SDSC's Installation and Development of Kerberos

Wayne Schroeder,
San Diego Supercomputer Center, San Diego, California

ABSTRACT: *SDSC is in the process of installing and porting MIT's Kerberos 5 Beta 4 network security software to SDSC systems including the C90, Intel Paragon, SUNs, DEC Alphas, SGIs and RS6000s, and purchasing compatible software for our Macintoshes and PCs. By using Kerberized clients (telnet, rlogin, etc.), we will be able to keep plain-text passwords off the network and thereby mitigate the most significant security problem in our distributed environment. This paper will discuss Kerberos features, advantages over DCE authentication, how Kerberos functions in our environment, and the current status of our project. This paper is online at http://www.sdsc.edu/SDSC/Staff/schroede/kerberos_cug.html*

Introduction

Computer security is a serious concern for centers like SDSC. Although we do not carry classified information on our systems, controlled access to our industrial and academic users' data and to the Center's resources is essential. Yet our compute resources need to be readily accessible to academic researchers across the United States. And we are a very visible target for computer break-in attempts.

Many of these attempts are equivalent to someone walking down the street and casually rattling the knobs of every door to see if one has been left unlocked. Some are sophisticated and well-targeted assaults. Some of these have been serious enough to cause us to escalate our own monitoring and security practices.

An incident on December 25, 1994 eventually led to the arrest of Kevin D. Mitnick after some brilliant cyber-sleuthing by SDSC senior fellow Tsutomu Shimomura and SDSC staff member Andrew Gross. This was reported in the New York Times on February 16, 1995: "How a Computer Sleuth Traced a Digital Trail." [1] The incident was first reported in a front-page article in the Monday, January 23, 1995 New York Times.

Since 1993, SDSC has been interested in running Kerberos as one method of improving our security. In late 1994 we began the project, led by SDSC security specialist Tom Hutton, with various members of the SDSC Systems Department participating. All participants had other responsibilities and were not able to commit large amounts of time to the project.

It is a fairly large project, with impact on users, and required a phased-in approach.

Why Kerberos?

The most significant security problem in our distributed environment, and a common problem on the Internet, is the monitoring of network packets for passwords. Intruders who manage to gain root access to most Unix platforms are then able to monitor ethernet traffic and can watch for passwords. There are other serious security problems that we concern ourselves with, but transmission of plain-text passwords on the network is clearly the weakest link in our user authentication system.

Kerberos [2] was designed to deal with this problem. Through a sophisticated protocol built on DES encryption, plain-text passwords need never be transmitted across the network. And, via a ticket-granting sevice (TGS), users need only enter their passwords once for an entire work period. For example, users can register (kinit) in the morning and then can repeatedly connect (ktelnet) to other systems throughout the day without entering a password again.

Also, Kerberos rlogin has an option (-x) to encrypt an entire session. Encrypting a session causes all entered input and all terminal output to be encrypted while traversing the network. This is particularly useful for users who wish to protect information which is entered on their keyboard or viewed on their monitor.

CRI provides an older version of Kerberos (4) as part of Unicos 8.0, but reportedly has no plans to port and provide Kerberos Version 5.

What About DCE Authentication?

CRI provides OSF DCE with Unicos and, initially, recommended that we use DCE Authentication instead of Kerberos on the Cray. Since some hosts do not have DCE, we would need a

blended environment, with both DCE Authentication and Kerberos. HP, as part of OSF/DCE, is working on interoperability issues, and OSF plans to release an interoperable solution within a year or so. Our hope was that we could use vendor-supported DCE-ized telnetd, rlogind, etc., on the Cray and have them interoperate in our environment.

Further investigation revealed, however, that ESnet staff (a network managed and funded by the U.S. Department of Energy Office of Energy Research (DOE / OER)), had examined Kerberos and DCE Authentication and had developed a proposal to use them for intersite authentication in a wide-area network. This Distributed Informatics, Computing & Collaborative Environment (DICCE) proposal "Authentication Infrastructure of the DICCE" [3], included a description of some current limitations:

"DICCE is being based on OSF/DCE which is currently missing a number of the key components needed for a wide area collaborative environment. These include:

- Hierarchical intersite authentication
- Authenticated/Secure Telnet
- Authenticated/Secure FTP
- Authenticated/Secure "r" commands
- Integrated login, including DCE, AFS, Kerberos
- Interoperability of non-DCE clients with a DCE cell

"The goal of the project is to use a mixture of OSF/DCE and Kerberos Version 5, utilizing the best features of each. This will provide for greater flexibility in implementing other applications on both DCE and non-DCE platforms."

