



Building Secure Sun Fire™ Link Interconnect Networks Using Sun Fire™ 15K and Sun Fire™ 12K Servers

*Joe Higgins and Steven Spadaccini, Enterprise Server
Products, High End Software*

Sun BluePrints™ OnLine—August 2003



<http://www.sun.com/blueprints>

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 817-3344-10
Revision A, August 2003

Copyright 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

This document and the product to which it pertains are distributed under licenses restricting their use, copying, distribution, and decompilation. No part of the product or of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Sun BluePrints, Sun Fire, Sun HPC ClusterTools, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the US and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government Rights—Commercial use. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuels relatants à la technologie qui est décrit dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuels peuvent inclure un ou plus des brevets américains énumérés à <http://www.sun.com/patents> et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats-Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Sun BluePrints, Sun Fire, Sun HPC ClusterTools, Java, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciées de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



Please
Recycle



Adobe PostScript

Building Secure Sun Fire™ Link Interconnect Networks Using Sun Fire™ 15K and Sun Fire™ 12K Servers

In a distributed computer system, data is sent from one computer over a network to another computer. The data that is being sent across the network may be readable by unauthorized users. Data transmitted over the network is sensitive to privacy, authenticity, and point of origin attacks so it must be protected. The Sun Fire Link interconnect software is part of a distributed computer system, so it must be fortified against these attacks.

Deploying a secure distributed computer system can be difficult. This article describes how to install and deploy the Sun Fire Link product so that it can be securely managed and operated and documents the software architecture and the steps needed to secure the Sun Fire Link interconnect. The commands used in configuration steps are either Fire Link Manager (FM) or Solaris™ Operating Environment (Solaris OE) tools. This article requires a general knowledge of Solaris OE system administration and is written for advanced system administrators.

The article also includes a section on how to create, configure, and secure a Sun Fire Link fabric. The Sun Fire Link fabric is a collection of remote shared memory (RSM) partitions, compute nodes, and switch nodes.

This article covers the following topics:

- “Sun Fire Link Hardware Overview” on page 2
- “Sun Fire Link Software Overview” on page 4
- “System Configuration” on page 7
- “Fabric Configuration” on page 18

The main recommendations are:

- Follow the guidelines in [1] “Building Secure Sun Fire Link Interconnect Networks Using Midframe Servers” at:
<http://www.sun.com/solutions/blueprints/0203/817-1656.pdf>.
- Follow the guidelines in [2] “Securing the Sun Fire™ 15K and 12K System Controllers” article at:
<http://www.sun.com/solutions/blueprints/0203/817-1358.pdf>.
- Configure the `wcaa` to use the Secure Sockets Layer (SSL).
- Modify the FM keystore to include `wcapp` version 1.2 10/28/99.

The procedures for implementing these recommendations are located in the sections on “System Configuration” on page 7 and “Fabric Configuration” on page 18 following the Sun Fire Link hardware and software overviews.

Sun Fire Link Hardware Overview

The Sun Fire Link is a high-bandwidth, low-latency cluster interconnect used with Sun Fire™ 6800, Sun Fire 15K, and Sun Fire 12K servers to expand the high-end Sun Fire series system capabilities beyond the chassis. A Sun Fire Link cluster consists of up to eight Sun Fire 6800 and/or Sun Fire 15K and Sun Fire 12K nodes, connected to each other by a Sun Fire Link optical network. Each node has a separate instance of the Solaris OE running under a layer of clustering software, which can be either Sun™ Cluster software or Sun HPC ClusterTools™ software. This separate instance of the Solaris OE is also referred to as a domain. For some configurations, the interconnect hardware will include Sun Fire Link switches as well. A Sun Fire Link cluster also requires an Ethernet network to carry cluster administration traffic. This network connects all cluster components that exchange control and status or error information. A dedicated server to run the required management software is also recommended. The “Securing the Sun Fire Midframe System Controller” article discusses the midframe service processor (MSP). The MSP is a dedicated server that restricts access to the private System Controller (SC) network.

The Sun Cluster software and the Sun HPC ClusterTools software use the remote shared memory (RSM) interface for internode communication across a Sun Fire Link network. The RSM is a Sun messaging interface that is highly efficient for remote memory operations. For Sun Fire Link clusters of two or three nodes, the network connections can be either point to point (direct-connect topology) (FIGURE 1) or through Sun Fire Link switches (FIGURE 2). For larger clusters (four to eight nodes), a Sun Fire Link switch is required.

