enviroment applications must be different from your Kerberos password.

For FRHL 7.1.1 or RedHat 7.1, the installation instructions include steps to make all of the desktop enviroment applications use your Kerberos password. If you want these passwords to be the same as they were before, skip the last couple of steps as noted on the instructions (below). An alternative to configuring PAM is to allow only text-based logins and use the FNAL `login.krb5`.

The PAM for RedHat 7.3 and 8.0 (also used in FRHL 7.3.1) has been improved, and your Kerberos password is used by default for all local authentication.

15.1.6 SSH and OpenSSH

If you are running Scientific Linux Fermi, the Fermi OpenSSH rpms should be already installed. If you are running some other distribution, look for something compatable at `ftp://linux.fnal.gov/linux/contrib/openssh/`.

15.2 Kerberos and OpenSSH RPM Installation

Log in as *root* to perform the installations. Follow the instructions at `http://www-oss.fnal.gov/projects/fermilinux/common/kerberos.html` for the OS version you have. Descriptions of the Fermi Kerberos and ssh RPMs can also be found on that page.

For more information, or to do a custom install, see the various README files that come with the Fermi ssh and Kerberos products (go to `ftp://ftp.fnal.gov:8021/products/kerberos/`, choose a version, and continue down the branch to find the files).

Chapter 16: The Kerberos Configuration File: krb5.conf

In this chapter we describe the Kerberos configuration file `krb5.conf`.

A `krb5.conf` file must exist in the `/etc` directory on each UNIX node that is running Kerberos. We provide a template for this file in the **krb5conf** product in KITS (under `ftp://ftp.fnal.gov:8021/products/krb5conf/`).

If you install Fermi **kerberos** from KITS using UPS/UPD or RPM for Linux, the **krb5conf** product (and file) gets installed automatically for you. If you obtain Kerberos from another source, you must obtain this file yourself, edit it as necessary, and copy it into the `/etc` directory of your machine.

You may need to update your `krb5.conf` file from time to time as the template in KITS gets updated. New versions are announced on the *kerberos-users@fnal.gov* mailing list.

If you need to change a setting in `krb5.conf` but cannot or don't want to change the file in `/etc`, you can copy `/etc/krb5.conf` to a new file and edit this copy. Then set the environment variable $KRB5_CONFIG to the full name of your copy. Your copy will be honored by client programs such as **kinit** or **rlogin**, but not by programs that need a trusted configuration file, e.g., **ksu** and the service daemons.

16.1 What does krb5.conf Control?

The file consists of several stanzas, each of which controls certain aspects of the installation:

- `[libdefaults]` sets defaults for Kerberos on your system, e.g., default realm, default ticket lifetime
- `[realms]` tells where to find the KDCs for each realm
- `[instancemapping]` maps client principal properly (for things like cron jobs which require a special principal)
- `[domain_realm]` maps domains to realms
- `[logging]` tells Kerberos where and how to log errors

- `[appdefaults]` lists default settings for outgoing Kerberized network connection applications and for incoming portal mode connections

In section 16.4 *krb5.conf.template* we list the template `krb5.conf` file (current as of November '01) with annotations.

16.2 Reinstall krb5conf Using UPD

To reinstall **krb5conf** and thus update your `/etc/krb5.conf` file using UPS/UPD, log in as *root* (or any login id with permissions to write in `/etc`), and run:

```
% upd install krb5conf -G -c
```

Then on all nodes in the cluster (including the original node), run the command:

```
% ups installAsRoot krb5conf
```

Or instead of the **`ups installAsRoot`** command, after running **`upd install`**, you may manually set the SOURCE_FILE environmental variable to point to the `krb5.conf.template` script:

```
% SOURCE_FILE=/path/to/krb5/ups/krb5.conf.template
```

and then invoke the `install` script

```
% /path/to/krb5/ups/install
```

16.3 Obtain krb5conf without Using UPD

If you're not running **UPS/UPD**, go to `ftp://ftp.fnal.gov:8021/products/krb5conf/vx_y/NULL/krb5conf_vx_y_NULL.tar` (where `x_y` is `1_9` as of January 2005). Download and untar the file. Look at the top of the `installAsRoot` script for instructions on how to install it without **UPS**. If you're not running AFS, check to be sure that the `installAsRoot` script changes the following line in `/etc/krb5.conf` to "false":

```
krb5_run_aklog = false
```

16.4 krb5.conf.template

For reference, we provide the `krb5.conf.template` file contents for version v1_9, with some explanations inserted. If you install the **krb5conf** product using UPD, the necessary name substitutions will be made as part of the installation; otherwise, you need to edit this file manually.

`[libdefaults]`

This section sets defaults for Kerberos on your system.

```
ticket_lifetime = 1560
```

There are some implementations of Kerberos that read the above number as seconds, and is equivalent to 26 hours. In MIT-derived code (which Fermi's is), it's read as minutes.

```
default_realm = xMYREALMx
```

The UPD installation process changes `xMYREALMx` to `FNAL.GOV`. (In Kerberos transactions, this `default_realm` is assumed when you mention any principal without its "`@REALM`" part.)

```
checksum_type = 1
ccache_type = 3
default_tgs_enctypes = des-cbc-crc
default_tkt_enctypes = des-cbc-crc
```

`[realms]`

This section lists the realms, and for each the KDCs, admin server (master KDC), the `default_domain` for converting between Kerberos v4 and Kerberos v5 service names, and principal-to-account name matching info.

If and when we cross-authenticate with some other site, each host that wants to initiate connections *to* the other site will have to list that site's realm information here. (We think it won't be necessary for accepting connections *from* that site.)

```
FNAL.GOV = {
        kdc = krb-fnal-1.fnal.gov:88
        kdc = krb-fnal-2.fnal.gov:88
        kdc = krb-fnal-3.fnal.gov:88
        kdc = krb-fnal-4.fnal.gov:88
        kdc = krb-fnal-5.fnal.gov:88
        kdc = krb-fnal-6.fnal.gov:88
        admin_server = krb-fnal-admin.fnal.gov
        master_kdc = krb-fnal-admin.fnal.gov:88
        default_domain = fnal.gov
WIN.FNAL.GOV = {
        kdc = littlebird.win.fnal.gov:88
        kdc = bigbird.win.fnal.gov:88
        default_domain = fnal.gov
```

```
        }
...
[instancemapping]
```

This deals with the instance portion of a principal (see *principal* in the *Glossary*). The lines that follow instruct Kerberos to strip a trailing `/cron/*` or `/cms/*` portion of the client principal when generating a Kerberos v4 ticket for the service called `afs`.

```
        afs = {
                cron/* = ""
                cms/* = ""
                afs/* = “”
        }
```

```
[logging]
```

This section tells Kerberos where and how to log errors; through syslog or directly to file.

```
        kdc = SYSLOG:info:local1
        admin-server = SYSLOG:info:local2
        default = SYSLOG:err:auth
```

```
[domain_realm]
```

In this section the domains get mapped to the realms. (This determines the realm in which you need to get a service ticket to log into a Kerberized host in a particular domain.) For individual machines in a domain that need to be mapped to a different realm than the domain as a whole, list each machine separately, mapped to the correct realm. Make your changes in the lower part of this section as noted below.

```
        .fnal.gov = FNAL.GOV
        .cdms-soudan.org = FNAL.GOV
        .minos-soudan.org = FNAL.GOV
        .dhcp.fnal.gov = FNAL.GOV
        ...
        .cs.ttu.edu = FNAL.GOV
        ...
        .harvard.edu = FNAL.GOV
        ... (other “friends and family” by request)
# The whole "top half" is replaced during "ups installAsRoot
krb5conf", so:
# It would probably be a bad idea to change anything on or above
this line

# If you need to add any .domains or hosts, put them here
[domain_realm]
                mojo.lunet.edu = FNAL.GOV
```

`[appdefaults]`

This section lists default application settings (ticket attributes, login parameters, etc.). Each of the applications listed may have additional attributes set (e.g., ticket lifetime, and so on). All of these defaults (or nearly all) can be overridden by a command-line flag. The `krb5.conf` file just sets the defaults for the host. (The ftp client does not look for defaults here; the ftpd ticket lifetime set in the file is invoked for CRYPTOCard FTP access and kcron.)

```
default_lifetime = 7d
retain_ccache = false
```

`retain_ccache` determines whether tickets in a user's ticket cache on a particular host get saved (`true`) or destroyed (`false`) when the user closes his session on that host.

```
autologin = true
forward = true
forwardable = true
```

`forward` should in most cases be set to `true`, in order to forward tickets obtained as "forwardable" to remote hosts by default.

```
renewable = true
encrypt = true
krb5_aklog_path = /usr/krb5/bin/aklog
```

The initial list is for common settings. These values are used by all the applications except when an overriding value is listed for a particular application; see below.

```
telnet = {
}
```

Telnet uses the common settings; no overrides.

```
rcp = {
        forward = false
        encrypt = false
        allow_fallback = true
}
```

Whereas rcp sets two overrides (the first of which is unnecessary) and one additional parameter.

```
rsh = {
        allow_fallback = true
}
rlogin = {
        allow_fallback = false
}
```

```
login = {
        forwardable = true
        krb5_run_aklog = true
        krb5_get_tickets = true
        krb4_get_tickets = false
        krb4_convert = false
}
```

`login` is invoked by telnetd (not telnet) and sshd (not ssh), and may be invoked by the OS for a local (console) login. CRYPTOCard logins use these settings.

```
kinit = {
        forwardable = true
        krb5_run_aklog = true
}

pam = {
        forwardable = true
}

rshd = {
        krb5_run_aklog = true
}

ftpd = {
       krb5_run_aklog = true
       default_lifetime = 10h
}
```

Chapter 17: Kerberized UNIX System Administration Issues

In this chapter we discuss some UNIX system administration issues related to the installation of Kerberos software.

17.1 Alterations Made to your System when Fermi Kerberos is Installed

When you issue one of the **`ups install ...`** commands to complete the installation of the **kerberos** product (e.g., see step 5 of section 14.2 *Installing Fermi Kerberos using UPS/UPD*), the following changes are made to your environment:

- new directory `/usr/krb5` and directories/files underneath are created
- some service (port) definitions are added to `/etc/services` (if not already present). Note that these changes must be made known to the system: for Sun and Linux, make sure `/etc/nsswitch.conf` points to the correct `services` file.
- `/etc/krb5.conf` and `/etc/krb5.keytab` files are added
- `/etc/inetd.conf` is altered to enable Kerberized services and disable non-Kerberized ones, then `SIGHUP` is sent to `inetd`[1]. The default kerberos install preserves pre-existing usage of tcpwrappers on a service-by-service basis.
- if **`ups install-keep-ssh`** isn't chosen, `/etc/sshd_config` is also altered

1. If two **`inetd`** processes are running, some nonKerberized services like rlogin may get handled by a different file than `/etc/inetd.conf` and thus won't be disabled by the **kerberos** installation script as they should be.

17.2 Setting Defaults for Tickets/Applications

The `/etc/krb5.conf` file, described in Chapter 16: *The Kerberos Configuration File: krb5.conf*, contains configuration information needed by the Kerberos V5 library. This includes information describing the default Kerberos realm, and the location of the KDC. You can use it to set default flags for tickets (e.g., forwardable, renewable) and application parameters (e.g., tell application to forward "forwardable" tickets). If your machine is in a domain other than fnal.gov, you'll need to add your domain to the `[domain_realm]` section of the file (see section Chapter 18: *Additional UNIX Sysadmin Information for Off-Site Installations*). For complete information, refer to `http://www.osxfaq.com/man/5/krb5.conf.ws`.

Note that as of January 2001, the current `krb5.conf` file available from KITS does not turn on ticket forwarding by default (for applications that check this file, e.g., **telnet**, **rlogin**, **FTP**, **rsh**). This was changed in response to users' concerns about inadvertently forwarding credentials to an untrustworthy machine. However if the sysadmin turns it on by editing `krb5.conf`, a later update or re-installation will leave that change alone.

17.3 The /etc/hosts File

In the `/etc/hosts` file, the first-listed name for the local system must be the full name, including the domain, and must not be a nickname. The line should be of the form:

```
<IP address> <node>.<domain> <node>
```

E.g.,:

```
<131.225.11.11> mynode.fnal.gov mynode
```

or, depending on your home institution, something like

```
<111.111.11.11> mynode.myuniv.edu mynode
```

Note regarding tcpwrappers: If in `/etc/hosts.deny` there is an entry `ALL : ALL`, then all tcp connections are disabled, unless explicitly enabled in `/etc/hosts.allow`.

17.4 Portal Mode Configuration

A UNIX host running **kerberos v1_0** or later performs the portal function by default when accessed via telnet or FTP from the untrusted realm, unless this mode is specifically disabled. Host and **FTP** principals must exist for the node in order to enable portal mode.

In the `inetd.conf` file (which resides in either `/etc` or `/etc/inet`) you should find a line for **telnet** similar to:

```
telnet stream tcp nowait root /usr/krb5/sbin/telnetd telnetd
-Pa valid
```

And for **FTP**:

```
ftp stream tcp nowait root /usr/krb5/sbin/ftpd ftpd -aP
```

The **`P`** flag in these lines enables portal mode. To disable this mode, remove the **`P`** flag. (This still leaves unencrypted **rsh** and **rlogin** open[1].)

17.5 Register yourself as an Administrator

If you need to allow remote logins to your machine or offer services, you need host and ftp principals for the machine. First register yourself in the database of system administrators. Go to *System Administrator Registration* at `http://miscomp.fnal.gov/sysadmindb/` to register.

17.6 User Accounts and Passwords

17.6.1 User Account Names

Set up each user's account such that the account name (login id) is the same as the person's principal. Otherwise, the user is subject to the problems listed in C.2 *If your Principal and Login Name do not Match*.

1. To eliminate those, take out the klogin service from inetd (leave eklogin) and add an 'e' flag to the kshell service.

17.6.2 Determine if a Particular Principal Exists

If you need to check whether a principal has been created for a user, run the **`kinit`** command with the principal name you want to test. Enter at least one character at the password prompt. The text of the error message will indicate whether the principal exists or not. If the principal exists, it will give a message indicating the password is wrong:

```
% kinit realuser
```

```
Password for realuser@FNAL.GOV: x
kinit: Preauthentication failed while getting initial
credentials
```

If the principal doesn't exist, it will give a "Client not found..." message instead:

```
% kinit nosuchuser
```

```
Password for nosuchuser@FNAL.GOV: x
kinit: Client not found in Kerberos database while getting
initial credentials
```

17.6.3 User Passwords

A Kerberized machine uses the Kerberos login program by default, and that login program accepts Kerberos passwords. Standard UNIX passwords can be used for non-Kerberos-authenticated login at the console. If a user will only access the gateway remotely, the user's account doesn't need a local UNIX password. Using `!!` in the password field for that account in `/etc/shadow` will disable local login, while leaving remote Kerberos login available.

Disable NIS passwords and AFS passwords. There should be no passwords in the yp password files. Standard UNIX passwords can be used for non-Kerberos-authenticated login at the console.

17.6.4 Providing Access to Sensitive Accounts

You as the system administrator can choose to require that users of the *root* account and/or any other sensitive accounts obtain a *root instance* of their principal. This is described in section 9.4 *Using Root Instance of your Principal*.

To allow authorized users to log in directly to a sensitive account via ssh, telnet, rsh, rlogin or ftp, add the person's principal (or the person's */root* principal if you use that method) to the `.k5login` file in the account's home directory (`/root/` for Linux, `/` for the other supported flavors). This file is described in section 9.3 *Account Access by Multiple Users*.

For the *root* account, an alternative is for the authorized user to log in to the machine under his own login id, and use **`ksu`** (instead of **`su`**) to run as *root*. For this, the user must have a forwardable ticket on the machine and his principal must be entered in the root account's `.k5login` or `.k5users` file.

17.7 Non-user Accounts

There are often accounts maintained for file ownership/permissions reasons, and people don't log into these accounts. Typically these accounts have names that don't correspond to user names (e.g., "products"), but it is best to prevent accidental login in case a user's principal matches this account name. To do so, create an empty `.k5login` file in the account's home directory (see section 9.3.1 *The .k5login File*).

17.8 Searching KDC Log Files and the Principal List

The KDC log files and the list of principals are available in AFS space for users who are registered system administrators (see section 17.5 *Register yourself as an Administrator*). If you are a registered system administrator and can't access the KDC logs as described here, please contact *nightwatch@fnal.gov*.[1]

The AFS directory `/afs/fnal.gov/files/data/k5logs` contains various KDC log files and a list of KDC principals. These files can be used by system administrators to understand error messages and to diagnose problems. All the directories referred to below reside under this directory.

