



Scalix Setup and Configuration Guide

Version 11

Revision 4

Scalix Setup and Configuration Guide

Published by Scalix, Inc.
149 Madison Ave. Suite 302
New York, NY 10016
USA

Copyright © 2008 Scalix, Inc.
All rights reserved.

Product Version: 11
Document Revision: 4

Notice for Open-Source Software

Copyright (c) 1998-2007 The OpenSSL Project. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Notices

The information contained in this document is subject to change without notice.

Scalix Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Scalix Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

UNIX is a registered trademark of The Open Group.

Linux is a registered trademark of Linus Torvalds.

Fedora is a registered trademark of Red Hat, Inc. Red Hat is a registered trademark of Red Hat Software, Inc. RPM is a trademark of Red Hat Software, Inc.

SUSE is a registered trademark of Novell, Inc.

Java is a registered trademark of Sun Microsystems, Inc.

Microsoft, Windows 2000, Windows NT, Exchange, Outlook, and Active Directory are registered trademarks or trademarks of Microsoft Corporation.

Mozilla is a trademark of the Mozilla Foundation.

Eudora is a registered trademark of QUALCOMM Incorporated.

McAfee and VirusScan are registered trademarks of McAfee, Inc.

Trend Micro, InterScan, and VirusWall are registered trademarks of Trend Micro Incorporated.

ClamAV is a trademark of Sourcefire, Inc.

SpamAssassin is a trademark of the Apache Software Foundation.

Thawte and VeriSign are registered trademarks of VeriSign, Inc.

Veritas is a registered trademark of Symantec Operating Corporation.

Xandros

All other company names, product names, service marks, fonts, and logos are trademarks or registered trademarks of their respective companies.

Xandros Server is a trademark of Xandros, Inc.

Restricted Rights Legend

Use, duplication, or disclosure is subject to restrictions as set forth in contract subdivision (c)(1)(ii) of the Rights in Technical Data and Computer Software clause 52.227-FAR14.

Contents

Introduction to This Guide	7
About This Guide	7
How to Use This Guide	7
Related Documents	8
Getting Help	8
Basic Setup and Configuration	9
Verifying Connectivity.	11
Testing Connectivity Inside the System	11
Testing Connectivity with Outside Systems	12
Virus Protection	13
Introduction	13
Installing Anti-Virus Protection	15
Configuring Anti-Virus Protection	17
Testing the Anti-Virus Installation	20
Updating the Anti-Virus Installation	21
Using the Microsoft Outlook Security Model.	22
Improving Performance	22
Spam Protection	23
Introduction	23
Where to Install SpamAssassin	24
System Requirements and Installation	24
Configuration	25
SMTP Authentication and Spam Protection	28
Using a DNS Block List	42
Authentication	43
Introduction	43
Using Pluggable Authentication Modules (PAM)	44
Configuring Scalix for LDAP Authentication	49
Configuring Scalix for Windows NT Authentication	52
Configuring Scalix for Kerberos Authentication	53
Securing Scalix	61
Overview	61

Internal Security Precautions	62
Using a VPN	63
Using an Apache Proxy Server.	64
Using stunnel	68
Using Certificates.	74
Other Forms of Security	75
Advanced Setup and Configuration	77
Integrating with Active Directory.	78
Integrating with Active Directory	78
Installing the Schema Extensions.	79
Installing the GUI Extensions	80
Setting Up Synchronization Agreements	81
Using Active Directory to Manage Scalix Mailboxes and Groups	85
Scalix Active Directory Extensions	90
Integrating with an LDAP Directory	92
About the LDAP Server and Directories.	92
Configuring the LDAP Server	93
Starting and Stopping the LDAP Server	94
LDAP and Scalix Attribute Type Mappings	94
LDAP Commands	94
Multiserver Environments.	96
Distributed Architecture	96
Setting Up High Availability	97
Routing Mail	102
Synchronizing Directories	103
Synchronizing Public Folders	106
Configuring Outbound Internet Messages	106
Server Trust Relationships	107
Localizing Scalix	110
Introduction	110
Localizing Microsoft Outlook	111
Localizing Scalix Web Access.	112
Localizing the Scalix Search and Index Service	113
Localizing for Japanese Language Characters	115
Hosting	116
Introduction	116

Restrictions118
Managing Hosting119
Managing User Accounts120
Moving a Company121

Introduction to This Guide

About This Guide

This guide outlines setup and configuration for a Scalix server, including anti-virus and anti-spam protection, routing between multiple servers, and integrating directories. There are two sections: basic and advanced setup.

For information on creating and managing users and groups, calendars, contact lists, backups, public folder maintenance, and so on, see the *Scalix Administration Guide*.

For an introduction to Scalix, see the *Introduction to Scalix* and *Scalix Architecture* chapters of the *Scalix Installation Guide*.

