
HPCMP PKINIT Users Guide

	Revision History
Revision 20070711_PKINIT_04	November 16, 2007
Revision 20070711_PKINIT_05	December 04, 2007
Revision 3.2.0_20071210	December 10, 2007
Revision 3.2.0_20080114	January 14, 2008
Revision	April 16, 2008
HPCMP_RELEASE_20080416	
Revision	October 06, 2008
HPCMP_RELEASE_20081006	
Revision	March 17, 2009
HPCMP_RELEASE_20090317	

Table of Contents

General Information	1
PKINIT Notes	2
Release Notes	2
Obtaining the Software	2
Client Configuration	2
Linux	3
Mac OS X	9
Microsoft Windows	19
Building From Source	37
Troubleshooting pkinit	41
Manpages	43
Definitions	62

General Information

General information about *Kerberos* is available on the *MIT Kerberos web site* [<http://web.mit.edu/kerberos/www/>] and in the *Kerberos V5 UNIX User's Guide* [<http://web.mit.edu/kerberos/krb5-1.6/krb5-1.6.3/doc/krb5-user.html>]

Specific Information about Kerberos as it is used in the *HPCMP*, is available here at the *HPCMP web site* [<http://www.hpcmo.hpc.mil>] and the *HPCMP Kerberos and SecurID Center* [<http://www.hpcmo.hpc.mil/security/kerberos/>].

Frequently Asked Questions (FAQ's) for Kerberos can be found *here*. [<http://www.hpcmo.hpc.mil/security/kerberos/docs/krb-faq.html>] Frequently Asked Questions (FAQ's) for HPCMP Kerberos can be found *here* [<http://www.hpcmo.hpc.mil/security/kerberos/docs/krb-faq-hpcmp.html>].

Help for solving specific problems related to Kerberos and the use of Kerberos-aware programs for HPCMP users is available from the following:

Table 1. Site Specific Help

Kerberos Realm	Points of Contact	Phone Number	E-Mail Address	Hours of Operation
ARSC.EDU [http://www.arsc.edu/]	CCAC	(877) 222-2039	<help@ccac.hpc.mil>	04:00 to 19:00 AST weekdays

Kerberos Realm	Points of Contact	Phone Number	E-Mail Address	Hours of Operation
<i>ASC.HPC.MIL</i> [http://www.afrl.hpc.mil]	CCAC	(877) 222-2039	< help@ccac.hpc.mil >	08:00 to 23:00 EST weekdays
<i>WES.HPC.MIL</i> [http://www.erd.hpc.mil]	CCAC	(877) 222-2039	< help@ccac.hpc.mil >	07:00 to 22:00 CST weekdays
<i>HPCMP.HPC.MIL</i> & <i>HERCULES.HPC.MIL</i> [http://www.ccac.hpc.mil]	CCAC	(877) 222-2039	< help@ccac.hpc.mil >	08:00 to 23:00 EST weekdays
<i>MHPCC.HPC.MIL</i> [http://www.mhpcc.hpc.mil/]	CCAC	(877) 222-2039	< help@ccac.hpc.mil >	02:00 to 17:00 HST weekdays
<i>NAVO.HPC.MIL</i> [http://www.navo.hpc.mil/]	CCAC	(877) 222-2039	< help@ccac.hpc.mil >	07:00 to 22:00 CST weekdays
<i>NRL.NAVY.MIL</i> [http://www.cmf.nrl.navy.mil/CCS/hpc-nrl.html]	Support Services	(202) 404-7337	< ccshelp@nrl.navy.mil >	Not Specified

PKINIT Notes

PKINIT replaces the use of a password with a digital certificate. The digital certificate can be stored in a file or on a *Smart Card*. The Smart Cards that are supported include the DoD *CAC* and the HPCMP distributed *hToken*. The HPCMP *hToken* has been implemented using the Aladdin *eToken*. Questions about the HPCMP implementation can be sent to <pkinit@hpcmo.hpc.mil>. *PKINIT* is documented in *RFC 4556* [<http://www.ietf.org/rfc/rfc4556.txt>]. University of Michigan was funded by Sandia to implement *PKINIT* for MIT Kerberos. This work was based on the MIT Kerberos 1.6 branch. The HPCMP has back-ported this work to the NRL branch of Kerberos. For the HPCMP the digital certificate is stored on a Smart Card and is accessed via *PKCS#11*. This requires a *PKCS#11 Provider* be installed on the Desktop machine that will translate the generic *PKCS#11 API* calls to the specifics of the smart card.

Release Notes

- Windows Kits are based on MIT Kerberos for Windows (KFW) 3.2.3.
- *UNIX* and OS X Kits are based on HPCMP_RELEASE_20090211 and back-ported 1.6.3.

Obtaining the Software

All source code and binary client kits (including *PKINIT* capable kits) can be found on the *Kerberos & SecurID Information Center* [<https://www.hpcmo.hpc.mil/security/kerberos>]. Unauthenticated access to binary only client kits is available via this *url* [<https://www.hpcmo.hpc.mil/security/kerberos/software>]. Authenticated access to source code and binary kits is available under the *software* [<https://www.hpcmo.hpc.mil/security/kerberos/private/software>] link of the *Kerberos & SecurID Information Center*.

Client Configuration

Before you install the HPCMP *PKINIT* software, you should ensure that the proper drivers for your Smart Card and the *PKCS#11 Provider* have been installed on your system. The HPCMP has put together documents to assist you in configuring your Smart Card under various Operating Systems. If you are using the DoD *CAC* as your Smart Card, the *CAC Configuration Guide* [<https://www.hpcmo.hpc.mil/security/>]

kerberos/docs/Configuring_CAC.pdf] can be found on the HPCMP Kerberos website in the documentation section. If you are using the HPCMP supplied hToken as your Smart Card, the *eToken Install Guide* [https://www.hpcmo.hpc.mil/security/kerberos/docs/etoken_install.pdf] can also be found in the documentation section of the HPCMP Kerberos website.

To ensure your Smart Card is hooked up properly and the PKCS#11 Provider is installed correctly, insert your Smart Card and go to the *HPCMP Kerberos Site* [https://www.hpcmo.hpc.mil/security/kerberos]

Try logging into the site using the *PKI login* option. A successful PKI login will show that you are logged in as your CAC or hToken identity just under the title bar of the resulting page. If you cannot achieve a successful login, please contact your system administrator or local support to help get your Smart Card working in your machine.

Additional CAC client configuration help:

- An excellent resource for Linux systems is *Using DoD CAC and Smart Card Readers on Linux* [http://symbolik.wordpress.com/2007/02/25/using-dod-cac-and-smartcard-readers-on-linux/].
- An excellent resource for Macintosh systems is *Using DoD CAC and Smart Card Readers on Mac OS X* [http://images.apple.com/server/pdfs/Smart_Card_Setup_Guide.pdf].

Linux

Requirements

This section will take you through the steps needed to install the PKINIT software for the Linux system.

Note

If you have a previous version of the HPCMP PKINIT software installed on your system, you will need to remove the old directory structure (typically `/usr/local/krb5`) or rename it. By renaming the directory to `krb5.OLD` you can preserve the contents until you have fully tested the new version.

Uncompressing the GNU zipped tar file

Download the latest HPCMP PKINIT software for Linux from *HPCMP Kerberos Site* [https://www.hpcmo.hpc.mil/security/kerberos/]. The software is in a gnu zipped tar file. The file will be named based on the system it was built on and the Kerberos version and PKINIT version. For example: `HPCMP_RELEASE_20070711_2.6.15-29-386-Linux.tar.gz`. This shows that the binaries were built on a 386 based Linux system running kernel version 2.6.15 and the Kerberos used was HPCMP Version 20090211. The binaries will work on most of the current Linux systems. When downloaded, the file should be placed in a temporary directory that you have access to. Our example will use `/tmp` as the download directory.

Before unpacking the HPCMP PKINIT software for Linux on your local system, login as root. Set the `umask` so that the files preserve their permissions correctly.

```
#umask 022
```

Now go to the directory that you want the software to be installed. When uncompressing the software package, a subdirectory called `krb5` will be created. In our example we will be installing the software in `/usr/local`.

```
#cd /usr/local
```

To uncompress the gnu zipped tar file use the following command:

```
tar zxvf /tmp/HPCMP_RELEASE_20090211_2.6.15-29-386-Linux.tar.gz
```

The software has now been installed on the Linux systems in the `krb5` subdirectory.

Kerberos Binaries

The Kerberos software is contained in the `krb5` subdirectory. The kit contains binaries, configuration files, scripts and documentation files in PDF format.

After unpacking the archive, the following files will be located in `krb5`:

krb5 directory contents

<code>Configuring_CAC.pdf</code>	PDF file containing instructions for using CACs.
<code>pkinit_kdc_guide.pdf</code>	PDF file containing instructions on setting up a <i>KDC</i> with PKINIT.
<code>pkinit_user_guide.pdf</code>	PDF file containing instructions on setting up the PKINIT client kits.
<code>krb5.conf</code>	Kerberos configuration file
<code>bin</code>	Directory containing Kerberos client binaries: <ul style="list-style-type: none">• <code>bin/compile_et</code> - Error table compiler.• <code>bin/ftp</code> - Symbolic link to <code>kftp</code>.• <code>bin/kdestroy</code> - Will destroy Kerberos <i>tickets</i>.• <code>bin/kftp</code> - Kerberized file transfer program.• <code>bin/kinit</code> - Obtain and cache Kerberos ticket-granting ticket.• <code>bin/klist</code> - List cached Kerberos tickets.• <code>bin/kpasswd</code> - Change a user's Kerberos password.• <code>bin/krb5-config</code> - Tool for linking against MIT Kerberos libraries.• <code>bin/krcp</code> - Kerberized remote file copy.• <code>bin/krlogin</code> - Kerberized remote login.• <code>bin/krsh</code> - Kerberized remote shell.• <code>bin/kshell</code> - Creates a subshell for handling memory-based <i>credential caches</i>.• <code>bin/ktelnet</code> - User interface to the TELNET protocol.• <code>bin/kvno</code> - Print key version numbers of Kerberos <i>principals</i>.• <code>bin/telnet</code> - Symbolic link to <code>ktelnet</code>.• <code>bin/kxf</code> - Kerberized X Window forwarding.• <code>bin/pkinit</code> - Symbolic link to <code>kinit</code>. Used to get a ticket using a Smart Card.

	<ul style="list-style-type: none"> • bin/rcp - Symbolic link to krcp. • bin/rlogin - Symbolic link to krlogin. • bin/rsh - Symbolic link to krsh.
sbin	<p>Directory containing Kerberos server binaries:</p> <ul style="list-style-type: none"> • sbin/ftpd - Kerberized File Transfer Protocol server. • sbin/k5srvutil - Host key table (keytab) manipulation utility. • sbin/kadmin - Kerberos V5 database administration program. • sbin/kadmind - KADM5 administration server. • sbin/kadmin - Kerberos V5 database administration program. • sbin/kdb5_util - Kerberos database maintenance utility. • sbin/klogind - Kerberized remote login server. • sbin/kprop - Propagate a Kerberos V5 principal database to a slave server. • sbin/kpropd - Kerberos V5 slave KDC update server. • sbin/krb5kdc - Kerberos V5 KDC. • sbin/kshd - Kerberized remote shell server. • sbin/ktutil - Kerberos keytab file maintenance utility. • sbin/login.krb5 - Kerberos enhanced login program. • sbin/telnetd - DARPA TELNET protocol server.
etc	Directory containing DOD Certificates
include	Directory containing Kerberos include files
lib	Directory containing Kerberos library files
share	Directory Misc shared files, contains man pages

In order to use these binaries, you will need to ensure that the PATH variable has been set up to use the new Kerberos binaries. For example, if the binaries are in /usr/local/krb5, you would change the users' PATH variable as follows.

If you are using tcsh/csh:

```
% set path=($path /usr/local/krb5/bin)
```

If you are using ksh/sh:

```
% PATH="$PATH:/usr/local/krb5/bin"
```

Please note that some systems include a form of Kerberos already distributed with the system (Solaris or Linux is one example). In the majority of these cases, these Kerberos binaries will not work with HPCMP

system, so you must ensure that you're running the correct versions of **pkinit**, **kinit**, **klist**, and **kdestroy**. You may find out that these Kerberos binaries do not ask you to enter your pin or passcode.

Kerberos Client Only Binaries

For sites that do not want to install the server binaries, include files, and lib files, an abbreviated 'client version' of the kit is available. This kit uses the same naming convention as the full kit, but adds the word client in the title. For example: `HPCMP_RELEASE_20090211_client-2.6.15-29-386-Linux.tar.gz`.

After unpacking the archive, the following files will be located in the `krb5_client` subdirectory:

krb5_client directory contents

<code>Configuring_CAC.pdf</code>	PDF file containing instructions for using CACs.
<code>pkinit_kdc_guide.pdf</code>	PDF file containing instructions on setting up a KDC with PKINIT.
<code>pkinit_user_guide.pdf</code>	PDF file containing instructions on setting up the PKINIT client kits.
<code>krb5.conf</code>	Kerberos configuration file
<code>bin</code>	Directory containing Kerberos client binaries <ul style="list-style-type: none"> • <code>bin/compile_et</code> - Error table compiler. • <code>bin/ftp</code> - Symbolic link to <code>kftp</code>. • <code>bin/kdestroy</code> - Will destroy Kerberos tickets. • <code>bin/kftp</code> - Kerberized file transfer program. • <code>bin/kinit</code> - Obtain and cache Kerberos ticket-granting ticket. • <code>bin/klist</code> - List cached Kerberos tickets. • <code>bin/kpasswd</code> - Change a user's Kerberos password. • <code>bin/krb5-config</code> - Tool for linking against MIT Kerberos libraries. • <code>bin/krcp</code> - Kerberized remote file copy. • <code>bin/krlogin</code> - Kerberized remote login. • <code>bin/krsh</code> - Kerberized remote shell. • <code>bin/kshell</code> - Creates a subshell for handling memory-based credential caches. • <code>bin/ktelnet</code> - User interface to the TELNET protocol. • <code>bin/kvno</code> - Print key version numbers of Kerberos principals. • <code>bin/kxf</code> - Kerberized X Window forwarding. • <code>bin/pkinit</code> - Symbolic link to <code>kinit</code>. Used to get a ticket using a Smart Card. • <code>bin/rcp</code> - Symbolic link to <code>krcp</code>.

- bin/rlogin - Symbolic link to krlogin.
- bin/rsh - Symbolic link to krsh.
- bin/telnet - Symbolic link to ktelnet.

etc Directory containing DOD Certificates

In order to use these binaries, you will need to ensure that the PATH variable has been set up to use the new krb5 client binaries. For example, if the binaries are in /usr/local/krb5_client/bin, you would change the users' PATH variable as follows.

If you are using tcsh/csh:

```
% set path=($path /usr/local/krb5_client/bin)
```

If you are using ksh/sh:

```
% PATH="$PATH:/usr/local/krb5_client/bin"
```

Setting up the /etc/krb5.conf file

The krb5.conf file included with the software is located in the krb5 directory. If you have root on the system, install the krb5.conf file in /etc/krb5.conf. If you don't have root or prefer not to modify the system you're using, place the krb5.conf file wherever you want, but set the environment variable KRB5_CONFIG to point to this location.

For example, if you place the krb5.conf file in /usr/people/somebody/krb5.conf, you should do:

If you are using tcsh/csh:

```
% setenv KRB5_CONFIG /usr/people/somebody/krb5.conf
```

If you are using ksh/sh:

```
$ KRB5_CONFIG=/usr/people/somebody/krb5.conf; export KRB5_CONFIG
```

Several changes may have to be made to the krb5.conf file, based on which directory the software was placed in.

If your home realm is not HPCMP.HPC.MIL, you will need to modify the default realm (under the [libdefaults] section in the krb5.conf file) to your home realm. If you don't know your home realm, please contact the center from which you received your Kerberos login for the information.

The kit includes the required DOD and HPCMP CA certificates for proper authentication. If you put the krb5 directory in a directory other than /usr/local/krb5, you will have to change the following two lines in the krb5.conf file:

```
pkinit_anchors = DIR:/usr/local/krb5/etc/CA
pkinit_pool = DIR:/usr/local/krb5/etc/CERTS
```

The krb5.conf file included in the kit looks for a PKCS11 module listed by the environment variable 'PKINIT_IDENTITY'. If the variable is empty or has an invalid PKCS11 module entry, it will look in several standard directory locations. It will use the first module that it finds and not look at any remaining entries. If you are using a PKCS11 module that is not listed in the krb5.conf file or is in a different location, you can add your entry at the beginning of the list or use the environment variable 'PKINIT_IDENTITY'.

```
pkinit_identities = ENV:PKINIT_IDENTITY
pkinit_identities = KEYCHAIN:
pkinit_identities = PKCS11:/usr/local/lib/pkcs11/libcoolkeypk11.so
pkinit_identities = PKCS11:/lib/libcoolkeypk11.so
pkinit_identities = PKCS11:/usr/lib/libcoolkeypk11.so
pkinit_identities = PKCS11:/usr/lib/libetPkc11.so
```

Note

The KEYCHAIN entry is only used for Mac OS X. It has no effect on the Linux systems.

The PKINIT_IDENTITY variable can be set as follows.

If you are using tcsh/csh:

```
% setenv PKINIT_IDENTITY PKCS11:/usr/local/lib/pkcs11/libcoolkeypk11.so
```

If you are using ksh/sh:

```
$ PKINIT_IDENTITY=PKCS11:/usr/local/lib/pkcs11/libcoolkeypk11.so
export PKINIT_IDENTITY
```

The following line in the krb5.conf file is needed to choose the proper certificate on the Smart Card:

```
pkinit_cert_match = <SAN>^[0-9]{10}@mil$
pkinit_cert_match = <SAN>^[0-9]{10}@hpcmp$
```

Getting a Kerberos Ticket Using pkinit

After the software has been installed and the krb5.conf file has been configured, you need to perform the following steps to login to the HPCMP systems:

1. Launch a terminal or console.

2. type **kshell** and hit Return.

```
$ kshell
```

3. Run **pkinit** with optional USERNAME and REALM. Hit Return.

```
$ pkinit tproue@HPCMP.HPC.MIL
```

4. Enter your PIN associated with your Smart Card and hit Return.

```
PROUE.THOMAS.MARK.1212298626 PIN:
$
```

5. After inputting the PIN associated with your Smart Card, pkinit should return with no errors. To actually see your Kerberos Ticket Granting Ticket, or any other tickets you may have acquired, run klist.

```
$ klist
Ticket cache: PIPE:1018
Default principal: tproue@HPCMP.HPC.MIL
```

```
Valid starting    Expires          Service principal
07/12/07 19:22:46 07/13/07 05:21:55 krbtgt/HPCMP.HPC.MIL@HPCMP.HPC.MIL
```


- Next, login to the appropriate system at an HPCMP center. You should use a Kerberized **ssh**, which can be found at the *HPCMP Kerberos Site* [<http://www.hpcmo.hpc.mil/security/kerberos/>]. You could also use either **krlogin** or **ktelnet** if the remote server allows it. Not all of the systems in the HPCMP centers are set up to accept incoming telnet or r-command connections. You can log in to any HPCMP resource you have an account on.

```
$ ssh hostname
or
$ ktelnet hostname
or
$ krlogin hostname
```

Note

If you do not know the hostname, please contact system administrator for the information.

Note

You can also use **sftp**, **scp**, **kftp** or **krftp** to transfer files into HPCMP systems, and **kpasswd** to change your password.

Mac OS X

Requirements

Setting up Mac OS X

This section will take you through the steps needed to install the pkinit software for the Mac OS X 10.4.x or 10.5.x system.