The `princ/`, `kdc/`, `log/` and `adm/` directories contain subdirectories for the year and month. The format for the names of these directories is YYYY-MM (e.g., 2001-08). Under each YYYY-MM directory are the actual log files as listed here:

`princ/` contains the weekly list of KDC principals, plus the `diag_user.pl` which allows you to look at yesterday's log file.

`kdc/` contains the daily KDC transaction log files (the transaction records for each KDC are maintained in separate files)

1. If your AFS username is different than your email username, it's likely that the script that built the AFS group that controls access to the KDC log files doesn't have your correct username and you can't access the files.

`log/` contains the daily KDC log files (not much here)

`adm/` contains the daily KDC administration log files

The format for the names of the log files in these directories is `i-krb-<n>.YYYY-MM-DD` (e.g.,. `i-krb-3.2001-08-15`). The meaning of `i-krb-<n>` is the DNS CNAME for a KDC as follows:

i-krb-2	Pilot realm (PILOT.FNAL.GOV) master KDC (alias krb-pilot-1)
i-krb-3	Production realm (FNAL.GOV) master KDC (alias krb-fnal-1)
i-krb-4	FNAL.GOV realm backup KDC (alias krb-fnal-2)
i-krb-5	FNAL realm backup located in D0 (alias krb-fnal-5)
i-krb-6	FNAL realm backup located in CDF (alias krb-fnal-4)
i-krb-7	FNAL realm backup located in BD (alias krb-fnal-3)
i-krb-8	FNAL realm backup located in Soudan (alias krb-fnal-6)

The list of principals under the `princ/` directory is only maintained for the master KDCs, i-krb-2 and i-krb-3. The list of principals includes the attributes for each principal and the expiration dates for the principal and password. Each principal record has comma-separated fields. The format of the records is as follows:

Field number	Field value	Description
1	principal name	full principal name including realm
2	principal expiration	number of days till principal expires, "*" for no expiration, "E" for expired
3	password expiration	same as for principal expiration
4 and beyond	principal attributes	

Most principal attributes are self explanatory such as "DISALLOW_FORWARDABLE". The attribute "DISALLOW_ALL_TIX" is used to disable a principal (except in the case of CRYPTOCard principals[1]).

The KDC transaction log files reside under the `tmp/` and `kdc/` directories:

1. Every user in possession of a CRYPTOCard has an "RB1" instance associated with his or her principal (e.g., username/RB1@FNAL.GOV), which we call a "CRYPTOCard principal". CRYPTOCard principals are given the "DISALLOW_ALL_TIX" attribute because the credentials obtained via a CRYPTOCard are associated with the principal name "username@FNAL.GOV".

`tmp/` contains the real-time KDC transaction log file, plus recent historical transaction log files, so look there to diagnose a problem in real-time.

`kdc/` contains the KDC transaction log files which are at least one day old.

The format of a KDC transaction log file is variable. The `diag_user.pl` perl script in the `tmp/` directory can be used to view the KDC transaction log file for a specific user. For example, if user johndoe is having a problem, try the command (from the directory `tmp/`):

```
% ./diag_user.pl johndoe
```

This command uses **`grep`** to search the current KDC transaction log file `kdc.log` for records with the string `johndoe`. The command will also output specific error records from the log file that pertain to "johndoe" transactions. The error records appear immediately before the transaction record and are missed if the standard **`grep`** command is used. Interpreting these KDC error messages is more art than science(!) For example, here is an error that indicates johndoe is using the wrong password (from the `tmp/` directory)

```
% ./diag_user.pl johndoe

    ERROR->No such file or directory - pa verify failure
    08:30:31=>AS_REQ from fnkerb.fnal.gov(131.225.68.13) PREAUTH_FAILED
    johndoe@FNAL.GOV for krbtgt/FNAL.GOV@FNAL.GOV, Preauthentication failed
```

The "No such file or directory" output means wrong password. The next record containing "Preauthentication failed" is the message user johndoe receives.

There is another version of the `diag_user.pl` tool in the `princ/` directory. If used from there, the tool defaults to looking at yesterday's log file.

17.9 Changing a Machine's Node Name

If you need to change the node name of a Kerberized machine, the host and **FTP** service principals and keys, if any, must also be changed. There is no "rename" function on the principal database, so the old service keys must be deleted and new ones added. Request new service principals `host/<newname>.<domain>` and `ftp/<newname>.<domain>` using the form at `http://computing.fnal.gov/cd/forms/extra_kerb_req_form.html`. When you get them, follow one of these procedures to change your node name.

17.9.1 Using UPS

If you have installed Fermi Kerberos, have **UPS** running and don't mind an interruption, the easiest way to change your node name is to:

1) Change the node name

2) Delete `/etc/krb5.keytab`

3) Run the command: **`ups install-hostkeys kerberos`** and provide the new password(s) when prompted.

17.9.2 Using Kerberos Utilities

If you're not running UPS, you'll need to use the native Kerberos utilities. You can avoid interruptions of service during the name change if you want to prepare in advance.

Once you get your new service principals, follow the procedure outlined in section 17.10 *Installing Service Host Keys* to install the new keys.

Then change the node name, and reboot as necessary. You may delete the old host and **FTP** keys from the `keytab` using the **`ktutil`** command:

```
% /usr/krb5/sbin/ktutil

ktutil:  rkt /etc/krb5.keytab

ktutil:  list

                slot KVNO Principal
                ---- ---- ---------------------------------------------------
                   1    2     host/oldname.domain@FNAL.GOV
                   2    2      ftp/oldname.domain@FNAL.GOV
                   3    2     host/newname.domain@FNAL.GOV
                   4    2      ftp/newname.domain@FNAL.GOV

ktutil:  delent 2

ktutil:  delent 1
```

Note: Delete entry 2 before entry 1 because they all drop down a slot after **`delent`**. Continue:

```
ktutil:  wkt /etc/krb5.keytab.new

ktutil:  quit

% mv /etc/krb5.keytab.new /etc/krb5.keytab
```

Done!

17.10 Installing Service Host Keys

With new host and FTP service principals and their assigned password(s) in hand, log in as *root* and run the `/usr/krb5/config/makehostkeys` script. Or, run the **`kadmin`** command as shown below to install the keys (use appropriate values of **`hostname`**, **`domain`** and **`REALM`**). Note that Kerberos clients append the machine's default realm to the principal names typed in the **`kadmin`** command (**`hostname.domain`**). If the default realm of the machine does not match the realm for which the principals/keys were created, then include the **`-r <REALM>`** option.

```
% /usr/krb5/sbin/kadmin -p host/<hostname.domain> \
  -q "ktadd host/<hostname.domain>" [-r <REALM>]
   Enter password: <type in host principal's password>
% /usr/krb5/sbin/kadmin -p ftp/<hostname.domain> \
  -q "ktadd ftp/<hostname.domain>" [-r <REALM>]
   Enter password: <type in ftp principal's password>
```

17.11 Configuration to allow use of CRYPTO-Card with OpenSSH

Configuring OpenSSH to work with a CRYPTOCard requires the use of a PAM configuration file. The following is a file `/etc/pam.d/sshd` for Linux. (Solaris uses a `/etc/pam.conf` file. The `sshd` lines will be similar to those of the linux `/etc/pam.d/sshd`.)

```
#%PAM-1.0
auth       required     pam_env.so
auth             sufficient      pam_krb5.so  use_opt_hwauth
use_shmem=sshd existing_ticket=true
auth       required     pam_nologin.so
account    required     pam_stack.so service=system-auth
password   required     pam_stack.so service=system-auth
session    required     pam_limits.so
session    required     pam_unix.so
session    sufficient    pam_krb5.so external use_shmem=sshd
session    required     pam_loginuid.so
```

The `use_shmem=sshd existing_ticket=true` line is a continuation of the previous, and in the file, should continue on the same line.

Descriptions:

`use_opt_hwauth` Tells pam to do CRYPTOCards with kerberos.

`use_shmem=sshd` Tells pam to store the ticket from the KDC (in shared memory) during the 'auth' step. This information is then needed during the 'session' step.

`existing_ticket` Tells pam to skip the CRYPTOCard login if the user already has valid credentials.

In addition, the following must be set in the `/etc/ssh/sshd_config` file.

```
PasswordAuthentication no
ChallengeResponseAuthentication yes
```

By default they are set with the `yes` and `no` values swapped. The shown settings are required in order for the system to display the challenge to the user.

17.12 Static IP vs. DHCP Addresses

You can get host and FTP principals for a DHCP-based machine, but your service principals will work only for your nominal node name (e.g., `host/mynode.dhcp.fnal.gov` and `ftp/mynode.dhcp.fnal.gov`). Whenever that name does not resolve to your current IP address, then the service principal is of no use, and you can't authenticate to your host (you can still authenticate yourself to other hosts). A different machine using your node name cannot impersonate your node or steal Kerberized connections intended for your machine, so there's no risk, just inconvenience. However, if you plan to offer reliable services, a static IP address is the better solution.

17.13 Multiple IP Addresses or Node Names

If your machine is configured to have two or more active (static) IP addresses, as long as there's just one node name, you do not need multiple service principals. Just make sure all the IP addresses are listed in DNS. There should be no problems using credentials which have been *forwarded to* such a single-named host.

If you have multiple node names (which are not nicknames), get a host service principal for each name. This will take care of telnet and the r-commands. FTP will not work properly under these circumstances, and credentials forwarded to such a host will be only partly usable.

17.14 Laptops

The feature that sets laptops apart as regards authentication is the fact that they may have different host names and/or IP addresses depending on where they're being used. Install the Kerberos product on it as you would on any other machine, but first decide whether you want a static IP address or if you want to use DHCP.

Chapter 18: Additional UNIX Sysadmin Information for Off-Site Installations

In this chapter, we discuss some miscellaneous issues that sysadmins of off-site Kerberos installations should be aware of. Also see Chapter 6: *Logging In from Off-Site*.

18.1 root access to /usr

The binaries for the **kerberos** product go into `/usr/krb5`, so you don't need access to `/usr/local`. As long as you have *root* access to `/usr`, you can install the product.

18.2 Obtaining the krb5.conf File

We recommend that you use the most recent **UPS** tar file for `krb5.conf` from `ftp://ftp.fnal.gov:8021/products/krb5conf/` (as of this writing, January 2005, this would be `ftp://ftp.fnal.gov:8021/products/krb5conf/v1_9/NULL/krb5conf_v1_9_NULL.tar`). The `krb5.conf` template is updated from time to time. These updates are announced on the *kerberos-announce* mailing list.

If you're not running **UPS**, untar it and look at the top of the `installAsRoot` script for instructions on how to install it without **UPS**. If you're not running AFS, check to be sure that the `installAsRoot` script changes the line in `/etc/krb5.conf` to:

```
krb5_run_aklog = false
```

The `krb5.conf.template` file from the krb5conf product now has lines containing xMYREALMx and xMYNODEx which have to be edited if doing a manual installation. To join the FNAL.GOV production realm, change xMYREALMx to FNAL.GOV and xMYNODEx to the fully-qualified name of host.

18.3 When your Node is in a Different Domain

If your machine is part of a different domain than .fnal.gov, you need to inform applications (e.g., **rsh**, **rlogin**, **telnet**, **FTP**) that it is part of the FNAL.GOV strengthened realm. There are two ways to do this:

The First Way:

In the `[domain_realm]` section of the `/etc/krb5.conf` file on the systems from which you'll be logging on, add lines of the form:

```
<domain> = FNAL.GOV
```

with and without the leading dot, e.g.,

```
.myuniv.edu = FNAL.GOV
 myuniv.edu = FNAL.GOV
```

(You only need to add the domain without the leading dot if the undotted form is the name of some host, which is sometimes the case.) This tells applications that any node in this domain should be assumed to be in the FNAL.GOV realm. Otherwise the host's realm is taken to be the hostname's domain portion converted to upper case.

Since the **krb5conf** product can be updated independently of each new release of the Fermi **kerberos** product, you can send mail to *nightwatch@fnal.gov* to request that your domain be added to the template.

The Second Way:

Whenever you run one of the network connection applications (except FTP), just add **`-k FNAL.GOV`** to the command line, e.g.,:

```
% telnet -x -k FNAL.GOV mynode.myuniv.edu
```

18.4 Connecting from One Off-Site Domain to Another

This concerns connections between two Kerberized machines in the FNAL.GOV strengthened realm where neither is in the fnal.gov domain and they are in different domains from each other, e.g., *mynode.myuniv.edu* and *yournode.youruniv.edu*. In order for one of these Kerberized machines to connect directly to the other via **telnet** or **FTP**, the `/etc/krb5.conf` file

on each must contain the `[domain_realm]` mapping for both off-site domains. This does not concern portal mode where the client machine is unstrengthened.

Chapter 19: Installing and Configuring WRQ® Reflection on a Windows System

In this chapter we describe how to install and configure the **WRQ® Reflection** software on your Windows system (Windows 2000, NT4, XP[1]) in order to authenticate to Kerberos from your Windows desktop, access Kerberized machines, and optionally encrypt your data transmissions. This has been updated for **WRQ® Reflection v10.0.0**.

As of January 2005, WRQ v12 is available. This chapter has not been updated, but you can find the software under \\Pseekits\WRQ.

19.1 Getting Ready

First, verify that you have administrator privileges on the PC. Next, you need to obtain a Kerberos principal and initial password for the FNAL.GOV realm. See section 3.1 *Your Kerberos Principal*.

For PCs running Windows Windows 2000 (also called W2k), XP, NT4, 95 or 98, you need to install two **WRQ® Reflection** software products, **Reflection Kerberos Manager** which runs the **Kerberos Manager** on your PC, and **Reflection X** which is a terminal emulation package similar to **Hummingbird eXceed**, but with Kerberos authentication added.

- Notes:You need a license for the **WRQ® Reflection** software; contact your group's PC administrator or your local W2k/NT server administrator to request one.
- You do not need to remove previous versions of the software before installing these components (except on XP).
- Installing the recommended components of the **WRQ® Reflection v12.0.0** product will consume about 65 MB of disk space.
- It is possible to run **WRQ® Reflection** with other terminal emulation products, however the Computing Division may not support combined installs.

1. The procedures are expected to work also on Windows ME, 98 and 95, although these operating systems have not been tested.

- After installing this software you will still log into your PC the same way as before (e.g., for the W2k Kerberized domain, use your W2k Kerberos password). You will need to provide your FNAL realm principal and Kerberos password only when you run the **Kerberos Manager** or attempt to connect to a Kerberized node over the network from your PC.

- You can configure the **Reflection** software to access nonKerberized nodes, or (as of version 10.0.0) to access **ssh**-only nodes.
- The **Reflection X** portion of the software must be installed before the **Security Components** portion; the automated install takes care of this.

Subscribe to the *wrq-users@fnal.gov* mailing list to receive announcements about this product, to benefit from other users' experiences and to share your own, or to ask questions.

19.2 Automated Installation of WRQ® Reflection v12.0.0

A script is available that performs an automated installation of both portions of the **WRQ® Reflection** software: **Reflection X**, and **Reflection Security Components**. It has been successfully tested on NT4, XP and Win 2000. It may work on Windows ME, 98 and 95 as well, but has not been tested.

The **WRQ® Reflection** v12.0.0 installation script is located at `\\PSeeKits\WRQ\Reflection_12\Install_WRQ.bat`. Read the `README.txt` file.

There is a helpful discussion in the *wrq-users@fnal.gov* list with the subject "Configuring WRQ Reflection X 12.0", that took place in late March 2005. Many useful tips!

Run the `Install_WRQ.bat` file by double-clicking on it. You will need to respond to a series of questions, reproduced here. Answer each with a "y" for "yes", as shown. A series of windows will appear and provide status information.

```
This will install WRQ Reflection 12.0
Do you want to continue [Y,N]?y
Installing WRQ Reflection
Wait for the installation window to disappear, then
Press any key to continue ...
Do you wish to install the default FNAL realms[Y,N]?y
Writing the realm defaults into the Registry
Do you wish to update your services file[Y,N]?y
(If you're upgrading, you'll get a different message here "you already have
a saved copy of the services file..."
Install of Reflection X has completed.
ECHO is off.
Please reboot!
Press any key to continue ...
```

Reboot as instructed. The **Reflection** products will appear in your **START** menu under **PROGRAMS**. The **Kerberos Manager** configuration should reflect the FNAL production realm when done.