So, since DCE does not currently provide Authenticated/Secure Telnet or "r" commands (rlogin, rsh, remsh, rcp), SDSC would not be able to leverage off DCE services for our needs; we would need to install/port the MIT Kerberos release ourselves. It is likely that the OSF/DCE will provide an interoperable authentication system, but probably not for a year or more.

Also, DCE Authentication may not be adopted by all vendors. OSF/DCE is available for SUN platforms, for example, only through a third-party (TransArc-IBM). And Microsoft has stated that it intends to replace the existing security server in Windows NT with MIT Kerberos. [4]

CRI has been looking at solutions to the Single Login issue for quite some time. The main problem for any of this work is the fact that there is no protocol support for passing authentication information (tickets) with the standard unix login (telnet, rlogin) commands or "r" commands. There needs to be an industry-wide solution for this problem, but to date there is none. If one existed, every vendor would have adopted it long ago .

Currently there are no standard DCE-ized login commands such as telnet, ftp, fta, rlogin, rsh, rexec, and rcp. As part of its 1.0.2.1/1.0.3.1 DCE/DFS product, CRI has integrated DCE authentication into most of the standard login commands (telnet, rlogin, rexec, ftp fta, nqe). This allows users to pick up DCE authentication at login time without the need to take an additional step of doing a dce_login. This is handy for things like placing a user's home directory out in DFS space. But the interaction occurs at login time, within the Cray, and passwords are still sent to the Cray as plain-text.

CRI has been looking at adding DCE network authentication to these commands as well, but in doing so, CRI would be solving this issue only for a collaborative environment of its own. Other vendors would need to adopt the same protocol for passing DCE authentication in order for this feature to be of any use for non-CRAY machines.

The solution that CRI has been leaning toward is using GSSAPI as the authentication service, since this is perhaps a larger standard, and more vendors might be willing to GSSAPI-ize their commands in an attempt to solve these issues. GSSAPI will increase portability and maintainability of commands needed to provide authentication, authorization, and privacy. Underneath GSSAPI there must be vendor-supplied mechanisms providing these functions. Unless vendors supply them (Kerberos V, DCE), sites still will not have interoperablity. Perhaps what is needed is some coordination among vendors to supply these mechanisms.

OSF has stated that its DCE 1.2 release will include implementations of telnet, rlogin, and ftp that have been integrated with the DCE environment, and will use Kerberos-based protocols to avoid exposing users' passwords on a network. CRI is eager to see this code, as it has been stated that mods to the protocol will be handed back for potential inclusion in the IETF standards. This is a good step toward an industry-wide solution.

Kerberos Shortcomings

At the Spring '95 Cray User Group meeting, Bryan Koch of CRI held a tutorial on network/Internet security, and discussed (mostly during Q&A) some of the shortcomings of Kerberos.

He cited three main problems:

- The server doesn't know if tickets are used or not, so a program could repeatedly ask for tickets and attempt to decrypt them by guessing passwords. The Kerberos ticket server is stateless and so just gives tickets in response to requests. [On the Cray, even encrypted passwords are not accessible to general users in the /etc/passwd file, preventing such an attack.]
- Since users pick their own passwords under Kerberos, passwords can often be easy to guess.
- There are too many sites to trust Kerberos tokens moving from one organization to another.

To deal with the first problem, SDSC is adding logging and reporting logic in the Key Distribution Center (KDC, Kerberos Server), which will help us monitor this. This will require additional investigation and perhaps some software development.

SDSC is dealing with the second problem by adding public-domain software from other password systems to the KDC. Passwords that do not meet the criteria will be rejected with an explanation of the rules. These include that there be at

least one non-alphabetic character and that the password does not match the username or dictionary words.

