The Sun Enterprise Authentication Mechanism™
John Douglass
GSEC Certification v2.0

Summary

Modern computer systems provide service to multiple users across an ever-increasing web of inter-networked systems. These interactions require the ability to reliably authenticate the user making the request, and determine the legitimacy of the request (i.e. does the user have the appropriate privileges for the operation).

This paper introduces the Sun Enterprise Authentication Mechanism™ (SEAM) software. The paper begins by providing a background discussion of the Kerberos authentication service, which forms the underpinnings of the SEAM product. Next, an introduction to the SEAM product is presented. Information on the obtaining, installation, and general application of the product is given. Finally, performance and interoperability with other implementations of the Kerberos protocol are discussed.

The Kerberos Authentication Mechanism

Kerberos is an authentication mechanism developed at the Massachusetts Institute of Technology (MIT) as part of Project Athena. Currently, Kerberos is in Version 5, release 1.2 and is available at its MIT website [13].

Traditional authentication, both local and network oriented, have relied upon a password based mechanism. One of the major shortcomings of this mechanism in a networked environment is that, sans cryptography, an attacker listening to the network is able to intercept all knowledge necessary to gain illicit access to the computer resources. Password collection programs are widely available on the Internet, and several high profile incidents involving this form of attack have been reported over the past decade.

The key to the Kerberos protocol is the ability of a user, called a principal in Kerberos terminology, to prove his or her identity to a verifier without the need to send data across that network that would allow an attacker to later impersonate the user. "Though conceptually, Kerberos authentication proves that a client running on behalf of a particular user, a more precise statement is that the client has knowledge of an encryption key that is known by only the user and the authentication server [9]."

The Kerberos mechanism utilizes a series of encrypted messages in order to provide this proof. The exchange protocol is based in part on the work of Needham and Schroeder, which outlined a secure mechanism for symmetric connections. Additional changes have been made to address some of the weaknesses that have been discovered in the protocol and a complete discussion of the protocol is beyond the scope of this paper. Further information regarding the protocol mechanisms can be found at the home website for the Kerberos project [7].

Two of the most important features of Kerberos are that it is freely provided under copyrights similar to that of the X Window System (also originally developed at MIT) and standardized as defined by many Internet Request For Comments (RFC) including RFC-1510 (The Kerberos Network Authentication Service, Version 5) and RFC-1964 (The Kerberos Version 5 GSS-API Mechanism). This has allowed Kerberos to become the de facto industry standard for secure authentication with both vendor and free implementations widely available. The standardization of the Kerberos protocols allows interoperability among these systems making Kerberos the ideal choice for providing secure authentication.

References to discussion of the limitations of the Kerberos protocol can be found in the "Papers about the Kerberos Protocol" section of the MIT project website. It is important to note in any discussion of the protocol that Kerberos does not provide effective protection against guessable passwords. [7, 9]. Thus, any installation implementing the Kerberos protocol must ensure that a strong password policy is in place to protect against the choice of poor passwords.

Introduction to Sun Enterprise Authentication Mechanism™ (SEAM)

The Sun Enterprise Authentication Mechanism™ (SEAM) is a Kerberos v5 compliant software package designed to integrate into the Solaris Operating Environment (OE). The SEAM software provides both server (Key Distribution Center or KDC) and client (kerberized applications such as telnet, rlogin, NFS, etc.) side services which together provide not only a secure authentication protocol, but the ability to encrypt data streams for remote applications.

SEAM is a *single-sign-on* system, meaning that the user only needs to authenticate once per session. All subsequent transactions during the session are automatically secured. Once authenticated by SEAM the user does not need to authenticate every time they use one of the kerberized applications provide by SEAM. This means the user's password will not be sent over the network, where it can be intercepted, each time they use these services [11, pg 374].

In addition, the SEAM software provides an implementation of the RPCSEC_GSS security protocol as defined in Internet RFC-2203. This application program interface (API) allows developers and software vendors to adopt evolving security technologies by producing standards compliant software. Future security products from Sun or third parties can utilize the interface without requiring modifications to the applications programmed to the standard.