19.3 Configuration for Addressless Tickets

If you plan to use Reflection from off-site through a local area router with NAT (for information about NAT, see section 6.5 *Network Address Translation*), you'll need to configure your system to get addressless tickets. To do so,

1) Start WRQ Reflection Kerberos Manager

2) Pull down Configuration > Configure Realms

3) Select FNAL.GOV

4) Click on Properties

5) Choose Realm Defaluts

6) Clear the IP address in the Ticket Address box

7) OK

8) OK

Now when you authenticate (see section 4.6 *Logging In Through WRQ® Reflection Software from Windows*), you'll get an addressless ticket.

Your IP address will appear in the Realm properties, even if you successfully get an addressless ticket. Reflection puts it there in case you want to reset to getting an addressed ticket. To verify that you've gotten an addressless ticket, first get one, then right click on it, and click on properties. The resulting box will have the address cleared if it's addressless.

What if you get the message the message "Kerberos Ticket forwarding failed." and you're sure you checked "Forwardable Ticket" in the configuration? Here are a couple of possible reasons:

1) You actually have an addressed ticket and are going through a NAT router. To check, right click on the ticket, chose properties and check that the ticket is addressless. If it isn't, clear the address from: Kerberos Manager -> Configuration -> Configure Realms -> FNAL.GOV -> Properties -> Realm Defaults, then reauthenticate.

2) Some versions of the software on the host will not accept forwarded tickets. This is sensitive to:

   a) defaults and program versions on the host

b) which one of the WRQ programs you are using: WRQ Reflection for UNIX and Digital (WRQ's terminal emulator) or WRQ Reflection X Manager (pop an xterm).

c) the protocol you are using on your end: Kerberized Telnet, OPENssh, etc.

If you use the X manager with OPENSSH to pop an xterm, make sure you've set ssh to tunnel. This gets through (not around, through) all the problems with NAT, routers etc. You might also try going to a different machine!

19.4 Time Synchronization

Kerberos requires a time sync within five minutes, each machine to its local time zone. Version 10.0.0 of the **WRQ® Reflection** software includes time sync software (versions 9.0.0 and 7.0.2 also did; version 8.0.0 did not).

19.4.1 WRQ® Reflection 10.0.0

- Navigate to **START > PROGRAMS > REFLECTION > UTILITIES > REFLECTION TIMESYNC** to open the **Reflection TimeSync** application.
- Make sure the *Synchronize* tab is selected.
- Under **Time Servers** enter the IP addresses of the default primary and secondary time servers. Use the Fermilab core router 131.225.8.200 as primary and 131.225.17.200 as secondary. Check `NTP` for both.
- Under **Time synchronization**, check `Automatically synchronize time:` and check `Once at system startup`, or if you don't restart your machine frequently, the other option is better (the default 1000 mSec accuracy is fine).
- Again under **Time synchronization**, click `Synchronize Now` to set the system clock and check the time server setting.
- Click **OK**.

19.4.2 WRQ® Reflection 8.0.0

Windows 2000 Host

If you first want to see what your Time service is set to on your Win2K machine, pull up the command prompt, and query the setting by issuing:

```
% net time /querysntp
```

To synchronize the time, issue the following command:

```
% net time /setsntp:131.225.xx.200
```

where **131.225.xx.200** is the IP address of your network gateway at Fermilab. Stop and restart the network time service, by running:

```
% net stop "windows time"
% net start "windows time"
```

Windows NT Host

To synchronize the time on an NT machine, we recommend the MicroSoft utility TIMESERV. This is part of the Windows NT resource kit, and called `Timeserv.exe`. The servers are configured to look at the gateway given in the IP request.

19.5 Configuring WRQ® Reflection Kerberos Manager v12.0.

1) Bring up your Kerberos Manager: Start -> Programs -> WRQ Reflection -> Utilities -> Kerberos Manager

2) Then Configuration -> Configure Realms

3) If the Realm list does **not** say: FNAL.GOV, then add it. The KDC host is krb-fnal-1.fnal.gov. Click OK.

4) Highlight FNAL.GOV -> Properties

5) Choose the KDC tab: Add to your KDC list any missing from krb-fnal-1.fnal.gov through krb-fnal-5.fnal.gov.

6) Kadmin server should be: krb-fnal-admin.fnal.gov

7) Hosts tab: krb-fnal-1.fnal.gov through krb-fnal-5.fnal.gov

8) Realm Defaults tab: Ticket Lifetime, anything greater than 26 hours will give you 26 hours, the realm maximum. Ticket renew time, enter anything greater than 7 days. Pre-Authentication: choose "Encrypted timestamp". Check "Forwardable ticket". For addressless tickets, clear "Ticket Address".

9) Encryption tab: the default of RSA_MD5 in both boxes is fine.

10) Click OK

Back in the Configuration box:

11) User Defaults tab: Default realm = FNAL.GOV. Default storage, chose "File". Click OK.

Back at WRQ Reflection Kerberos Manager:

12) Authenticate (Chose 26 hours, Forwardable, 7 days (should be set at or greater than these already). Click OK.

13) It should ask for your kerberos password. Type it in and a krbtgt/FNAL.GOV@FNAL.GOV ticket should appear in the text box. Right click on it, chose "Properties". Check the flags (Initial, Renewable, Forwardable, Pre-Auth should be set but greyed out). Check the expiration times. For addressless tickets, check that the Addresses text box is empty.

19.6 Configuring WRQ® Reflection X

This section has purposely not been updated for v10.0.0 and following; we encourage you to use the automated install, in which case you don't need to configure the software manually. You DO need to configure individual X connections; see section ???

For version 9 and below:

1) Invoke the **Reflection X Client Manager** using the **START** menu. 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 **NO** if you don't use XDMCP (X Display Manager Control Protocol) to start clients.

3) Now you have the option to let the client wizard create **Reflection X** client files for you. If you say yes, follow the wizard's instructions.

4) At the bottom of the **Reflection X Client Manager** window, click `Never close client starter connection` under the **ADVANCED** button. Also select `KERBERIZED TELNET` as the method.

5) If you logged on as `Administrator`, log off and log back on with your normal userid.

6) 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.

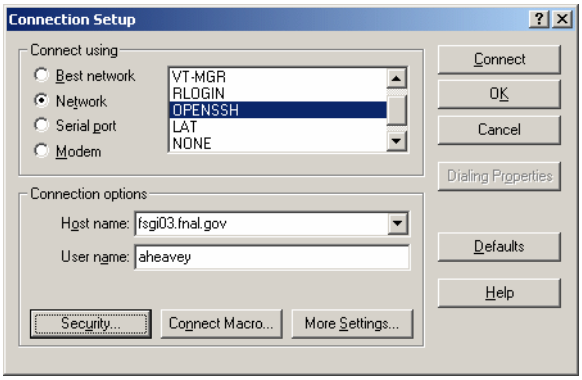
19.7 Configuring WRQ® Reflection OpenSSH Connections

You can define an OpenSSH configuration (profile) specific to each host you need to access, and save each one to a file. To run OpenSSH to a particular host, you just run its corresponding profile (see section 4.6 *Logging In Through WRQ® Reflection Software from Windows*).

19.7.1 For Kerberized Host

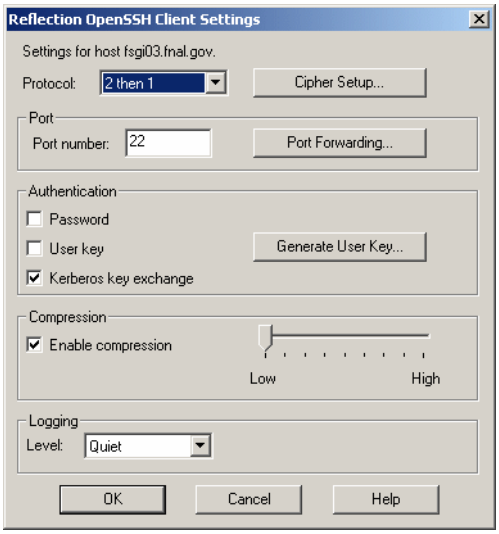
1) To configure the **Reflection OpenSSH** client to access a remote Kerberized system, first open **START > PROGRAMS > REFLECTION > HOST - UNIX AND DIGITAL**.

2) To configure your profile, 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 `OPENSSH` is highlighted in the scroll box:



3) Fill in the **HOST NAME** of your target Kerberos system

4) **Very important!!!** Click **SECURITY**.



5) The default **PROTOCOL** and **PORT NUMBERS** are fine. Change **AUTHENTICATION** to **KERBEROS KEY EXCHANGE**. **COMPRESSION** and **LOGGING LEVEL** settings are optional. Click **OK**.

6) Back on the **CONNECTION SETUP** window, click **CONNECT**.

19.7.2 For nonKerberized Host

Follow the same procedure as in section 19.7.1 *For Kerberized Host*, but on the **REFLECTION OPENSSH CLIENT SETTINGS** window, choose the **AUTHENTICATION** method appropriately for the target system.

19.7.3 Create a Template Configuration

To create a template **OpenSSH** profile, first create and save a model profile for any Kerberized or nonKerberized host, as appropriate, as described in the preceding sections. Pull up that profile, use it to 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 profile, bring up the template, add the desired host name, and save it to a different file with a host-specific name.

19.8 Configuring WRQ® Reflection telnet Connections

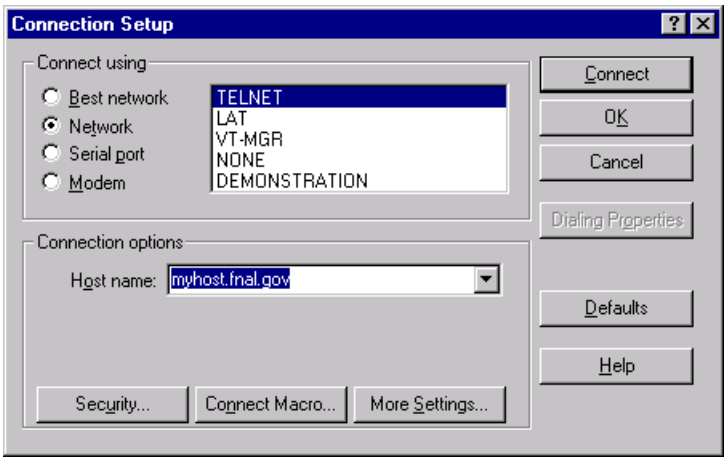
You can define a telnet configuration (profile) specific to each host you need to access, and save each one to a file. To run telnet to a particular host, you just run its corresponding profile (see section 4.6 *Logging In Through WRQ® Reflection Software from Windows*).

19.8.1 For Kerberized Host

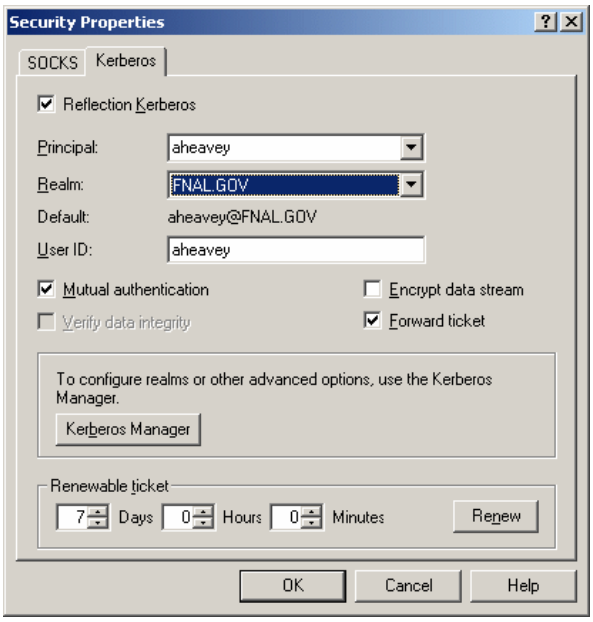
1) To configure the **Reflection telnet** client to access a remote Kerberos system, first open **START > PROGRAMS > REFLECTION > HOST - UNIX AND DIGITAL**.

2) To configure your profile, 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:

Fill in the **HOST NAME** of your target Kerberos system



3) **Very important!!!** Click **SECURITY**.



4) Select the *Kerberos* tab. Check `Reflection Kerberos`.

Principal: Select your FNAL principal name from the pull-down list.

Realm: Assuming the target host is in the FNAL.GOV realm and FNAL.GOV is the default realm set in **Kerberos Manager**, select either (default) or FNAL.GOV.

User ID: If your user id on the target host doesn't match your principal, fill in the user ID.

Mutual authentication should be checked by default; leave it checked.

Check just `Forward ticket`, or check both that and `Encrypt data stream`. If you have forwardable tickets and choose `Forward tickets`, then you can make further connections to other Kerberized machines without having to type your Kerberos password over the net, so you may not need an encrypted connection. (Whenever you authenticate via the **Kerberos Manager**, you will need to check **FORWARDABLE** in order to obtain tickets that can be forwarded by this telnet connection.) Conversely, if you don't forward tickets, then you must make sure not to do anything that involves typing your Kerberos password over the net, even if you check `Encrypt data stream`.

To request a renewable ticket (maximum lifetime at Fermilab defined as seven days), enter a non-zero lifetime value under `Renewable ticket`. Seven days is provided as a default. (Whenever you authenticate via the **Kerberos Manager**, you will need to specify a non-zero **RENEWABLE LIFETIME** in order to get tickets that can be renewed. The lesser of the two renewable lifetimes value is used.)

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. If you don't want to connect now, just click **OK**.

6) Optional: From the **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 host name (use `&h`). For instructions, in the **SETUP > DISPLAY... > OPTIONS** dialog box, click on the **?** (upper right corner, as usual), then on **WINDOW TITLE > DETAILS**.

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 **PROGRAMS\REFLECTIONS** folder.

8) To start a telnet session to the host for which the profile was created, navigate to **START > PROGRAMS > REFLECTION > HOST - UNIX AND DIGITAL**. Pull down the **FILE** menu, select **OPEN**, and double-click the configuration file name. If you haven't yet authenticated, you will need to provide your Kerberos password. It does not go over the net when typed at this point.

19.8.2 For 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** profile, follow the same steps as in section 19.8.1 *For Kerberized Host*, but make sure the host name is a nonKerberized node, and eliminate step (3) which sets the Kerberos security.

19.8.3 Create a Template Configuration

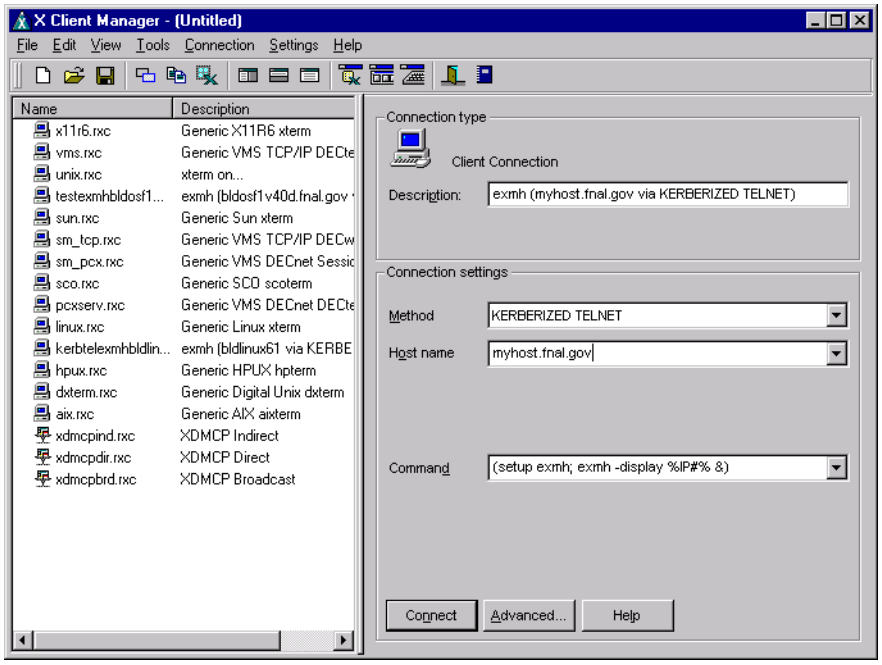
Follow the procedure described in section 19.7.3 *Create a Template Configuration*.