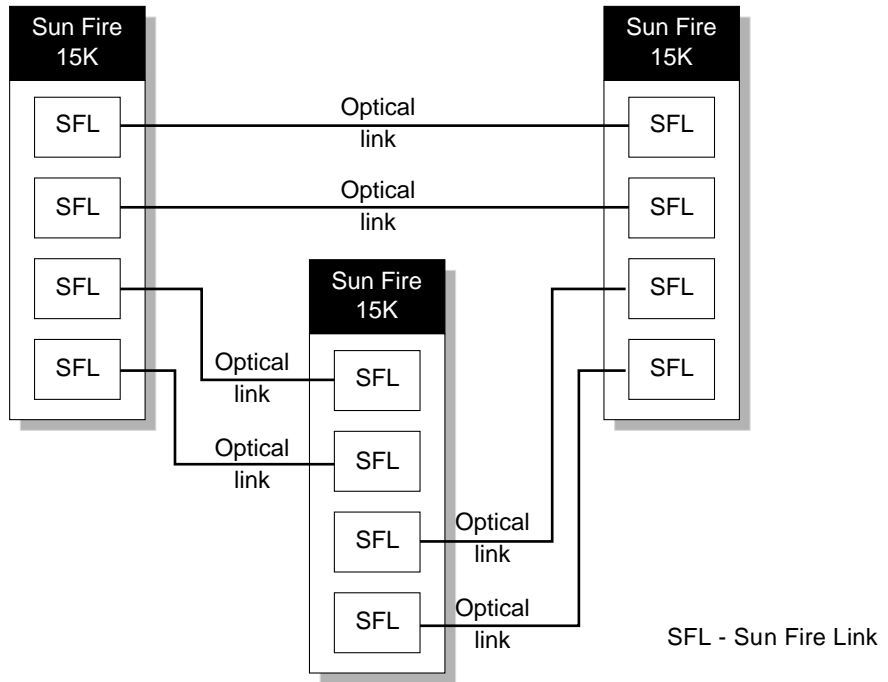


FIGURE 1 Direct-Connect Topology

The server's interface to the Sun Fire Link network is provided by a Sun Fire Link-specific I/O subsystem called the Sun Fire Link assembly. These assemblies are installed in standard server I/O slots. Each Sun Fire Link assembly contains two optical transceiver modules called Sun Fire Link optical modules. Each optical module supports a full-duplex optical link. The Sun Fire Link assemblies are installed in pairs to enhance availability and to support message striping for higher bandwidth.

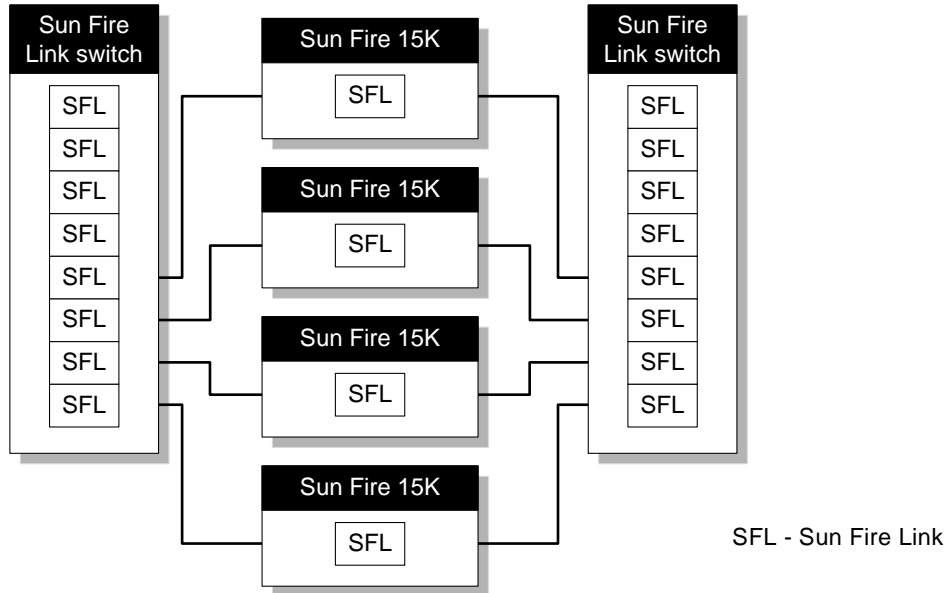


FIGURE 2 Switched Topology

Sun Fire Link Software Overview

The elements of the Sun Fire Link interconnect software stack that are configured in this article are:

- Sun Fire 15K or Sun Fire 12K `wcapp` software.
- Sun Fire Link Administration software

Other components of the software stack are described in detail in [1] “Building Secure Sun Fire Link Interconnect Networks Using Midframe Servers” at:

<http://www.sun.com/solutions/blueprints/0203/817-1656.pdf>.

Sun Fire 15K and Sun Fire 12K `wcapp` Software

The `wcapp` software is a daemon responsible for implementing Sun Fire Link clustering functionality and so forth. This software runs on the domain, SC, MSP, and so on.

The SC in the Sun Fire 15K and Sun Fire 12K systems controls the assignment of resources. Resource assignment includes which domains are on or off and which components (such as CPUs, I/O cards, and memory) are associated with domains. All of the server's configuration is stored in the SC. Network discovery and fabric configuration services are exported to the FM software through a private Java™ remote method invocation (RMI in the domain, SC, and MSP) interface. The RMI allows client applications to locate remote server objects and execute methods on those objects as though they were local objects. The RMI is the object equivalent of remote procedure calls (RPCs). The RMI interface can use the Secure Sockets Layer (SSL), which will provide integrity and confidentiality across networks in addition to providing authentication.