Obtaining the SEAM software

SEAM is available for Solaris 2.6, Solaris 7, and Solaris 8 for both the SPARC™ and Intel architecture platforms. For Solaris 2.6 and Solaris 7 the SEAM 1.0 software is available as part of the Solaris Easy Access Server (EAS) 3.0 package. This package is not available for download, however it can be ordered from Sun or an authorized reseller.

The Solaris 8 OE contains the client side software. The entire SEAM 1.0.1 release is available as part of the Solaris 8 OE Admin Pack. This software is available for download from http://www.sun.com/bigadmin/content/adminPack/index.html.

The SEAM software packages available for download for the Solaris 8 OE provide only the secure authentication functionality. In order to provide the ability for users to exchange their data privately an additional data-encryption module is necessary. The privacy enhancing packages are available in a separate downloaded packaged available at http://www.sun.com/solaris/encryption. The encryption package enables users to use the NFS protocol with privacy protection as well as encrypt data streams in the kerberized remote applications provided with the SEAM software.

In the upcoming Solaris 9 release, a full release of the SEAM software minus the remote applications has been included as part of the standard distribution of the operating system. Preference has been given to the use of Secure Shell, also now included as part of the standard distribution, as a replacement for the remote applications that the SEAM application previously provided.

Installation of the SEAM software

The installation procedure given below assumes that the SEAM product is contained on a CDROM. This would be the case if the Solaris EAS CDROM was purchased, or the entire Solaris 8 Admin Pack was downloaded as an ISO image and burned on to a CDR. The SEAM product is available as a separate download, installation of that download is not presented here. Furthermore, the procedure assumes an available NFS server for shared use of installation information, alternative installation methods are available and can be found in the SEAM documentation set available online at `http://docs.sun.com/` under the “Collections” tab.

The first step for the installation procedure would be to plan the installation. Significant planning is required for the use of the SEAM product. Items that should be considered are the Realms to be used (a Realm is a group of systems under the same master KDC), the location of the master KDC and one or more slave KDCs, the client and service principal names, database propagation, et. al. Due to the complexity of this planning and its site-specific nature this step in the procedure is not fully covered but instead only mentioned as a necessity.

The second step in the procedure covers the actual installation of the necessary packages. It is comprised of the following sub steps:

1. Copy the SEAM image to the NFS server to be used.
2. Define preconfiguration information, this is not necessary but for sites with many server and/or client installs to perform this may save significant effort.
3. Choose and install the SEAM product. The installation mechanism allows for the installation of master KDC, slave KDC, and client configurations.

After becoming `root` on the NFS server to be used to export the SEAM image and mounting of the CDRom (if necessary) the following commands can be used to copy the image:

```
# cd /export
# mkdir SEAM
# cd /cdrom
# find .install products/Sun_Enterprise_Authentication_Mechanism_1.0 -print|\
cpio -dump /export/SEAM
```

[12]

Note the above example shows specifically the copying of the SEAM 1.0 product, minor adjustments must be made for later releases.

After completion of the above step we next need to enter the preconfiguration information. The SEAM product provides a graphical interface for this procedure. This installer will allow for the easy configuration of the parameters necessary for installation and use of SEAM on a network. Parameters to be configured include: hardware configuration (sparc, i86), realm name, DNS domain name, master kdc, slave kdc, online help locations (answerbook server, URL), and directory path of file system hold the SEAM packages. To start the graphical configuration tool use the following commands:

```
# cd /export/SEAM/products/Sun_Enterprise_Authentication_Mechanism_1.0
# ./installer
```

[12] As above the example is specific to the SEAM 1.0 product. A complete step-by-step example for use of this configuration tool is also given in [12]. The installation tool provides an intuitive interface and if the proper planning has been followed the tool should require no further documentation. After the "Welcome" screen and selection of locale, please choose "Select Software Components to be Installed." The screens following this selection will allow for the selection of components to be installed (these may be left blank if the NFS server being used will not be a KDC or client) and the definition of the site default information. The result of this step is a file containing all necessary preconfiguration information.