19.9 Configure X Connection to Host

Here we describe how to create a profile to use for making an X connection to a host.

1) Use **START > PROGRAMS > REFLECTION > REFLECTION X** to start the **Reflection X Client Manager**.

2) On the left side, click Client Templates, then Client Startup. Choose the startup template that corresponds to the target OS. This puts the appropriate command in the Command field on the right hand side.



3) 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.)

4) Enter a description in the Description field, and your username on the host in the User Name field.

5) On the right hand side, under **CONNECTION SETTINGS** pull down **METHOD**, and scroll down and select `KERBERIZED TELNET` or `OPENSSH`, depending on what service is installed on the target machine.

If you select Kerberized Telnet:

1) Click the Advanced button.

a) Check the "Show host response" box (it's helpful in case the connection fails).

b) Click Configure Kerberos, and make sure Reflection Kerberos, Mutual Authentication, and Forward Ticket are all checked. Click OK (twice).

2) For either Kerberized Telnet or Openssh, click the Settings button at the bottom of the window.

a) Under Category on the left of the X Settings window, click Security.

b) On the right, click Host Based security for "Security Mode", and Refuse Connection for "If client cannot be authorized".

c) Edit the host access security file: scroll to the bottom and add the fully-qualified host name for the target machine if it's not there yet. Save it and exit.

d) Ideally you should set Security Mode back to User-based security, but you may have connection problems on some hosts. If you leave it as Host-based security, you're more likely to connect successfully, but the CD security scans may pick you up and ask you to close your session (you can answer No, but you risk being blocked!).

3) 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.)

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

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.

19.9.1 Connect to Host with X Application Startup

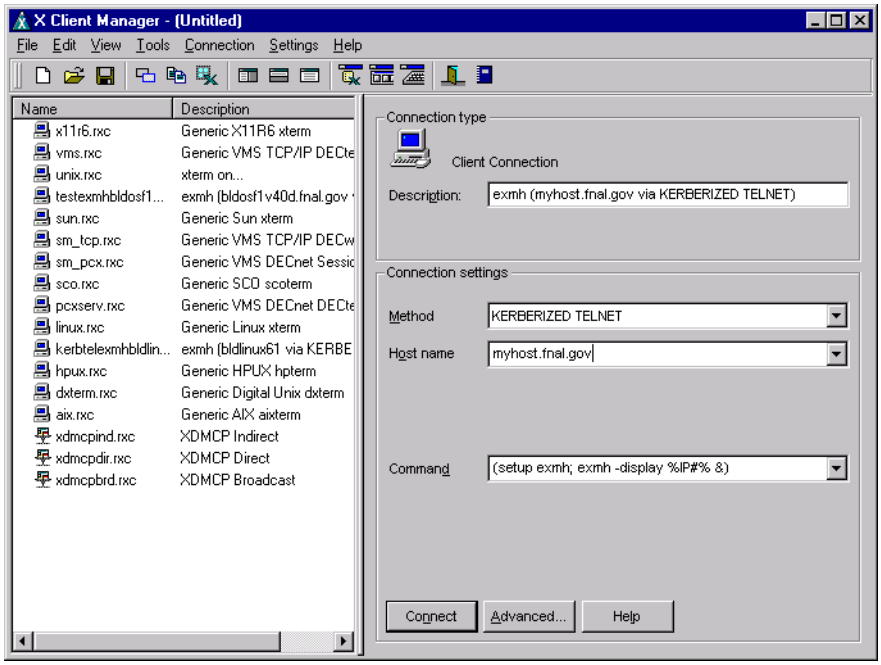
Here we describe how to create a profile to use for connecting to a host and starting a generic X application. (This procedure is somewhat dependent on the target OS.)**Be aware that this method provides unencrypted connections only, so use this only for applications that don't require Kerberos authentication.**

The easiest way to create a profile is to use the X client wizard. Go to **START > PROGRAMS > REFLECTION > WIZARDS > X CLIENTWIZARD** and follow the instructions. To do it manually, follow the instructions that follow here.

1) Use **START > PROGRAMS > REFLECTION > REFLECTION X** to start the **Reflection X Client Manager** if it isn't already running.

2) Use **FILE > NEW...** to open the **NEW CONNECTION** dialog, and select `Client Connection` and click **OK**; *or* (in "Split Window Vertically" view) highlight an existing connection in the left pane of the

X CLIENT MANAGER window to use as a template.



3) On the right hand side, 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...**.

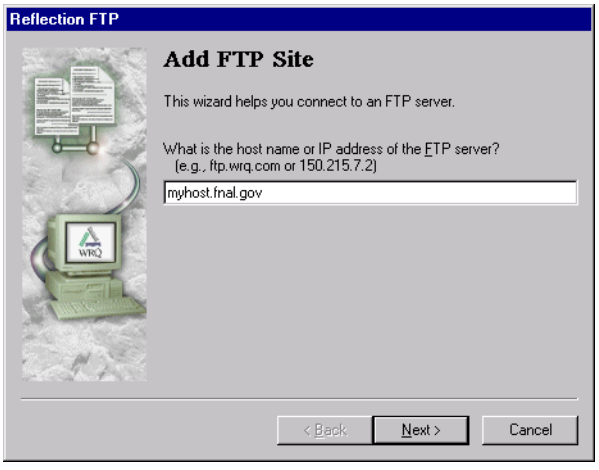
There is extensive on-line help for other problems or applications.

19.10 Configuring WRQ® Reflection FTP Connections

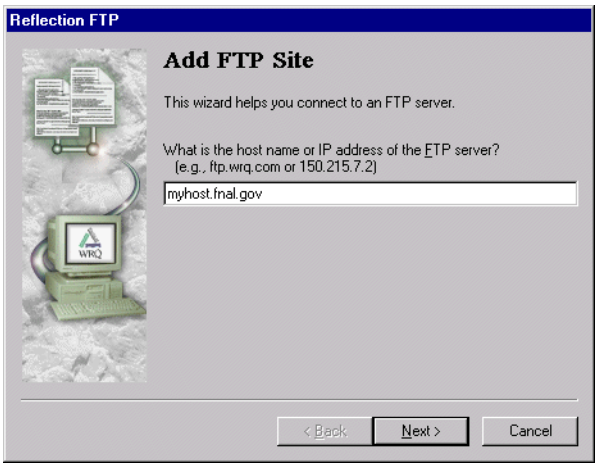
19.10.1 Create a Profile for FTP to Kerberized Host

1) Navigate to **START > PROGRAMS > REFLECTION > FTP CLIENT**.

2) Click **NEW** in the **CONNECT TO FTP CLIENT** screen. This brings you to the FTP wizard. On the **ADD FTP SITE** screen, fill in the name or IP address of the Kerberized host and click **NEXT >**.



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



4) With the **GENERAL** tab selected, click **SECURITY** to get to the **SECURITY PROPERTIES** screen. Select the *Kerberos* tab. The screen is similar to the **SECURITY** screen for configuring telnet connections in section 19.8 *Configuring WRQ® Reflection telnet Connections*.

   Check `Reflection Kerberos`.

   For a target host in the FNAL.GOV realm, enter your FNAL.GOV principal name and select either (default) or FNAL.GOV for the realm.

   If your user id on the target host doesn't match your principal, fill in the user ID.

   `Mutual authentication` and `Verify data integrity` should be checked by default; leave them checked.

   You may check `Encrypt data stream`, but it usually isn't necessary.

   Check `Forward tickets`. Version 10.0.0 is the first version of Reflection's FTP client for which this option is available!

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

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

7) On the **CONNECT** screen, verify the name of the **FTP** host, choose whether you want to connect immediately, then click **FINISH**. Note that in order to connect, the default realm set in **USER PREFERENCES** (see number [2] in section 19.5 *Configuring WRQ® Reflection Kerberos Manager v12.0.*) must be set to the default realm of the target FTP host.

19.10.2 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 profile, follow the same steps as in section 19.10.1 *Create a Profile for FTP to Kerberized Host*, but make sure the host name is a nonKerberized node, and don't bother with **ADVANCED...** in step (3). Instead, click **NEXT >** and continue from step (6).

19.10.3 Edit an FTP Setup

1) Open **START > PROGRAMS > REFLECTION > FTP CLIENT**.

2) In the **CONNECT TO FTP SITE** screen, select a configuration file and click **PROPERTIES**.

Part V System Administrator's Guide "B": Community-Supported Implementations

Chapter 20: *Installing Kerberos on a non-Fermi-Supported Linux System*

In this chapter we discuss Kerberizing a machine running a Linux OS other than FRHL, using the Fermi Kerberos source code from the MIT Kerberos product. The instructions provided here should help non-UPS/UPD Linux users achieve a fully-functional Fermilab Kerberos implementation.

Chapter 21: *Installing MIT Kerberos on Windows, for use with Exceed 7 and FileZilla*

In this chapter we describe how to install and configure the MIT Kerberos software to Kerberize your Hummingbird Exceed 7.0 telnet connections on your Windows system (Win2k, NT4, 95, or 98). The MIT Kerberos software for Windows systems comes with a GUI configuration interface called **Leash32**. Installation of the Kerberos software will allow you to connect to Kerberized machines and encrypt your data transmissions.

Chapter 22: *Installing Heimdal Kerberos for use with Cygwin*

In this chapter we get you started installing the Heimdal Kerberos software in order to Kerberize your network connections from a Windows Cygwin system (Win2k or NT4, or other OS running NTFS). Currently, MIT Kerberos and Fermi Kerberos do not run on Cygwin without tweaking and recompiling. Installation of the Heimdal Kerberos software will allow you to connect to Kerberized machines and encrypt your data transmissions.

Chapter 23: *Installing and Configuring MIT Kerberos on a Macintosh System*

In this chapter we describe how to install and configure the **MIT Kerberos for Macintosh 3.5** software on your Macintosh system in order to access Kerberized machines and encrypt your data transmissions.

Chapter 20: Installing Kerberos on a non-Fermi-Supported Linux System

In this chapter we discuss Kerberizing a machine running a Linux OS other than FRHL, using the Fermi Kerberos source code from the MIT Kerberos product[1]. The instructions provided here should help non-UPS/UPD Linux users achieve a fully-functional Fermilab Kerberos implementation.

The Computing Division does not support these types of installations explicitly, but you can request help on the *kerberos-users@fnal.gov* mailing list (and usually obtain it!).

20.1 Before You Install Kerberos

20.1.1 Obtain a Kerberos Principal

Strictly speaking, you don't need a Kerberos principal to just install the software. It will be difficult to judge your results without one, however. You'll need to get one (plus an initial password) to have access to the FNAL.GOV realm. See section 3.1 *Your Kerberos Principal* for information. Use the online *Request Form for Computing Username and Primary Accounts* at `http://computing.fnal.gov/cd/forms/acctreq_form.html`.

20.1.2 Do you Need to Allow Incoming Kerberos Connections?

For any machine on which services will be offered and which therefore must allow incoming Kerberos connections (including portal mode connections) you must get a service principal for the host, and one for **FTP** if that is an offered service. These service principal names are of the form `host/<full.node.name>` and `ftp/<full.node.name>` (e.g.,

1. Kerberos V5 is available from other sources as well, and these instructions should work for the general case, except of course for MIT-specific comments.

`host/mynode.fnal.gov` and `ftp/mynode.fnal.gov`, or something like `host/mynode.myuniv.edu` and `ftp/mynode.myuniv.edu`, depending on your institution's domain).

Before installing **kerberos** on a machine the first time, request these host-specific service principals (plus initial passwords) for that machine, using the form at `http://computing.fnal.gov/cd/forms/extra_kerb_req_form.html`. You will need to provide the full hostname of the machine.

Notes:

- For a machine with two or more active (static) IP addresses or multiple node names, see section 17.13 *Multiple IP Addresses or Node Names*.
- If you are reinstalling **kerberos** on a machine, you should keep the same host and **FTP** principals. If the `krb5.keytab` is not lost, there's nothing you have to do for these principals. If it is lost, contact *compdiv@fnal.gov* to get password resets on the principals.

If you don't intend to allow incoming connections, don't request these service principals, and just answer "no" when asked if you have the passwords for them during installation of the **kerberos** product. You can request and install them at a later date, if needed (see section 17.10 *Installing Service Host Keys*).

20.1.3 Create an Account that Matches your Principal

We strongly recommend that you create an account/login name on the machine that matches the "primary" (the username part) of your user principal. See section C.2 *If your Principal and Login Name do not Match* under section Appendix C: *More about Choosing a Principal Name*. Note that even if your login name and principal don't match you can still get into your machine (console) after it's Kerberized, as long as your UNIX password is there.

20.1.4 Synchronize your Machine with Time Server

When using Kerberos, the client and server must be time-synchronized with each other, each in its local time zone. A wrong system clock is the single most common authentication problem (it typically appears as a "preauthentication failed" message). 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. But beware: AFS doesn't set the hardware clock, so, for example, when daylight savings time starts or ends, your clock may be an hour off. Choose ONLY ONE of the following solutions:

- start **xntp**, let it sync the clock, then turn it off
- see if the **afsd** has a `-nosettime` option; if so, set it and run **xntp** to handle the timekeeping instead
- (Linux) make sure the date is correct, then run `/sbin/hwclock --systohc` to change the hardware clock to match the system clock (or edit your `crontab` to run the above command at some frequency; e.g., to sync it up once a month, add the line `33 3 3 * * /sbin/hwclock --systohc`)

20.2 Installing MIT Kerberos

1) Bring up the **MIT Kerberos** web page, at URL `web.mit.edu/kerberos/www/`. Select Kerberos V5, the latest release (this section was originally written for 1.2).

2) Follow the links to the MIT Kerberos Distribution page. You'll need to download the Kerberos source. Scroll down to Kerberos V5 Release 1.2 Source Distributions, and download (the latest; shown here for 1.2) `krb5-1.2.x.tar.gz, 5240k`.

3) Login as *root*.

4) Unzip and untar the file, creating the directory `krb5-1.2.x`

5) In the `krb5-1.2.x` directory, run `./configure` (use all defaults).

6) Still in `krb5-1.2.x`, run `make` and `make install`. Now, the software is configured, compiled and installed.

7) Get the latest `krb5.conf` file from Fermi KITS `ftp://ftp.fnal.gov:8021/KITS/GENERIC_UNIX/krb5conf/`. The `krb5.conf.template` file from the krb5conf product now has lines containing xMYREALMx and xMYNODEx which have to be edited. To join the production realm, change xMYREALMx to FNAL.GOV and xMYNODEx to the fully-qualified name of host. At this point, you should be able to authenticate to the Fermilab strengthened realm from your machine.

8) In the `/etc/inetd.conf` file, disable the default FTP, telnet,

rlogin[1], etc., on your machine, and enable the Kerberized versions. Also comment out or delete the lines starting with “shell”, “login”, “rexec” and insert new lines for kshell, klogin and eklogin:

```
## ftp  stream  tcp   nowait  root   /usr/local/sbin/ftpd    ftpd -a
ftp     stream  tcp   nowait  root   /usr/krb5/sbin/ftpd    ftpd -aOP
...
kshell  stream  tcp     nowait  root    /usr/krb5/sbin/kshd kshd -5c
klogin  stream  tcp     nowait  root    /usr/krb5/sbin/klogind klogind -5c
eklogin stream  tcp     nowait  root    /usr/krb5/sbin/klogind klogind -5ce
```

9) Run **kadmin**, and use **ktadd** to add host and FTP principals to the /etc/krb5.keytab file. Run **kadmin** as follows (supplying host and FTP passwords as needed):

```
% /usr/krb5/sbin/kadmin -p host/hostname.domain \
  -q “ktadd host/hostname.domain”
% /usr/krb5/sbin/kadmin -p ftp/hostname.domain \
  -q “ktadd ftp/hostname.domain”
kadmin: ktadd host/hostname.domain
kadmin: ktadd ftp/hostname.domain
```

At this point, you can FTP and telnet *into* your machine, as well as *from* it. Now, it’s time to replace the default login program with the Kerberized version. The typical RedHat login program is PAM-aware, but there is no PAM support in MIT Kerberos v1.2.2. In the RH Linux login file (/etc/pam.d/login) there is a line:

```
session    optional
  /lib/security/pam_console.so
```

The pam_console.so module is responsible for changing the ownership and permissions on the console devices. We recommend that you modify the source for the Kerberos login.krb5 program, krb5-1.2.x/src/appl/bsd/login.c, to be PAM-aware.