How to Use This Guide

This guide uses the following typographical conventions.

Table 1: Conventions Used

Convention	Explanation
<Angle Brackets>	Values that you need to supply are sometimes shown using angle brackets. For example: http://<server_name>/webmail
Numbered and alphabetized lists versus bullets	Numbered and alphabetized lists denote steps to be followed while bullets provide information.
Buttons	The boldface font indicates a button, a link, a field, or other user interface element to click or press as well as a keyboard stroke. For example: Click Finish or type in the Username field.
Code	This smaller font indicates code to write or run. For example: <code>./scalix-installer</code>
<i>Italics</i>	Indicates a document or section, a directory path, a file, or the name of a window. For example: Open the <i>/var/opt/scalix</i> folder. Or: The <i>Reply</i> screen appears.

Related Documents

Scalix manuals include:

- *Scalix Release Notes*
- *Scalix Installation Guide*
- *Scalix Migration Guide*
- *Scalix Setup and Configuration Guide*
- *Scalix Client Deployment Guide*
- *Scalix Administration Guide*
- *Scalix API Guide*

In addition, there are online help systems in:

- Scalix Management Console
- Scalix Web Access
- Microsoft Outlook (when enabled for the Scalix connector)

Getting Help

For help with installation, contact technical support at **support@scalix.com**

For the latest documents, see

<http://www.scalix.com/community/downloads/documentation.php>

For documents, a knowledge base, and forums, see

<http://www.scalix.com/community/resources/>

Basic Setup and Configuration

The following chapters outline basic setup.

Section Contents

This section includes the following chapters:

- “Verifying Connectivity” on page 11
- “Virus Protection” on page 13
- “Spam Protection” on page 23
- “Authentication” on page 43
- “Securing Scalix” on page 61

Advanced tasks are outlined in their own section.

Verifying Connectivity

This chapter outlines tests to ensure connectivity between servers and clients.

Contents

This chapter includes the following information:

- “Testing Connectivity Inside the System” on page 11
- “Testing Connectivity with Outside Systems” on page 12

Testing Connectivity Inside the System

Check whether you can send and receive messages inside the Scalix system.

To test for connectivity inside the system

- 1 In Scalix Management Console (SAC), create two user accounts by clicking the **Users** icon on the toolbar and then the **Create Users** button at the bottom of the window.
- 2 In a Web browser, log in to Scalix Web Access as the first user and send a message to the second user. The login format is <username>@<yourcompany.com>/webmail, for example jane.rogers@xandros.com/webmail
- 3 Log in to Scalix Web Access as the second user and verify that the message arrived.
- 4 Reply to the message.
- 5 Return to Scalix Web Access as the first user and check that the reply arrived. (Click **Send/Receive** to get mail.)

Testing Connectivity with Outside Systems

Check whether you can send and receive messages to clients outside the Scalix system.

To test connectivity with outside clients

- 1 With a user account created as outlined in the previous procedure, log in to Scalix Web Access as that user.
- 2 Send a message to an outside user account.
- 3 Log in to the outside account and verify that the message arrived.
- 4 Reply to the message and check that the message arrived.

Virus Protection

This chapter describes which anti-virus products work with Scalix and how to install and configure them. If you do not intend to use virus protection, skip this chapter.

Contents

This chapter includes the following information:

- “Introduction” on page 13
- “Installing Anti-Virus Protection” on page 15
- “Configuring Anti-Virus Protection” on page 17
- “Testing the Anti-Virus Installation” on page 20
- “Updating the Anti-Virus Installation” on page 21
- “Using the Microsoft Outlook Security Model” on page 22
- “Improving Performance” on page 22

Introduction

The Scalix virus-protection framework can integrate the following third-party anti-virus applications:

- Clam Anti-Virus (ClamAV)
- McAfee VirusScan for Linux
- Trend Micro InterScan VirusWall

Scanning is performed within the service router, which is superior to gateway solutions because it also scans internal e-mail. Scalix accomplishes this by extending message delivery rules to include additional rule sets and a special “mapper” script that detect and delete infected messages.

Message Delivery Rules

Scalix runs anti-virus software as a set of rules. The rules tell the service router to test a message and carry out specific actions based on the results. In the case of anti-virus software, the most effective rule is simply the following: If infected, delete the message.

The way Scalix is set up, rules are contained in “rule sets,” which are text files located in the `/var/opt/scalix/<nn>/s/rules` folder in a file to be named ALL-ROUTES.VIR. Each rule set can be associated with one or more Scalix routes, but the virus scanning rule set applies to all routes.

Mapper Scripts

All incoming messages pass through the service router, which you configure to perform virus-scanning tasks based upon rules. The router instructs a “mapper” script (`omvscan.map`) to invoke the third-party anti-virus software, which performs the scan and returns the results to the router.