Once the preconfiguration information has been entered, a system administrator can now quickly deploy both servers and clients. The following command examples assume that the SEAM packages are available at `/mnt/SEAM/`. The same graphical interface used to perform the preconfiguration is used to install and configure the master and slave KDCs and the client software. Use the following commands to run the installer:

```
# cd /mnt/SEAM/products/Sun_Enterprise_Authentication_Mechanism_1.0
# ./installer
```

[12] Again, the above example is specific to the SEAM 1.0 product. Following the graphical screens will allow for the choice of what type of installation (Master, Slave, Client) and allow for "Use previously configured site information." Note, the installation tool can also be used to reconfigure site information.

Although the graphical interface can be used for client installations, a non-graphical installation procedure is provided to allow for easy scripting of the client installation once the preconfiguration step has been completed. Perform the following command as `root` on the client to complete the software installation, as before the example assumes the SEAM packages are available at `/mnt/SEAM/`.

```
# cd /mnt/SEAM/products/Sun_Enterprise_Authentication_Mechanism_1.0
# ./installer -nodisplay
```

[11]

SEAM Kerberos Server Security Considerations

A typical server installation of the SEAM product includes the Key Distribution Center (KDC), the Kerberos Administration Daemon, and the Kerberos database on a single machine; although, this is not a requirement. The KDC is a trusted server that issues the Kerberos "tickets" which are what allow the clients and servers to communicate securely. The Kerberos Administration Daemon handles administrative tasks such as the addition and deletion of principals in the Kerberos database.

Because this single system contains the database and the keys to the database, it is critical that the KDC system be secured and monitored closely [6]. Many "best practice" methods for securing a Solaris system are available. In addition, many tools such as Titan, YASSP, and the Center for Internet Security's Benchmark are available to help secure your server [13, 15, 1].

The synchronization of system clocks deserves special note here. Due to the time dependent nature of the Kerberos v5 protocol, failure to maintain reasonable synchronization of time across systems may result in a service denial. Solaris OE includes client and server software that implement the Network Time Protocol (NTP). For more information regarding this software see [2, 3, 4].

Key Distribution Center (KDC) Configuration

The `/etc/krb5/kdc.conf` file provides configuration information for both the KDC and the Kerberos administration daemons. The `kdc.conf` file includes parameters that describe the locations of various files and ports to use for accessing the KDC and the administration daemon. The parameters are defaulted to a secure setting and generally do not need to be modified. For a complete and detailed discussion of these options please see the `kdc.conf(4)` man page [6].

As mentioned previously, Kerberos does not protect against guessable passwords. SEAM allows for the establishment of a custom password policy and different policies can be defined for individual principals or groups of principals. A password policy defines the following parameters:

- Minimum password lifetime – This is the minimum time, defined in seconds, that a password must be used before it can be changed.
- Maximum password lifetime – This is the maximum time, defined in seconds, that a password can be used before it must be changed.
- Saved password history – This is the number of previous passwords used by the principal that cannot be reused.

- Maximum password classes – This is the number of different character classes that must be used to make up the password. The three character classes defined are letters, numbers, and punctuation.
- Minimum password length – This is the minimum number of characters that must be in a password. Note, the maximum password length supported is 255 characters. The higher character limit allows the principal to choose an easy to remember phrase instead of the single word password common under the Unix authentication mechanism.

Although the above parameters may not be flexible enough to fit all organizations password policies, proper settings can greatly improve password security under Kerberos. Although it was stated above that most of the parameters in the kdc.conf did not need modification, one parameter does deserve mention here -- dict_file. The dict_file parameter defines the location of a dictionary file containing strings that are not allowed as passwords. The parameter applies to any principal who has a password policy association, however it is undefined by default.

The format of the dictionary is one word or phrase per line. Many “hacker” dictionaries are available on line and could be used for this purpose. Alternatively the Solaris OE ships with a default system dictionary located in /usr/share/lib/dict/words.

Use of at least one password policy for every principal, in conjunction with definition of the dict_file parameter in the kdc.conf file, will greatly enhance the password security for the system.

An example is given below on how to modify the default password policy using kadmin:

```
kadmin: addpol -minlife "1 hour" -maxlife "180 days" -minlength 8
-minclasses 3 -history 4 default
kadmin: getpol default
Policy: default
Maximum password life: 15552000
Minimum password life: 3600
Minimum password length: 8
Minimum number of password character classes: 3
Number of old keys kept: 4
Reference count: 35
```

This policy will be applied to all new principals that are not given the same password as the principal name when they are created. The reference count value in the policy is the number of principals that are configured to use this policy.