10) To do this, **cd** to the the krb5-1.2.x/src/appl/bsd/ directory, make a copy of login.c (to be safe!), copy the patch shown below into a file in this directory (we call it patchfile), and run it:

```
% patch -p0 < patchfile
```

Now the MIT Kerberos login.c will call the pam_console.so that came with RH Linux.

11) To link to the pam and pam_misc libraries, modify the Makefile in krb5-1.2.2/src/appl/bsd. Replace

1. Note that klogind replaces rlogind, and kshd repalaces rshd.

```
LOGINLIBS =
```

with

```
LOGINLIBS = -lpam -lpam_misc
```

The Patch

```
--- login.c.origTue Mar  6 15:13:27 2001
+++ login.cWed Mar  7 15:44:56 2001
@@ -81,6 +81,10 @@

 #include <libpty.h>

+/* begin pam stuff */
+#include <security/pam_appl.h>
+#include <security/pam_misc.h>
+/* end pam stuff */
 #ifdef HAVE_UNISTD_H
 #include <unistd.h>
 #endif
@@ -1004,6 +1008,11 @@
     }
 }

+/* begin pam stuff */
+  int retcode;
+  pam_handle_t *pamh = NULL;
+  struct pam_conv conv = { misc_conv, NULL };
+/* end pam stuff */
 int main(argc, argv)
      int argc;
      char **argv;
@@ -1438,6 +1447,11 @@
     quietlog = access(HUSHLOGIN, F_OK) == 0;
     dolastlog(quietlog, tty);

+/* begin pam stuff */
+    retcode  =  pam_start("login.krb5",  username,  &conv,
&pamh);
+  pam_set_item(pamh, PAM_TTY, tty);
+  pam_open_session(pamh, PAM_SILENT);
+/* end pam stuff */
     if (!hflag && !rflag && !kflag && !Kflag && !eflag) {/*
XXX */
 static struct winsize win = { 0, 0, 0, 0 };
```

```
@@ -2394,6 +2408,10 @@
 #ifdef _IBMR2
     update_ref_count(-1);
 #endif
+/* begin pam stuff */
+   pam_close_session(pamh, PAM_SILENT);
+   pam_end(pamh, PAM_SUCCESS);
+/* end pam stuff */
```

This patch only enables the session module-type. If you add auth, account and/or password module-types, you may compromise the Kerberos security.

20.3 Installing Fermi Kerberos

20.3.1 Download Modified Source from CVS

Instead of installing non-Fermi Kerberos software and enabling the locally-added features of Kerberos, you can download the modified source from the Computing Division CVS repository:

```
% cvs -d :pserver:kpilot@cdcvs.fnal.gov:/cvs/cd co kerberos
```

Read (and be sure you understand!) the `README.*` files in the `ups/` directory. Then configure, compile and install.

20.3.2 Download Tar File from KITS

If you're running a Fermi-supported OS but not UPS/UPD, you can fetch the **kerberos** product tar file from fnkits.fnal.gov, untar it into `/usr/krb5`, then carry out the `/etc/services`, `/etc/inetd.conf` and `/etc/krb5.keytab` steps by hand, and get the `krb5.conf` file from the **krb5conf** product or from another system.

Assuming that you're logged on as *root* and `/usr/krb5/sbin` is in your PATH, the command to do the keytab file is:

```
kadmin -q "ktadd host/<node>.fnal.gov" -p host/<node>.fnal.gov
```

```
kadmin -q "ktadd ftp/<node>.fnal.gov" -p ftp/<node>.fnal.gov
```

and provide the passwords.

Chapter 21: Installing MIT Kerberos on Windows, for use with Exceed 7 and FileZilla

In this chapter we describe how to install and configure the MIT Kerberos software on your Windows system (Win2k, NT4, 95, or 98). This software, when used with the Hummingbird Exceed 7.0 telnet client and the FileZilla FTP client, allows you to authenticate to Kerberos, open Kerberized connections to remote machines, and encrypt your data transmissions. The MIT Kerberos software for Windows systems comes with a GUI called **Leash32**.

Note that while the configuration described in this chapter complies with the Fermilab Policy on Computing and some divisions are recommending and supporting it, it is not formally supported by the Computing Division.

21.1 Getting Ready

21.1.1 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 FNAL.GOV realm. See section 3.1 *Your Kerberos Principal*. Use the online *Request Form for Computing Username and Primary Accounts* at `http://computing.fnal.gov/cd/forms/acctreq_form.html`.

21.1.2 Install Exceed and FileZilla

Exceed 7.0[1]

Exceed is a licensed product. We do not describe the installation process in this document. Versions prior to 7 do not support Kerberos. Version 7.0.0.0 must be patched, since it has a number of severe bugs. You can check the

1. The Exceed version information presented here was taken from the Beams Division documentation at `http://www-bdnew.fnal.gov/networking/`.

Exceed version number by starting Exceed. The startup screen shows 7.0.0.0 for unpatched systems. The correct version shows 7.0.0.12 when starting Exceed, and 7.0.0.5 when starting Exceed host explorer.

Hummingbird Exceed 7.0 FTP connections cannot be Kerberized.

FileZilla 1.93

FileZilla is a small (791k) but powerful freeware FTP client that supports Kerberos (as well as firewalls and proxy connections). It claims to work on virtually all the Windows platforms: W2k/NT/9x/ME/XP. The software includes a site manager to store all your connection details and logins as well as an Explorer-style interface that shows the local and remote folders and can be customized independently. Additional features include keep alive and auto ascii/binary transfer.

Download the software from `\\Pseekits\DesktopTools\Apps\FileZilla_1.6\FileZilla_1_6setup.exe`. Instructions are provided in the same directory. We do not describe the installation process in this document. However, we want to draw your attention to a couple of configuration issues. Under **EDIT > SETTINGS > CONNECTION >**

- **GSS SUPPORT**: Check `Enable Kerberos GSS support`, and add `FNAL.GOV` to the **GSS ENABLED SERVERS** list (you can remove `mit.edu`).
- **FIREWALL SETTINGS**: Check `Passive Mode`

21.1.3 Caveats

Although it appears that you can use **Leash32** to configure Kerberos for multiple realms, we have only gotten this software to work reliably when configured for accessing a single realm.

As mentioned above, Hummingbird Exceed 7.0 FTP connections cannot be Kerberized; use FileZilla's FTP client.

21.2 Installing Kerberos

1) Log into an account with administrator privileges.

2) Download the Kerberos client software from MIT. First browse to: `http://web.mit.edu/network/kerberos-form.html`.

This brings you to the **MIT Kerberos Distribution Page**. Scroll down to the latest *MIT Kerberos for Windows* and click. Next click on the file listed next to Installer. Save the file to disk. The default location it chooses is `C:\Program Files\Accessories`.

3) Once this file is copied on to your machine, execute it to install the Kerberos program. You will be asked a series of questions, but you can safely use the defaults, and just click through the screens. Checking the time synchronization when prompted is a good idea. The software gets installed under `C:\Program Files\Kerberos` by default.

4) After installing the files, it will ask if it's OK to restart your computer. Say yes.

21.3 Configuring Kerberos using Leash32

1) Log back on to the same account.

2) Create the configuration file `krb5.ini` as listed in section 21.6 *krb5.ini for FNAL.GOV*, and put it in your Kerberos folder. (If you accepted the default installation values, this folder is under `C:\Program Files`.) The `krb5.ini` file is comparable to the `krb5.conf` on UNIX.

3) Find where **Exceed 7** has installed the file `krbv4w32.dll` (should be the Kerberos folder), and delete this file.

4) Navigate to **START** > **PROGRAMS** > **KERBEROS UTILITIES** > **LEASH32**. (**Leash32** is a GUI for your Kerberos client.)

5) On the **LEASH32** window, go to the **OPTIONS** menu and select **KERBEROS PROPERTIES**.

6) Under **TICKET LIFETIME**, choose how long you would like your tickets to last (in minutes). 1500 is a good choice. The rest of the configuration under this heading is done for you.

7) Back on the **LEASH32** window, go to the **OPTIONS** menu and select **KERBEROS V5 PROPERTIES**. Under the *Configuration Options* tab, check **FORWARDABLE** to make your Kerberos tickets forwardable to remote Kerberized hosts. Under the *File Location* tab, check that the configuration file path is correct.

8) Also on the **OPTIONS** menu, select **DESTROY TICKETS/TOKENS ON EXIT**.

21.4 Getting a Ticket

To authenticate locally using the **Leash32** utility, select **GET TICKET(S)** on the **ACTION** menu. You will be required to enter your Kerberos password. Ignore the CRYPTOCard prompt that may follow (press **CANCEL**). You ticket will appear in the **Leash32** window. Click on the Windows Explorer-style plus signs (+) to get details.

Alternatively, you can invoke the command prompt and type **`kinit -5`** to request a ticket. You will be required to enter your Kerberos password. Ignore the CRYPTOCard prompt that may follow (just press **ENTER**). To verify the ticket and its flags, either bring up the **Leash32** window, or type **`klist -f`** at the command prompt.

21.5 Configuring the Exceed 7 Telnet Application

21.5.1 Create a new Telnet Profile for Kerberized Host

You should create one profile for each Kerberized host you wish to access.

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

2) In the **OPEN SESSION** window, click on the icon in the upper right corner (second from right) that has the blue screen inside the box with the yellow stripe over it (Rollover text is: `Create New Profile`). Set the following values:

   a) Profile Name = any name to identify the profile (e.g., target host name)

   b) Profile Type = VT

   c) Connect by = Telnet

   d) Hostname = the fully qualified name or IP address of name of the target host (e.g., myhost.fnal.gov or 131.225.876.54)

3) Back on the **OPEN SESSION** window, right-click on the profile you just created and select **PROPERTIES**.

4) In the **SETTINGS GROUP** area of the session profile, expand the **SECURITY** folder, and select **KERBEROS**.

a) Change the **KERBEROS VERSION** to Kerberos 5 from the pulldown menu.

b) In the **COMMON KERBEROS OPTIONS** field, check both Authentication and Encryption.

c) In the **KERBEROS 5 OPTIONS**, check Forwarding. If your user name on the target machine is different from your principal, enter your user name under Alternate User Name.

d) Click **OK**.

21.5.2 Create a new Telnet Profile for nonKerberized Host

You should create one profile for each host you wish to access.

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

2) In the **OPEN SESSION** window, click on the icon in the upper right corner (second from right) that has the blue screen inside the box with the yellow stripe over it (Rollover text is: `Create New Profile`). Set the following values:

   a) Profile Name = any name to identify the profile (e.g., target host name)

   b) Profile Type = VT

   c) Connect by = Telnet

   d) Hostname = the fully qualified name or IP address of name of the target host (e.g., myhost.fnal.gov or 131.225.876.54)

   e) Click **OK**.

21.5.3 Connect to Kerberized Host using Telnet Profile

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

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

21.5.4 Connect to nonKerberized Host using Telnet Profile

On the **OPEN SESSION** window, with a nonKerberized profile selected, the target host name or IP address should appear in the Host Name window. To connect, click on the **CONNECT** button. You will need to log in normally.

21.6 krb5.ini for FNAL.GOV

Make sure you have tabs in front of the items in each stanza, not a series of spaces.

```
[domain_realm]

        fnal.gov = FNAL.GOV



[libdefaults]

        default_realm = FNAL.GOV

        default_tgs_enctypes = des-cbc-crc

        default_tkt_enctypes = des-cbc-crc

        forwardable = true

        proxiable = true



[login]

        krb4_convert = true

        krb4_get_tickets = true



[realms]

        FNAL.GOV = {
                kdc = krb-fnal-1.fnal.gov:88
```

```
kdc = i-krb-7.fnal.gov:88
kdc = krb-fnal-2.fnal.gov:88
kdc = krb-fnal-3.fnal.gov:88
kdc = krb-fnal-4.fnal.gov:88
kdc = krb-fnal-5.fnal.gov:88
admin_server = krb-fnal-admin.fnal.gov
default_domain = fnal.gov        }
```

Chapter 22: Installing Heimdal Kerberos for use with Cygwin

In this chapter we get you started installing the Heimdal Kerberos software in order to Kerberize your network connections from a Windows Cygwin system (Win2k or NT4, or other OS running NTFS). Currently, MIT Kerberos and Fermi Kerberos do not run on Cygwin without tweaking and recompiling. Installation of the Heimdal Kerberos software will allow you to connect to Kerberized machines and encrypt your data transmissions.

Notes:

- While the configuration described in this chapter complies with the Fermilab Policy on Computing and thus may be used, it is not supported at Fermilab.
- The documentation we are providing on this configuration is cursory.
- Work is being done on getting Fermi **kerberos** to compile under Cygwin. Stay tuned...
- Testing of Heimdal has been minimal.
- The Heimdal distribution includes Kerberized daemons that can be used for Kerberizing a Windows machine. However we restrict our discussion to setting the machine up as a Kerberos client only.

22.1 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 FNAL.GOV realm. See section 3.1 *Your Kerberos Principal*. Use the online *Request Form for Computing Username and Primary Accounts* at `http://computing.fnal.gov/cd/forms/acctreq.html`.

22.2 Install Cygwin

Cygwin runs on Win2K, and on NT using NTFS. This discussion is based on a Win2K install. The full Cygwin installation requires ~ 300 MB of space. This can be reduced by selecting only the tools desired from the installation.

22.2.1 Partial Installation

In order to run the Heimdal kerberos client software, you don't need to install the full Cygwin. The minimum installation for Kerberized telnet and ftp for Windows can be accomplished by downloading six files, all available for download from the URL `ftp://ftp.it.su.se/pub/kerberos/contrib/win32/`. The six necessary files are:

- `cygwin1.dll` (the DLL file necessary to run Cygwin executables under Windows)
- `telnet.exe`
- `rsh.exe`
- `ftp.exe`
- `kinit.exe`
- `kdestroy.exe`

The four executables and the DLL can be put into `C:\WINNT\SYSTEM32`[1] or into a directory of your choice, provided that the client executables can find the DLL file. We recommend that you copy the DLL file to one of the following locations: the same directory as the executables, `C:\WINNT\SYSTEM32`, or to some other directory in the PATH. If you choose a different location, make sure the directory containing the DLL is in your PATH[2] before you try running the programs.

22.2.2 Complete Installation

Cygwin can be installed from: `http://sources.redhat.com/cygwin/`. There is an icon on the upper right of this page that is titled **INSTALL CYGWIN NOW**. Click this icon to download the `setup.exe` program to your hard drive.

Run the `setup.exe` program to begin installation (Sorry, no screen-by-screen details!).

---

1. Assuming that `%SYSTEMROOT%` is `C:\WINNT`.
2. To get to the PATH, navigate to **START > SETTINGS > CONTROL PANEL > SYSTEM > ENVIRONMENT**.

22.3 Install Heimdal Kerberos

The Heimdal distribution of kerberos is available via a binary distribution at: `ftp://ftp.it.su.se/pub/kerberos/contrib/win32/`. The file of interest is `travelkit.zip`. This binary distribution is based on the Heimdal 0.3e source. The current source is 0.4b and is available via a link from the Heimdal page `http://www.pdc.kth.se/heimdal/`. (If you prefer to compile the current source under Cygwin, which requires some tweaking of the source, send a request to *kerberos-users@fnal.gov*.)

To install the `travelkit.zip`:

- Expand the zip file into the `/usr` directory (under Cygwin `/usr` becomes `//c/cygwin/usr`).

  This will populate the `/usr/heimdal` directory as well as drop a sample `krb5.conf` file in the `/usr/etc` directory.
- Remove the sample `krb5.conf` file.
- Obtain a standard Fermi `krb5.conf` (available from KITS as the product **krb5conf**, or just copy from a Kerberized UNIX machine), and copy it to the `/etc` directory.
- Put the `/usr/heimdal/bin` directory in your PATH.

In the `/usr/heimdal/bin` directory you will find the available client tools. There are Kerberized clients for telnet, FTP, rsh and rcp (rlogin is not yet available).

22.4 Using CVS under Cygwin

The Heimdal Kerberized **rsh** allows the Cygwin CVS client to work with Kerberos authentication. Put the Kerberized rsh in your $PATH, and set your CVSROOT variable to the appropriate value, e.g., `cvsuser@cdcvs.fnal.gov:/cvs/cd`. Authenticate to Kerberos, and then, for example, you can execute **`cvs co kerberos`** to get the kerberos source.