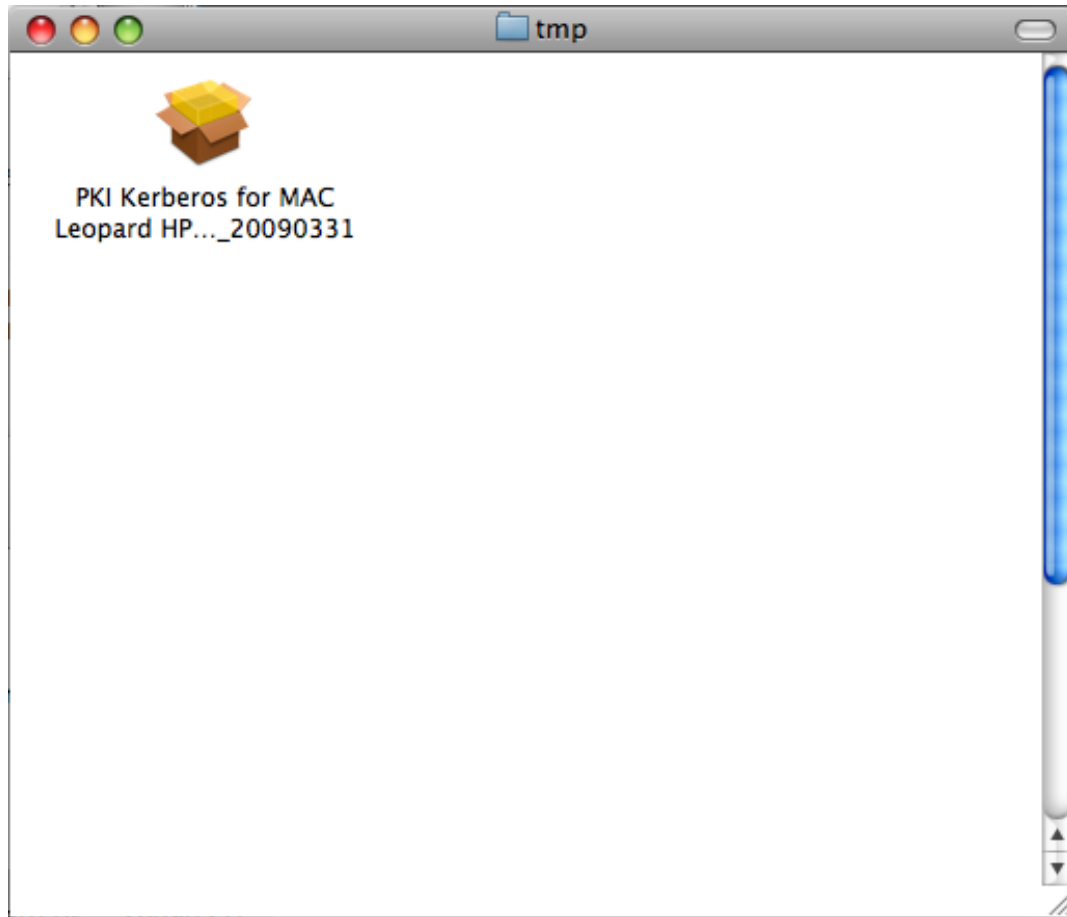
Note

If you have a previous version of the HPCMP pkinit software installed on your system, you will need to remove the old directory structure (typically `/usr/local/krb5`) or rename it. By renaming the directory to `krb5.OLD` you can preserve the contents until you have fully tested the new version. If you will be installing the newer version of the Kerberized SSH package, again you should remove or rename the previous version. The default location for the Kerberized SSH package is `/usr/local/ssh`.

Installing the Mac OS X Kerberos Package

Download the latest HPCMP pkinit software for Mac OS X from the *HPCMP Kerberos Site* [<https://www.hpcmo.hpc.mil/security/kerberos/>]. The software is contained in a MAC OS X installer package file. The package is named based on the system it was built on and the Kerberos version and PKINIT version. For example - PKI Kerberos for MAC Leopard HPCMP_RELEASE_20090331.pkg. This shows that the binaries were built on a Mac Leopard system using Kerberos HPCMP Version 20090331 code. The binaries were built as universal binaries and will work on both the Intel and PowerPC based Mac systems. When downloaded, the file should be placed in a temporary directory that you have access to. Our example will use `$HOME/tmp` as the download directory.

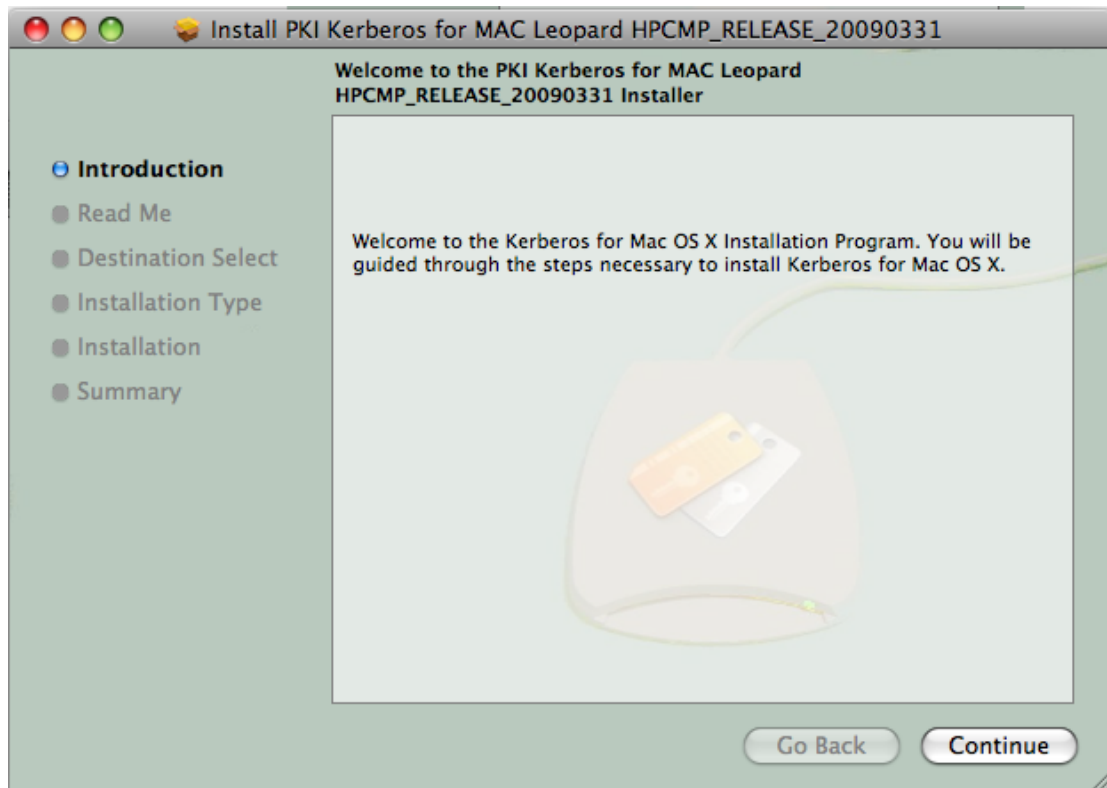
After the download is complete, use the File finder to go to the download directory. The installer package file will be displayed as shown in the figure below.

Figure 1. Mac OS X Installer Package File

Installing the Kerberos Software on the MAC

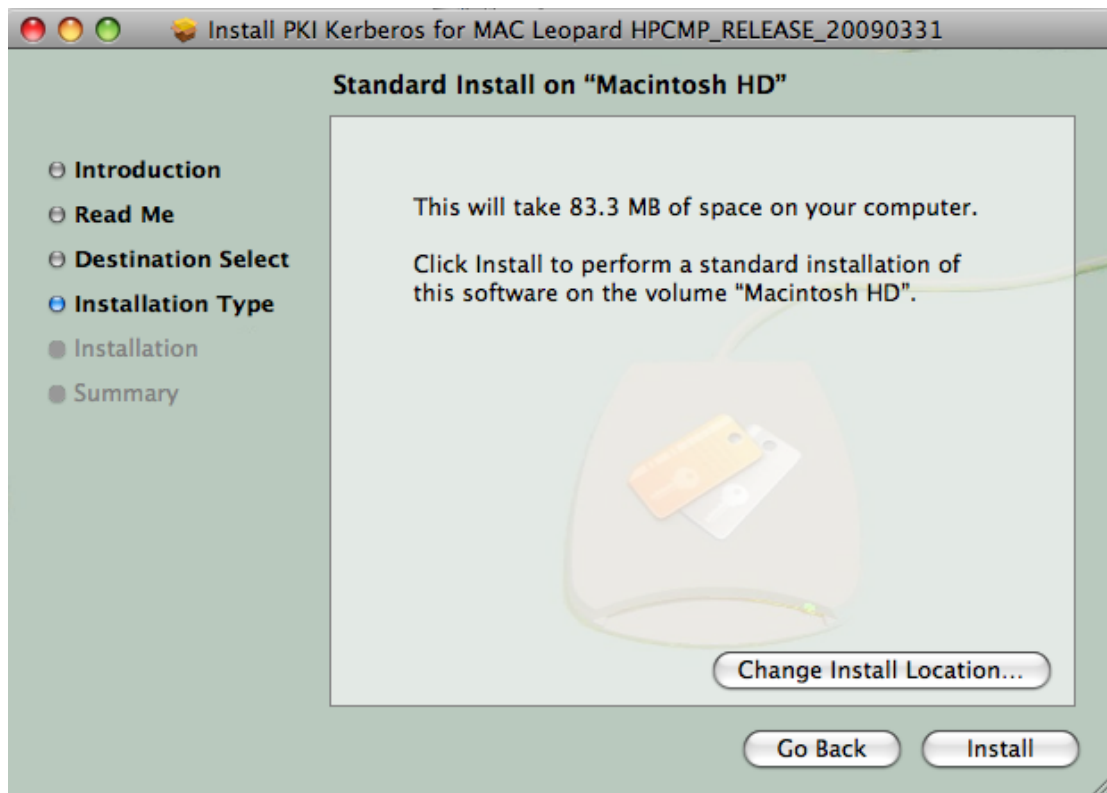
The Kerberos binary package contains compiled programs, ready to run. These programs were compiled as universal binaries and can be run on either PowerPC or x86 (Intel) based Macintosh computers. If you previously installed an older version of the Kerberos binaries, you should be able to upgrade by simply installing the newer version over the old one. Another option would be to install the binaries in a different directory.

To start the install process, double click on the Kerberos package(ie. PKI Kerberos for MAC Leopard HPCMP_RELEASE_20090331.pkg). A menu system will be displayed as shown in the figure below.

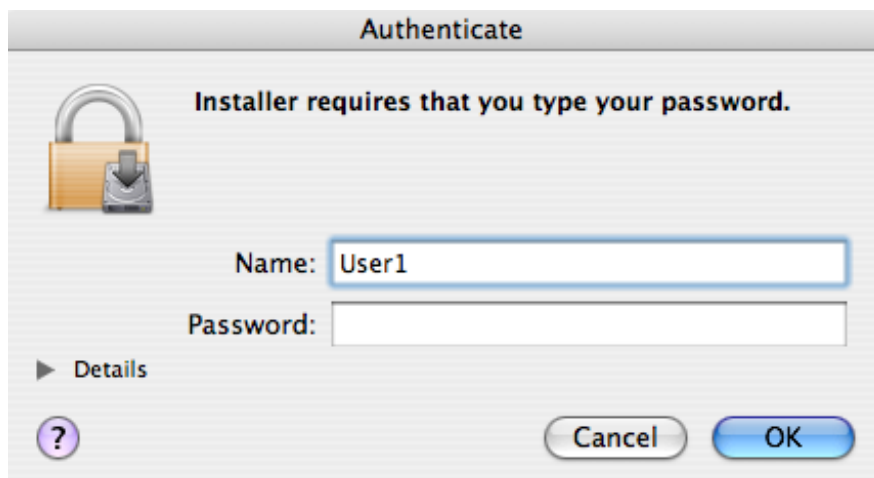
Figure 2. Install PKI Kerberos for Mac OS X Package

Click on the 'Continue' button and the README file will be displayed. This file explains what version of the software is being installed and default directory for the installation. Click on the 'Continue' button again.

You are now presented with a screen giving you the option of what filesystem and directory to install the Kerberos package in. Typically the software will be installed to the Macintosh Hard Drive under the default directory `/usr/local/krb5`. If you choose to test the Kerberos software before using it in a production environment, you can install it on a test filesystem or a different directory. After the testing is complete, you could either move the files to the default location, or reinstall them to the default location.

Figure 3. Select Destination Volume

After choosing the filesystem and directory to install the package, you should click on the 'Continue' button. The next screen verifies the drive and directory of the install. You have one last opportunity to change the destination filesystem and directory by clicking on the 'Go Back' button. The next screen presented is verifying that this is a 'basic install' as opposed to an upgrade. If you are ready to proceed with the install click on the 'Install' button. If you are installing to a system directory, such as the default directory, you will need admin privileges. A screen asking you to put in your password will be presented. If you do not have admin privileges you will either have to install the software in a directory you have permissions on or get an admin to install the software for you.

Figure 4. OS X Authenticate

If the software installed properly the next screen will display the message 'The software was successfully installed'. If the installation failed for any reason, ensure you were going to a valid filesystem and directory and that you have admin priviledges.

Kerberos Binaries

The Kerberos software is contained in the `/usr/local/krb5` subdirectory. The kit contains binaries, configuration files, scripts and three PDF documents.

Note

`/usr/local/krb5` is the default directory. During installation the admin may have chose to put the files in a different location.

After installing the Kerberos binary package, the following files will be located in `/usr/local/krb5`(install directory):

`/usr/local/krb5` directory contents

<code>Configuring_CAC.pdf</code>	PDF file containing instructions for using CACs.
<code>pkinit_kdc_guide.pdf</code>	PDF file containing instructions on setting up a KDC with PKINIT.
<code>pkinit_user_guide.pdf</code>	PDF file containing instructions on setting up the PKINIT client kits.
<code>krb5.conf</code>	Kerberos configuration file
<code>bin</code>	Directory containing Kerberos client binaries: <ul style="list-style-type: none">• <code>bin/compile_et</code> - Error table compiler.• <code>bin/ftp</code> - Symbolic link to <code>kftp</code>.• <code>bin/kdestroy</code> - Will destroy Kerberos tickets.• <code>bin/kftp</code> - Kerberized file transfer program.• <code>bin/kinit</code> - Obtain and cache Kerberos ticket-granting ticket.• <code>bin/klist</code> - List cached Kerberos tickets.• <code>bin/kpasswd</code> - Change a user's Kerberos password.• <code>bin/krb5-config</code> - Tool for linking against MIT Kerberos libraries.• <code>bin/krcp</code> - Kerberized remote file copy.• <code>bin/krlogin</code> - Kerberized remote login.• <code>bin/krsh</code> - Kerberized remote shell.• <code>bin/kshell</code> - Creates a subshell for handling memory-based credential caches.• <code>bin/ktelnet</code> - User interface to the TELNET protocol.

	<ul style="list-style-type: none"> • bin/kvno - Print key version numbers of Kerberos principals. • bin/kxf - Kerberized X Window forwarding. • bin/pkinit - Symbolic link to kinit. Used to get a ticket using a Smart Card. • bin/pkpasswd - Change a user's Kerberos password using PKI. • bin/rcp - Symbolic link to krcp. • bin/rlogin - Symbolic link to krlogin. • bin/rsh - Symbolic link to krsh. • bin/telnet - Symbolic link to ktelnet.
sbin	<p>Directory containing Kerberos server binaries:</p> <ul style="list-style-type: none"> • sbin/ftpd - Kerberized File Transfer Protocol server. • sbin/k5srvutil - Host key table (keytab) manipulation utility. • sbin/kadmin - Kerberos V5 database administration program. • sbin/ klogind - Kerberized remote login server. • sbin/kshd - Kerberized remote shell server. • sbin/ktutil - Kerberos keytab file maintenance utility. • sbin/login.krb5 - Kerberos enhanced login program. • sbin/pkadmin - Kerberos V5 database administration program using PKI. • sbin/telnetd - DARPA TELNET protocol server.
etc	<p>Directory containing DOD and HPCMP Certificates</p> <ul style="list-style-type: none"> • etc/CA - directory containing Certificate Authorities. • etc/CERTS - directory containing Intermediate Certificates.
include	Directory containing Kerberos include files
lib	Directory containing Kerberos library files
share	Directory of miscellaneous files, contains man pages

In order to use these binaries, you will need to ensure that the PATH variable has been set up to use the new krb5 client binaries. For example, if the binaries are in /usr/local/krb5/bin, you would change the users' PATH variable as follows.

If you are using tcsh/csh:

```
% set path=( $path /usr/local/krb5/bin )
```

If you are using ksh/sh:

```
% PATH="$PATH:/usr/local/krb5/bin"
```

Setting up the `/etc/krb5.conf` file

The `krb5.conf` file included with the software is located in the `krb5` directory. If you have root on the system, install the `krb5.conf` file in `/etc/krb5.conf`. If you don't have root or prefer not to modify the system you're using, place the `krb5.conf` file wherever you want, but set the environment variable `KRB5_CONFIG` to point to this location.

For example, if you place the `krb5.conf` file in `/usr/people/somebody/krb5.conf`, you should do:

If you are using tcsh/csh:

```
% setenv KRB5_CONFIG /usr/people/somebody/krb5.conf
```

If you are using ksh/sh:

```
$ KRB5_CONFIG=/usr/people/somebody/krb5.conf; export KRB5_CONFIG
```

Several changes may have to be made to the `krb5.conf` file based on which directory the software was placed in.

If your home realm is not `HPCMP.HPC.MIL`, you will need to modify the default realm (under `[libdefaults]` section in the `krb5.conf` file) to your home realm. If you don't know your home realm, please contact the center from which you received your Kerberos login for the information.

The kit includes the required DOD and HPCMP CA certificates for proper authentication. If you put the `krb5` directory in a directory other than `/usr/local/krb5`, you will have to change the following two line in `krb5.conf`:

```
pkinit_anchors = DIR:/usr/local/krb5/etc/CA
pkinit_pool = DIR:/usr/local/krb5/etc/CERTS
```

When using a MAC, all certificates should go through the MAC Keychain. If this does not happen, there may be conflicts between the PKINIT Kerberos application, the mail client, and the web application when using Smart Cards. In the `krb5.conf` file the following lines show that we first attempt to go through the environment variable 'PKINIT_IDENTITY' then the MAC keychain. If 'PKINIT_IDENTITY' is set to a valid value it will be used, otherwise an attempt is made to use the Keychain. If that fails, we will use the first valid PKCS11 module listed.

```
pkinit_identities = ENV:PKINIT_IDENTITY
pkinit_identities = KEYCHAIN:
pkinit_identities = PKCS11:/usr/local/lib/pkcs11/libcoolkeypk11.so
pkinit_identities = PKCS11:/lib/libcoolkeypk11.so
pkinit_identities = PKCS11:/usr/lib/libcoolkeypk11.so
pkinit_identities = PKCS11:/usr/lib/pkcs11/libcoolkeypk11.so
pkinit_identities = PKCS11:/usr/lib/libeTPkcs11.so
```

The following line in the `krb5.conf` file is needed to choose the proper key on the Smart Card:

```
pkinit_cert_match = <SAN>^[0-9]{10}@mil$
```

```
pkinit_cert_match = <SAN>^[0-9]{10}@hpcmp$
```

Getting a Kerberos Ticket Using pkinit

After the software has been installed and the `krb5.conf` file has been configured, you need to perform the following steps to login to the HPCMP systems:

1. Launch a terminal window.
2. Run **pkinit** with optional USERNAME and REALM. Hit Return

```
$ pkinit tproue@HPCMP.HPC.MIL
```

3. Enter your PIN for your Smart Card and hit Return

```
CAC-2050-5000-1028-0055-0883 PIN:
```

4. After inputting the PIN associated with your Smart Card, pkinit should return with no errors. To actually see your Kerberos Ticket Granting Ticket, or any other tickets you may have acquired, run **klist**

```
$ klist
Ticket cache: API:Initial default ccache
Default principal: tproue@HPCMP.HPC.MIL
```

```
Valid starting      Expires            Service principal
08/31/07 11:10:08  08/31/07 21:09:00  krbtgt/HPCMP.HPC.MIL@HPCMP.HPC.MIL
```

5. Next, login to the appropriate system at an HPCMP center. You should use a Kerberized **ssh**, which can be found at the *HPCMP Kerberos Site* [<http://www.hpcmo.hpc.mil/security/kerberos/>]. You could also use either **krlogin** or **ktelnet** if the remote server allows it. Not all of the systems in the HPCMP centers are set up to accept incoming telnet or r-command connections. You can log in to any HPCMP resource you have an account on.

```
$ ssh hostname
or
$ ktelnet hostname
or
$ krlogin hostname
```

Note

If you do not know the hostname, please contact system administrator for the information.