For the third, SDSC plans to allow only inter-realm authentication between the other NSF centers and SDSC; so we will be dealing with only a few known and trusted system administrators. Most of our users (i.e., those not in Kerberos NSF center realms) will install our clients (i.e., ktelnet, krlogin, kinit, and kpasswd) and authenticate to our KDC without requiring a "shared secret" to function.

Presumably, a set of Kerberos clients will be able to interact with the SDSC realm, and at the same time another set (or the same set, maybe) could interact with another realm. This way, a user now (or in the future) using Kerberos at University XYZ could use our ktelnet to access SDSC and another ktelnet to access his/her local realm.

User View

While Kerberos is in some ways inconvenient for users, it is in other ways more convenient. Users enter their password once via kinit, and they can repeatedly use ktelnet, krlogin, rsh, rexec, and rcp to interact with systems at SDSC in a secure manner without entering another password for the rest of the workday.

Regular rlogin can be used in a passwordless manner, but SDSC has not encouraged its use (although we do allow it, and some users use it). rlogin is not particularly secure because of the "trusted host" method associated with the .rhosts entries. If someone were to break into a workstation that was in a user's Cray .rhosts file, the intruder would have access to the user's Cray account. Or if someone were to "spoof" the network into "believing" that his host was that user's workstation, access would also be obtained. Also, rlogin is inefficient, increases system time, and reduces interactivity because it generally sends a network message for every character typed.

With Kerberos, users can have efficient, passwordless, and secure interaction with SDSC systems via ktelnet and other kerberized utilities.

Another advantage, for users, as mentioned earlier, is the encryption option of krlogin to encrypt the terminal traffic of an entire session. This will occasionally be useful.

Users run kinit to get authenticated in the Kerberos system:

```
number9 1% kinit -f -l10
Password for schroede@TEST.SDSC.EDU:
```

This creates a ticket-granting ticket that can be listed via the klist command:

```
number9 2% klist
Ticket cache: /tmp/krb5cc_122
Default principal: schroede@TEST.SDSC.EDU

   Valid starting   Expires Service principal
   7-Sep-95 08:44:38   7-Sep-95 18:45:55
   krbtgt/TEST.SDSC.EDU@TEST.SDSC.EDU
```

The -f option on kinit makes the ticket forwardable to other systems. By using the ktelnet -F or -f option (or krlogin -F or -f), the Kerberos software forwards a copy of the ticket in a secure way, so that the user does not have to kinit on the remote host (kinit would take a plain-text password). These tickets are viewable via klist. The -l option specifies the number of hours that the ticket-granting ticket (and all tickets generated by it) are valid. Once tickets expire, they are no longer valid, but existing ktelnet sessions, etc., will continue.

A sample session follows.

```
number9 3% ktelnet -F c90
Trying 132.249.40.32...
Connected to c90.sdsc.edu.
Escape character is '^]'.
[ Kerberos V5 accepts you as ``schroede@TEST.SDSC.EDU'' ]
[ Kerberos V5 accepted forwarded credentials ]

               San Diego Supercomputer Center
           CRAY C90 with 8 CPUs and 256 MW running UNICOS 8.0.3.2

Last successful login was : Thu Sep  7 15:44:29 from number9.sdsc.edu
Account (? for available accounts) [sys200]:

   See 'news sdsc.important.cray' for more details!

         If you have any questions, please contact the SDSC consultants at
                    (619)534-5100 or e-mail to consult@sdsc.edu
                     or visit SDSC's User Services web page at
                http://www.sdsc.edu/Services/Consult/consult.html

c90 1% klist
Ticket cache: /tmp/krb5cc_122
Default principal: schroede@TEST.SDSC.EDU

   Valid starting        Expires          Service principal
7-Sep-95 08:46:53   7-Sep-95 16:48:13      krbtgt/TEST.SDSC.EDU@TEST.SDSC.EDU

c90 2% logout
Connection closed by foreign host.
number9 4%
```