Kerberos and Pluggable Authentication Modules (PAM)

Solaris supports a pluggable authentication module for Kerberos v5 that can be used by both kerberized and non-kerberized applications. The /etc/pam.conf file is used to configure the use of the PAM modules. Use of the PAM-KRB5 module with non-kerberized applications will expose the users password to network snooping attacks because the password will be passed over the wire in plain text.

Kerberized Clients and Remote Application Servers

The SEAM software package provides kerberized versions of the following commonly used client applications: telnet, rlogin/rsh, rcp, login, and ftp. In addition, kerberized telnetd, rlogind, and ftpd servers are provided. These applications provide the user with not only Kerberos authentication, but with the ability to encrypt all traffic between the client and the remote server, as well.

Default configuration options for the above applications are contained in the `/etc/krb5/krb5.conf` file. Details regarding the options can be found in the `krb5.conf(4)` man page. Command line options for many of the configurable parameters are available in the client applications. These command line options override the default configuration file settings. To ensure that the traffic between the client and remote server is encrypted, a `-x` command line option may be used with the `rcp, rlogin, rsh,` and `telnet` commands. The kerberized `ftp` client provided with SEAM supports three levels of protection: clear, safe, and private. With the "clear" level, the data is unprotected and unencrypted. The "safe" level provides for integrity checking through the use of cryptographic checksum. The "private" level encrypts all transmitted data and provides integrity checking through cryptographic checksums.

In order to configure the remote application servers to utilized Kerberos, command line options must be passed to the server. To enable Kerberos in the telnet and ftp daemons you would modify the `telnetd` and `ftpd` lines in the `/etc/inetd.conf` file to look like the following:

```
telnet   stream  tcp    nowait  root   /usr/krb5/lib/telnetd  telnetd -a user
ftp      stream  tcp    nowait  root   /usr/krb5/lib/ftpd    ftpd -a
```

In this instance the location of the daemons has changed to the kerberized versions and command line options "`-a user`" and "`-a`" have been added to `telnetd` and `ftpd` lines respectively. These options specify that only Kerberos authentication be accepted. Control over whether the data stream is encrypted lies on the client side.

NFS and Kerberos

NFS provides the capability to share filesystems over a network. The SEAM software package provides three levels of Kerberos protection for NFS: `krb5, krb5i,` and `krb5p`. These levels of protection correspond to the clear, safe, and private protection levels available with the `ftp` command. The first level of protection provides only an authentication mechanism to verify the user prior to allowing access to the NFS share. The second level of protection not only verifies the user but also verifies the integrity of the data packets using cryptographic checksums.

The third level of protection is the most secure, in this level not only is authentication performed and integrity checked, but all NFS traffic is encrypted between the client and the server. The following `share` command example demonstrates how the server can specify the level of protection that will be used:

```
share -F nfs -o sec=krb5p -d "KRB5 protected home directories" /export/home
```

Note, multiple security modes may be specified for a given share; thus allowing clients to request the level of protection they desire. The recommended method of implementing NFS with Kerberos is to utilize the automounter. If the `automount` program is configured to perform all mounts that a user will need, only users with valid Kerberos credentials will be able to access the protected mounts.

Performance Degradation of NFS using Kerberos

The three levels of protection afforded by the utilization of Kerberos with NFS come at the cost of performance. The degradation in performance varies widely with the choice of protection level. The table below summarizes the results provided by Sun at a talk given by Mike Eisler at the 1999 Connectathon Conference [5].

Security Flavor	Throughput (megabytes per second)	Throughput degradation vs. AUTH_SYS	CPU Utilization on Server
AUTH_SYS	5.40	N/A	69%
Kerberos V5 - just authentication	5.26	2.6%	70%
Kerberos V5 - with integrity - MD5	4.44	17.7%	77%
Kerberos V5 - with privacy and integrity - 56bit DES/MD5	1.45	73.1%	99% (more likely 100% pegged)

These results were for a NFS copy of a 200Mb file to the server's `tmpfs` filesystem. The client and server utilized NFSv3 implemented over TCP. The machines were both 270Mhz Ultra 5s with 128Mb of RAM. The machines were connected using a 100baseT network

The results presented above clearly demonstrate that there is a high overhead cost for ensuring privacy and integrity. While individual sites can determine their security requirements with respect to privacy and integrity, some thought must be given to the processing requirements and performance degradation imposed by choosing the highest level of security.