Note

You can also use **sftp**, **scp**, **kftp** or **krftp** to transfer files into HPCMP systems, and **kpasswd** to change your password.

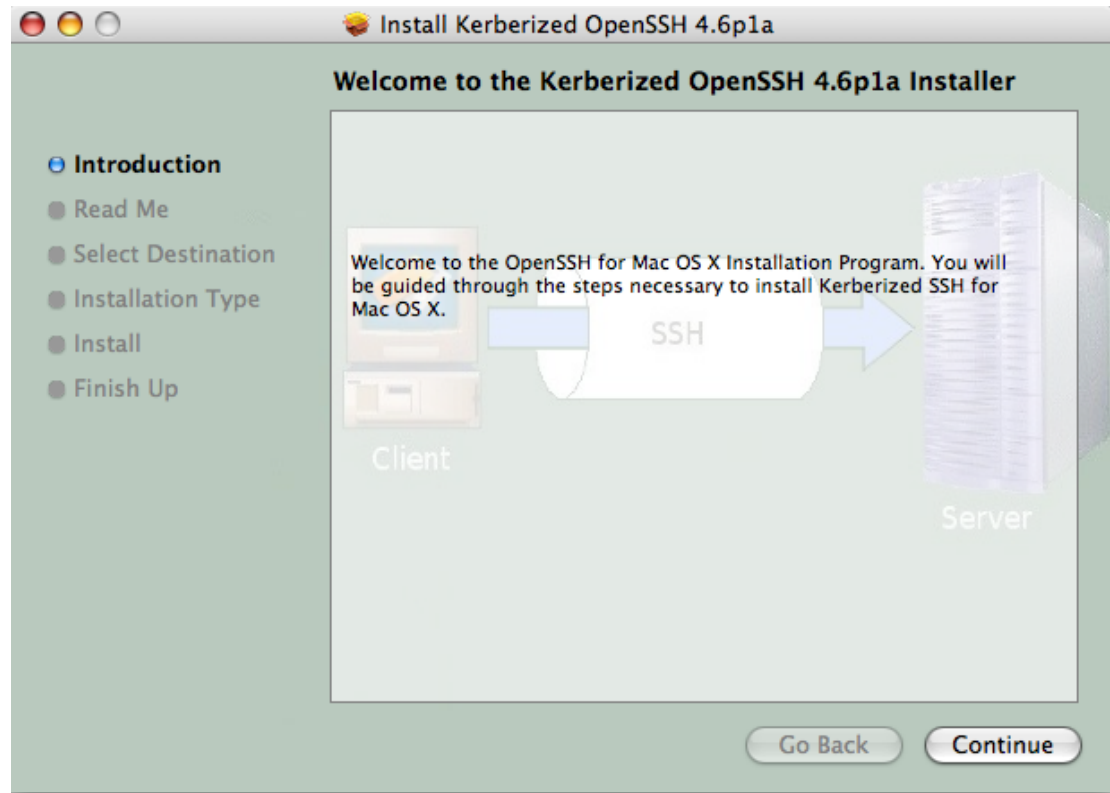
Installing the OpenSSH Software on the MAC

The OpenSSH binary package contains compiled programs, ready to run. These programs were compiled as universal binaries and can be run on either PowerPC or x86 (Intel) based Macintosh computers. If you previously installed an older version of the OpenSSH binaries, you should be able to upgrade by simply

installing the newer version over the old one. Another option would be to install the binaries in a different directory.

To start the install process, double click on the OpenSSH package(ie. Kerberized OpenSSH 4.6p1a). A menu system will be displayed as shown in the figure below.

Figure 5. Install OSSH for Mac OS X Package



Click on the 'Continue' button and the README file will be displayed. This file explains what version of the software is being installed and default directory for the installation. Click on the 'Continue' button again.

You are now presented with a screen giving you the option of what filesystem and directory to install the Kerberos package in. Typically the software will be installed to the Macintosh Hard Drive under the default directory `/usr/local/ssh`. If you choose to test the Kerberos software before using it in a production environment, you can install it on a test filesystem or a different directory. After the testing is complete, you could either move the files to the default location, or reinstall them to the default location.

After choosing the filesystem and directory to install the package, you should click on the 'Continue' button. The next screen just verifies that you would like to perform the install. You have one last opportunity to change the destination filesystem and directory by clicking on the 'Go Back' button. If you are ready to proceed with the install click on the 'Install' button. If you are installing to a system directory, such as the default directory, you will need admin privileges. A screen asking you to put in your password will be presented. If you do not have admin privileges you will either have to install the software in a directory you have permissions on or get an admin to install the software for you.

If the software installed properly the next screen will display the message 'The software was successfully installed'. If the installation failed for any reason, ensure you were going to a valid filesystem and directory and that you have admin privileges.

OpenSSH Binaries

The OpenSSH software is contained in the `/usr/local/ssh` subdirectory. The kit contains binaries, configuration files, scripts and man pages.

Note

`/usr/local/ssh` is the default directory. During installation the admin may have chose to put the files in a different location.

After installing the Kerberos binary package, the following files will be located in `/usr/local/ssh(install directory)`:

OpenSSH Directory Contents

`/usr/local/ssh` directory contents

<code>bin</code>	Directory containing OpenSSH client binaries: <ul style="list-style-type: none">• <code>bin/scp</code> - secure copy (remote file copy program)• <code>bin/sftp</code> - secure file transfer program• <code>bin/slogin</code> - Symbolic link to <code>ssh</code>• <code>bin/ssh</code> - OpenSSH SSH client (remote login program)• <code>bin/ssh-add</code> - adds <i>RSA</i> or <i>DSA</i> identities to the authentication agent• <code>bin/ssh-agent</code> - authentication agent• <code>bin/ssh-keygen</code> - authentication key generation, management and conversion• <code>bin/ssh-keyscan</code> - gather ssh public keys
<code>sbin</code>	Directory containing OpenSSH server binaries: <ul style="list-style-type: none">• <code>sbin/sshd</code> - OpenSSH SSH daemon
<code>etc</code>	Directory containing <code>ssh</code> and <code>sshd</code> config files
<code>libexec</code>	Directory containing <code>sftp-server</code> (SFTP server subsystem) and <code>ssh-keysign</code> (ssh helper program for host-based authentication)
<code>share</code>	Directory containing shared files and man pages

In order to use these binaries, you will need to ensure that the `PATH` variable has been set up to use the new `ssh` client binaries. For example, if the binaries are in `/usr/local/ssh/bin`, you would change the users' `PATH` variable as follows.

If you are using `tcsh/csh`:

```
% set path=( $path /usr/local/ssh/bin )
```

If you are using `ksh/sh`:

```
% PATH="$PATH:/usr/local/ssh/bin"
```

Microsoft Windows

Getting an initial Kerberos *TGT*, listing tickets, deleting tickets and changing your Kerberos password are all supported by the *GUI* application *krb5.exe*. The same *CLI* tools that are used under Linux are also available, please refer to the section called “Getting a Kerberos Ticket Using *pkinit*” for documentation on those tools. For host connectivity via the telnet or ssh protocol, *PuTTY.exe* is supplied. For file transfers using FTP or SFTP, *Filezilla.exe* is available. The following sections will describe how to install configure and run these applications.

Requirements

Supported Windows Versions

- Windows 2000 SP4
- Windows XP SP2
- Windows Vista 32-bit
- Windows Vista 64-bit
- Windows Server 2003
- Windows Server 2003 R2

Supported PKCS#11 Providers

- *ActivCard Gold CAC*
- *ActivClient CAC 32-bit*
- *ActivClient CAC 64-bit*
- *Aladdin PKI Client*

Installing the Software

The latest HPCMP Kerberos for Windows Client kit can be found with the other PKINIT enabled software. See the section called “Obtaining the Software”

The Client Kit is available as both a Windows Installer and a Self Extracting Zip archive.

Note

Administrator privileges are required to install using the installer. Use the self extracting zip file if you do not have administrator rights.

To install using the Windows Installer simply double click on it install as you would any other Windows Application.

To install using the Self Extracting Zip archive, copy the file to the desired location and double click on it. The files will be unpacked in the current directory.

Configuring krb5.exe

Before using krb5.exe to acquire a tgt using PKINIT you must first configure krb5.exe with your Kerberos information and select PKINIT as your form of authentication. If PKINIT authentication fails, krb5.exe will fall back to using your Kerberos Password and SecurID. The following steps illustrate how to configure krb5.exe for PKINIT authentication.

1. Click on the Start Menu # All Programs # HPCMP Kerberos # krb5.exe Icon or double click on the

krb5.exe application.

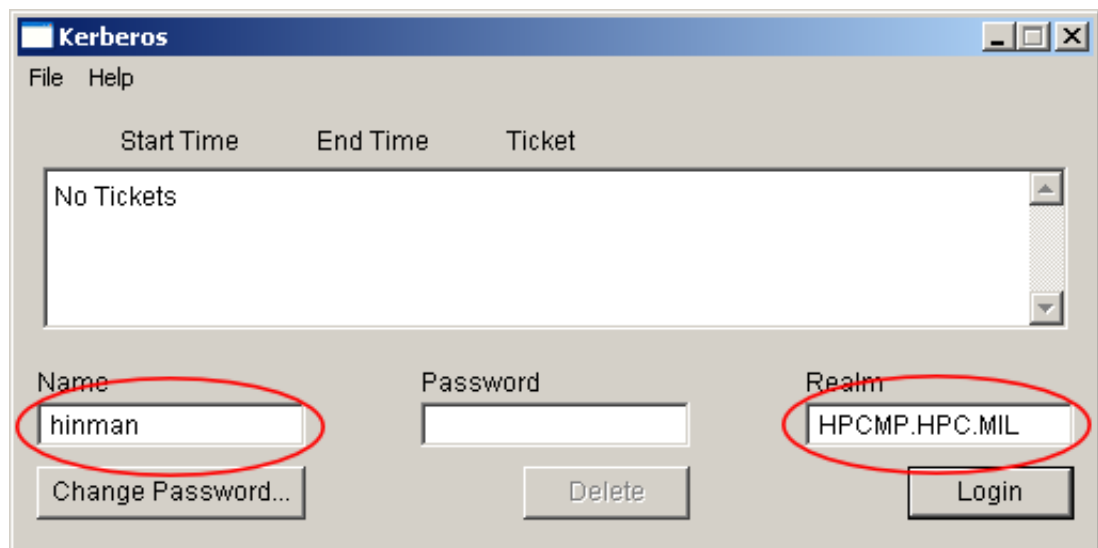


2. Fill in your Name and Kerberos Realm. See Figure 6, “Specify Kerberos Name and Realm”

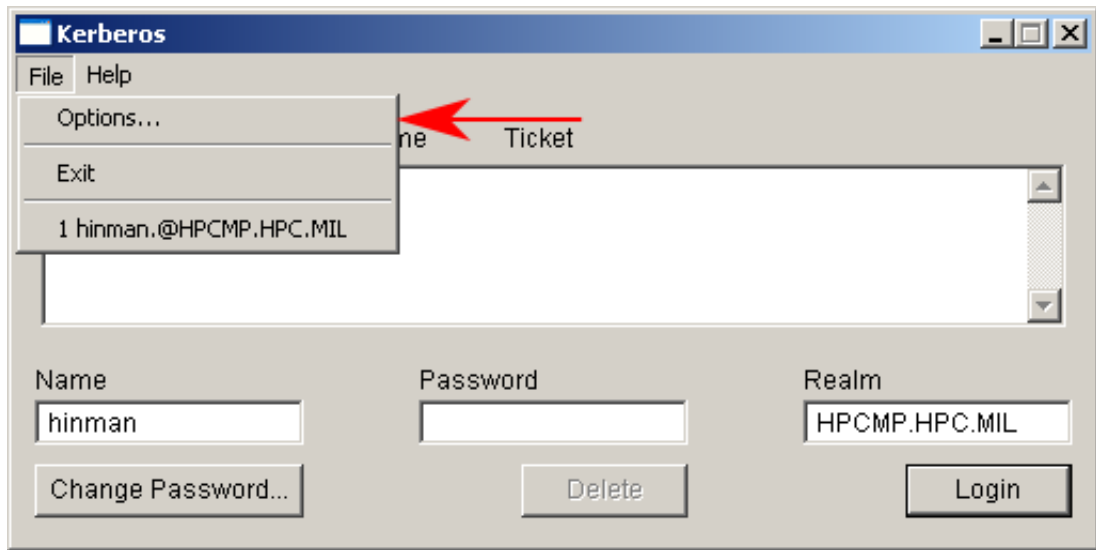
Important

The Kerberos Realm must be in ALL CAPS.

Figure 6. Specify Kerberos Name and Realm



3. Select File # Options .. from the drop down menu. See Figure 7, “Selecting krb5.exe Options”

Figure 7. Selecting krb5.exe Options

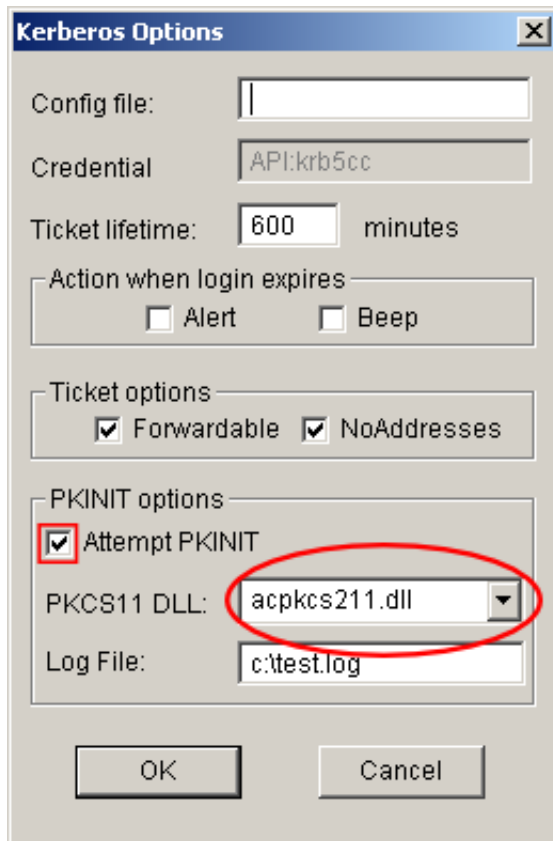
4. First Click on the Attempt PKINIT Check box. Second, Select your PKCS11 *DLL* from the drop down list. The drop down list is populated with known good DLLs that were found on your system. Please try one of these DLLs first. However, if your DLL is not in the list you can type it in the Text Box. See Figure 8, "Kerberos Options". Common values for PKCS11 DLL are:

- If your PKCS#11 Provider is ActivCard Gold it is *acpkcs.dll*.
- If your PKCS#11 Provider is ActivClient (32 or 64 bit) it is *acpkc211.dll*.
- If your PKCS#11 Provider is Aladdin PKI Client it is *eTpkcs11.dll*.

If you are experiencing difficulty getting a tgt with *krb5.exe* you can enable a log file to capture additional debugging output. Simply type in the name of the file in the Log File Text Box.

Note

The contents of this file will be overwritten every time you attempt to get a tgt.

Figure 8. Kerberos OptionsThe image shows a Windows-style dialog box titled "Kerberos Options". It contains several fields and checkboxes. The "Config file:" field is empty. The "Credential" field contains "API:krb5cc". The "Ticket lifetime:" field contains "600" and "minutes". There are two checkboxes under "Action when login expires": "Alert" and "Beep", both are unchecked. Under "Ticket options", there are two checkboxes: "Forwardable" and "NoAddresses", both are checked. Under "PKINIT options", there is a checked checkbox for "Attempt PKINIT". Below this is a dropdown menu for "PKCS11 DLL:" which is currently set to "acpkcs211.dll"; this dropdown is circled in red. The "Log File:" field contains "c:\test.log". At the bottom are "OK" and "Cancel" buttons.

5. Click OK.

Using PKINIT to get a Ticket with krb5.exe

To obtain a tgt using PKINIT for the Hardware Preauthentication with krb5.exe follow these steps:

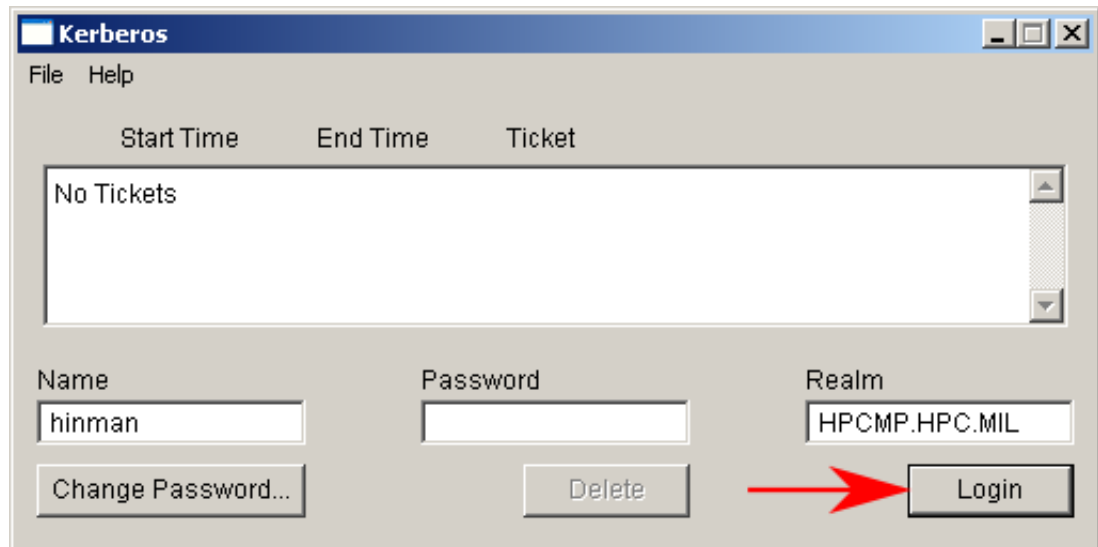
1. Click on the Start Menu # All Programs # HPCMP Kerberos # krb5.exe Icon or double click on the

krb5.exe application. 