This article describes how to enable use of SSL to access the RMI interface.

Sun Fire Link Administration Software

The Sun Fire Link software includes tools for administrating Sun Fire Link networks. Administration of Fire Link networks includes the following tasks:

- Configuring and reconfiguring Sun Fire Link partitions
- Dynamically adding nodes to and removing nodes from partitions
- Bringing up and taking down optical links
- Enforcing domain topology constraints
- Monitoring a configured cluster for faults, such as link failures

The major components of the Sun Fire Link Administration software are:

- Sun Fire Link Manager
- FM proxies
- Sun™ Management Center agents
- Sun Management Center console

FIGURE 3 shows where the software is located and how it communicates.

Sun Fire Link Manager

The Sun Fire Link Manager (FM) is installed on a host that is external to the Sun Fire Link cluster. The FM must be installed on the MSP. The FM is a Java application that communicates with the managed entities (compute nodes and switches) through the RMI. It is responsible for managing and configuring the Sun Fire Link fabric. The fabric is a collection of RSM partitions, compute nodes, and switch nodes.

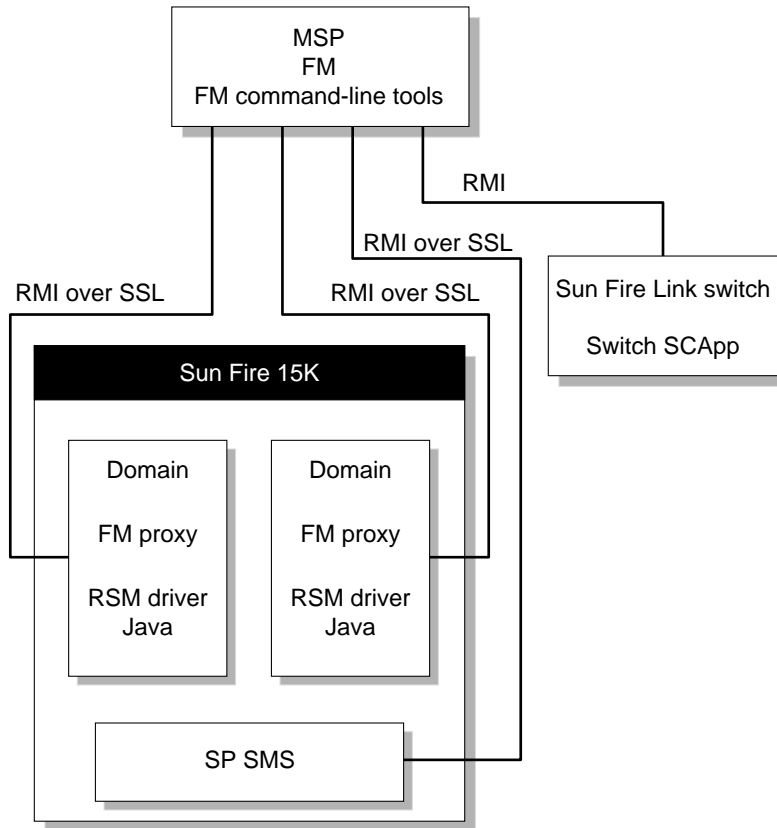


FIGURE 3 Sun Fire Link Software Location and Communication

The major functions of the FM are:

- Creating fabrics
- Creating switched and direct-connect topologies
- Adding and removing nodes in the switch topology
- Modifying the striping level of a partition

Given the requested topology, stripe level, and node membership, the FM computes configuration information for each node. The FM then distributes these configurations to every node of the fabric. This configuration information contains items such as striping level (the number of links between each node) and the cluster ID. The configuration data is stored in the FM configuration file. The FM data files represent the persistent form of the FM. If the FM is stopped and restarted, the FM configuration file restores the memory-resident data structures. This file contains the nodes in the fabrics, which partitions exist, and what links are used in which partitions.

Another file that the FM manages is the password file, which contains password information for the domains, switch SCs, and midframe SCs. This data is very sensitive and should always be guarded. The FM has a set of command-line tools that are used in the example given later. These tools allow the FM functionally to be accessed. Role-based access control (RBAC) is used to control access to these files. These files should never be copied to a unsecure location. Copies should never be made using unsecure communication channels. It is also important to treat these files as sensitive data when they are backed up.

The FM executes its functionality through an RMI interface that is called by the FM command-line interface tools. It is the access point when the Sun Management Center executes the FM functionality.

All FM command-line interface tools must be executed on the MSP. Access to the RMI interface is protected by a community password, and the FM RMI interface refuses connections from systems other than the MSP. The `wcapp` configuration must be modified so that it uses RMI between the FM and `wcapp`.

System Configuration

The system configuration procedures are:

- “Following the Midframe Article” on page 7
- “Creating the `wcappKeyStore`” on page 8
- “Installing the FM Public Key In the `wcappKeyStore`” on page 11
- “Installing the `wcapp` Public Key In the `fmKeyStore`” on page 13
- “Configuring `wcapp` To Use SSL” on page 14

Following the Midframe Article

To build a secure Sun Fire Link network, follow the recommendations of the “Building Secure Sun Fire Link Interconnect Networks Using Midframe Servers” article at:

<http://www.sun.com/solutions/blueprints/0203/817-1656.pdf>.