Chapter 23: Installing and Configuring MIT Kerberos on a Macintosh System

In this chapter we describe how to install and configure Kerberos for Mac OS X 10 in order to access Kerberized machines and encrypt your data transmissions. If you use an older Macintosh OS, we recommend that you upgrade. In case upgrading is not convenient for you, we still provide instructions for installing the MIT Kerberos 4.0x for pre-OS X Macintosh software.

Computing Division Macintosh Strategy

We quote from the (2001) Computing Division policy on Macintosh support. It is still valid as of January 2005.

"The Macintosh Operating System is no longer a supported operating system from the Computing Division and is not a strategic operating system for future plans...

... Specifically regarding the Strong Authentication realm, the supported access method from Macintoshes will be via the CRYPTOCard. Kerberos clients may be available and used, but there will be no effort expended to select, test or distribute them."

That said, there is some community support for the Macintosh, primarily through *kerberos-users@fnal.gov* and *macusers@fnal.gov*.

23.1 Kerberos on Mac OS X 10

23.1.1 Install and Configure

MIT Kerberos for Macintosh is shipped as part of Mac OS X (as of the OS X 10.1 "Cheetah" update). There is a kit of "extras" for OS X 10.1 and later with some additions to what gets shipped with the OS.

1) Update to OS X 10.3 "Panther" if you have not already done so. It has the best Kerberos support, especially for connecting to Windows services

2) Optionally install MIT's "Kerberos extras for Macintosh". It provides a shortcut in /Applications/Utilities to the GUI ticket manager which is already present in /System/Library/CoreServices, and provides a support library for CFM-based applications such as Fetch and Eudora.

To install it, go to `http://web.mit.edu/macdev/KfM/Common/Documentation/osx-kerberos-extras.html` and scroll down about 1/3 the way to "Where can I get Mac OS X Kerberos Extras?". Click as indicated to download: `http://web.mit.edu/macdev/Download/Mac_OS_X_Kerberos_Extras.dmg`. Once it's on your desktop, open it and run the installer (called Mac OS X Kerberos Extras).

3) Obtain the Fermilab Kerberos configuration file template `krb5.conf.template` from the FNAL ftp server. As an example, if the latest version is v1_9, the URL will be `ftp://ftp.fnal.gov:8021/products/krb5conf/v1_9/NULL/krb5conf_v1_9_NULL/ups/krb5.conf.template`.

Edit the [libdefaults] section of `krb5.conf.template` to contain only:

```
[libdefaults]

default_realm = FNAL.GOV

dns_lookup_realm = TRUE
```

If you commonly work from behind a NAT, as is typical of many cable and DSL internet users, you should also add to the [libdefaults] section:

```
noaddresses = TRUE
```

4) The system expects to find this configuration file in one, and only one, of two places. Check for the existence of either of the following two files. (`/etc` is a private directory, requires sysadmin privileges):

```
/etc/krb5.conf

/Library/Preferences/edu.mit.Kerberos
```

If the second one is the only one there, overwrite it with your edited `krb5.conf.template` file, renaming the file `edu.mit.Kerberos`. In all other cases, delete the second one (if there) and overwrite the first with your edited `krb5.conf.template` file, renaming it `krb5.conf`.

Make sure it only exists in one of the two places!

5) If you only want Kerberos access from your Mac to other services (e.g., to log into Unix and/or connect to Windows file servers) you're done. If you want more, keep reading.

6) For AFS access: Download the latest "Darwin" release of OpenAFS from `http://www.openafs.org/release/latest.html`. Click on Mac OS 10.3, then on `OpenAFS.pkg.tar.gz`. This unzips it, downloads the tar file, and untars it using Stuff-it.

7) Click on the untarred file (pkg file), and an installer pops up; run the installer.

8) Go to `/var/db/openafs/etc/` (requires sysadmin privileges) and edit `ThisCell.sample` such that it contains only the single line:

```
fnal.gov
```

9) Save it as `ThisCell`.

10) Restart your computer.

11) Edit `/etc/sshd_config` and make sure that every non-comment line containing the string "Authentication" (in any combination of upper and lower case!) ends with a "no", except this one:

```
KerberosAuthentication yes
```

In particular, make sure that

```
PasswordAuthentication no
```

Is NOT commented out with a '#'.

12) If your Mac is a DHCP client, make sure it gets a stable hostname when connected: Go to System Preferences, click "Network", choose each network interface in turn that you intend to use (probably "Built-in Ethernet" and "Airport"). For each one, click Configure, go to the TCP/IP tab, and fill in the "DHCP Client ID" box with just your hostname (not the fully qualified name). E.g., let's suppose you've registered in MISNET with the hostname mackinac. Just put mackinac in the box, even though your full domain name is mackinac.dhcp.fnal.gov.

13) Go to `http://computing.fnal.gov/cd/forms/extra_kerb_req_form.html` and request a "host principal" for that name. In the additional info box at the bottom, specify that you do NOT need an ftp principal.

14) When you get email back with an initial host principal password, open a Terminal session (Applications > Utilities > Terminal), and under an administrator account run this command:

```
sudo /usr/sbin/kadmin -p host/mackinac.dhcp.fnal.gov -q
  "ktadd  host/mackinac.dhcp.fnal.gov"
```

Provide the password when prompted -- it can only be used one time. If successful the terminal will display a message to the effect of "Entry for principal host/mackinac.dhcp.fnal.gov ... added to keytab WRFILE:/etc/krb5.keytab."

15) Open System Preferences, pick "Sharing" and with the "Services" tab selecte, click "Remote Login" to enable incoming ssh. Make sure your correct hostname (not the fully qualified name) is in the Computer Name field.

16) Add a `.k5login` file to the home directory of any account to which you want to be able to log in remotely, and include the appropriate principals (full principal with no spaces). This file must be writable only by the account itself and/or root.

Once you have set up Kerberos, you have:

- Kerberos telnet and ssh clients
- A Kerberos ssh server (if you completed steps 8-11)
- Kerberos access to WIN.FNAL.GOV Windows 2000 servers

You will not have Kerberos ftp, rlogin, and rsh.

23.1.2 Kerberized Ftp Client

You may be able to get by with sftp, an ssh subsystem, if the servers you use support it. Otherwise, you can get Fetch, an easy-to-use, full-featured FTP client for the Apple Macintosh. As of this writing (Dec 04), 4.0.3 is the latest version. Download it from `http://www.fetchsoftworks.com/`.

23.1.3 X Client

Download the X Client from the Apple site:
`http://www.apple.com/macosx/features/x11/`.

23.1.4 Authenticate to Kerberos

To authenticate, use either the command line **`kinit`** as you would on a Unix system, or the OS X GUI that will now (after installing the "Extras") be visible as "Kerberos" in the /Applications/Utilities folder.

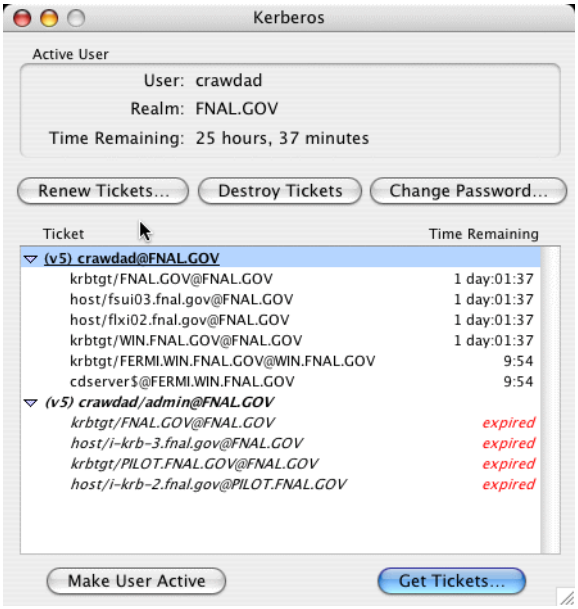
Command Line kinit

Open a terminal window and run the command **`kinit`**. See section 12.1 *kinit*.

GUI

1) Open /Applications/Utilities/Kerberos or, if you did not install Kerberos Extras, point to /System/Library/Core Services/Kerberos.

2) Click Get Tickets.



3) Check that your username is right and the realm is FNAL.GOV. Optionally click Show Options. Enter your Kerberos password and click OK.

4) You'll see your principal name appear and a Time Remaining for your tickets. You can click the triangle to reveal a list of the tickets.

5) Now you are ready to connect to a Unix system with telnet or ssh, or to a Windows 2000 domain file server with the Finder's Command-K function. You can quit the Kerberos GUI application without losing your tickets.

23.1.5 Time Synchronization

If[1] you get the error "KDC reply did not match expectations", your computer's date and time are different than the date and time on the Kerberos server. Should you see this error, make sure your date and time are correct.

On a Macintosh, the Date and Time in the System Preferences or Control Panel has an option for using a network time server. To set the date and time:

1. This text is adapted from MIT IS&T Stock Answer #5897.

- First quit all Kerberos-using applications.
- On Mac OS 10.3 or higher, open System Preferences and click the Date & Time. Check the field "Set Date & Time automatically", then in the field to the right of it, enter the Fermilab core router 131.225.8.200 as the time server. (You could use the secondary router 131.225.17.200.)
- On Mac OS 10.2 or previous, click the Network Time tab and enter the time server to in the field. Click the "Set Time Now" button.

If the problem persists, restart your computer.

23.2 Installing MIT Kerberos for Mac OS 9 and Earlier

Before you go on, remember that the best advice is to upgrade to OS X! If you choose not to, then follow these instructions which apply only to OS 9 and earlier. First, obtain a Kerberos principal and initial password for the FNAL.GOV realm. See section 3.1 *Your Kerberos Principal*.

> This section was originally written for version 3.5 of the MIT Kerberos software for Macintosh. Various versions 4.x have since been made available. Installation can be accomplished by clicking on the "Kerberos for Mac 4.0" installer application. This should install everything into the disk containing your System Folder. This version includes the Kerberos Floating Window (for status), and Kerberos Menu on the menubar (a quick way to create/destroy tickets and to open the Kerberos Control Panel). You will need to reboot probably twice, then, assuming your Kerberos Preferences file is configured properly, you should successfully get a ticket for your principal.

23.2.1 Changes in MIT Kerberos for Macintosh 4.0

See `http://web.mit.edu/macdev/Development/MITKerberos/Common/Documentation/release-4.0.html`. A big change is better OS X support. User interface changes relative to v3.5 include:

- The **KERBEROS CONTROL PANEL** is a changed version of the **KERBEROS MANAGER**.
- The **KERBEROS MENU** on the menu bar shows the status of the active user's TGT and can be used to quickly get/destroy/renew tickets or open the control panel.

- The **KERBEROS CONTROL STRIP** is similar to the **KERBEROS MENU** but a module in the control strip.
- Kerberos Floating Window
- Optional status display of all user's TGTs.

Regarding installation, version 4.0 includes two installer programs, one for OS X and the other for OS 8/9 (supports 8.1 through 9.2.1) but is otherwise much the same as version 3.5.

23.2.2 Download Kerberos from the MIT Web Site

1) Bring up the **MIT Kerberos for Macintosh** web page, at URL `http://web.mit.edu/macdev/www/kerberos.html`.

2) Select *Getting MIT Kerberos for Macintosh*.

3) On this page, look for the paragraph that starts "If you are outside of MIT but still in the US or Canada...". Click on the *download page* link in that paragraph.

4) This brings you to the **Kerberos Distribution Authorization Form**. Answer the three questions, and submit the form to arrive at the download page. (There is a link on this page for Canadian users, which we have not tried or documented.)

5) Click on the link for MIT Kerberos for Macintosh 4.

6) Under the small heading "Binaries and SDKs", click *Binhexed self mounting disk image*.

23.2.3 Items that Appear on your Desktop

You'll find three new items on your desktop once the transfer finishes (This section has not been updated since v3.5; you will find similar things for v4.0.):

- `MIT Kerberos for the Mac` folder
- `MIT_Kerberos_for_Mac_3.5.hqx` file
- `MIT Kerberos for Mac 3.5.smi` file

There will also be a new disk volume from mounting the `.smi` (if the disk is not present, double-click the `.smi` file).

Discard the `hqx` file, and open the `MIT Kerberos for the Mac` folder. This folder contains:

- two subfolders:

- `Mac OS 9 Binaries 3.5`, which contains four sub-subfolders labelled as per their destination folders (the names are of the form `->Into <Foldername>`)
- `Mac OS 9 SDK 3.5`; this is the software development kit and can be ignored.

• one application program **Kerberos for Mac 3.5**

• three links/HTML files: to the MIT Kerberos home page, to the Kerberos for Macintosh Bugs page, and to the KfM 3.5 Release Notes.

• one text file `KfM 3.5 Read Me`, which contains installation instructions

The Kerberos for Macintosh 4.0 disk will have similar contents with the addition of the "Kerberos for Mac OS X 4.0" application and a link "Mac OS X SDK Information". Note that 4.0 supports both Mac OS 8.1 through 9.1 as well as Mac OS X.

23.2.4 Installation Instructions

(This section has not been updated since v3.5; v4.0 is similiar.) We refer you to the `Read Me` file to complete the installation of `MIT Kerberos for the Mac`, but we provide a few clarifications here:

• On the MIT download page, double-click the `Kerberos for Mac 3.5` application to install.

• The downloaded files no longer need to be copied manually into folders under the `System Folder` on your system.

• The `->Into Preferences` folder contains three subfolders. Choose `Kerberos Preferences v5`.

After installation, if you get the error message “preauthentication fails” when you attempt login via the **GET TICKETS** button, it is most likely caused by a password or time-sync error. First verify your password is correct. Then, synchronize your machine with the network time (see section 23.7.3 *Time Synchronization (Pre-OS X 10)*).

23.3 Configuring the Kerberos Software v4 for Mac

23.3.1 The Preferences File

The `Kerberos Preferences` file needs to contain information for Fermilab's strengthened realm(s). Edit the file or just replace the initial contents with that of the `krb5.conf` file from either the **krb5conf** product in KITS or a machine in the Fermilab FNAL.GOV realm (note that pasting text directly from a web browser may cause end-of-line problems). A Fermi-configured Preferences file is now available for download from `http://www.fnal.gov/docs/strongauth/ps/` (see `Kerberos_Preferences.sit` for the StuffIt archive file, or `Kerberos_Preferences.hqx` for the BinHexed (ASCII encoding) version of that file). We reproduce the text of the file here:

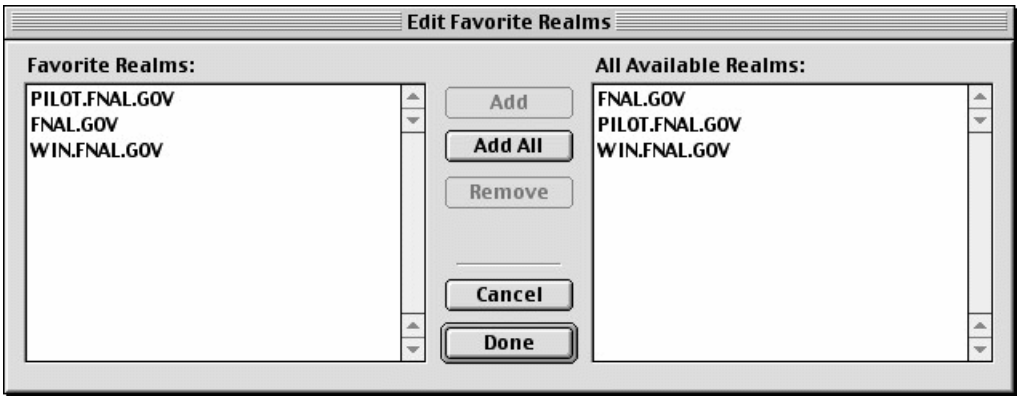
```
[libdefaults]
         default_realm = FNAL.GOV
         ticket_lifetime =1560
         checksum_type = 1
         ccache_type = 2
         default_tkt_enctypes = des-cbc-crc
         default_tgs_enctypes = des-cbc-crc
         noaddresses = true

[realms]
        FNAL.GOV = {
                kdc = krb-fnal-1.fnal.gov:88
                kdc = krb-fnal-2.fnal.gov:88
                kdc = krb-fnal-3.fnal.gov:88
                kdc = krb-fnal-4.fnal.gov:88
                kdc = krb-fnal-5.fnal.gov:88
                admin_server = krb-fnal-admin.fnal.gov
                master_kdc = krb-fnal-admin.fnal.gov:88
                default_domain = fnal.gov
        }
        WIN.FNAL.GOV = {
                kdc = newpckits.fnal.gov:88
                admin_server = newpckits.fnal.gov
                default_domain = fnal.gov
        }

[domain_realm]
        .fnal.gov = FNAL.GOV
```

```
.hep.net = FNAL.GOV
.minos-soudan.org = FNAL.GOV
```

Note: if you have to deal with Network Address Translation (NAT), see section 6.5 *Network Address Translation*.