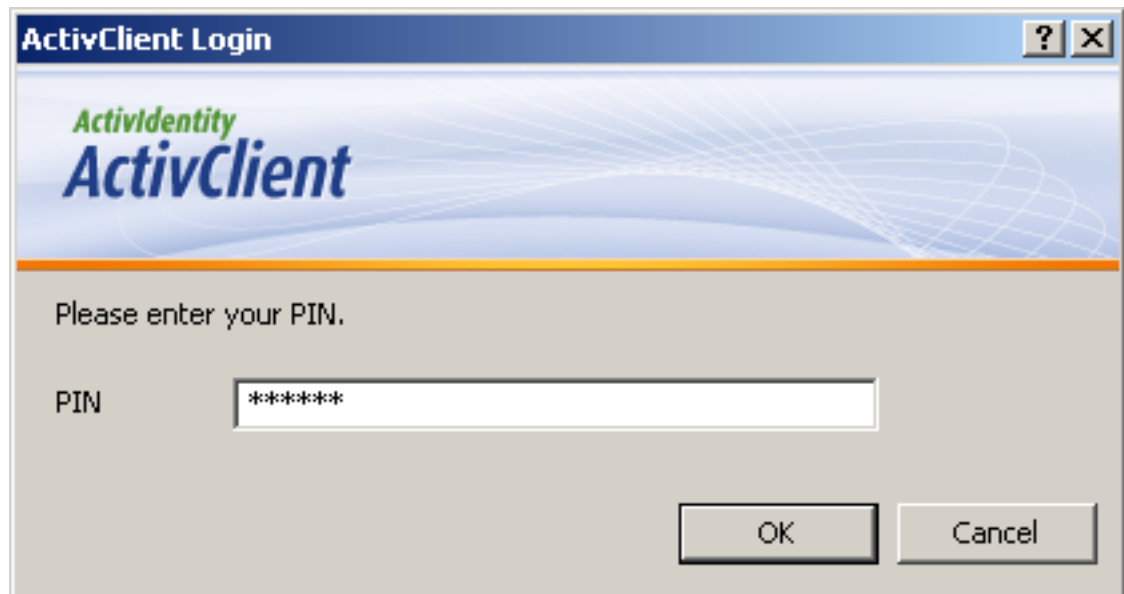
2. Click Login. See Figure 9, "Obtain TGT with krb5.exe"

Warning

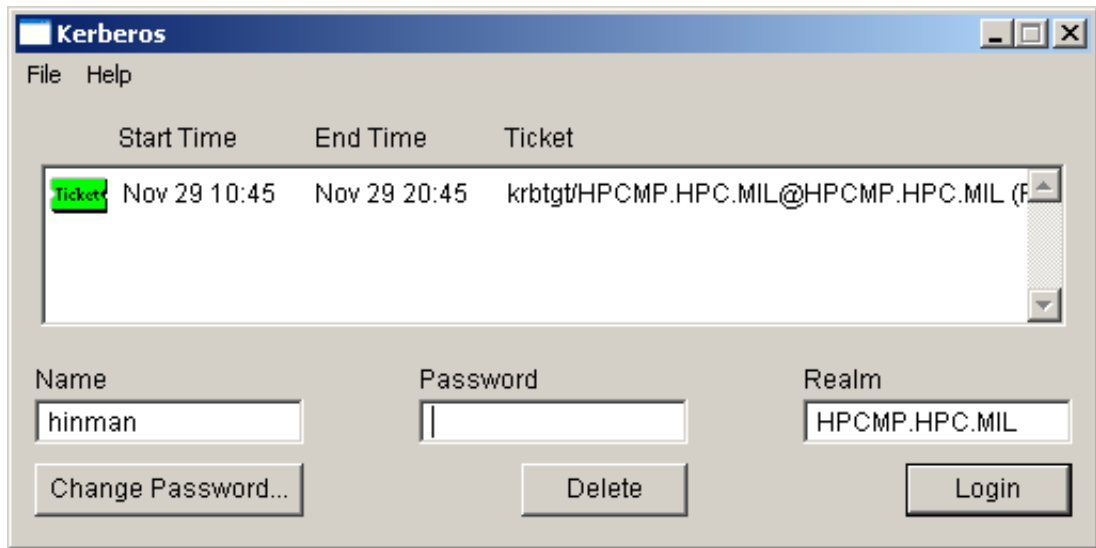
Do not fill in a Password, PKINIT does not use your Kerberos Password. If you do supply one, you will use SecurID for HW preauthentication.

Figure 9. Obtain TGT with krb5.exe

3. Enter your PIN when prompted and Click the OK Button. See Figure 10, "Enter your PIN"

Figure 10. Enter your PIN

4. Enjoy your new tgt. See Figure 11, "Ticket Listing"

Figure 11. Ticket Listing

Using SecurID to get a Ticket with krb5.exe

To obtain a tgt using SecurID for the Hardware Preauthentication with krb5.exe follow these steps:

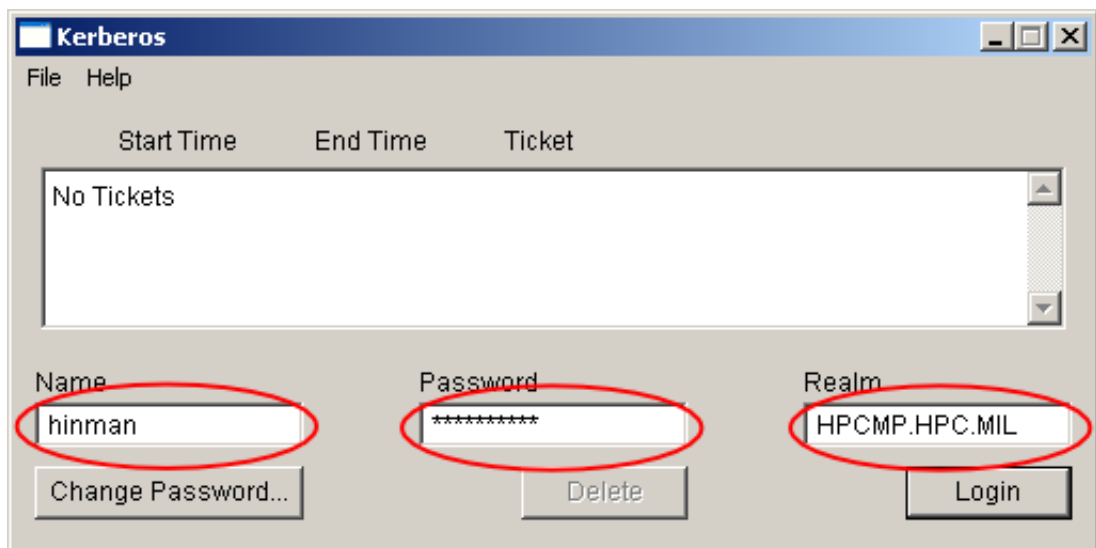
1. Click on the Start Menu # All Programs # HPCMP Kerberos # krb5.exe Icon or double click on the

krb5.exe application.



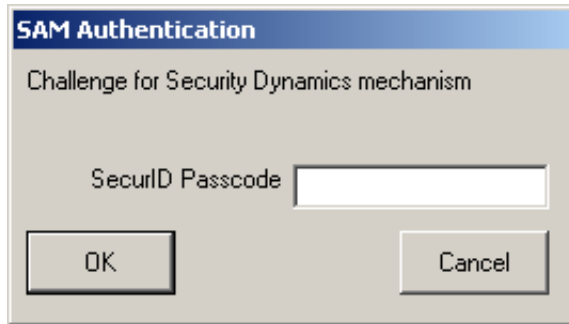
2. Select File # Options .. from the drop down menu. See Figure 7, “Selecting krb5.exe Options”

3. Fill in your Username, Password and Realm. See Figure 12, “Obtain TGT with SecurID and krb5.exe”

Figure 12. Obtain TGT with SecurID and krb5.exe

4. Click the Login Button. See Figure 9, “Obtain TGT with krb5.exe”
5. Enter your SecurID PIN into your SecurID Card. Type the response into the SecurID Passcode Text Box, then click the OK Button. See Figure 13, “Enter your SecurID Passcode”

Figure 13. Enter your SecurID Passcode



6. Enjoy your new tgt. See Figure 11, “Ticket Listing”

Change Kerberos Password

This section will take you through the steps needed to change your Kerberos password.


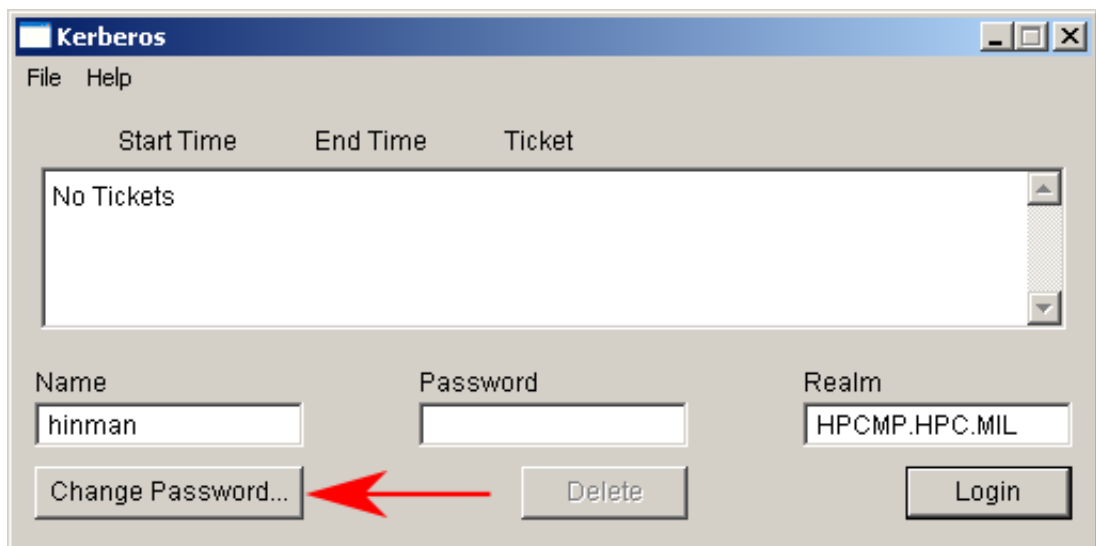
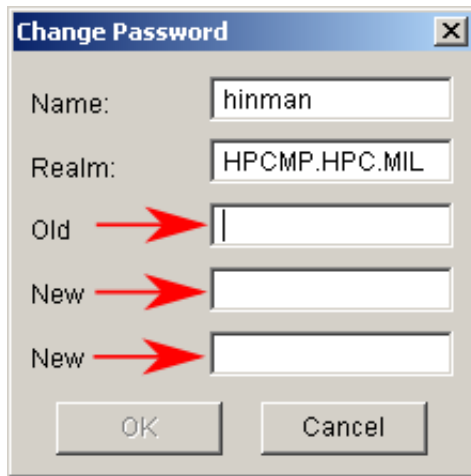
1. Click on the Start Menu # All Programs # HPCMP Kerberos # krb5.exe Icon or double click on the  krb5.exe application.
2. Click Change Password.... See Figure 14, “Change Kerberos Password with krb5.exe”

Figure 14. Change Kerberos Password with krb5.exe



3. Fill in the Old and New Password, then click on the OK Button. See Figure 15, “Change Kerberos Password Form”

Figure 15. Change Kerberos Password FormA Windows-style dialog box titled "Change Password" with a close button (X) in the top right corner. It contains five input fields: "Name:" with the text "hinman", "Realm:" with the text "HPCMP.HPC.MIL", "Old" with a red arrow pointing to it, "New" with a red arrow pointing to it, and another "New" with a red arrow pointing to it. At the bottom are "OK" and "Cancel" buttons.

4. If you are using PKINIT enter your Smart Card PIN when prompted and Click the OK Button. See Figure 10, "Enter your PIN"

If you are using SecurID enter your SecurID PIN into your SecurID card and enter the resulting Passcode in the SecurID Passcode Text box when prompted and Click the OK Button. See Figure 13, "Enter your SecurID Passcode"

5. Click the OK Button on the Password Changed Confirmation Window. See Figure 16, "Change Kerberos Password Confirmation"

Figure 16. Change Kerberos Password Confirmation

Delete all Kerberos Tickets

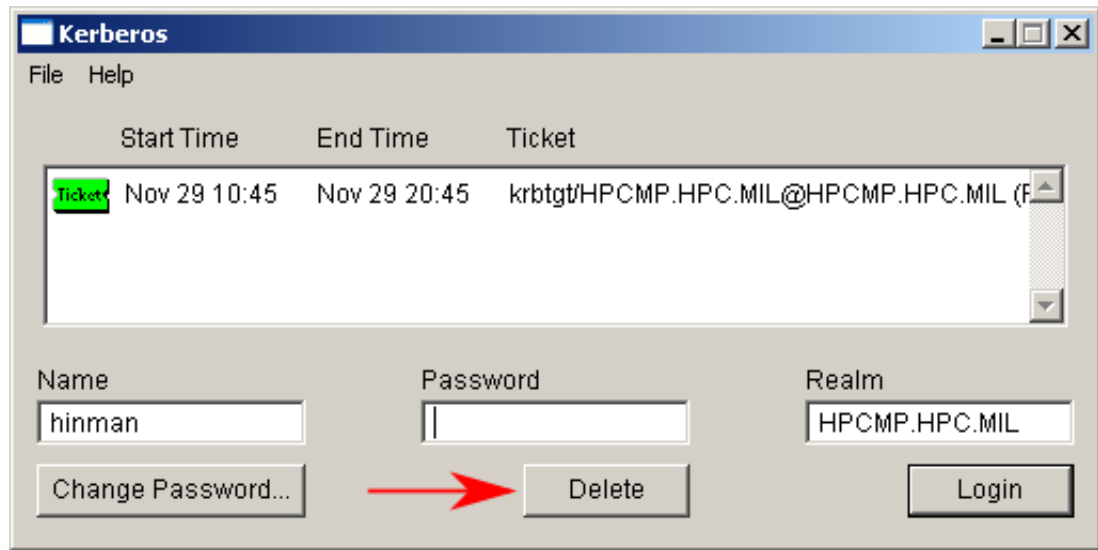
This section will take you through the steps needed to delete all of your Kerberos Tickets.

1. Click on the Start Menu # All Programs # HPCMP Kerberos # krb5.exe Icon or double click on the

krb5.exe application.



2. Click Delete. See Figure 17, "Delete all Kerberos Tickets with krb5.exe"

Figure 17. Delete all Kerberos Tickets with krb5.exe

Putty

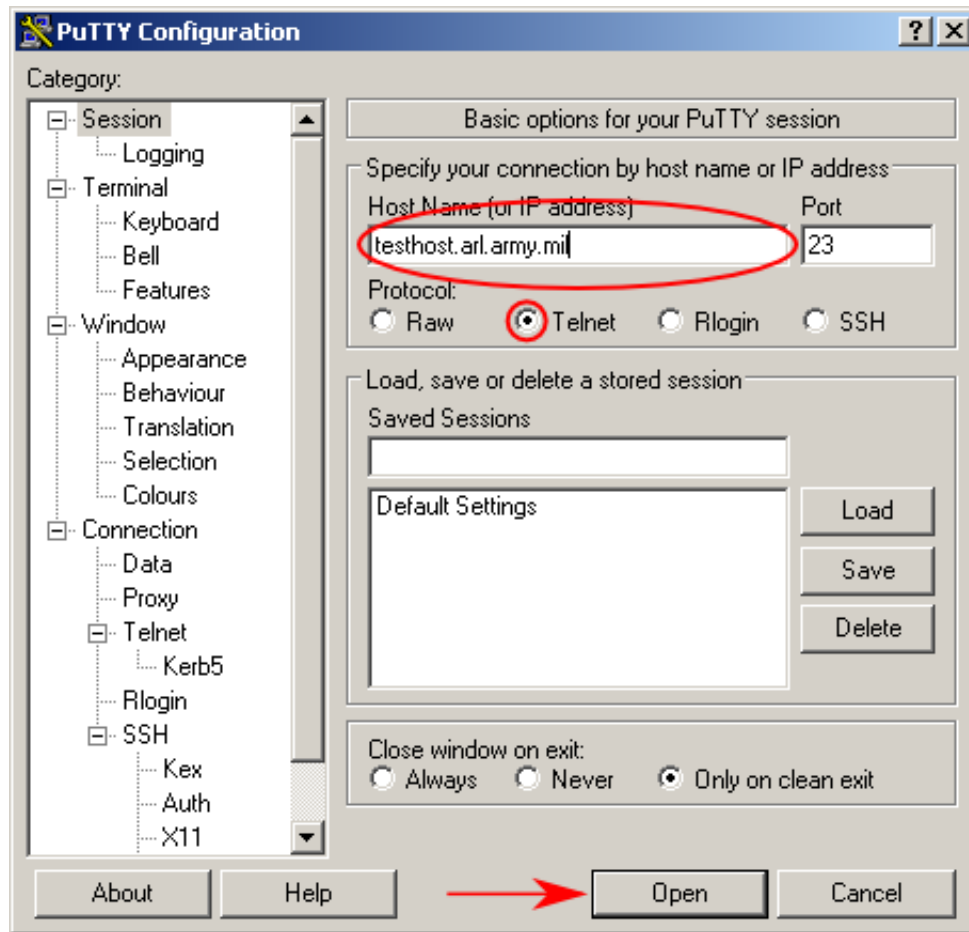
The following sections describe how to use Putty to connect to another host using either the Telnet or SSH protocol.

Telnet

1. Click on the Start Menu # All Programs # HPCMP Kerberos # Putty.exe Icon or double click on the

putty.exe application. 

2. Fill in the Host Name Text Box and Select the Telnet Radio Button. See Figure 18, "Putty Telnet"

Figure 18. Putty Telnet

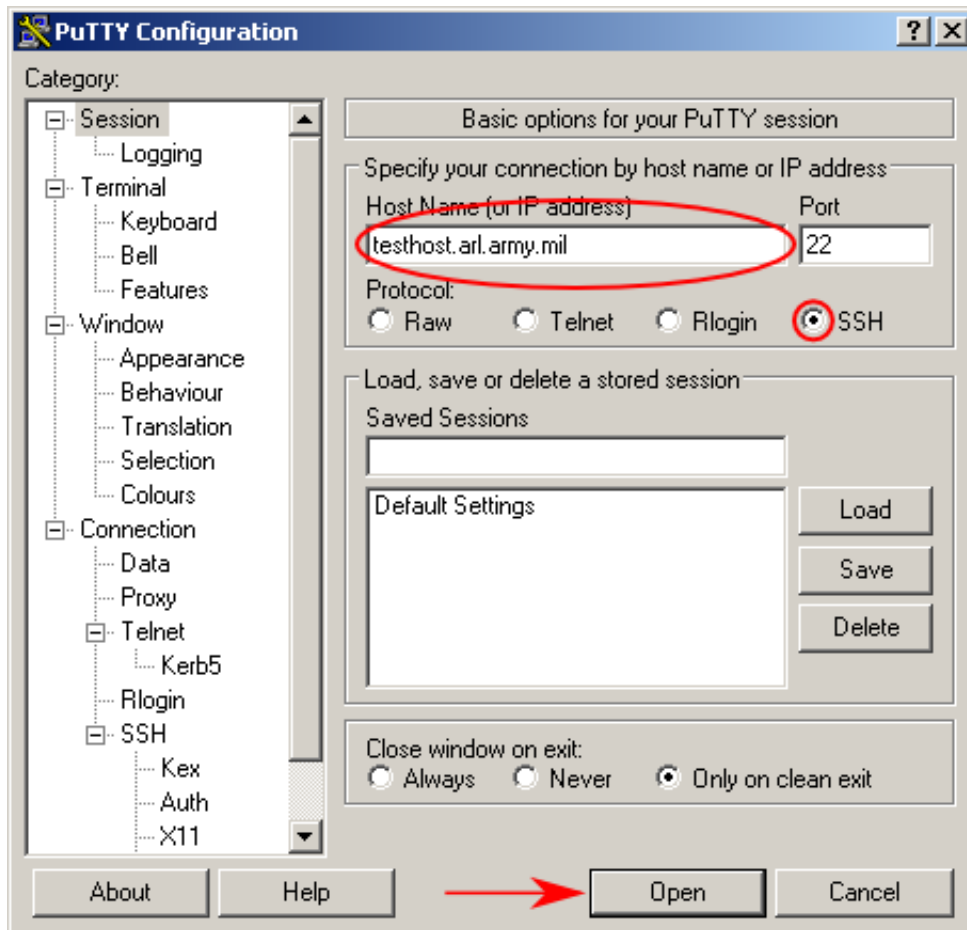
3. Click on the Open Button.
4. Fill in your username on the remote system if prompted.

SSH

1. Click on the Start Menu # All Programs # HPCMP Kerberos # Putty.exe Icon or double click on the

putty.exe application. 