If you are building a direct-connect topology, an MSP is not required. However, you still must set up the FM’s proxy on each domain to use SSL for communication between it and the FM. If you do not deploy an MSP, you must select the machine where the FM resides with caution.

Creating the wcappKeyStore

“Building Secure Sun Fire Link Interconnect Networks Using Midframe Servers” discusses the motivation and technology behind SSL. For an overview of SSL and public key cryptography, refer to that article.

Two keystores were created: the FM Proxy keystore and the FM keystore. In this article, an additional keystore and another key to the FM keystore are created. FIGURE 4 shows where the certificate (public key) and private keys are distributed.

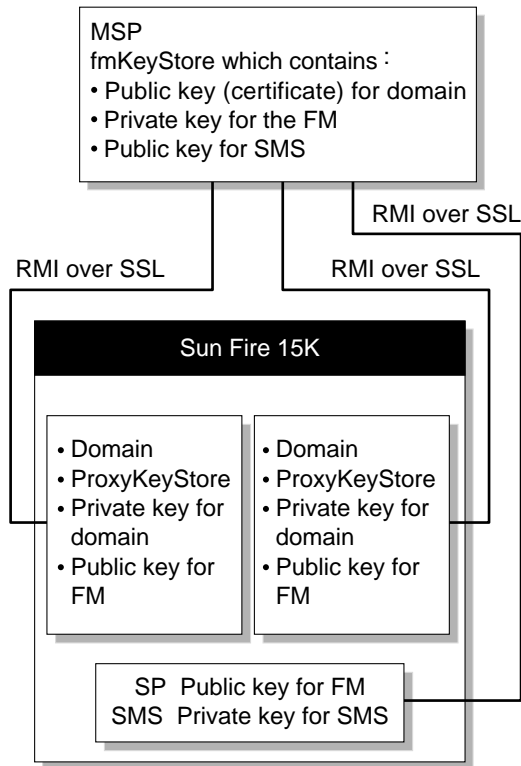


FIGURE 4 Public and Private Key Locations

▼ To Create the `wcappKeyStore`

1. Create the keystore and generate the key pair by typing the following command in a safe private directory on the MSP.

Note – You must type this command as a single line. Multiple lines are used in the examples for legibility purposes only. Press return for the `fmkey` password.

```
# /usr/bin/keytool -genkey -dname "cn=CommonName, ou=OrganizationName, o=CompanyName,  
s=SN c=US" -alias wcappKey -keystore wcappKeyStore -validity 180  
Enter keystore password: YourPassword  
Enter key password for fmKey: Password  
(RETURN if same as keystore password):
```

The preceding command creates the keystore named `fmKeyStore` in the working directory, and assigns it the password *YourPassword*. Substitute a password for *YourPassword*. This password must be kept secret to the administrators. This password is referred to as the keystore password. You must remember this password because it is used in other steps.

The italicized items in the preceding command represent a value and the keywords are abbreviations for the following:

TABLE 1 X.500 Distinguished Names

KeyWord	X.500 Distinguished Names	Example
cn	CommonName (Name of person)	John Smith
ou	OrganizationUnit (department)	Purchasing
on	OrganizationName (company)	ABC Systems, Inc.
ln	LocalityName (city name)	Burlington
s	Statement (state)	MA
c	Country	US

The distinguished names are used to identify entities, such as those named by the subject and issuer.

Caution – Keytool generates a public and private key pair for the entity `fmkeys`. The generated key expires in 180 days. Every 180 days you must generate a new private and public key and replace the private key on the proxy and public key on the FM.

2. Verify that the keystore was correctly created and the key entry is contained in the keystore.

Substitute your password for the keystore password you specified in the previous command.

```
# keytool -list -keystore wcappKeyStore
Enter keystore password: YourPassword

Keystore type: jks
Keystore provider: SUN

Your keystore contains 1 entry:

wcappkeys, Fri Apr 11 12:11:44 EDT 2003, keyEntry,
Certificate fingerprint (MD5): F1:11:FF:90:B0:D8:C6:DE:23:CE:36:3F:81:B2:30:36
```

The MD5 fingerprint and date will be will be different than those shown.

3. Generate a file that contains the certificate (public key) for the `wcapp` key.

This public key will be installed in the `fmKeyStore`.

```
# /usr/bin/keytool -export -alias wcappKey -keystore wcappKeyStore >
wcAppCert
Enter keystore password: YourPassword
```

4. Verify that the public key was created:

```
# /usr/bin/keytool -printcert -file wcAppCert
Owner: CN=CommonName, OU=OrganizationName, O=CompanyName, ST="SN c=US"
Issuer: CN=CommonName,ls -OU=OrganizationName, O=CompanyName, ST="SN c=US"
Serial number: 3cb70740
Valid from: Fri Apr 11 12:11:44 EDT 2003 until: Thus Oct 09 12:11:44 EDT
2003
Certificate fingerprints:
    MD5:  F1:11:FF:90:B0:D8:C6:DE:23:CE:36:3F:81:B2:30:36
    SHA1: 6C:76:5D:E9:64:84:08:E2:95:0B:64:95:70:6D:3F:E9:F5:D5:87:7E
```

Assume that the strings for CommonName and the other keywords are what you entered in the `-genkey` keytool in Step 1. The certificate fingerprint and date should be the same as that displayed when the `fmKeyStore` was listed in Step 2.