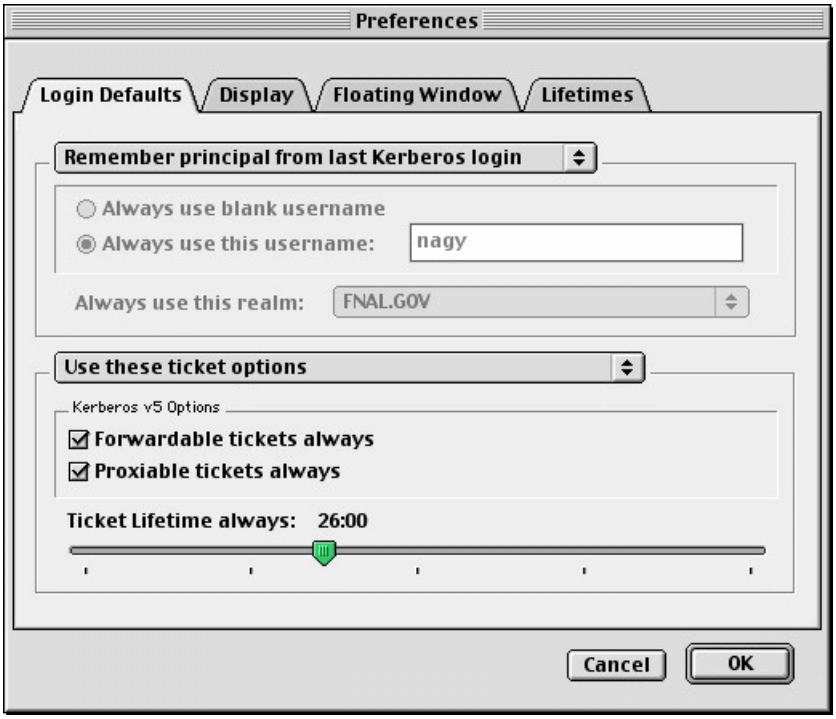
23.3.2 Select Favorite Realms

After modifying the **KERBEROS PREFERENCES**, start the **KERBEROS CONTROL PANEL** and select the **FAVORITE REALMS** item from the **EDIT** menu. Use the dialog box to copy your favorite realms from the right to the left-hand side of the screen.

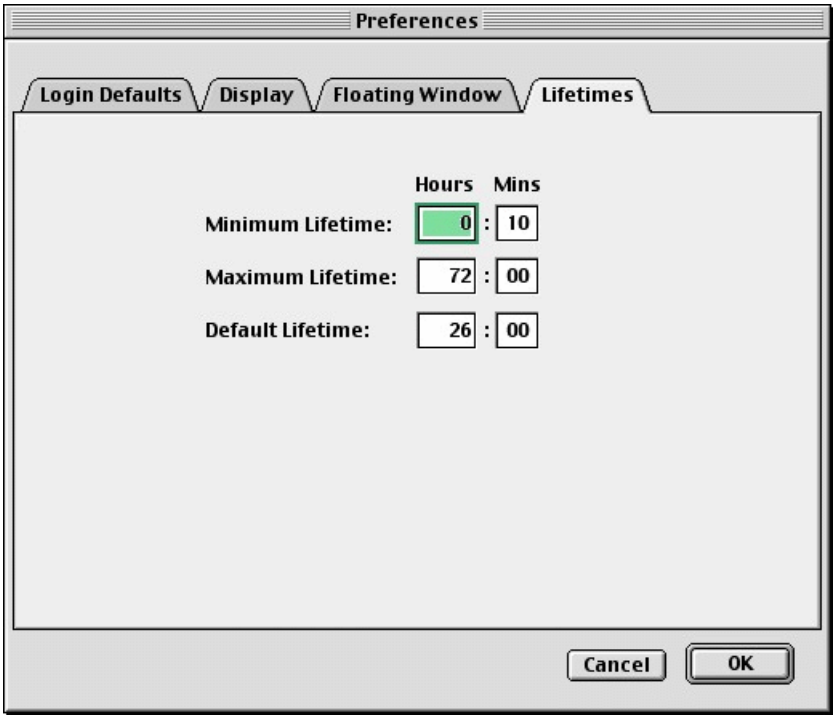


23.3.3 Edit Preferences

Edit your login preferences, and make sure you check **FORWARDABLE TICKETS ALWAYS**:



Edit your ticket lifetime preferences (the KDC limits the ticket lifetime to 26 hours):



23.4 Installing Telnet Client

BetterTelnet and **NiftyTelnet** with Kerberos v5 support are the only **telnet** products that we know of at this time that work on the Macintosh. We document **BetterTelnet** here. You'll need both it and an associated plug-in installed on your machine.

1) Bring up the **MIT Kerberos for Macintosh** web page, at URL `http://web.mit.edu/macdev/www/kerberos.html`. Select *Frequently Asked Questions*.

2) Look for the Q/A that discusses **telnet** (you can search on "BetterTelnet"). Click on the link *BetterTelnet and Kerberos plugin*. This brings you to the FTP site:

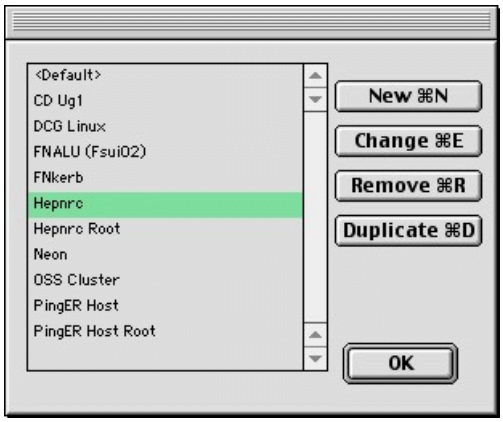
`ftp://ftp.cmf.nrl.navy.mil/pub/chas/MIT_Kerberos_3.5/`.

3) If you don't already have **BetterTelnet** installed, click on `BetterTelnet 2.0f...` and install this software first.

4) Once **BetterTelnet** is installed, download `Telnet_Plugin.bin` from the same **FTP** site and copy it to the **BetterTelnet** folder on your machine.

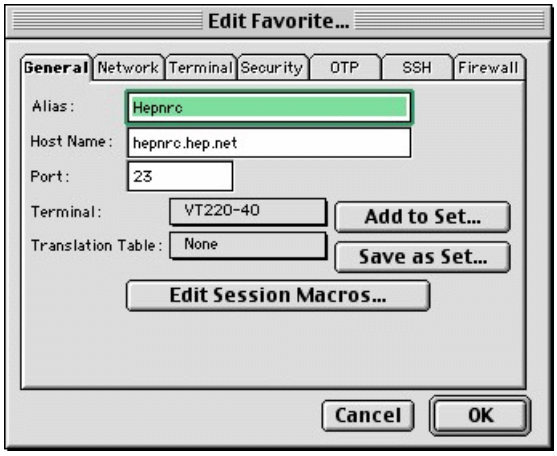
23.5 Configuring Telnet

1) Invoke **BetterTelnet**. On the **FAVORITES** menu, choose **EDIT FAVORITES**. You should create one configuration for each strengthened host you plan to access.

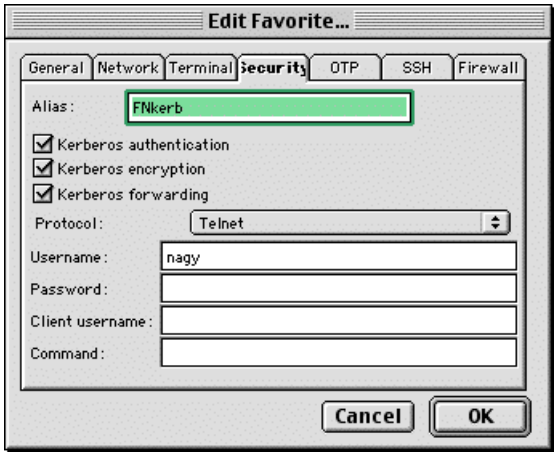


2) To create a new configuration, on the pop-up screen, click **NEW**. Then,

with the **GENERAL** tab selected, type in an **ALIAS** which will be used to identify the host (this can be any string) and the **HOST NAME**.



3) **Very important!!** Change to the **SECURITY** tab, check `Kerberos authentication` and `Kerberos encryption`. `Kerberos forwarding` is recommended. The protocol should be left as `telnet` (the default). Filling in other fields is optional (even if you fill in your Kerberos password, you need to provide it again when you authenticate). Click **OK** to save the configuration.



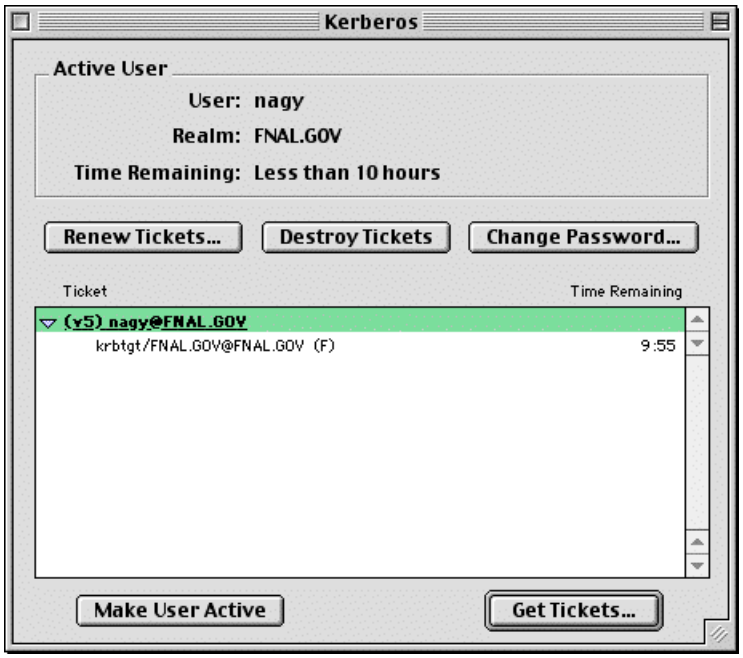
23.6 Kerberized FTP Client

Fetch is an easy-to-use, full-featured FTP client for the Apple Macintosh. As of this writing (Dec 04), 4.0.3 is the latest version. Download it from `http://www.fetchsoftworks.com/`.

23.7 Authenticating to Kerberos

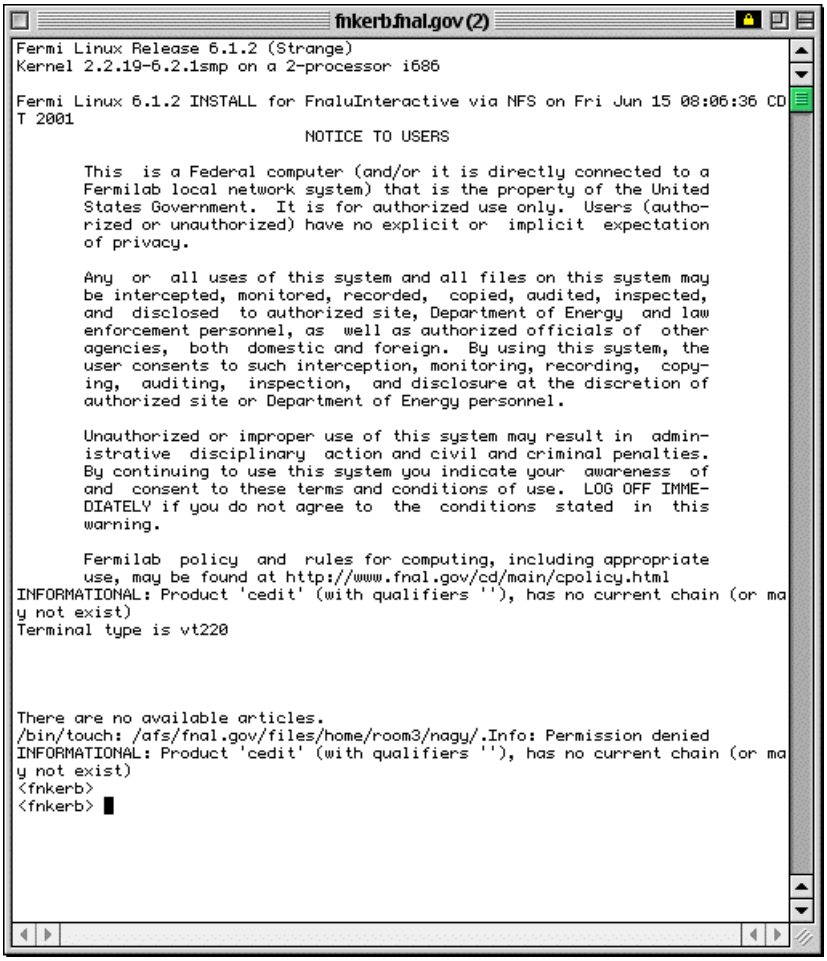
23.7.1 Authenticate via Kerberos Control Panel

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



- Select principal, and click **GET TICKETS**.
- Enter your Kerberos password on the pop-up screen.

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



23.7.2 Authenticate at Login

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

23.7.3 Time Synchronization (Pre-OS X 10)

If[1] you get the error “KDC reply did not match expectations”, your computer’s date and time are different than the date and time on the Kerberos server. Should you see this error, make sure your date and time are correct.

1. This text is adapted from MIT IS&T Stock Answer #5897.

On a Macintosh, the Date and Time in the System Preferences or Control Panel has a setting for using a network time server. For a time server, use the Fermilab core router 131.225.8.200 as primary and 131.225.17.200 as secondary. First quit all Kerberos-using applications. On Mac OS 9, click the 'Network Time Server' button and add the time server to the list. Click the 'Set Time Now' button to sync your computer. If the problem persists, restart your computer.

Part VI Appendices

Appendix A: *Implementation Details of Strong Authentication at Fermilab*

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

Appendix B: *About the Kerberos Network Authentication Service V5*

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

Appendix C: *More about Choosing a Principal Name*

In this appendix, we present information for users who have pre-existing account names and/or an email address at Fermilab, and for whom the guidelines in Chapter 3: *Kerberos Principals and Passwords* are not straightforward to follow.

Appendix A. Implementation Details of Strong Authentication at Fermilab

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

A.1 What is "Strong Authentication"?

A.1.1 Definition

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" rather than transmit a password. 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].

A.2 Goals of Strong Authentication at Fermilab

Fermilab must demonstrate to the DOE that it is implementing a computer security system that exercises tight control over who uses the lab's computers and network (which are owned by the government). The Computing Division has been charged with implementing Strong Authentication to meet Fermilab's obligation.

---

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 (e.g., all of them in one or both directions, until the client gives up).

A primary goal of this effort is to essentially eliminate the transmission of clear text reusable passwords over the network and their storage on local systems. It is impossible to entirely prevent the transmission of clear text passwords, but we are implementing a solution that removes the most common opportunities as well as most of the necessity for typing a password.

Other important goals for us include:

- Providing a single sign-on environment for users
- Providing access to users who have no specialized software (this necessitates an unencrypted mode of access)
- Integrating existing accounts
- Centralizing account maintenance
- Consistently enforcing password policies such as length, quality and lifetime

A.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 : *About the Kerberos Network Authentication Service V5*. In this section we describe the model more generally.

A.3.1 The Realms

The model employed at Fermilab 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 involve passwords are allowed. Weak authentication (standard security) is allowed for local access only, i.e., via the console or locally attached display.

The production realm at Fermilab for UNIX machines is called FNAL.GOV[1], and for Windows 2000, there is FERMI.WIN.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

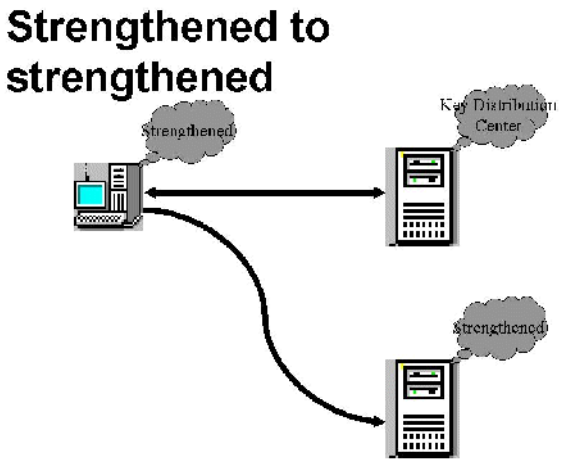
1. As long as the PILOT.FNAL.GOV realm is operating in parallel, the information in this manual applies equally to both realms.