2. Fill in the Host Name Text Box and Select the SSH Radio Button. See Figure 19, "Putty SSH"

Figure 19. Putty SSH

3. Click on the Open Button.
4. Fill in your username on the remote system if prompted.

For further information on using Putty please consult the Putty Help Files.

Filezilla

The following sections describe how to use Filezilla to transfer files to or from another host using either the FTP or SFTP protocol.

FTP

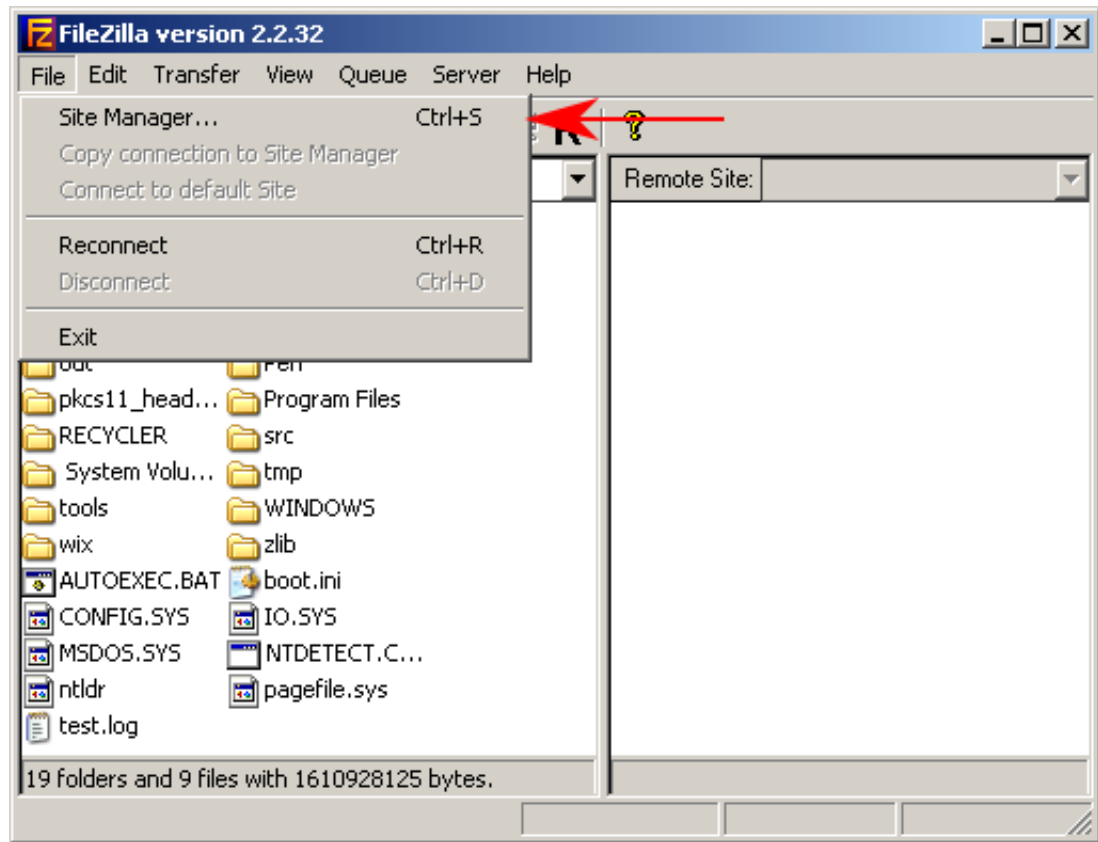
1. Filezilla is configured to use GSSAPI for authentication to sites that end in .mil or arsc.edu. If you are attempting to use GSSAPI authentication to another site, please see the section "GSS Support" in the Filezilla Help.
2. Click on the Start Menu # All Programs # HPCMP Kerberos # Filezilla.exe Icon or double click on

the FileZilla.exe application.

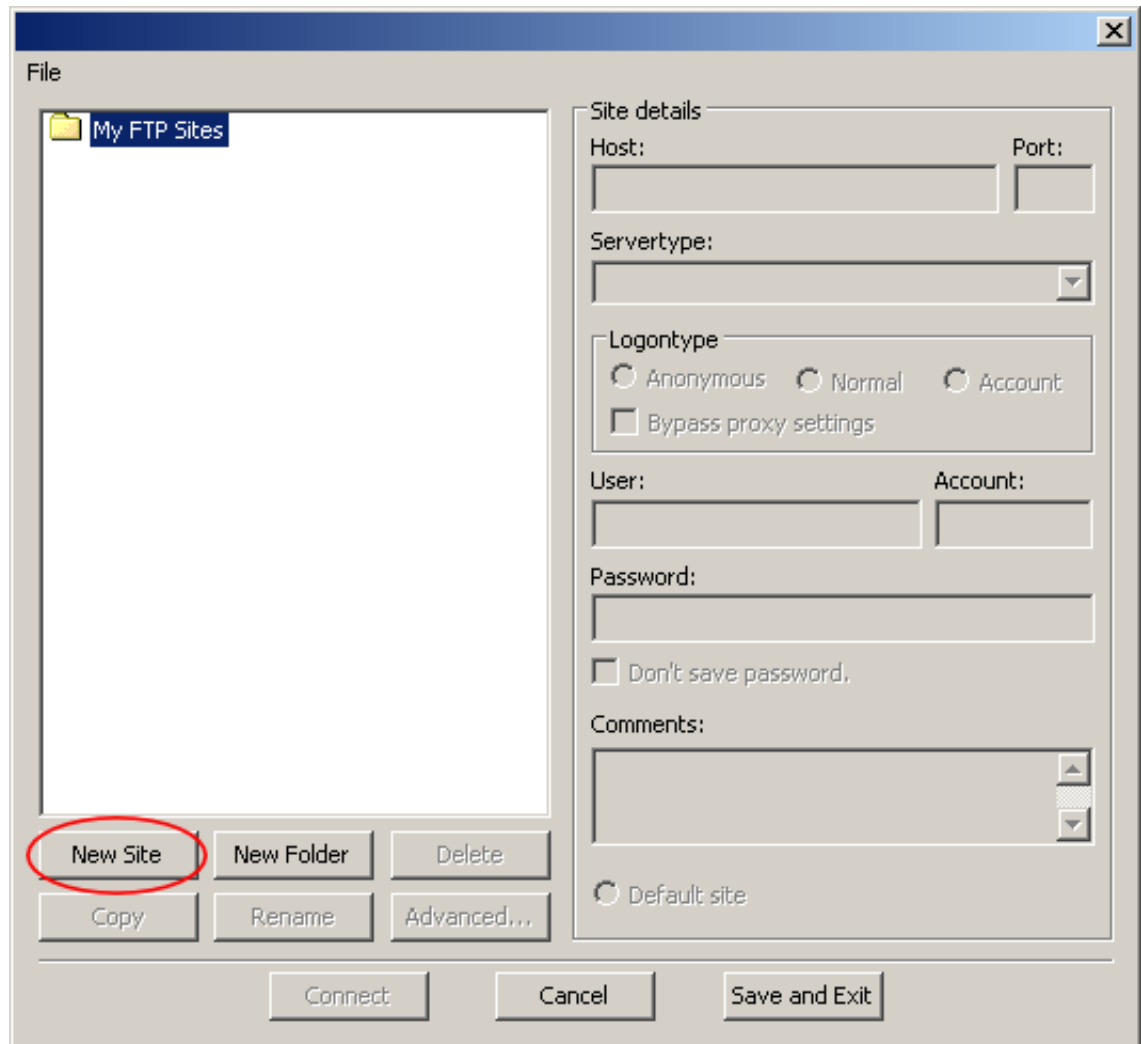


3. Select File # Site Manager... from the drop down menu. See Figure 20, “FileZilla SiteManager”

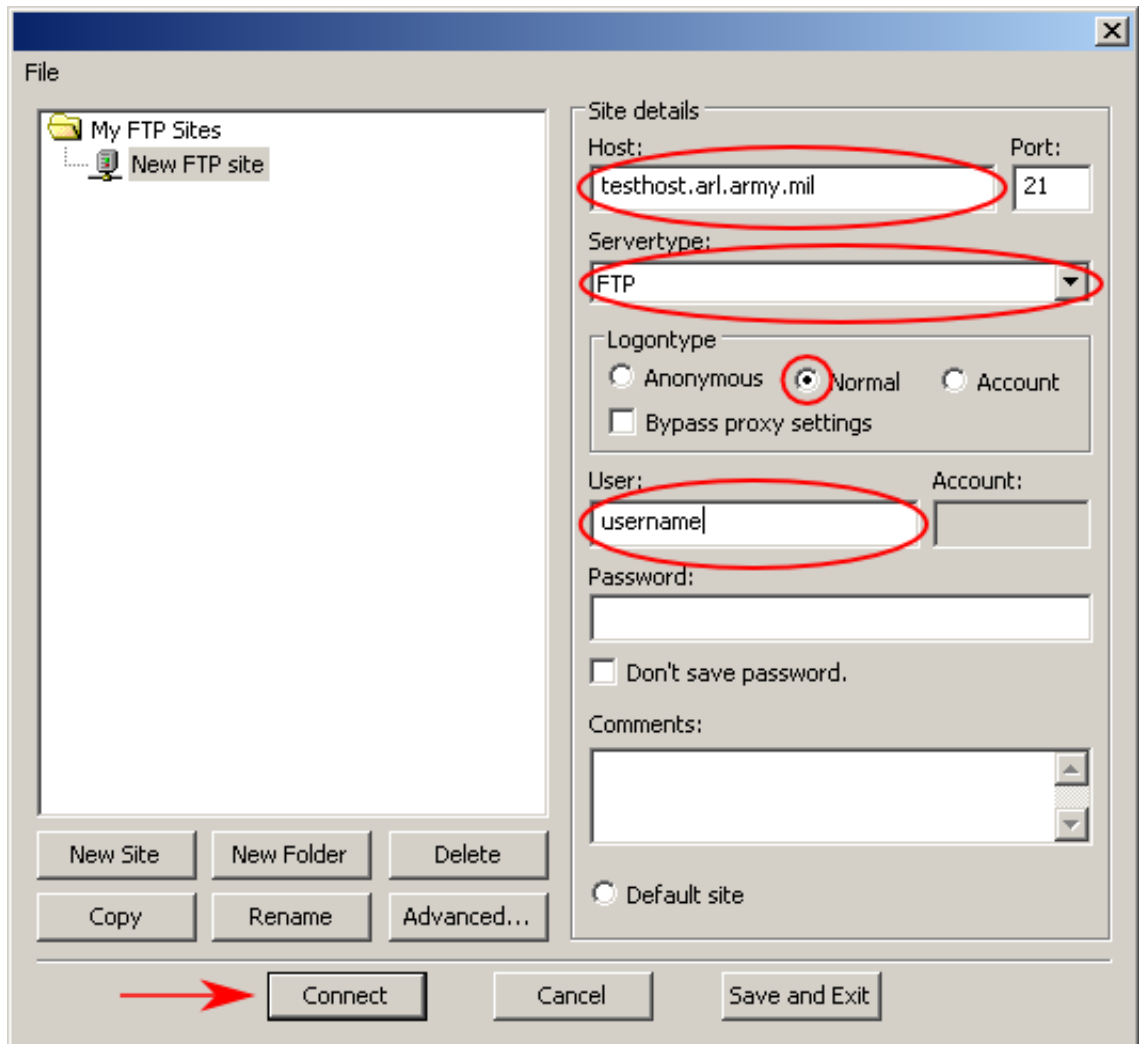
Figure 20. FileZilla SiteManager



4. Click on the New Site Button. See Figure 21, “Filezilla New Site”

Figure 21. Filezilla New Site

5. Fill in the Host: Text Box, Select FTP from the Servertype: Drop Down list, Click on the Normal Radio Button, and fill in your remote username in the User: Text Box. Then Click on the Connect Button. See Figure 22, "Filezilla Connect FTP" If desired the New FTP Site label can be renamed by clicking on the Rename Button.

Figure 22. Filezilla Connect FTP

SFTP


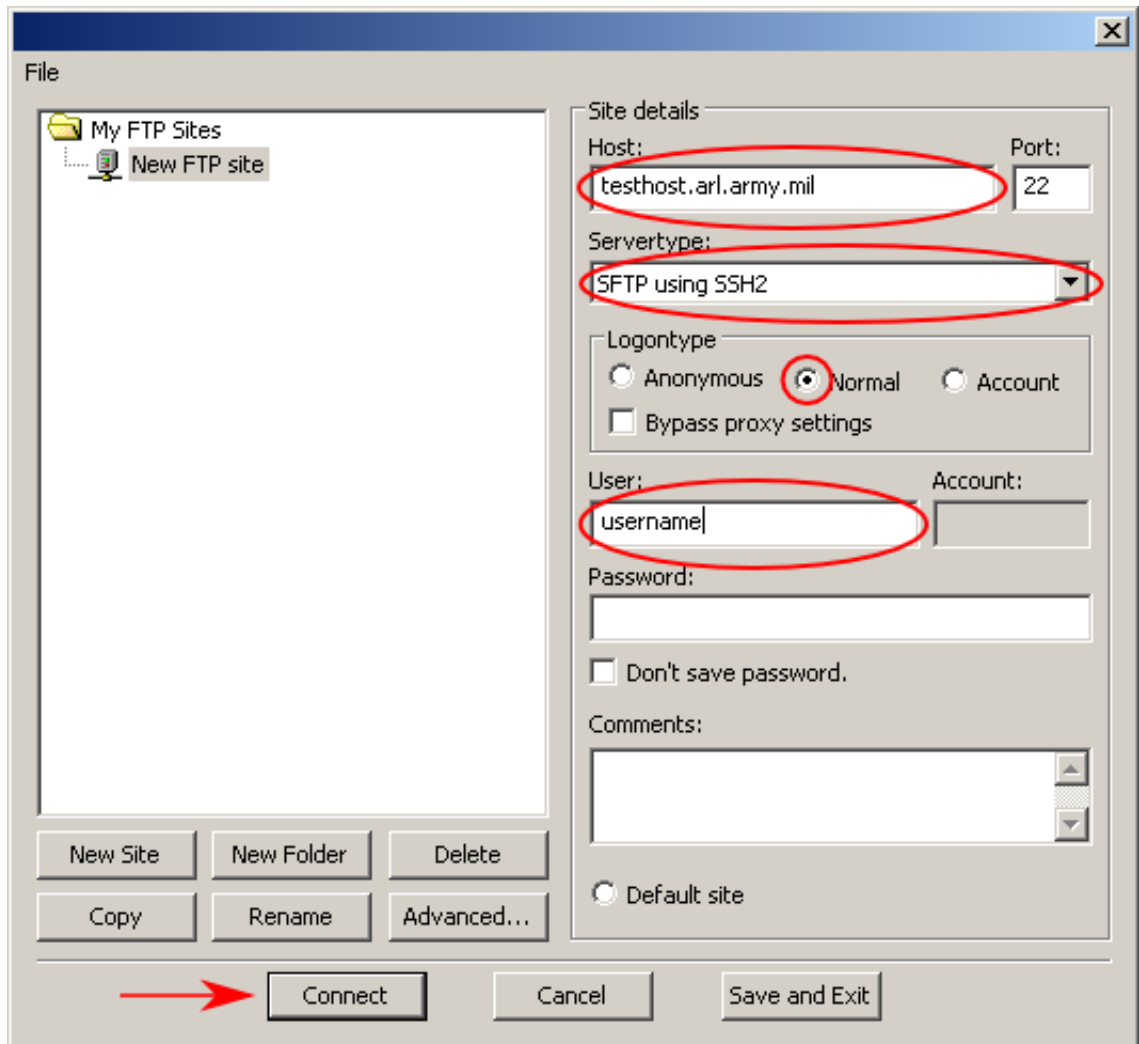
1. Click on the Start Menu # All Programs # HPCMP Kerberos # Filezilla.exe Icon or double click on the FileZilla.exe application. 
2. Select File # Site Manager... from the drop down menu. See Figure 20, “FileZilla SiteManager”
3. Click on the New Site Button. See Figure 21, “Filezilla New Site”
4. Fill in the Host: Text Box, Select SFTP using SSH2 from the Srvertype: Drop Down list, Click on the Normal Radio Button, and fill in your remote username in the User: Text Box. Then Click on the Connect Button. See Figure 23, “Filezilla Connect SFTP” If desired the New FTP Site can be renamed by clicking on the Rename Button.

Figure 23. Filezilla Connect SFTP

For further information on using Filezilla please consult the Filezilla Help Files.

Network Identity Manager

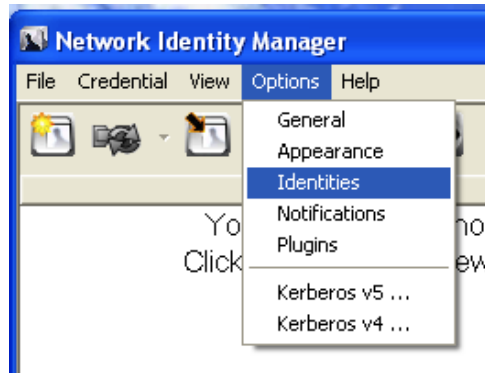
Secure Endpoints Inc. has begun working on a new program to acquire and delete Kerberos tgts. This new program is called "Network Identity Manager" and it is distributed with the MIT Kerberos for Windows.

Warning

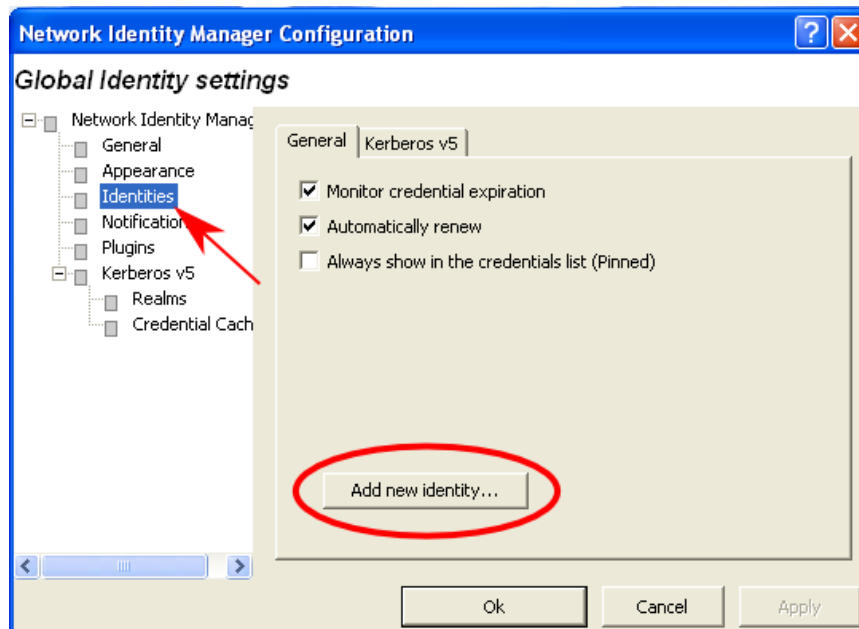
The HPCMP has begun work on modifying Network Identity Manager to work with our Hardware Preauth and Pkinit. However these efforts are in their earliest phases and this software should be considered beta quality. krb5.exe is still the only officially HPCMP supported way to get Kerberos tickets on the Microsoft Windows platform.

The following section describes configuring Network Identity Manager to acquire Kerberos tickets using PKINIT.

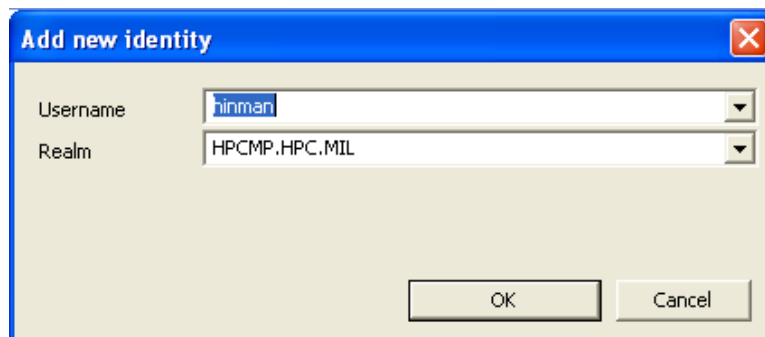
1. Click on the Start Menu # All Programs # HPCMP Kerberos # netidmgr Icon or double click on the netidmgr.exe application.
2. Click on the Options # Identities Menu.