Installing the FM Public Key In the `wcappKeyStore`

For the `wcapp` program to authenticate the SSL connection between the FM and `wcapp` the FM public key must be placed in the `wcappKeyStore`. First you must extract the public key from the FM keystore. For the following command to work, you must have followed the instructions in the article “Building Secure Sun Fire Link Interconnect Networks Using Midframe Servers.”

▼ To Install the FM Public Key in the `wcappKeyStore`

1. **Generate a file that contains the certificate (public key) for the `proxyKey`.**

This public key will be installed in the `fmKeyStore`.

```
# /usr/bin/keytool -export -alias fmKey -keystore /opt/SUNWwcfm/classes/
fmKeyStore > fmCert
Enter keystore password: YourPassword
```

2. Verify that the public key was created.

```
# /usr/bin/keytool -printcert -file fmCert
Owner: CN=CommonName, OU=OrganizationName, O=CompanyName, ST="SN
c=US"
Issuer: CN=CommonName,ls -OU=OrganizationName, O=CompanyName,
ST="SN c=US"
Serial number: 3cb70740
Valid from: Fri Apr 11 12:15:44 EDT 2003 until: Thus Oct 09
12:11:44 EDT 2003
Certificate fingerprints:
MD5: F2:11:FF:90:B0:D8:C6:DE:23:CE:36:3F:81:B2:30:36
SHA1: 6C:76:5D:E9:64:84:08:E2:95:0B:64:95:70:6D:3F:E9:F5:D5:87:7E
```

Assume that the strings for CommonName and the other keywords are what you entered in the `-genkey` keytool in Step 1 "Creating the `wcappKeyStore`" on page 8.

3. Import the FM public key into `wcappKeyStore`:

```
# /usr/bin/keytool -import -file fmCert -keystore wcappKeyStore

Enter keystore password: YourPassword

Owner: CN=CommonName, OU=OrganizationName, O=CompanyName,
ST="SN c=US"

Issuer: CN=CommonName, OU=OrganizationName, O=CompanyName,
ST="SN c=US"
Valid from: Fri Apr 11 12:11:44 EDT 2003 until: Thus Oct 09
12:11:44 EDT 2003
Certificate fingerprints:
MD5: F1:11:FF:90:B0:D8:C6:DE:23:CE:36:3F:81:B2:30:36
SHA1:6C:76:5D:E9:64:84:08:E2:95:0B:64:95:70:6D:3F:E9:F5:D5:87:7

Trust this certificate? [no]: yes

Certificate was added to keystore
```

4. Validate that `wcappKeyStore` contains the private key for the proxy and the public key for the F:

```
# /usr/bin/keytool -list -keystore wcappKeyStore
keystore password: YourPassword
Keystore type: jks
Keystore provider: SUN
Your keystore contains 2 entries:

wcapp Key, Fri Apr 11 12:15:44 EDT 2003
Certificate fingerprints(MD5)
F2:11:FF:90:B0:D8:C6:DE:23:CE:36:3F:81:B2:30:36
fmkey, Fri Apr 11 12:11:44 EDT 2003 trustedCertEntry,
Certificate fingerprint (MD5):
F1:11:FF:90:B0:D8:C6:DE:23:CE:36:3F:81:B2:30:36
```

Installing the wcapp Public Key In the fmKeystore

▼ To Install the wcapp public key In the fmKeyStore

1. Type the following command to install the fmKeyStore.

The fmKeyStore needs a public key for the wcapp key pair.

```
# /usr/bin/keytool -import -file wcappCert -keystore fmKeyStore
```

```
Enter keystore password: YourPassword
```

```
Owner: CN=CommonName, OU=OrganizationName, O=CompanyName,  
ST="SN c=US"
```

```
Serial Number:84848484
```

```
Issuer: CN=CommonName, OU=OrganizationName, O=CompanyName,  
ST="SN c=US"
```

```
Valid from: Fri Apr 11 12:15:44 EDT 2003 until: Thus Oct 09  
12:11:44 EDT 2003
```

```
Certificate fingerprints:
```

```
MD5: F2:11:FF:90:B0:D8:C6:DE:23:CE:36:3F:81:B2:30:36
```

```
SHA1:6C:76:5D:E9:64:84:08:E2:95:0B:64:95:70:6D:3F:E9:F5:D5:87:7
```

```
Trust this certificate? [no]: yes
```

```
Certificate was added to keystore
```