No password was needed. Likewise, krlogin can be used:

```
number9 4% krlogin c90

               San Diego Supercomputer Center
           CRAY C90 with 8 CPUs and 256 MW running UNICOS 8.0.3.2

Last successful login was : Thu Sep  7 15:47:42 from ibm-6000.sdsc.edu
Account (? for available accounts) [sys200]:
```

Or ktelnet/krlogin to other systems:

```
number9 5% krlogin graywolf
Last login: Thu Sep  7 08:52:54 from number9.sdsc.edu
SunOS Release 4.1.3 (SDSC_SPARCSTATION) #6: Tue Apr 26 23:41:03 PDT 1994
Thu Sep  7 08:54:12 PDT 1995
graywolf 1%
```

Kerberos Overview / Features

The Scientific American article, "Secure Distributed Computing" [2], provides an excellent general description of how Kerberos works. What follows is a brief summary that presents the basic ideas.

The Key Distribution Center (KDC) and Ticket-Granting Service (TGS) are two parts of the Kerberos Server. At SDSC, we are running this on a Sun in the machine room (i.e., physically secure). The initial user passwords are known to the KDC and securely given to users (e.g., through paper mail).

When a user runs kinit, a message is sent to the KDC with the user id. The KDC responds with a message that is DES (Data Encryption Standard) encoded using the user's key (which is based on the user's password). When the user enters his password to kinit, it then uses that to make a key and attempt to decrypt the message from the KDC. If it decrypts properly, then the user has entered the password correctly and kinit saves encrypted information, a "ticket-granting ticket," in a ticket file. This is normally just a user-owned file in /tmp, i.e. /tmp/krb5cc_122. If someone broke into the user's workstation, access to his or her Kerberos accounts would be gained until the ticket-granting ticket expired.

The second part of the Kerberos server is the Ticket-Granting Service (TGS). The ticket sent from the KDC to kinit contains the user name, current time, duration of validation, name of workstation, etc., and a DES session key to be used for this session. All this is encoded with the TGS secret key, and then the session key is appended, and all this is encoded with the user's key. Thus if kinit can decrypt the packet, it has a session key, and it has a ticket that is encrypted with a key that it does not know (or need to). Later processes can pass this encrypted ticket, and information encrypted in the session key, to the TGS to begin similar series of authenticated communication transactions.

These methods result in a system in which passwords are only briefly needed on the local system, various keys and encrypted packets are used to authenticate, and secret keys are used but not known by clients.

There are some secret keys that must be set up by hand on some systems to identify each other. These are kept in uid 0 files (with no group or other access permissions), e.g., /etc/v5srvtab, for systems that are servers. The SDSC Cray and most SDSC workstations have v5srvtab files, as these are needed for the Kerberized server processes, i.e., when users ktelnet, or krlogin, etc., into one of these systems.

These secret keys are not needed for the client services, such as kinit, ktelnet, and krlogin, so they will not be needed for most of our remote users using SDSC-provided kerberoized clients for access to SDSC hosts.

Cray Port

At SDSC we decided to use the current (at the time, December 1994) Kerberos 5 beta 4 release 3 version.

The port of the Kerberos libraries to the Cray C90 went fairly well, but the port of various daemons and clients were more involved. After getting a few library compile errors resolved, many of the basic tests worked fine. SDSC staff member Andrew Gross ran some additional tests, found and fixed some problems, and got the basic message passing between the Cray and a test server working.

Our entire Cray port probably took about two man-months (spread over many months), as we had many problems to resolve with the ktelnet, krlogin, and other "r" command daemons. But at this point, we have telnetd (and telnet), krlogind (and krlogin), and the other "r" commands and daemons working well.

Previous versions of Kerberos 5 required the ISO Development Environment, but, fortunately, 5.4 did not. (Sandia had to port a version of the ISODE to the Cray for their Kerberos 5.2 work.) I found that I needed to use GNU make instead of CRI's make for the Kerberos build environment. Also, we found the 5.4 GNU automatic configuration system (which builds the Makefiles) sometimes difficult to work with, and for quite a few problems I patched Makefiles (Kerberos 5.5 has a revised configuration system).