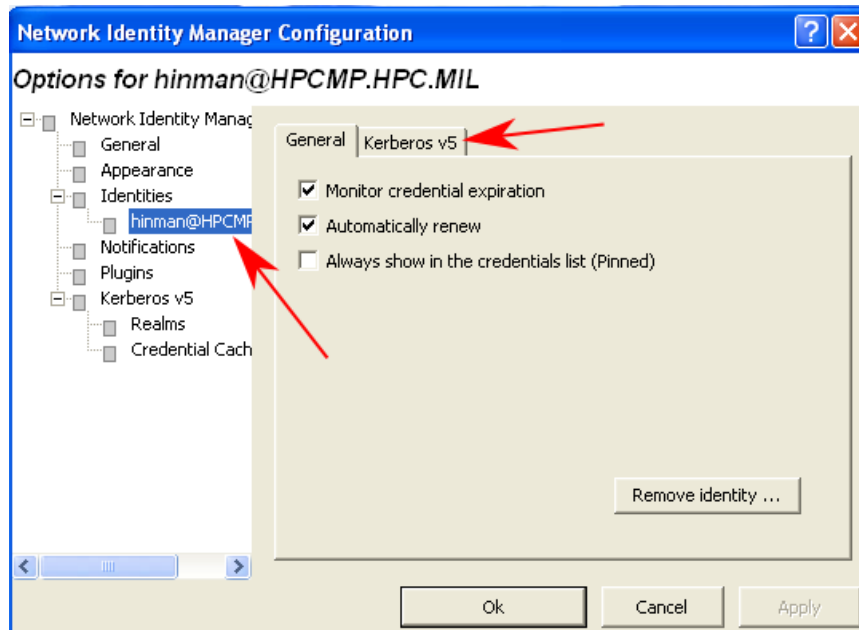
3. Click on the Identities in the left hand pane and click on the Add new identity... Button.



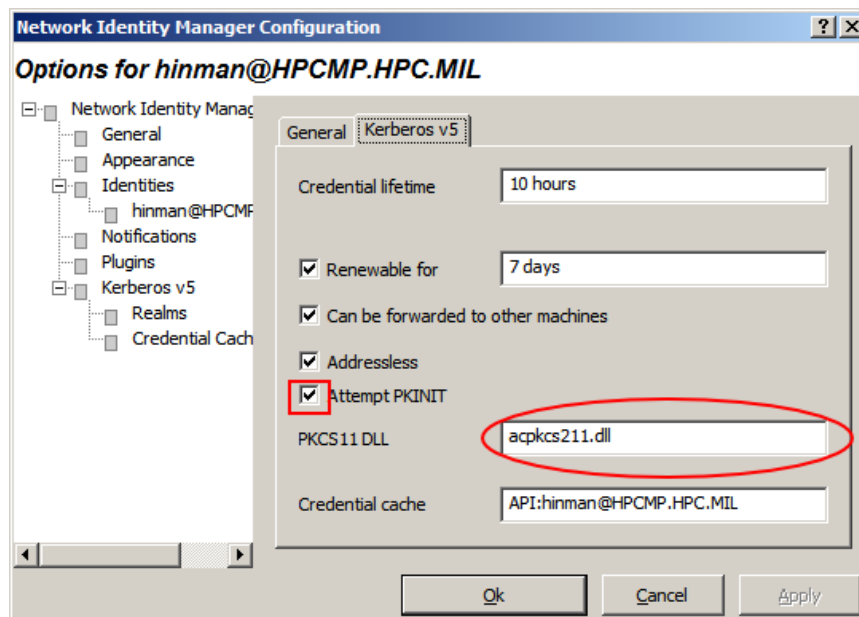
4. Enter your Username and Realm in the text boxes and click OK.



5. Due to a display bug you must now click OK to close the Options dialog box and then re-open it.
6. Select your newly created identity in the left hand pane and click on the Kerberos v5 tab in the right pane.



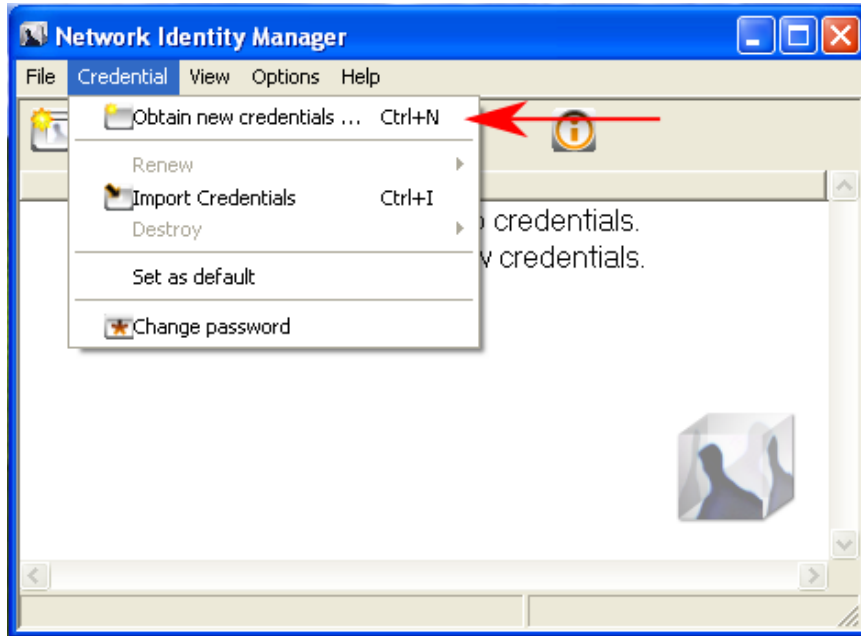
7. Select the Attempt PKINIT check box and fill in the PKCS11 DLL text box with the same value as found in your krb5.exe File # Options # PKCS11 DLL Textbox.



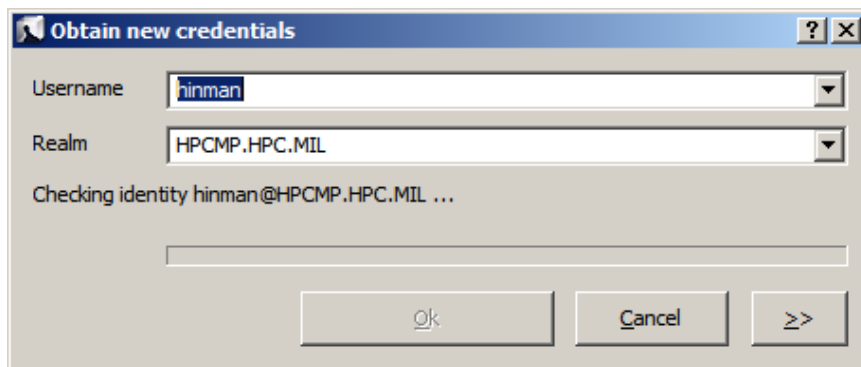
8. Click OK to close the dialog box.

To use Network Identity Manager to obtain a ticket follow these steps.

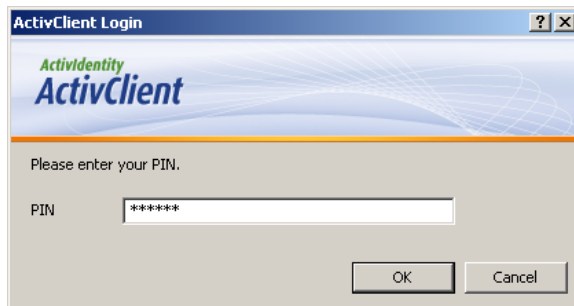
1. Click on the Credential # Obtain new credentials ... Menu Item.



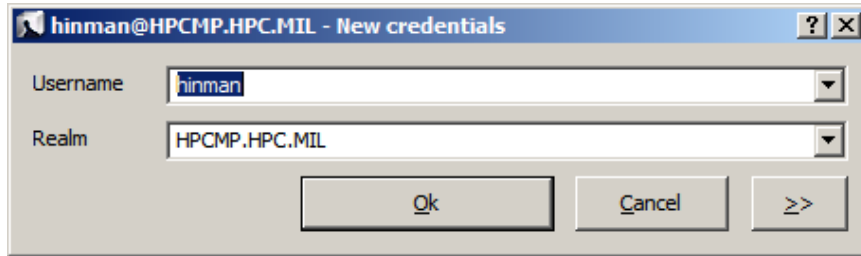
2. Fill in the username and realm text boxes. Network Identity Manager should show the text "Checking identity USERNAME@REALM".



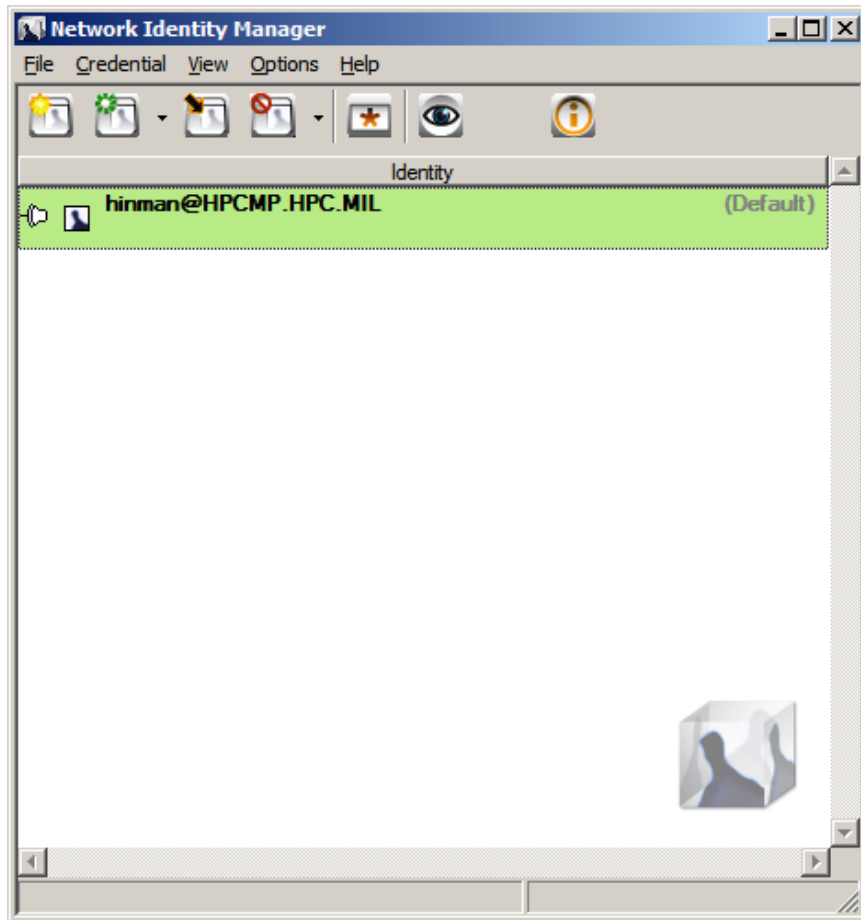
3. Enter your Smart Card PIN when prompted.



4. Click the OK button.



5. Finished.



Building From Source

Requirements

This section will take you through the steps needed to build and install the pkinit software on a UNIX system.

The following SmartCard components have been tested on a Solaris 10 system:

- libusb - user-space USB device management library.
- libccid - PC/SC driver for USB CCID Smart Card readers.

- `pcscd` - Middleware to access a Smart Card using PC/SC (daemon side).
- *CoolKey* - an open-source PKCS#11 smartcard library that can recognize CAC (Common Access Card).

Building PKINIT Kerberos on a UNIX System From Source

PKINIT Kerberos has been successfully built on the following systems; MAC OS X 10.4.x, MAC OS X 10.5.x, Linux, Solaris (Intel and Sparc), and IRIX.

The following steps will explain how to build the PKINIT kerberos binaries on a UNIX system using the source available at <http://www.hpcmo.hpc.mil/security/kerberos/>.

Note

The build of PKINIT Kerberos has been tested using the following development tools; autoconf version 2.61, automake version 1.9.6, and gcc version 3.4.6

1. Download the PKINIT Kerberos gzipped source tarball from <http://www.hpcmo.hpc.mil/security/kerberos/>.

2. Uncompress the source file into a working directory:

```
$ mkdir $HOME/source
$ cd $HOME/source
$ mv $HOME/HPCMP_RELEASE_20090211.tar.gz .
$ gunzip < HPCMP_RELEASE_20090211.tar.gz | tar xvf -
```

3. Go to the top of the source tree:

```
$ cd HPCMP_RELEASE_20090211
```

4. Create the configure scripts:

```
$ ./util/reconf
```

5. Create a build directory and go there:

```
$ mkdir pkinit_build
$ cd pkinit_build
```

6. Run the configure scripts from the build directory:

```
$ ../configure \
CFLAGS=-g --prefix=<dir> \
--enable-pkinit --with-pkinit-openssl=<dir> \
--enable-pkinit-matching --enable-pkinit-ocsp \
--with-pkinit-pkcs11=cryptoki \
--without-krb4 --enable-btree-db \
--enable-login-print-issue --enable-dns \
--enable-log-preauth-logins --with-kadmin-cryptocard \
--disable-krb4-compat --with-mulithomed-fixes \
--with-cryptocard-validate --disable-shared \
--enable-require-telnet-encrypt \
--enable-pipe-appserver --disable-file-cache
```

7. Compile the source from the build directory:

```
$ make
```

8. After ensuring there were no errors during the make, install the binaries:

```
$ umask 022
make install
```

Ensure that kinit and pkinit are in <DESTDIR>/bin. kinit is used to get a Kerberos ticket using SecurID or CryptoCard, pkinit is used to get a Kerberos ticket using a Smart Card.

You will need to get a tarball of the DOD certificates and put them in a directory that will be referenced by the krb5.conf file. Without access to these certificates, the pkinit software has no way of verifying the certificates on the Smart Card. This will result in an error when trying to get a Kerberos ticket using the Smart Card.

Setting up the /etc/krb5.conf file

The krb5.conf file can be downloaded from <https://www.hpcmo.hpc.mil/security/kerberos/private/> and should be placed in the /etc directory. If you have root on the system, install the krb5.conf file in /etc/krb5.conf. If you don't have root or prefer not to modify the system you're using, place the krb5.conf file wherever you want, but set the environment variable KRB5_CONFIG to point to this location.

For example, if you place the krb5.conf file in /usr/people/somebody/krb5.conf, you should do:

If you are using tcsh/csh:

```
% setenv KRB5_CONFIG /usr/people/somebody/krb5.conf
```

If you are using ksh/sh:

```
$ KRB5_CONFIG=/usr/people/somebody/krb5.conf
export KRB5_CONFIG
```

Several changes may have to be made to the krb5.conf file based on which directory the software was placed in.

If your home realm is not HPCMP.HPC.MIL, you will need to modify the default realm (under [libdefaults] section in the krb5.conf file) to your home realm. If you don't know your home realm, please contact the center from which you received your Kerberos login for the information.

The kit includes the required DOD and HPCMP CA certificates for proper authentication. If you put the krb5 directory in a directory other than /usr/local/krb5, you will have to change the following two lines in krb5.conf:

```
pkinit_anchors = DIR:/usr/local/krb5/etc/CA
pkinit_pool = DIR:/usr/local/krb5/etc/CERTS
```

If you are using coolkey as the PKCS11 module, in the krb5.conf file you need to identify the directory the coolkey library is located in.

```
pkinit_identities = PKCS11:/usr/lib/pkcs11/libcoolkeypk11.so
```

The following line in the krb5.conf file is needed to choose the proper key on the Smart Card:

```
pkinit_cert_match = <SAN>^[0-9]{10}@mil$
```

```
pkinit_cert_match = <SAN>^[0-9]{10}@hpcmp$
```

Getting a Kerberos Ticket Using pkinit

After the software has been installed and the `krb5.conf` file has been configured, you need to perform the following steps to login to the HPCMP systems:

1. Launch a terminal or console.

2. type **kshell** and hit Return>

```
$ kshell
```

3. Run **pkinit** with optional USERNAME and REALM. Hit Return>

```
$ pkinit tproue@HPCMP.HPC.MIL
```

4. Enter you Smart Card PIN and hit Return

```
PROUE.THOMAS.MARK.1212298626 PIN:
```

5. **Note**

After inputting the PIN associated with your Smart Card, pkinit should return with no errors. To actually see your Kerberos Ticket Granting Ticket, or any other tickets you may have acquired, run **klist**

```
$ klist
```

```
Ticket cache: PIPE:1023
```

```
Default principal: tproue@HPCMP.HPC.MIL
```

```
Valid starting      Expires              Service principal
09/06/07 11:20:56   09/06/07 21:20:56   krbtgt/HPCMP.HPC.MIL@HPCMP.HPC.MIL
renew until 09/06/07 11:20:56
```

6. Next, login to the appropriate system at an HPCMP center. You should use a Kerberized **ssh**, which can be found at the *HPCMP Kerberos Site* [<http://www.hpcmo.hpc.mil/security/kerberos/>]. You could also use either **krlogin** or **ktelnet** if the remote server allows it. Not all of the systems in the HPCMP centers are set up to accept incoming telnet or r-command connections. You can log in to any HPCMP resource you have an account on.

```
$ ssh hostname
```

```
or
```

```
$ ktelnet hostname
```

```
or
```

```
$ krlogin hostname
```

Note

If you do not know the hostname, please contact system administrator for the information.

Note

You can also use **sftp**, **scp**, **kftp** or **krftp** to transfer files into HPCMP systems, and **kpasswd** to change your password.

Troubleshooting pkinit

This section will look at some common problems that may be seen when using pkinit.

Note

It is possible to get verbose debugging information regarding PKINIT. For the CLI application pkinit add the following option: `-X pkinit_debug=stdout`, this works for Unix variants and Microsoft Windows. For the Microsoft Windows GUI application krb5.exe, specify a log file in the File # Options .. Dialog box.

Before running pkinit, you should ensure that the proper software and hardware has been installed on the workstation. A very good reference for looking at PKI and CAC can be found at *NRL DoD PKI notes* [<https://airborne.nrl.navy.mil/PKI/>].

Common Problems and Solutions

1. The following error is displayed:

```
pkinit(v5): Preauthentication failed while getting initial credentials
```

SOLUTION: Check to see if your card is plugged in.

2. The following error is displayed:

```
pkinit(v5): Client not trusted while getting initial credentials
```

SOLUTION: Check for the lines:

```
pkinit_cert_match = <SAN>^[0-9]{10}@mil$  
pkinit_cert_match = <SAN>^[0-9]{10}@hpcmp$
```

in /etc/krb5.conf or krb5.ini

3. The following error is displayed:

```
pkinit(v5): Generic preauthentication failure while getting initial credentials
```

SOLUTION: Your Kerberos kit may be too old and you don't have all of the DoD certificate authority certificates installed. Download and install the latest kit from the *HPCMP Kerberos and SecurID Center* [<http://www.hpcmo.hpc.mil/security/kerberos/>] or download/install the latest certificates from the *HPCMP Certificate Repository* [<http://www.hpcmo.hpc.mil/security/pki/crt/pem/>].

4. You are prompted for your PIN twice then a failure message is displayed:

```
Common Access Card PIN:
```

```
Common Access Card PIN:
```

```
pkinit(v5): Permission denied while getting initial credentials
```

SOLUTION 1: Make sure the permissions are correct in 'pkinit_anchors' and 'pkinit_pool'. The permissions on the certificates on a UNIX based system should be 644. For Windows they must be readable by the user.