2. **Validate that `wcappKeyStore` contains the private key for the private key for the proxy and public key for the FM:**

```
# /usr/bin/keytool -list -alias -keystore /opt/SUNWwcfm/classes/fmKeyStore
keystore password: YourPassword
Keystore type: jks
Keystore provider: SUN

Your keystore contains 2 entries:
fmKey, Fri Apr 11 12:11:44 EDT 2003
Certificate fingerprints(MD5)
F1:11:FF:90:B0:D8:C6:DE:23:CE:36:3F:81:B2:30:36
wcappKey, Fri Apr 11 12:15:44 EDT 2003 trustedCertEntry,
Certificate fingerprint (MD5):
F2:11:FF:90:B0:D8:C6:DE:23:CE:36:3F:81:B2:30:36
```

Configuring `wcapp` To Use SSL

These steps must be performed on every SC (both the primary and the backup) in your configuration. For example, if you have two Sun Fire 15K or Sun Fire 12K systems you must repeat the following installation steps on all four SCs:

- “To Configure `wcapp` Java Virtual Machine to use SSL” on page 14
- “To Install the `wcappKeyStore`” on page 15
- “To Create the `ssl.info` File” on page 17
- “To Stop and Restart `wcapp`” on page 17

▼ To Configure `wcapp` Java Virtual Machine to use SSL

1. **Download the Java™ Secure Socket Extension (JSSE) 1.0.3 program.**

You can download the file to any location on your local disk. Note that JSSE 1.0.3 requires that you have Java 1.2.1 or greater already installed. You can download the file from:

<http://java.sun.com/products>.

The file name is `jsse-1_0_3-do.zip`.

2. Uncompress and extract the downloaded file by typing the following command to unzip the download.

This will create a directory named `jsse1.0.2`, with two subdirectories named `doc` and `lib`. The following command will unzip the download:

```
# unzip jsse-1_0_3-do.zip
Archive:  jsse-1_0_3-do.zip
  inflating: jsse1.0.3/BUGS.html
  inflating: jsse1.0.3/CHANGES.txt
  inflating: jsse1.0.3/COPYRIGHT.ht
... Many more files are listed
```

3. The JSSE lib subdirectory contains the extension files `jsse.jar`, `jcert.jar` and `jnet.jar`. Copy these files into the `/usr/java1.2/lib/ext` (installed extension) directory:

```
# cp lib/jsse.jar JRE/lib/ext/jsse.jar
# cp lib/jcert.jar JRE/lib/ext/jcert.jar
# cp lib/jnet.jar JRE/lib/ext/jnet.jar
```

4. Verify that the files exist and they are owned by root.

```
# ls -l $JRE/lib/ext
-rw-r--r--  1 root      root 7637 Feb 20 10:17 jcert.jar
-rw-r--r--  1 root      root 3098 Feb 20 10:17 jnet.jar
-rw-r--r--  1 root      root 463471 Feb 20 10:17 jsse.jar
```

5. Register the Sun JSSE provider.

The standard JSSE comes with a cryptographic service provider (provider for short) named SunJSSE. Although the SunJSSE provider must be configured explicitly, this provider should be registered statically. The registration is done by editing the security properties file, which is located at:

`JRE/lib/security/java.security`.

One of the types of properties contained in the `java.security` file is of the following form:

```
security.provider.n=providerClassName
```

This line declares the security provider and its preference.

6. Add a new line to that section and install the standard provider shipped with the Java run-time environment (JRE).

The entries should now look like:

```
security.provider.1=sun.security.provider.Sun
security.provider.2=com.sun.net.ssl.internal.ssl.Provider
```

▼ **To Install the wcappKeyStore**

The previously created keystore (wcappKeyStore) must be distributed to each SC domain. To move the key, it is recommended that you use an encrypted file, that is, *scp*, to copy the file onto your system.

1. Execute the following command.

Substitute the machine name of *MSP* for the server that is acting as your MSP.

```
# scp MSP:/privatedir/wcappKeyStore /opt/SUNWSMS/SMS1.3/classes/wcappKeyStore
```

2. Verify that this file is root read only.

It is important to protect private key.

```
# ls -l /opt/SUNWSMS/SMS1.3/wcappKeyStore
-rw----- 1 root    root 7637 Feb 20 10:17 /opt/SUNWSMS/SMS1.3/classes/
wcappKeyStore
```

3. Edit the java.policy file so that the proxy has access to the fmKeyStore file.

Add the following lines above the `};` in the policy file. The wcapp java.policy file is located in:

`/opt/SUNWSMS/SMS1.3/classes/com/sun/wildcat/common/server.policy`

```
permission java.io.FilePermission "/opt/SUNWSMS/SMS1.3/classes/wcappKeyStore", "read";
permission java.io.FilePermission " /opt/SUNWSMS/SMS1.3/classes/ssl.info", "read";
```

▼ To Create the ssl.info File

The `ssl.info` file contains the information that `wcapp` needs to use the SSL.