Interoperability Information

Because of its widespread acceptance and implementation in other operating systems, Kerberos has become the de facto standard. These standard based implementations provide for cross-system operation through a heterogeneous organization. The SEAM software product has been tested in such an environment. In particular, the SEAM Interoperability Documentation [10] documents interoperability tests with the Microsoft Windows 2000 operating system. Multiple tests under multiple configurations were performed. Tests included using the SEAM product as a KDC server serving Windows 2000 and Solaris based clients, using a Windows 2000 KDC to serve Solaris based clients, etc. The results of these tests demonstrate that Solaris using the SEAM product is completely interoperational with Windows 2000 Kerberos functionality.

Complete results of these tests can be found at [10].

Conclusion

The Sun Enterprise Authentication Mechanism™ provides powerful tools to help protect a network from unauthorized access and to ensure the privacy and integrity of data on that network. Compliant with the Kerberos v5 standard, SEAM provides a robust platform upon which to base an enterprise wide Kerberos deployment. The Internet is an inherently insecure place. Widespread deployment of tools such as SEAM or other implementations of Kerberos is a first step in providing greater security in the future.

References

1. *Center for Internet Security Level 1 Benchmark and Scoring tool for Solaris.* URL: http://www.cisecurity.org/

2. David Deeths and Glenn Brunette *Using NTP to Control and Synchronize System Clocks – Part I: Introduction to NTP (July 2001).* URL: http://www.sun.com/blueprints/0701/NTP.pdf

3. David Deeths and Glenn Brunette *Using NTP to Control and Synchronize System Clocks – Part II: Basic NTP Administration and Architecture (August 2001).* URL: http://www.sun.com/blueprints/0801/NTPpt2.pdf

4. David Deeths and Glenn Brunette *Using NTP to Control and Synchronize System Clocks – Part III: NTP Monitoring and Troubleshooting (September 2001).* URL: http://www.sun.com/blueprints/0901/NTP.pdf

5. Mike Eisler. *SEAM: Sun Enterprise Authentication Mechanism (Kerberos V5 for solaris and Solaris NFS).* Connectathon, 1999. URL: http://www.connectathon.org/talks99/mre.pdf

6. Wyllys Ingersoll. *Kerberos Network Security in the Solaris™ Operating Environment (October 2001).* URL: http://www.sun.com/blueprints/1001/krb.pdf

7. Massachusetts Institute of Technology. *Kerberos: The Network Authentication Protocol.* URL: http://web.mit.edu/kerberos/www/

8. R. M. Needham and M.D. Schroeder. *Using encryption for authentication in large networks of computers.* Communications of the ACM, 21(12):993-999, December 1978

9. Clifford Neuman and Theodore Ts'o. *Kerberos: An Authentication Service for Computer Networks.* IEEE Communications Magazine 32(9): 33-38, September 1994. URL: http://www.isu.edu/gost/publications/kerberos-neuman-tso.html

10. Sun Microsystems. *SEAM 1.0 Interoperability Documentation.* URL: http://www.connectathon.org/seam1.0/

11. Sun Microsystems. *Solaris 8: System Administration Guide Volume II.* Palo Alto: Sun Microsystems, February 2000. 373-413.

12. Sun Microsystems. *Sun Enterprise Authentication Mechanism: SEAM Installation and Release Notes.* URL: http://docs.sun.com/

13. *Titan Release 4.0 Beta 1* URL: http://www.fish.com/titan/

14. Brian Tung. *The Moron's Guide to Kerberos, Version 1.2.2.* URL: http://www.isi.edu/gost/brian/security/kerberos.html

15. *YASSP: Yet Another Solaris Security package.* URL: http://www.yassp.org/

Solaris is a registered trademark of Sun Microsystems, Inc. in the United States and other countries.

Microsoft Windows 2000 is a trademark or registered trademark of the Microsoft Corporation.