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

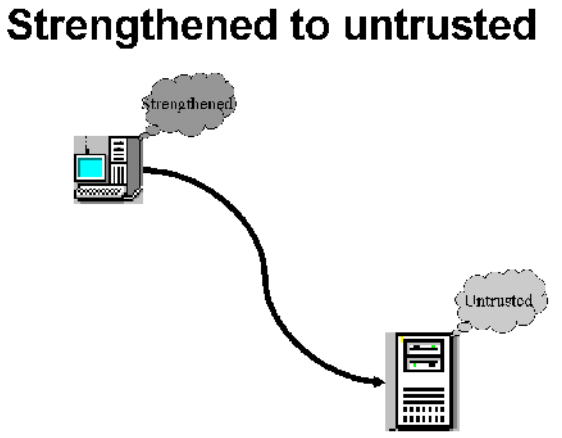
A.3.2 Relationships between the Realms

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

Direct connections between machines in the strengthened realm are allowed

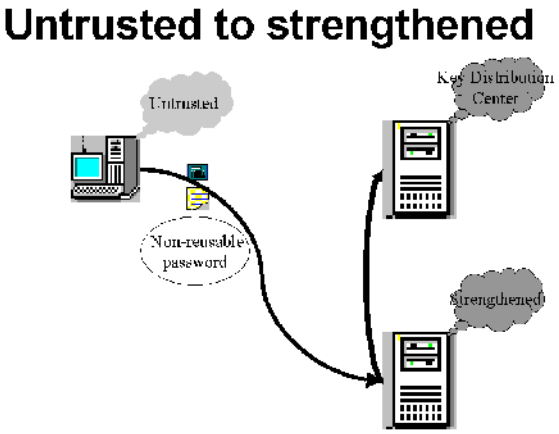


(the Key Distribution Center is involved in providing credentials to the client's machine which can be passed along to access the other strengthened machine).



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

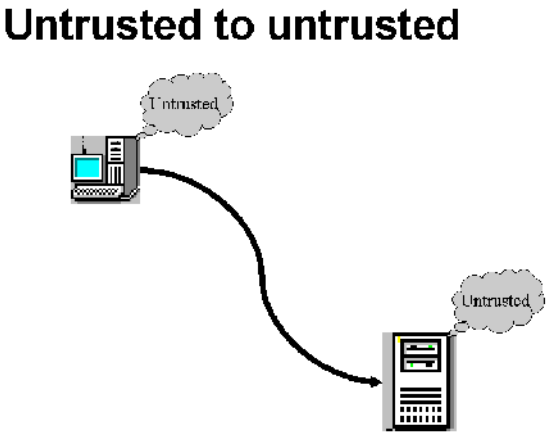
One-time passwords are used for direct connections from the untrusted to the



strengthened realm at Fermilab. Strengthened machines are configured to respond in *portal mode* when requests for access come from machines in the untrusted realm. In portal mode, the strengthened machine acts as a secure gateway into the strengthened realm, requiring a single-use password for authentication. This avoids transmission of reusable clear-text passwords over a potentially unprotected network.

Different programs exist for generating non-reusable passwords, and at Fermilab we currently support CRYPTOCard (described in Appendix : *Using your CRYPTOCard*). No special hardware or software is required on the untrusted system.

For connections between untrusted machines, strong authentication is not



involved. The standard network programs are used in the normal way.

A.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 and Macintosh systems[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 incorporated into the Run II software infrastructure
- 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 PalmPilot)

1. 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.

Appendix B. About the Kerberos Network Authentication Service V5

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

B.1 Introduction to Kerberos V5

B.1.1 Background

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 as connections to other machines are made.

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

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**, **rlogin** and **ssh**. 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.

B.1.2 About Kerberos Authentication

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 assure integrity and/or privacy of their communications with cryptographic methods.

B.1.3 How Secure is Kerberos?

Password Centralization

In Kerberos V5, the password-checking (authentication) happens in a central place for all the machines in the strengthened realm, not on the end systems. End systems need not store any information which can be used to try to guess a password, and they are not involved in password maintenance or quality control. Let's compare this to standard UNIX and (nonKerberized) ssh:

- For standard UNIX passwords, each end system has to store information sufficient to check the password, which is therefore also sufficient to try to *guess* the password. Another problem is that password changes must be repeated on each system or NIS cluster of systems, and quality, aging and reuse prevention are hard to ensure.
- The problems with UNIX passwords are also present with ssh, and ssh presents a couple of additional problems:
  - RSA keys[1] can give access to various accounts, and there's no way to know with certainty who possesses which keys. In the event of a compromise of a private key, there's no mechanism for locating every host on which the corresponding public key appears. The private keys are protected by passphrases which (a) are often no better than a very short password, (b) are sometimes typed in the clear, and (c) are sometimes completely lacking.
  - It is difficult to determine where a password/account resides. Consequently it is much more difficult to control access in a thorough way.
- The problems with UNIX passwords are also present with ssh. There is in addition the issue of RSA keys[2]. RSA keys can give access to various accounts, and there's no way to know with certainty who possesses which keys. In the event of a compromise of a private key, there's no mechanism for locating every host on which the corresponding public key appears. The private keys are protected by passphrases which (a) are often no better than a very short password, (b) are sometimes typed in the clear, and (c) are sometimes completely lacking.

---

1. RSA is an authentication method supported by ssh; it is based on public-key cryptography in which encryption and decryption are done using separate keys, and it is not possible to derive the decryption key from the encryption key. The idea is that each user creates a public/private key pair for authentication purposes. The server knows the public key, and only the user knows the private key.
2. RSA is an authentication method supported by ssh; it is based on public-key cryptography in which encryption and decryption are done using separate keys, and it is not possible to derive the decryption key from the encryption key. The idea is that each user creates a public/private key pair for authentication purposes. The server knows the public key, and only the user knows the private key.

Password Compromise

As noted in section A.2 *Goals of Strong Authentication at Fermilab*, it is impossible to entirely prevent the transmission of clear text passwords, but Kerberos V5 removes the most common opportunities as well as most of the necessity for typing a password. Our implementation of Kerberos allows an unencrypted mode of access in order to accommodate users who have no specialized software of any sort available. This was a requirement we had to meet. The down side is, it means that all users must pay attention to whether their connection is encrypted or not whenever they need to type their Kerberos password.

It is possible to issue your Kerberos password over an unencrypted connection, but this is a violation of common sense and FNAL policy! Please see Appendix : *Encrypted vs. Unencrypted Connections* for instructions on how to avoid doing this.

In the event a Kerberos password is stolen by eavesdropping, it's not impossible for the thief to use it, but there is one serious obstacle: Because a system configured according to our rules will not accept *any* password, correct or incorrect, for a network login (described in section B.4 *The Authentication Process*), the thief must first get onto a system in order to use the stolen password. If the thief installs Kerberos software on his or her own system in order to use the password, we have a record of exactly when and where the password was used.

Furthermore, once into a Fermilab system as a normal user, gaining root access is not necessarily any harder than on other systems, but doing so does not let the perpetrator harvest a password file to crack more passwords, nor exploit any ".rhosts" trusts that may exist. Some valid Kerberos credentials of other users could get stolen, but those are strictly time-limited in value and do not contain information which can be used to guess another password.

B.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

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.

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 a temporary secret encryption key, generated at random by the KDC to be shared between two principals (usually a user and a service). Its validity is limited to the lifetime of an accompanying ticket. The session key is used to authenticate the two principals to each other, possibly multiple times during the ticket lifetime. Its purpose is to limit the use of the permanent key (which for a user is derived from the password) over the network. If encryption or integrity protection of bulk data is required, yet 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[1]. Tickets generally contain the session key, the user and service ids and the client's IP address. Some of 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 ticket and the session key is called a *credential*.

The Authentication Service (AS) issues secret session keys and 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.

B.3 Fermi vs. Standard MIT Kerberos

The Computing Division at Fermilab has taken the MIT Kerberos V5 product and modified it to provide additional features. (Some of these in turn have been incorporated into MIT's releases.) The "Fermi Kerberos" is packaged as

1. 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.

the UPS/UPD product **kerberos** for Fermilab-supported UNIX systems and, recently, also in RPM format for FRHL. It is available in the central product repository, KITS[1]. The most important features that have been added include:

1) CRYPTOCard logins through telnet and FTP.

2) The tools to do authentication of users' cron jobs.

3) Flexible fallback to a non-Kerberized client if you default to encryption "on" but connect to a non-Kerberos server.

4) An FTP client that plays nicely with emacs' efs mode.

Users whose operating systems are not supported at Fermilab, or who don't use UPS/UPD for other reasons, have been installing Kerberos V5 from non-Fermilab sources.

B.4 The Authentication Process

When a user logs in to a strengthened machine, or runs **`kinit`** (described in section 9.2.1 *Obtaining Tickets (Authenticating to Kerberos)*), the Kerberos program transmits some short "behind-the-scenes" messages. First it sends a message, encrypted with (but not containing) your password, to the KDC. This message also contains a timestamp, to confirm that you gave the right password very recently. The KDC attempts to decrypt the message with its copy of your password. If it can do so, and if the timestamp is recent, the KDC believes you know the password, and that you are who you say you are. This portion of the exchange is called *preauthentication* (error messages generated in this portion of the exchange use this word).

Now that the KDC believes you are who you say you are, it makes a Ticket Granting Ticket (TGT), which is sent back to you (also encrypted), and which contains an encryption key for future ticket requests. This gets written in your credential cache, and is the first entry listed when you run **`klist`** (described in section 9.2.2 *Viewing Tickets*).

When you connect over the network from one Kerberized host to another, 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 (see third footnote in section B.2) 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.

1. There are a few related products that get installed automatically by UPD when kerberos is installed.

Kerberized hosts at Fermilab running AFS are configured to obtain AFS tokens automatically at login via the **aklog** program, provided that a Kerberos ticket has been forwarded to the system. If not, the **`kinit`** command obtains both a Kerberos ticket and an AFS token. The **aklog** program authenticates to a cell or directory in AFS.

Appendix C. More about Choosing a Principal Name

In this appendix, we present information for users who have pre-existing account names and/or an email address at Fermilab, and for whom the guidelines in Chapter 3: *Kerberos Principals and Passwords* are not straightforward to follow.

C.1 Guidelines for Choosing a Kerberos Principal

In Chapter 3: *Kerberos Principals and Passwords*, we provided the following guidelines for choosing a Kerberos principal and system login ids:

- New principals should be chosen to be eight or fewer characters, and may include a variety of characters. Please use only lowercase letters and any digits 0 through 9. **Do not use uppercase letters or any special characters in principal names**.
- (New users) Choose one login id (account name) common to all systems at Fermilab that you use, and use this id also as your Kerberos principal name.

If you have pre-existing accounts which make the above guidelines hard to follow, here are further guidelines:

1) If your existing primary system login name (UNIX and/or Windows) is eight or fewer characters, then use this login name for your Kerberos principal. Notes:
   - If your email address and your primary login name do not match, choose the login name as your principal, not your email address. The Computing Division will reserve this login name for you as an email address name. You may continue to use your existing email address on the mail server for a limited time (not yet specified); please transition to the new one. Separate forwards for the two will not be supported.
   - If your primary login name has ever been used as an email address by an individual besides yourself, you must choose a different name for your Kerberos principal. In fact you will need to relinquish the old login name on each system as it becomes Kerberized.

2) If your primary login name is longer than eight characters, then you can choose between the following two options:

- Choose a new name that is eight characters or less, and use it both as your principal and as a new, common login name for all systems. In this case you will have to move or rename your current accounts and files.

- Go ahead and use the long login name as your principal, but be aware that you will very likely have difficulty using some UNIX resources, and the problems may be hard to diagnose. For example, Solaris currently does not accept login names longer than eight characters.

C.2 If your Principal and Login Name do not Match

If your principal does not match your login name, then you need to be aware of the following:

- When connecting over the network (**ssh**, **rlogin**, **rsh**, **telnet**, etc.) you'll always have to give the **`-l <login_name>`** option (or **`login_name@host:`**... for **rcp**), and there will have to be a `.k5login` file in your home directory that lists your principal (see section 9.3.1 *The .k5login File*).
- If using a CRYPTOCard (described in Chapter 5: *Using your CRYPTOCard*), you must initially log in to a system on which your login id matches your principal name. If there are no systems for which this is true, you will not be able to log in with the CRYPTOCard. The portal mode login code assumes that the login name and principal match. For connecting from the initial machine to a second machine with a different login id, see the above note.

Glossary

addressless ticket

A Kerberos ticket that is not bound to a particular IP address, which can then be passed through a NAT-created “firewall”.

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 method

The method used to verify the claimed identity of a principal, e.g., Kerberos V5, CRYPTOCard.

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 ticket-granting tickets (TGTs) and other 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 service.

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 log in 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.

credential

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.

credential cache

Kerberos tickets are stored in a credential cache. The credential cache can be a file with restricted rights, or it can be a persistent memory location.

cross-authentication

This concerns 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 and authorization has been established in the other (e.g., via `.k5login`).

CRYPTOCard

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

ftp

`ftp` is a program to transfer files to and from a remote host.

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 encryption key 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.

kdestroy

The **`kdestroy`** utility destroys the user's active Kerberos credentials (tickets) by writing zeros to the specified credentials cache that contains them, and then deleting the cache.

Kerberized application

A software application that requires or performs Kerberos authentication.

Kerberized machine

A machine on which the Kerberos product has been installed and which requires Kerberos V5 authentication for access.

Kerberized ssh client

An ssh client application that requires or performs Kerberos V5 authentication, and which does not implement RSA keys, IP addresses + "privileged ports", or other non-Kerberos 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 tickets and other data.

keytab file

A keytab file is used by a service host to store keys.

kinit

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

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.

kpasswd

`klist` is used to change your Kerberos password (a second command of the same name, different path, is used to change your AFS password).

ksu

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.

kvno

The **`kvno`** command acquires a service ticket for the specified Kerberos principals and prints out the key version numbers of each.

Openssh

OpenSSH is a FREE version of the SSH connectivity tools. OpenSSH encrypts all traffic (including passwords) to effectively eliminate eavesdropping, connection hijacking, and other attacks. Additionally, OpenSSH provides secure tunneling capabilities and several authentication methods, and supports all SSH protocol versions.[1]

permanent secret key

See *secret key*.

---

1. Adapted from description at http://www.openssh.com/.

portal

A secure gateway between the untrusted and strengthened realms that requires non-reusable passwords. At Fermilab, any Kerberized host may be configured to respond in *portal mode* to requests for access from untrusted machines (see *portal mode*).

portal mode

(See *portal*.) 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.

preauthentication

This is the stage of authentication in which you prove to the KDC that you know the shared secret key (which is a function of your password) before the KDC delivers a ticket to decrypt with the key.

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`. For a user, the instance portion is generally 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 upper-case letters, to differentiate the realm from the internet domain.

rcp

`rcp` copies files between machines.

response

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

rlogin

`rlogin` connects your terminal on the current local host system to the remote host system, as specified.

rsh

`rsh` connects to the specified host, and executes the specified command.

scp

`scp` copies files between hosts on a network. It uses **`ssh`** for data transfer, and uses the same authentication and provides the same security as **`ssh`**.

secret key

A long-term (permanent) 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 the user's 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.

slogin

`slogin` is the "login" part of **`ssh`**; if the <command> argument is left off, **`ssh`** runs **`slogin`**.

ssh

`ssh` (SSH client) is a program for logging into a remote machine and for executing commands on a remote machine. It is intended to replace **`rlogin`** and **`rsh`**, and provide secure encrypted communications between two untrusted hosts over an insecure network. Also see Openssh.

strengthened application

A software application that requires Kerberos 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 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.

telnet

A program used to communicate with another host using the TELNET protocol.

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.

transport

The program/protocol used to make a network connection and transport commands, data, etc. across the network. A transport program must implement an authentication method.

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.

Index

Symbols

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y