We decided to use Concurrent Versions System (CVS) for our source management (with separate subdirectories for each architecture) and set that up on a fileserver accessible from each relevant host. CVS works well for this type of project, but we encountered various problems with it on the Cray and Alphas in particular.

It was found preferable to use CRI's login process (for both ktelnetd and krlogind) instead of the MIT Kerberos login. There were too many enhancements missing from the Kerberos login to make its use viable, including accessing the UDB, checking and prompting for multiple accounts, using the CRI ia_user routine, etc. Certain restrictions accompany a CRI login, however, such as the requirement of a .rhost entry for krlogind logins.

Most problems had to do with word size or include file and system definitions. In many cases it took a while to track a problem to the key section of code. In some cases small sections of CRI code were added to the Kerberos release (such as PTY handling in krlogind).

There were problems in krlogind interface to login (or init), errors in writing wtmp and utmp entries, problems in reading keytab files, errors in the window-size message processing, md4 encryption/decryption, "no access to tty," the Kerberos syslog system, removing credentials files, rcp buffer sizes, a srandom/srand48 problem, a runaway ktelnetd problem, an EOF tty setup problem, etc. Code was also added to krlogind to purge $TMPDIR directories on logout.

Other Ports/Installs

So far, we've been working with Kerberos primarily on five architectures/systems: the Cray C90 Unicos 8.0.3.2, Sun SunOS 4.1.3, DEC Alpha DEC OSF/1 V2.0 , IBM RS6000, and SGI IRIX 5.2. Total effort for each port/install is estimated to be (from most to least): Cray, SGI, Alpha, IBM, Sun. Work has begun on Intel Paragon Kerberos, and we expect it to fall somewhere in the mid-range (both Intel and SDSC are independently

investigating issues relating to porting and supporting Kerberos 5.5 under Paragon OS).

The Alpha required as many similar changes as the Cray for 64-bit words, although since the Alpha can allocate in smaller units, the actual changes differ. For example, changes were needed in routines des_read and des_write in krlogin.c and krlogind.c to handle the 4 byte network stream that is the length of the encrypted message that follows. For the Cray, these reads and writes were offset into the word. On the Alpha, the variable was declared to be an int instead of a long.

Much like the Cray, the SGI port required quite a few corrections to the system interface routines and utilities (telnet, "r" commands, etc.), although the Kerberos libraries themselves installed fairly easily. For krlogind on the SGI, we fixed a set of problems with the following changes:

1. If you attempt a "resize," you get "can't open terminal /dev/tty."

   This was fixed when #2 was fixed.

2. Doing a "ps," no processes appear. (We had a similar problem on the Cray, but it doesn't appear to be due to the same cause on the SGIs.)

   This is fixed by having krlogind.c do a setsid() call like the Sun and POSIX systems do, before the open(line, O_RDWR) call.

3. The output of a "last" command (which shows recent logins), does not include info in all the fields for a SGI krlogin or ktelnet session.

   Various problems with writing utmp and wtmp file records were corrected to resolve this.

4. One does not get the normal SGI rlogin greeting messages:

```
number9 12% rlogin startrek
IRIX Release 5.2 IP22 voyager
Copyright 1987-1994 Silicon Graphics, Inc.
 All Rights Reserved.
Last login: Thu Jun 29 09:13:52 PDT 1995 by
   UNKNOWN@number9.sdsc.edu
Thu Jun 29 09:19:26 PDT 1995
voyager 1% logout
Connection closed.

number9 13% krlogin voyager
Thu Jun 29 16:19:37 GMT 1995
voyager 1%
```

   This was fixed by having krlogind run SGI's login instead of the MIT-released "login.krb5."

5. The TZ environment variable, and probably others, are not set.

   This was also fixed by having krlogind run SGI's login instead of the MIT released "login.krb5."

6. The Last login message (see #4 above) sometimes shows "UNKNOWN@number9.sdsc.edu" as the last login.

   This happens when one logs into an SGI with the regular telnet (as well as ktelnet). The rlogin and krlogin daemons set something correctly so that the user name appears instead of UNKNOWN (on the next login). Since SGI's telnet doesn't handle it right, we are not making it worse with ktelnet. This information may be stored in the SGI extended utmp and wtmp files (utmpx and wtmpx).