SOLUTION 2: Check to see if your Smart Card is LOCKED. If you have been locked out your CAC PIN (after three incorrect tries) you should contact your local help desk to determine the location of the nearest CAC PIN Reset (CPR) station. You should go to an ID Card Issuance Facility (your central

processing/badge office or Local Registration Authority) to have your PIN reset only if a CPR station is unavailable. If your hToken is locked please contact CCAC.

5. Sometimes after the screen lock activates on Windows or MAC, the CAC is not recognized.

SOLUTION: Pull out the CAC and put it back in.

6. The light on the Reader sometimes continues to blink after the CAC is pulled out.

SOLUTION: Unplug the Reader from the USB port and plug it back in.

7. The CAC is plugged into the reader, but the light does not blink.

SOLUTION: On Unix Systems, ensure that pcsd is running. For Microsoft Windows, ensure the ActivClient Software is started.

8. The following error is displayed:

```
pkinit(v5): KDC name mismatch while getting initial credentials
```

SOLUTION: Make sure the `pkinit_krb_hostname` of the realm that you are connecting to is the canonical name of the kdc in the `krb5.conf` or `krb5.ini`. Example: `kdc1.hpcmp.hpc.mil` is mapped to `kdc5.asc.hpc.mil`. Therefore, the `pkinit_krb5_hostname` must equal `kdc5.asc.hpc.mil`.

9. The following error is displayed:

```
pkinit(v5): Client name mismatch while getting initial credentials
```

SOLUTION: Ensure the CAC has been registered via pIE.

The CAC registration process is as follows:

- a. Go to the Portal for Information Environment (pIE) at <http://www.hpcmo.hpc.mil/Htdocs/IE/index.html>
- b. Login via Kerberos/SecurID
- c. Select "User Information Environment"
- d. Select "Register My Smart Card (CAC)"

After the user registers their CAC, it may take up to 48 hours before the CAC will be associated with their Kerberos principal. If the user has already registered their CAC and waited 48 hours, check with the KDC admins to see if their CN(Common Name) was added to their Kerberos Account. If it has not, have the KDC admins find and run the directives for that user. If no directive exists, the user will have to re-register.

10. On a MAC after updating the firmware on the reader Firefox crashes every time you try to go to a PKI site. This sometimes occurs with a 64K CAC and may need to have the ATR values in the `Info.plist` updated in the `commonAccessCard.bundle`.

After removing the CAC from the reader do the following:

```
# pcsctool
Select the appropriate token driver:
-----
1.      commonAccessCard.bundle
```

2. GSCISPlugin.bundle
3. mscMuscleCard.bundle
4. slbCryptoflex.bundle

Enter the number: 1

Insert your token in: CCID Smart Card Reader 0 0

Token support updated successfully

Manpages

Name

`ftp` — ARPANET file transfer program

Synopsis

`ftp [-v] [-d] [-i] [-n] [-g] [-k realm] [-f] [-x] [-u] [-t] [host]`

DESCRIPTION

FTP is the user interface to the ARPANET standard File Transfer Protocol. The program allows a user to transfer files to and from a remote network site.

OPTIONS

Options may be specified at the command line, or to the command interpreter.

- `-v` Verbose option forces **ftp** to show all responses from the remote server, as well as report on data transfer statistics.
- `-n` Restrains **ftp** from attempting “auto-login” upon initial connection. If auto-login is enabled, **ftp** will check the `.netrc` (see below) file in the user's home directory for an entry describing an account on the remote machine. If no entry exists, **ftp** will prompt for the remote machine login name (default is the user identity on the local machine), and, if necessary, prompt for a password and an account with which to login.
- `-u` Restrains **ftp** from attempting “auto-authentication” upon initial connection. If auto-authentication is enabled, **ftp** attempts to authenticate to the FTP server by sending the AUTH command, using whichever authentication types are locally supported. Once an authentication type is accepted, an authentication protocol will proceed by issuing ADAT commands. This option also disables auto-login.
- `-i` Turns off interactive prompting during multiple file transfers.
- `-d` Enables debugging.
- `-g` Disables file name globbing.
- `-k realm` When using Kerberos v4 authentication, gets tickets in *realm*.
- `-f` Causes credentials to be forwarded to the remote host.
- `-x` Causes the client to attempt to negotiate encryption (data and command protection levels “private”) immediately after successfully authenticating.
- `-t` Enables packet tracing.

COMMANDS

The client host with which **ftp** is to communicate may be specified on the command line. If this is done, **ftp** will immediately attempt to establish a connection to an FTP server on that host; otherwise, **ftp** will enter its command interpreter and await instructions from the user. When **ftp** is awaiting commands from the user the prompt “ftp>” is provided to the user. The following commands are recognized by **ftp**:

- `args[]` Invoke an interactive shell on the local machine. If there are arguments, the first is taken to be a command to execute directly, with the rest of the arguments as its arguments.

<i>args</i>]	Execute the macro <i>macro-name</i> that was defined with the <i>macdef</i> command. Arguments are passed to the macro unglobbed.
<i>passwd</i>]	Supply a supplemental password required by a remote system for access to resources once a login has been successfully completed. If no argument is included, the user will be prompted for an account password in a non-echoing input mode.
<i>remote-file</i>]	Append a local file to a file on the remote machine. If <i>remote-file</i> is left unspecified, the local file name is used in naming the remote file after being altered by any <i>ntrans</i> or <i>nmap</i> setting. File transfer uses the current settings for <i>type</i> , <i>format</i> , <i>mode</i> , and <i>structure</i> .
<i>ascii</i>	Set the file transfer <i>type</i> to network <i>ASCII</i> . This is the default type.
<i>auth</i>	Uses the <i>AUTH/ADAT</i> mechanism (see above) to authenticate the user <i>AUTH</i>
<i>bell</i>	Arrange that a bell be sounded after each file transfer command is completed.
<i>binary</i>	Set the file transfer <i>type</i> to support binary file transfer.
<i>bye</i>	Terminate the FTP session with the remote server and exit ftp . An end of file will also terminate the session and exit.
<i>case</i>	Toggle remote computer file name case mapping during <i>mget</i> commands. When <i>case</i> is on (default is off), remote computer file names with all letters in upper case are written in the local directory with the letters mapped to lower case.
<i>ccc</i>	Turn off integrity protection on the command channel. This command must be sent integrity protected, and must be preceded by a successful <i>ADAT</i> command. Since turning off integrity protection potentially allows an attacker to insert commands onto the command channel, some FTP servers may refuse to honor this command.
<i>remote-directory</i>	Change the working directory on the remote machine to <i>remote-directory</i> .
<i>cdup</i>	Change the remote machine working directory to the parent of the current remote machine working directory.
<i>file-name</i>	Change the permission modes of the file <i>file-name</i> on the remote system to <i>mode</i> .
<i>clear</i>	Set the protection level on data transfers to “clear”. If no <i>ADAT</i> command succeeded, then this is the default protection level.
<i>close</i>	Terminate the FTP session with the remote server, and return to the command interpreter. Any defined macros are erased.
<i>protection-level</i>]	Set the protection level on commands to <i>protection-level</i> . The valid protection levels are “clear” for unprotected commands, “safe” for commands integrity protected by cryptographic checksum, and “private” for commands confidentiality and integrity protected by

encryption. If an ADAT command succeeded, then the default command protection level is “safe”, otherwise the only possible level is “clear”. If no level is specified, the current level is printed. *cprotect clear* is equivalent to the *ccc* command.

<i>cr</i>	Toggle carriage return stripping during ascii type file retrieval. Records are denoted by a carriage return/linefeed sequence during ascii type file transfer. When <i>cr</i> is on (the default), carriage returns are stripped from this sequence to conform with the UNIX single linefeed record delimiter. Records on non-UNIX remote systems may contain single linefeeds; when an ascii type transfer is made, these linefeeds may be distinguished from a record delimiter only when <i>cr</i> is off.
<i>remote-file</i>	Delete the file <i>remote-file</i> on the remote machine.
<i>debug-value</i>]	Toggle debugging mode. If an optional <i>debug-value</i> is specified it is used to set the debugging level. When debugging is on, ftp prints each command sent to the remote machine, preceded by the string `-->'
<i>local-file</i>]	Print a listing of the directory contents in the directory, <i>remote-directory</i> , and, optionally, placing the output in <i>local-file</i> . If interactive prompting is on, ftp will prompt the user to verify that the last argument is indeed the target local file for receiving <i>dir</i> output. If no directory is specified, the current working directory on the remote machine is used. If no local file is specified, or <i>local-file</i> is `-', output comes to the terminal.
<i>disconnect</i>	A synonym for <i>close</i> .
<i>format</i>	Set the file transfer <i>form</i> to <i>format</i> . The default format is “file”.
<i>local-file</i>]	Retrieve the file <i>remote-file</i> and store it on the local machine. If the local file name is not specified, it is given the same name it has on the remote machine, subject to alteration by the current <i>case</i> , <i>ntrans</i> , and <i>nmap</i> settings. The current settings for <i>type</i> , <i>form</i> , <i>mode</i> , and <i>structure</i> are used while transferring the file.
<i>glob</i>	Toggle filename expansion for <i>mdelete</i> , <i>mget</i> , and <i>mput</i> . If globbing is turned off with <i>glob</i> , the file name arguments are taken literally and not expanded. Globbing for <i>mput</i> is done as in <i>csh</i> (1). For <i>mdelete</i> and <i>mget</i> , each remote file name is expanded separately on the remote machine and the lists are not merged. Expansion of a directory name is likely to be different from expansion of the name of an ordinary file: the exact result depends on the foreign operating system and ftp server, and can be previewed by doing `m!s remote-files -' Note: <i>mget</i> and <i>mput</i> are not meant to transfer entire directory subtrees of files. That can be done by transferring a <i>tar</i> (1) archive of the subtree (in binary mode).
<i>hash</i>	Toggle hash-sign (“#”) printing for each data block transferred. The size of a data block is 1024 bytes.
<i>command</i>]	Print an informative message about the meaning of <i>command</i> . If no argument is given, ftp prints a list of the known commands.

<i>seconds</i>]	Set the inactivity timer on the remote server to <i>seconds</i> seconds. If <i>seconds</i> is omitted, the current inactivity timer is printed.
<i>bytes</i>]	Set the TCP buffer sizes on the local machine to <i>bytes</i> bytes. If <i>bytes</i> is zero, the TCP buffer sizes are not set; the system default is used. If <i>bytes</i> is omitted, the current TCP buffer sizes of the local machine are printed. Usually setting the TCP buffer sizes sets the TCP window size. Use with the <i>rbufsize</i> command.
<i>directory</i>]	Change the working directory on the local machine. If no <i>directory</i> is specified, the user's home directory is used.
<i>local-file</i>]	Print a listing of the contents of a directory on the remote machine. The listing includes any system-dependent information that the server chooses to include; for example, most UNIX systems will produce output from the command <code>`ls -l'</code> . (See also <i>nlist</i> .) If <i>remote-directory</i> is left unspecified, the current working directory is used. If interactive prompting is on, ftp will prompt the user to verify that the last argument is indeed the target local file for receiving <i>ls</i> output. If no local file is specified, or if <i>local-file</i> is <code>`-'</code> , the output is sent to the terminal.
<i>macro-name</i>	Define a macro. Subsequent lines are stored as the macro <i>macro-name</i> ; a null line (consecutive newline characters in a file or carriage returns from the terminal) terminates macro input mode. There is a limit of 16 macros and 4096 total characters in all defined macros. Macros remain defined until a <i>close</i> command is executed. The macro processor interprets <code>`\$'</code> and <code>`\'</code> as special characters. A <code>`\$'</code> followed by a number (or numbers) is replaced by the corresponding argument on the macro invocation command line. A <code>`\$'</code> followed by an <code>`i'</code> signals that macro processor that the executing macro is to be looped. On the first pass <code>`\$i'</code> is replaced by the first argument on the macro invocation command line, on the second pass it is replaced by the second argument, and so on. A <code>`\'</code> followed by any character is replaced by that character. Use the <code>`\'</code> to prevent special treatment of the <code>`\$'</code> .
<i>remote-files</i>]	Delete <i>remote-files</i> on the remote machine.
<i>local-file</i>	Like <i>dir</i> , except multiple remote files may be specified. If interactive prompting is on, ftp will prompt the user to verify that the last argument is indeed the target local file for receiving <i>mdir</i> output.
<i>remote-files</i>	Expand the <i>remote-files</i> on the remote machine and do a <i>get</i> for each file name thus produced. See <i>glob</i> for details on the filename expansion. Resulting file names will then be processed according to <i>case</i> , <i>ntrans</i> , and <i>nmap</i> settings. Files are transferred into the local working directory, which can be changed with <code>`lcd directory'</code> ; new local directories can be created with <code>`! mkdir directory'</code> .
<i>directory-name</i>	Make a directory on the remote machine.
<i>local-file</i>	Like <i>nlist</i> , except multiple remote files may be specified, and the <i>local-file</i> must be specified. If interactive prompting is on, ftp will

	prompt the user to verify that the last argument is indeed the target local file for receiving <i>mls</i> output.
<i>mode-name</i>]	Set the file transfer <i>mode</i> to <i>mode-name</i> . The default mode is “stream” mode.
<i>file-name</i>	Show the last modification time of the file on the remote machine.
<i>local-files</i>	Expand wild cards in the list of local files given as arguments and do a <i>put</i> for each file in the resulting list. See <i>glob</i> for details of filename expansion. Resulting file names will then be processed according to <i>ntrans</i> and <i>nmap</i> settings.
<i>file-name</i>	Get the file only if the modification time of the remote file is more recent than the file on the current system. If the file does not exist on the current system, the remote file is considered <i>newer</i> . Otherwise, this command is identical to <i>get</i> .
<i>local-file</i>]	Print a list of the files in a directory on the remote machine. If <i>remote-directory</i> is left unspecified, the current working directory is used. If interactive prompting is on, ftp will prompt the user to verify that the last argument is indeed the target local file for receiving <i>nlist</i> output. If no local file is specified, or if <i>local-file</i> is <i>`-'</i> , the output is sent to the terminal.
<i>outpattern</i>]	Set or unset the filename mapping mechanism. If no arguments are specified, the filename mapping mechanism is unset. If arguments are specified, remote filenames are mapped during <i>mput</i> commands and <i>put</i> commands issued without a specified remote target filename. If arguments are specified, local filenames are mapped during <i>mget</i> commands and <i>get</i> commands issued without a specified local target filename. This command is useful when connecting to non-UNIX remote computer with different file naming conventions or practices. The mapping follows the pattern set by <i>inpattern</i> and <i>outpattern</i> . [<i>Inpattern</i>] is a template for incoming filenames (which may have already been processed according to the <i>ntrans</i> and <i>case</i> settings). Variable templating is accomplished by including the sequences <i>`\$1'</i> , <i>`\$2'</i> , ..., <i>`\$9'</i> in <i>inpattern</i> . Use <i>`\'</i> to prevent this special treatment of the <i>`\$'</i> character. All other characters are treated literally, and are used to determine the <i>nmap</i> [<i>inpattern</i>] variable values. For example, given <i>inpattern</i> <i>\$1.\$2</i> and the remote file name "mydata.data", <i>\$1</i> would have the value "mydata", and <i>\$2</i> would have the value "data". The <i>outpattern</i> determines the resulting mapped filename. The sequences <i>`\$1'</i> , <i>`\$2'</i> , <i>inpattern</i> template. The sequence <i>`\$0'</i> is replaced by the original filename. Additionally, the sequence <i>`[seq2]'</i> is replaced by <i>[seq1]</i> if <i>seq1</i> is not a null string; otherwise it is replaced by <i>seq2</i> . For example, the command

`nmap $1.$2.$3 [$1,$2].[$2,file]`

would yield the output filename "myfile.data" for input filenames "myfile.data" and "myfile.data.old", "myfile.file" for the input filename "myfile", and "myfile.myfile" for the input filename

".myfile". Spaces may be included in *outpattern*, as in the example: ``nmap $1 sed "s/ *$/" > $1'`. Use the ``\'` character to prevent special treatment of the ``$'`, ``['`, and ``.'` characters.

outchars]]

Set or unset the filename character translation mechanism. If no arguments are specified, the filename character translation mechanism is unset. If arguments are specified, characters in remote filenames are translated during *mput* commands and *put* commands issued without a specified remote target filename. If arguments are specified, characters in local filenames are translated during *mget* commands and *get* commands issued without a specified local target filename. This command is useful when connecting to a non-UNIX remote computer with different file naming conventions or practices. Characters in a filename matching a character in *inchars* are replaced with the corresponding character in *outchars*. If the character's position in *inchars* is longer than the length of *outchars*, the character is deleted from the file name.

port] [-forward]

Establish a connection to the specified *host* FTP server. An optional port number may be supplied, in which case, **ftp** will attempt to contact an FTP server at that port. If the *auto-authenticate* option is on (default), **ftp** will attempt to authenticate to the FTP server by sending the AUTH command, using whichever authentication types which are locally supported. Once an authentication type is accepted, an authentication protocol will proceed by issuing ADAT commands. If the *auto-login* option is on (default), **ftp** will also attempt to automatically log the user in to the FTP server (see below). If the *-forward* option is specified, **ftp** will forward a copy of the user's Kerberos tickets to the remote host.

passive

Toggle passive data transfer mode. In passive mode, the client initiates the data connection by listening on the data port. Passive mode may be necessary for operation from behind firewalls which do not permit incoming connections.

private

Set the protection level on data transfers to "private". Data transmissions are confidentiality and integrity protected by encryption. If no ADAT command succeeded, then the only possible level is "clear".

prompt

Toggle interactive prompting. Interactive prompting occurs during multiple file transfers to allow the user to selectively retrieve or store files. If prompting is turned off (default is on), any *mget* or *mput* will transfer all files, and any *mdelete* will delete all files.

protection-level]

Set the protection level on data transfers to *protection-level*. The valid protection levels are "clear" for unprotected data transmissions, "safe" for data transmissions integrity protected by cryptographic checksum, and "private" for data transmissions confidentiality and integrity protected by encryption. If no ADAT command succeeded, then the only possible level is "clear". If no level is specified, the current level is printed. The default protection level is "clear".

ftp-command

Execute an ftp command on a secondary control connection. This command allows simultaneous connection to two remote ftp

servers for transferring files between the two servers. The first *proxy* command should be an *open*, to establish the secondary control connection. Enter the command "proxy ?" to see other ftp commands executable on the secondary connection. The following commands behave differently when prefaced by *proxy*: *open* will not define new macros during the auto-login process, *close* will not erase existing macro definitions, *get* and *mget* transfer files from the host on the primary control connection to the host on the secondary control connection, and *put*, *mput*, and *append* transfer files from the host on the secondary control connection to the host on the primary control connection. Third party file transfers depend upon support of the ftp protocol PASV command by the server on the secondary control connection.

<i>remote-file</i>]	Store a local file on the remote machine. If <i>remote-file</i> is left unspecified, the local file name is used after processing according to any <i>ntrans</i> or <i>nmap</i> settings in naming the remote file. File transfer uses the current settings for <i>type</i> , <i>format</i> , <i>mode</i> , and <i>structure</i> .
<i>pwd</i>	Print the name of the current working directory on the remote machine.
<i>quit</i>	A synonym for <i>bye</i> .
<i>arg2</i>] [. . .]	The arguments specified are sent, verbatim, to the remote FTP server.
<i>bytes</i>]	Set the TCP buffer sizes on the remote machine to <i>bytes</i> bytes. If <i>bytes</i> is zero, the TCP buffer sizes are not set; the system default is used. If <i>bytes</i> is omitted, the current TCP buffer sizes of the remote machine are printed. Usually setting the TCP buffer sizes sets the TCP window size. Use with the <i>lbufsize</i> command.
<i>local-file</i>]	A synonym for <i>get</i> .
<i>local-file</i>]	Reget acts like <i>get</i> , except that if <i>local-file</i> exists and is smaller than <i>remote-file</i> , <i>local-file</i> is presumed to be a partially transferred copy of <i>remote-file</i> and the transfer is continued from the apparent point of failure. This command is useful when transferring very large files over networks that are prone to dropping connections.
<i>command-name</i>]	Request help from the remote FTP server. If a <i>command-name</i> is specified it is supplied to the server as well.
<i>file-name</i>]	With no arguments, show status of remote machine. If <i>file-name</i> is specified, show status of <i>file-name</i> on remote machine.
<i>to</i>]	Rename the file <i>from</i> on the remote machine, to the file <i>to</i> .
<i>reset</i>	Clear reply queue. This command re-synchronizes command/reply sequencing with the remote ftp server. Resynchronization may be necessary following a violation of the ftp protocol by the remote server.
<i>marker</i>	Restart the immediately following <i>get</i> or <i>put</i> at the indicated <i>marker</i> . On UNIX systems, marker is usually a byte offset into the file.