1. **Create a file called file** `/opt/SUNWSMS/SMS1.3/classes/ssl.info` **containing the following lines.**

Substitute the KeyStore password for *YourPassword*.

```
KEY_STORE_PASSPHRASE=YourPassword
KEY_STORE_LOCATION=/opt/SUNWSMS/SMS1.3/classes/wcappKeyStore
```

2. **Verify that the file is root read only.**

This information is sensitive. Access to the file should be restricted to root. The following command will verify the access to the file:

```
# ls -l /opt/SUNWSMS/SMS1.3/classes/ssl.info
-rw----- 1 root    root 7637 Feb 20 10:17 /opt/SUNWSMS/SMS1.3/classes/ssl.info
```

▼ To Stop and Restart wcapp

To make `wcapp` use SSL and the enhanced security settings, you must stop and restart `wcapp` by stopping and restarting the entire SMS. If failover is enabled, the SC will failover when SMS is stopped so failover should be disabled and one SC modified and tested before you define the other SC as main. Modify and test the SC before you re-enable failover.

1. **Type the following command to stop the proxy:**

```
#!/etc/init.d/sms stop
```

2. **Type the following command to restart wcapp.**

```
#!/etc/inid.d/sms start
```

3. Verify that `wcapp` restarted with enhanced security.

Only the critical lines of output are listed. The rest are ignored for readability.

```
# tail /tmp/scapout.out
.
.
.
using SSL.
.
.
.
```

4. Repeat Steps 1 through 3 for each SC.

Fabric Configuration

This section contains an example topology and an example set of nodes. The example configuration (FIGURE 5) contains four nodes and a switch. The key step is the creation of the `.xml` file. The example file uses `password` for the string you should use to substitute a password. The following example has two compute nodes and four switches. The names of the compute nodes are: `sc1` and `sc2`. The switch nodes are `greatsandy`, `greatbasin`, `dryvalley`, and `saltflats`. This section contains the steps and configuration file to create a single partition called `part1`.

▼ To Configure the Fabric

The following example configures a fabric using the FM command-line tools.

1. Type the following command to switch to the `fmadmin` role.

If you are not using RBAC, skip this step.

```
# su fmadmin
```

2. Type the following command to configure the fabric:

```
# create_fabric sf1
```

Note – Fabric `sf1` was created when you followed the configure fabric procedures in “Building Secure Sun Fire Link Interconnect Networks Using Midframe Servers.”

FIGURE 5 shows the fabric cabling.

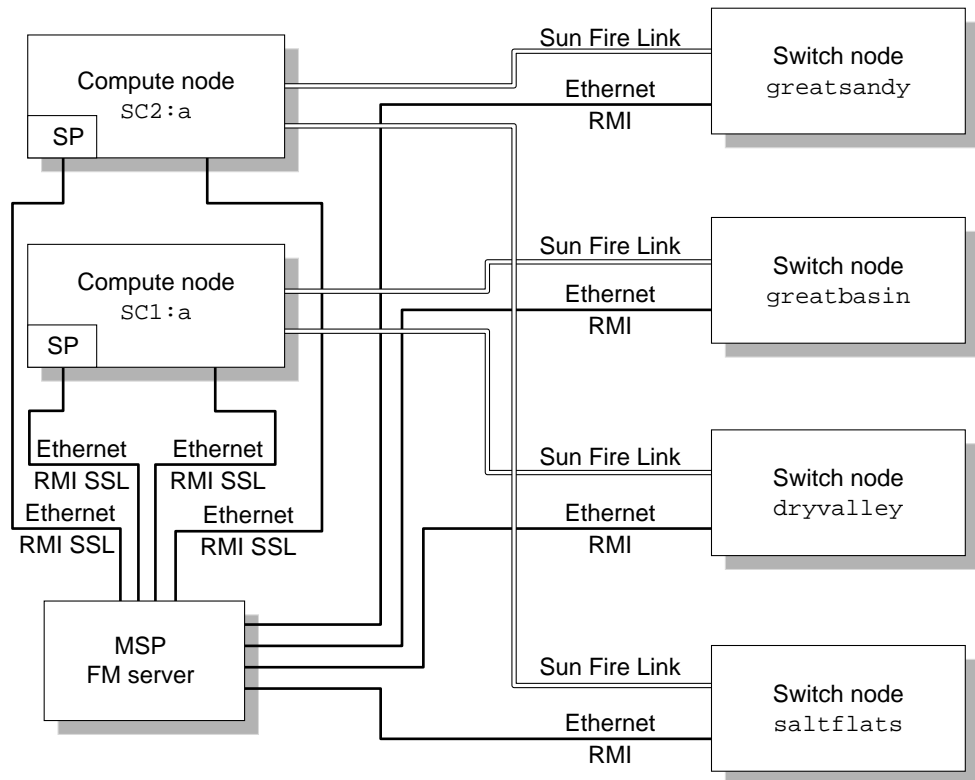


FIGURE 5 Fabric Cabling Diagram.

CODE EXAMPLE 1 shows the contents of the .xc_sfl.xml file for this topology.