MetaCenter Kerberos plan

The Kerberos work at SDSC is part of a plan to integrate Kerberos into the environment at each center of the NSF MetaCenter (CTC, NCSA, PSC, and SDSC). SDSC is leading this planning and development. Long-term plans may include incorporating other research centers into the Kerberos-authenticated realms.

Future

Although we have many kerberized functions (telnet, rlogin, rsh, rcp, remsh) working on many platforms, there are still a few problems to work out, and a specific phase-in plan to develop.

Over the next few months (fall, 1995), we plan to phase-in kerberized functions into the SDSC production environment. A primary goal will be to significantly reduce the plain-text passwords used by SDSC staff on the internal network. We also plan to eliminate plain-text root passwords on our network by using only encrypted rlogin sessions when we run 'su.' We also plan to support some kerberos client access (i.e., kinit, kpasswd, and ktelnet) by off-site SDSC researchers using workstation types that we have at SDSC.

We plan to eventually integrate Kerberos into all of our systems (including the Paragon) and into more functions (including ftp, UniTree, and HPSS).

NSL UniTree, our mass storage system, is currently accessed via modified ftp clients on SDSC systems. Currently, we allow only limited remote access to UniTree. When we add Kerberos authentication, we will be able to provide secure remote access to UniTree via kerberized ftp clients.

Longer term, we are interested in how DCE authentication will evolve and in its possible interoperation with Kerberos. At some point, we should be able to utilize vendor-supported software for a Kerberos/DCE environment that will secure passwords.

Our mods have been given to staff at ANL and NIH, who were particularly interested in our SGI fixes, and we have sent our mods back to MIT where many of them, presumably, will be incorporated in the next release.

Conclusion

Although it currently requires a substantial effort to integrate into an environment and local (non-vendor) support, Kerberos offers a significant improvement in distributed network security. By using kerberized clients (telnet, rlogin, etc), we will be able to keep plain-text passwords off the network and thereby mitigate the most significant security problem in our distributed

environment. While in some ways inconvenient for users, it is in other ways preferable, as users can enter their password once at their workstation and access systems repeatedly throughout the day. For centers like SDSC, we believe that Kerberos is the current best solution, and is likely to be at least part of the solution for the next few years and, perhaps, beyond.

Acknowledgments

The following people at SDSC were instrumental in progress to date with our installation and development of Kerberos.

Networking staff:
Tom Hutton
Andrew Gross
Jay Dombrowski
Systems staff:
Wayne Schroeder
Tom Perrine
Tom Sherwin
Larry Diegel
SDSC User Consultant:
Nancy Wilkins-Diehr

Bill Rahe, Walt VanDevander, and John Noe of Sandia Albuquerque provided us with valuable information (and source code) concerning their port of Kerberos 4 and 5.2. Chuck Athey and Rich Frobose of Livermore Computing shared valuable information on their experiences with Kerberos 5.2 (Sandia's port).

This work was funded in part by National Science Foundation Cooperative Agreement ASC-8902825.

References

[1] N.Y. Times News Service, February 16, 1995, "It takes a computer hacker to catch one," by John Markoff. Viewable at http://www.sdsc.edu/SDSC/Staff/tep/Mitnick/nyt-1.2.16.1995. Also see "Articles About Kevin Mitnick" at http://www.sdsc.edu/SDSC/Staff/tep/Mitnick/Index.html.

[2] Schiller, Jeffery I, "Secure Distributed Computing," Scientific American, Volume 271 Number 5, pp. 72-76, November 1994.

[3] "Authentication Infrastructure of the DICCE," Doug Engert(ANL), Jack Moore (PNL), Joe Ramus (NERSC), and Doug Brown (SANDIA). Viewable at http://www.es.net/pub/dccc/dicce-proposals/authentication.html.

[4] "Networking Windows NT," John D. Ruley, published by John Wiley and Sons, Inc., page 411.

For additional information, connect to SDSC's Home page at http://www.sdsc.edu and use the "SEARCH SDSC'S WWW SERVER" facility to search for "kerberos" to find this paper and other information.