<i>directory-name</i>	Delete a directory on the remote machine.
<i>runique</i>	Toggle storing of files on the local system with unique filenames. If a file already exists with a name equal to the target local filename for a <i>get</i> or <i>mget</i> command, a ".1" is appended to the name. If the resulting name matches another existing file, a ".2" is appended to the original name. If this process continues up to ".99", an error message is printed, and the transfer does not take place. The generated unique filename will be reported. Note that <i>runique</i> will not affect local files generated from a shell command (see below). The default value is off.
<i>safe</i>	Set the protection level on data transfers to "safe". Data transmissions are integrity-protected by cryptographic checksum. If no ADAT command succeeded, then the only possible level is "clear".
<i>remote-file]</i>	A synonym for <i>put</i> .
<i>sendport</i>	Toggle the use of PORT commands. By default, ftp will attempt to use a PORT command when establishing a connection for each data transfer. The use of PORT commands can prevent delays when performing multiple file transfers. If the PORT command fails, ftp will use the default data port. When the use of PORT commands is disabled, no attempt will be made to use PORT commands for each data transfer. This is useful for certain FTP implementations which do ignore PORT commands but, incorrectly, indicate they've been accepted.
<i>arg2] [. . .]</i>	The arguments specified are sent, verbatim, to the remote FTP server as a SITE command.
<i>file-name</i>	Return size of <i>file-name</i> on remote machine.
<i>status</i>	Show the current status of ftp .
<i>struct-name</i>	Set the file transfer <i>structure</i> to <i>struct-name</i> . By default "stream" structure is used.
<i>sunique</i>	Toggle storing of files on remote machine under unique file names. Remote ftp server must support ftp protocol STOU command for successful completion. The remote server will report unique name. Default value is off.
<i>system</i>	Show the type of operating system running on the remote machine.
<i>tenex</i>	Set the file transfer type to that needed to talk to TENEX machines.
<i>trace</i>	Toggle packet tracing.
<i>type-name]</i>	Set the file transfer <i>type</i> to <i>type-name</i> . If no type is specified, the current type is printed. The default type is network ASCII.
<i>newmask]</i>	Set the default umask on the remote server to <i>newmask</i> . If <i>newmask</i> is omitted, the current umask is printed.

<i>account</i>]	Identify yourself to the remote FTP server. If the <i>password</i> is not specified and the server requires it, ftp will prompt the user for it (after disabling local echo). If an <i>account</i> field is not specified, and the FTP server requires it, the user will be prompted for it. If an <i>account</i> field is specified, an account command will be relayed to the remote server after the login sequence is completed if the remote server did not require it for logging in. Unless ftp is invoked with “auto-login” disabled, this process is done automatically on initial connection to the FTP server.
<i>verbose</i>	Toggle verbose mode. In verbose mode, all responses from the FTP server are displayed to the user. In addition, if verbose is on, when a file transfer completes, statistics regarding the efficiency of the transfer are reported. By default, verbose is on.
<i>]</i>	A synonym for help.

Command arguments which have embedded spaces may be quoted with quote ``` marks.

ABORTING A FILE TRANSFER

To abort a file transfer, use the terminal interrupt key (usually Ctrl-C). Sending transfers will be immediately halted. Receiving transfers will be halted by sending a FTP protocol ABOR command to the remote server, and discarding any further data received. The speed at which this is accomplished depends upon the remote server's support for ABOR processing. If the remote server does not support the ABOR command, an ``ftp>` prompt will not appear until the remote server has completed sending the requested file.

The terminal interrupt key sequence will be ignored when **ftp** has completed any local processing and is awaiting a reply from the remote server. A long delay in this mode may result from the ABOR processing described above, or from unexpected behavior by the remote server, including violations of the ftp protocol. If the delay results from unexpected remote server behavior, the local **ftp** program must be killed by hand.

FILE NAMING CONVENTIONS

Files specified as arguments to **ftp** commands are processed according to the following rules.

1. If the file name ``-'` is specified, *stdin* (for reading) or *stdout* (for writing) is used.
2. If the first character of the file name is ``|'`, the remainder of the argument is interpreted as a shell command. *Ftp* then forks a shell, using `popen(3)` with the argument supplied, and reads from (writes to) *stdout* (*stdin*). If the shell command includes spaces, the argument must be quoted; e.g. `"" ls -lt""`. A particularly useful example of this mechanism is: `“dir more”`.
3. Failing the above checks, if “globbing” is enabled, local file names are expanded according to the rules used in `csh(1)`; c.f. the *glob* command. If the **ftp** command expects a single local file (e.g. *put*), only the first filename generated by the “globbing” operation is used.
4. For *mget* commands and *get* commands with unspecified local file names, the local filename is the remote filename, which may be altered by a *case*, *ntrans*, or *nmap* setting. The resulting filename may then be altered if *runique* is on.
5. For *mput* commands and *put* commands with unspecified remote file names, the remote filename is the local filename, which may be altered by a *ntrans* or *nmap* setting. The resulting filename may then be altered by the remote server if *sunique* is on.

FILE TRANSFER PARAMETERS

The FTP specification specifies many parameters which may affect a file transfer. The *type* may be one of “ascii”, “image” (binary), “ebcdic”, and “local byte size” (mostly for PDP-10's and PDP-20's). *Ftp* supports the ascii and image types of file transfer, plus local byte size 8 for *tenex* mode transfers.

Ftp supports only the default values for the remaining file transfer parameters: *mode*, *form*, and *struct*.

THE .netrc FILE

The .netrc file contains login and initialization information used by the auto-login process. It resides in the user's home directory. The following tokens are recognized; they may be separated by spaces, tabs, or new-lines:

<i>name</i>	Identify a remote machine <i>name</i> . The auto-login process searches the .netrc file for a <i>machine</i> token that matches the remote machine specified on the ftp command line or as an <i>open</i> command argument. Once a match is made, the subsequent .netrc tokens are processed, stopping when the end of file is reached or another <i>machine</i> or a <i>default</i> token is encountered.
<i>default</i>	This is the same as <i>machine name</i> except that <i>default</i> matches any name. There can be only one <i>default</i> token, and it must be after all <i>machine</i> tokens. This is normally used as: default login anonymous password user@site thereby giving the user <i>automatic</i> anonymous ftp login to machines not specified in .netrc. This can be overridden by using the -n flag to disable auto-login.
<i>name</i>	Identify a user on the remote machine. If this token is present, the auto-login process will initiate a login using the specified <i>name</i> .
<i>string</i>	Supply a password. If this token is present, the auto-login process will supply the specified string if the remote server requires a password as part of the login process. Note that if this token is present in the .netrc file for any user other than <i>anonymous</i> , ftp will abort the auto-login process if the .netrc is readable by anyone besides the user.
<i>string</i>	Supply an additional account password. If this token is present, the auto-login process will supply the specified string if the remote server requires an additional account password, or the auto-login process will initiate an ACCT command if it does not.
<i>name</i>	Define a macro. This token functions like the ftp macdef command functions. A macro is defined with the specified name; its contents begin with the next .netrc line and continue until a null line (consecutive new-line characters) is encountered. If a macro named <i>init</i> is defined, it is automatically executed as the last step in the auto-login process.

ENVIRONMENT

Ftp utilizes the following environment variables.

HOME For default location of a .netrc file, if one exists.

SHELL For default shell.

SEE ALSO

ftpd(8)

Lunt, S. J., FTP Security Extensions, Internet Draft, November 1993.

HISTORY

The **ftp** command appeared in 4.2BSD.

BUGS

Correct execution of many commands depends upon proper behavior by the remote server.

An error in the treatment of carriage returns in the 4.2BSD ascii-mode transfer code has been corrected. This correction may result in incorrect transfers of binary files to and from 4.2BSD servers using the ascii type. Avoid this problem by using the binary image type.

Name

`kdestroy` — destroy Kerberos tickets

Synopsis

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

DESCRIPTION

The **kdestroy** utility destroys the user's active Kerberos authorization tickets by writing zeros to the specified credentials cache that contains them. If the credentials cache is not specified, the default credentials cache is destroyed. If `kdestroy` was built with Kerberos 4 support, the default behavior is to destroy both Kerberos 5 and Kerberos 4 credentials. Otherwise, `kdestroy` will default to destroying only Kerberos 5 credentials.

OPTIONS

<code>-5</code>	destroy Kerberos 5 credentials. This overrides whatever the default built-in behavior may be. This option may be used with <code>-4</code>
<code>-4</code>	destroy Kerberos 4 credentials. This overrides whatever the default built-in behavior may be. This option is only available if <code>kinit</code> was built with Kerberos 4 compatibility. This option may be used with <code>-5</code>
<code>-q</code>	Run quietly. Normally kdestroy beeps if it fails to destroy the user's tickets. The <code>-q</code> flag suppresses this behavior.
<code>-c <i>cache_name</i></code>	use <i>cache_name</i> as the credentials (ticket) cache name and location; if this option is not used, the default cache name and location are used. The default credentials cache may vary between systems. If the <code>KRB5CCNAME</code> environment variable is set, its value is used to name the default ticket cache.

Most installations recommend that you place the **kdestroy** command in your `.logout` file, so that your tickets are destroyed automatically when you log out.

ENVIRONMENT

Kdestroy uses the following environment variables:

`KRB5CCNAME` Location of the Kerberos 5 credentials (ticket) cache.

`KRBTKFILE` Filename of the Kerberos 4 credentials (ticket) cache.

FILES

<code>/tmp/krb5cc_<i>[uid]</i></code>	default location of Kerberos 5 credentials cache (<i>[uid]</i> is the decimal UID of the user).
<code>/tmp/tkt<i>[uid]</i></code>	default location of Kerberos 4 credentials cache (<i>[uid]</i> is the decimal UID of the user).

SEE ALSO

kinit(1), klist(1), krb5(3)

BUGS

Only the tickets in the specified credentials cache are destroyed. Separate ticket caches are used to hold root instance and password changing tickets. These should probably be destroyed too, or all of a user's tickets kept in a single credentials cache.

Name

kinit — obtain and cache Kerberos ticket-granting ticket

Synopsis

```
kinit [-5] [-4] [-V] [-l lifetime] [-s start_time] [-r renewable_life] [-p | -P ] [-f | -F ] [-a] [-A] [-v] [-R] [ -k [-t keytab_file]] [-c cache_name] [-S service_name] [-X attribute [=value]] [principal]
```

DESCRIPTION

kinit obtains and caches an initial ticket-granting ticket for *principal*. The typical default behavior is to acquire only Kerberos 5 tickets. However, if kinit was built with both Kerberos 4 support and with the default behavior of acquiring both types of tickets, it will try to acquire both Kerberos 5 and Kerberos 4 by default. Any documentation particular to Kerberos 4 does not apply if Kerberos 4 support was not built into kinit.

OPTIONS

-5	get Kerberos 5 tickets. This overrides whatever the default built-in behavior may be. This option may be used with -4
-4	get Kerberos 4 tickets. This overrides whatever the default built-in behavior may be. This option is only available if kinit was built with Kerberos 4 compatibility. This option may be used with -5
-V	display verbose output.
-l <i>lifetime</i>	requests a ticket with the lifetime <i>lifetime</i> . The value for <i>lifetime</i> must be followed immediately by one of the following delimiters: <div style="margin-left: 40px;"> <i>s</i> seconds <i>m</i> minutes <i>h</i> hours <i>d</i> days </div> as in "kinit -l 90m". You cannot mix units; a value of `3h30m' will result in an error. If the -l option is not specified, the default ticket lifetime (configured by each site) is used. Specifying a ticket lifetime longer than the maximum ticket lifetime (configured by each site) results in a ticket with the maximum lifetime.
-s <i>start_time</i>	requests a postdated ticket, valid starting at <i>start_time</i> . Postdated tickets are issued with the <i>invalid</i> flag set, and need to be fed back to the kdc before use. (Not applicable to Kerberos 4.)
-r <i>renewable_life</i>	requests renewable tickets, with a total lifetime of <i>renewable_life</i> . The duration is in the same format as the -l option, with the same delimiters. (Not applicable to Kerberos 4.)
-f	request forwardable tickets. (Not applicable to Kerberos 4.)

<code>-F</code>	do not request forwardable tickets. (Not applicable to Kerberos 4.)
<code>-p</code>	request proxiabable tickets. (Not applicable to Kerberos 4.)
<code>-P</code>	do not request proxiabable tickets. (Not applicable to Kerberos 4.)
<code>-a</code>	request tickets with the local address[es]. (Not applicable to Kerberos 4.)
<code>-A</code>	request address-less tickets. (Not applicable to Kerberos 4.)
<code>-v</code>	requests that the ticket granting ticket in the cache (with the <i>invalid</i> flag set) be passed to the kdc for validation. If the ticket is within its requested time range, the cache is replaced with the validated ticket. (Not applicable to Kerberos 4.)
<code>-R</code>	requests renewal of the ticket-granting ticket. Note that an expired ticket cannot be renewed, even if the ticket is still within its renewable life. When using this option with Kerberos 4, the kdc must support Kerberos 5 to Kerberos 4 ticket conversion.
<code>-k [-t <i>keytab_file</i>]</code>	requests a host ticket, obtained from a key in the local host's <i>keytab</i> file. The name and location of the keytab file may be specified with the <code>-t <i>keytab_file</i></code> option; otherwise the default name and location will be used. When using this option with Kerberos 4, the kdc must support Kerberos 5 to Kerberos 4 ticket conversion.
<code>-c <i>cache_name</i></code>	use <i>cache_name</i> as the Kerberos 5 credentials (ticket) cache name and location; if this option is not used, the default cache name and location are used. The default credentials cache may vary between systems. If the <i>KRB5CCNAME</i> environment variable is set, its value is used to name the default ticket cache. Any existing contents of the cache are destroyed by kinit . (Note: The default name for Kerberos 4 comes from the <i>KRBTKFILE</i> environment variable. This option does not apply to Kerberos 4.)
<code>-S <i>service_name</i></code>	specify an alternate service name to use when getting initial tickets. (Applicable to Kerberos 5 or if using both Kerberos 5 and Kerberos 4 with a kdc that supports Kerberos 5 to Kerberos 4 ticket conversion.)
<code>-X <i>attribute</i> [=value]</code>	specify a pre-authentication attribute and value to be passed to pre-authentication plugins. The acceptable <i>attribute</i> and <i>value</i> values vary from pre-authentication plugin to plugin. This option may be specified multiple times to specify multiple attributes. If no <i>value</i> is specified, it is assumed to be "yes". The following attributes are recognized by the OpenSSL pkinit pre-authentication mechanism: <div style="display: flex; justify-content: space-between;"> <div><i>X509_user_identity</i> = value</div> <div>specify where to find user's X509 identity information</div> </div> <div style="display: flex; justify-content: space-between;"> <div><i>X509_anchors</i> = value</div> <div>specify where to find trusted X509 anchor information</div> </div>

<i>flag_RSA_PROTOCOL</i> [=yes]	specify use of RSA, rather than the default Diffie-Hellman protocol
<i>pkinit_debug</i> = <i>value</i>	specify a file to send debuggin output to, this can be stdout, stderr or a file on the system.

ENVIRONMENT

Kinit uses the following environment variables:

KRB5CCNAME Location of the Kerberos 5 credentials (ticket) cache.

KRBTKFILE Filename of the Kerberos 4 credentials (ticket) cache.

FILES

/tmp/krb5cc_*[uid]* default location of Kerberos 5 credentials cache (*[uid]* is the decimal UID of the user).

/tmp/tkt*[uid]* default location of Kerberos 4 credentials cache (*[uid]* is the decimal UID of the user).

/etc/krb5.keytab default location for the local host's *keytab* file.

SEE ALSO

klist(1), kdestroy(1), kerberos(1)

Name

klist — list cached Kerberos tickets

Synopsis

```
klist [-5] [-4] [-e] [[-c] [-f] [-s] [-a [-n]]] [-k [-t] [-K]] [cache_name | keytab_name]
```

DESCRIPTION

Klist lists the Kerberos principal and Kerberos tickets held in a credentials cache, or the keys held in a *keytab* file. If *klist* was built with Kerberos 4 support, the default behavior is to list both Kerberos 5 and Kerberos 4 credentials. Otherwise, *klist* will default to listing only Kerberos 5 credentials.

OPTIONS

- 5 list Kerberos 5 credentials. This overrides whatever the default built-in behavior may be. This option may be used with -4
- 4 list Kerberos 4 credentials. This overrides whatever the default built-in behavior may be. This option is only available if *kinit* was built with Kerberos 4 compatibility. This option may be used with -5
- e displays the encryption types of the session key and the ticket for each credential in the credential cache, or each key in the keytab file.
- c List tickets held in a credentials cache. This is the default if neither -c nor -k is specified.
- f shows the flags present in the credentials, using the following abbreviations:

F	Forwardable
f	forwarded
P	Proxiabile
p	proxy
D	post Dateable
d	post dated
R	Renewable
I	Initial
i	invalid
H	Hardware authenticated
A	pre Authenticated
T	Transit policy checked
O	Okay as delegate
a	anonymous
- s causes **klist** to run silently (produce no output), but to still set 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 or if the tickets are expired.
- a display list of addresses in credentials.
- n show numeric addresses instead of reverse-resolving addresses.
- k List keys held in a *keytab* file.
- t display the time entry timestamps for each keytab entry in the keytab file.

-K display the value of the encryption key in each keytab entry in the keytab file.

If *cache_name* or *keytab_name* is not specified, klist will display the credentials in the default credentials cache or keytab file as appropriate. If the KRB5CCNAME environment variable is set, its value is used to name the default ticket cache.

ENVIRONMENT

Klist uses the following environment variables:

KRB5CCNAME Location of the Kerberos 5 credentials (ticket) cache.

KRBTKFILE Filename of the Kerberos 4 credentials (ticket) cache.

FILES

/tmp/krb5cc_[uid] default location of Kerberos 5 credentials cache ([uid] is the decimal UID of the user).

/tmp/tkt[uid] default location of Kerberos 4 credentials cache ([uid] is the decimal UID of the user).

/etc/krb5.keytab default location for the local host's *keytab* file.

SEE ALSO

kinit(1), kdestroy(1), krb5(3)

Name

kpasswd — change a user's Kerberos password

Synopsis

kpasswd [*principal*]

DESCRIPTION

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

If the principal is governed by a policy that specifies the length and/or number of character classes required in the new password, the new password must conform to the policy. (The five character classes are lower case, upper case, numbers, punctuation, and all other characters.)

OPTIONS

principal change the password for the Kerberos principal *principal*. Otherwise, **kpasswd** uses the principal name from an existing ccache if there is one; if not, the principal is derived from the identity of the user invoking the **kpasswd** command.

PORTS

kpasswd looks first for kpasswd_server = host:port in the [realms] section of the krb5.conf file under the current realm. If that is missing, **kpasswd** looks for the admin_server entry, but substitutes 464 for the port.

SEE ALSO

kadmin(8), kadmind(8)

BUGS

kpasswd may not work with multi-homed hosts running on the Solaris platform.

Definitions

dummy This is a placeholder because a glossary can't be empty. It will be filled in from glossary.collection parameter