CODE EXAMPLE 1 File xc_sfl.xml

```
<?xml version= "1.0" encoding="UTF-8"?>
<!DOCTYPE fabric SYSTEM "fabric.dtd">
<fabric>
  <fname>sfl</fname>
  <config_file>
</config_file>
  <members>
    <switch_node>
      <node>
        <sc_name>greatsandy</sc_name>
        <sc_user_name>sfluser</sc_user_name>
        <sc_password>password</sc_password>
        <chassis_type>WCIX_SWITCH</chassis_type>
      </node>
    </switch_node>
    <switch_node>
      <node>
        <sc_name>greatbasin</sc_name>
        <sc_user_name>sfluser</sc_user_name>
        <sc_password>password</sc_password>
        <chassis_type>WCIX_SWITCH</chassis_type>
      </node>
    </switch_node>
    <switch_node>
      <node>
        <sc_name>dryvalley</sc_name>
        <sc_user_name>sfluser</sc_user_name>
        <sc_password>password</sc_password>
        <chassis_type>WCIX_SWITCH</chassis_type>
      </node>
    </switch_node>
    <switch_node>
      <node>
        <sc_name>saltflats</sc_name>
        <sc_user_name>sfluser</sc_user_name>
        <sc_password>password</sc_password>
        <chassis_type>WCIX_SWITCH</chassis_type>
      </node>
    </switch_node>
    <rsm_node>
      <node>
        <sc_name>sc1</sc_name>
```

CODE EXAMPLE 1 File (Continued)xc_sfl.xml

```
        <sc_user_name>sfluser</sc_user_name>
        <sc_password>password<sc_password>
        <chassis_type>S72</chassis_type>
    </node>
    <domain_name>a</domain_name>
    <hostname>scl-a</hostname>
    <host_user>sfluser</host_user>
    <host_password>password<host_password>
</rsm_node>
<rsm_node>
    <node>
        <sc_name>sc2</sc_name>
        <sc_user_name>sfluser</sc_user_name>
        <sc_password>password<sc_password>
        <chassis_type>S72</chassis_type>
    </node>
    <domain_name>a</domain_name>
    <hostname>scl-a</hostname>
    <host_user>sfluser</host_user>
    <host_password>password<host_password>
</rsm_node>
</members>
<partitions>
    <partition type="RSM" topology="WcixSwitch">
        <pname>part1</pname>
        <stripping_level>4 </stripping_level>
        <partition_members>
            <node_partition_member>
                <sc_name>scl</sc_name>
                <domain_name>a</domain_name>
            </node_partition_member>
            <node_partition_member>
                <sc_name>sc2</sc_name>
                <domain_name>a</domain_name>
            </node_partition_member>
            <switch_partition_member>
                <sc_name>greatsandy</sc_name>
            </switch_partition_member>
            <switch_partition_member>
                <sc_name>greatbasin</sc_name>
            </switch_partition_member>
            <switch_partition_member>
                <sc_name>dryvalley</sc_name>
```

CODE EXAMPLE 1 File (Continued)xc_sfl.xml

```
        </switch_partition_member>
        <switch_partition_member>
            <sc_name>saltflats</sc_name>
        </switch_partition_member>
    </partition_members>
</partition>
</partitions>
</fabric>
```

Note that the .xml file contains password strings and should only be placed in a secure directory. You must be very careful with this file.

References

[1] Higgins, Joe, "Building Secure Sun Fire Link Interconnect Networks Using Midframe Servers" *Sun BluePrints Online*, February 2003.

To access this article online, go to: <http://www.sun.com/solutions/blueprints/0203/817-1656.pdf>.

[2] Noordergraaf, Alex and Nimeh, Dina "Securing the Sun Fire™ 15K and 12K System Controllers" *Sun BluePrints Online*, July 2002.

To access this article online, go to: <http://www.sun.com/solutions/blueprints/0203/817-1358.pdf>.

Related Resources

"RBAC in the Solaris™ Operating Environment," Whitepapers, Sun Microsystems, Inc. at:

<http://www.sun.com/software/whitepapers/wp-rbac/#overview>.

This white paper provides additional information about using and implementing RBAC in the Solaris OE.

“Java™ Secure Socket Extension (JSSE) 1.0.3_02.” Java product document, Sun Microsystems, Inc. at:

<http://java.sun.com/products/jsse/index-103.html>.

This document provides additional details on JSSE version 1.0.3_02, including important features and updated information.

“keytool - Key and Certificate Management Tool,” Java product document, Sun Microsystems, Inc. at:

<http://java.sun.com/products/jdk/1.2/docs/tooldocs/solaris/keytool.html>

This manual explains how to manage keys and certificates using keytool.

The following articles are published as part of the Sun security blueprints. These articles provide additional details and best practices on securing the system.

To access these articles go to:

<http://www.sun.com/solutions/blueprints/pubs.html>.

Noordergraaf, Alex and Nimeh, Dina, “Securing the Sun Fire Midframe System Controller,” *Sun BluePrints OnLine*, February 2003.

Noordergraaf, Alex, “Securing Sun Enterprise™10000 System Service Processors,” *Sun BluePrints OnLine*, March 2002.

Noordergraaf, Alex and Nimeh, Dina, “Securing Sun Fire™ 15K and 12K System Controllers: updated for SMS 1.2,” *Sun BluePrints OnLine*, July 2002.

Weise, Joel "Public Key Infrastructure Overview," *Sun BluePrints OnLine*, August 2001.