When the anti-virus software detects a virus, the service router refers to the rule sets and they determine whether the message is discarded.

Process

The basic process for installing, configuring, and testing an anti-virus application on your Scalix server is:

- Acquire and install the anti-virus engine
- Set up the rules file to run messages through the anti-virus engine
- Set up and configure integration through a mapper script
- Restart the service router to activate these configuration changes

Each process is outlined here.

Installing Anti-Virus Protection

You can install and run ClamAV, McAfee VirusScan for Linux, or Trend Micro InterScan VirusWall. ClamAV is an open-source application. It and anti-virus updates are free, but installation can be lengthy. McAfee VirusScan for Linux is a command-line application that is very easy to install and has a free trial. Trend Micro InterScan VirusWall also has a free trial and has an easy user interface. Please note that at the time of writing, the latest version of Red Hat supported by Trend Micro was version 4 with update 2 and that it does not support Red Hat 5 (though it is known to install); check the product system requirements.

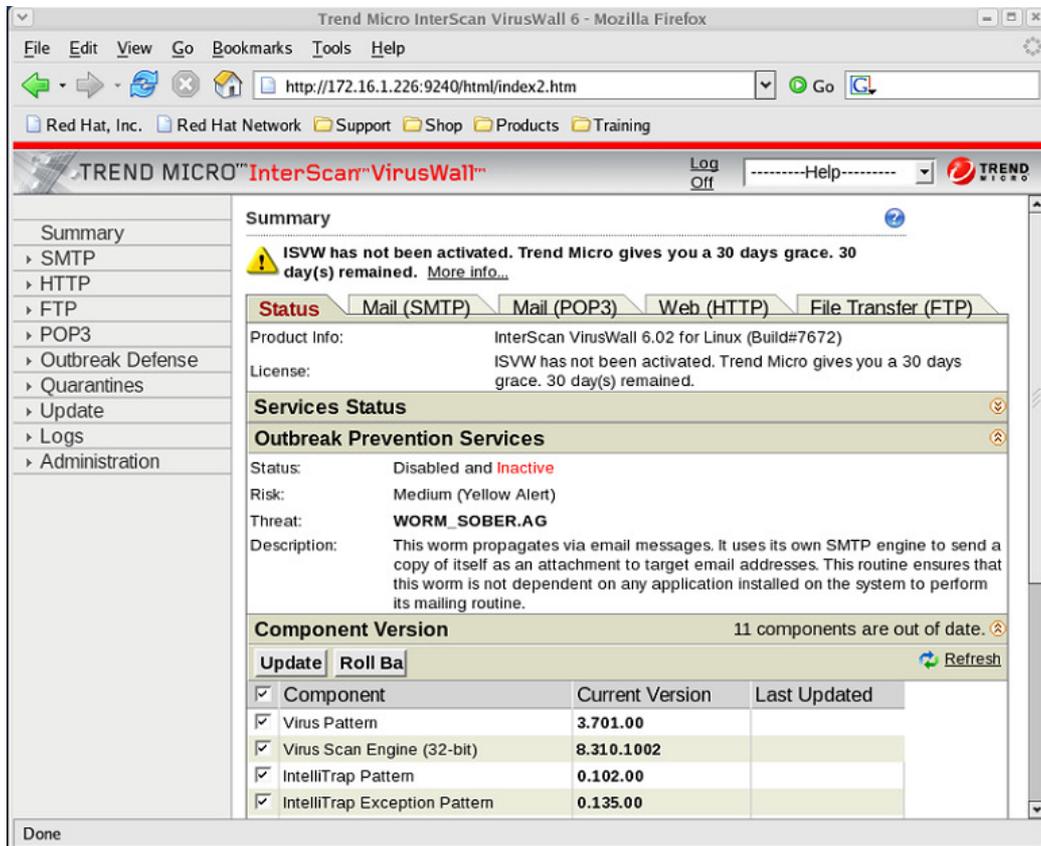


Figure 1: Trend Micro Web Browser Interface

Installing ClamAV

Scalix integrates ClamAV so that any message passing through the Scalix system is automatically scanned for viruses. You need to be comfortable installing applications and dependencies using the command line in order to install ClamAV. Follow the PDF document supplied by ClamAV to install the application.

To install ClamAV on Scalix

- 1 Download and install the ClamAV RPMs. They are readily available on the Internet at clamav.org, clamav.net and the rpmfind Web sites. Once downloaded, uncompress the file and change to the directory, for example

```
tar -xvf clamav-0.92.tar.gz
cd clamav-0.92
```

The general commands to install ClamAV are

```
./configure --disable-clamav
```

where the --disable-clamav extension avoids an error message that is generated for the new clamav on a Red Hat computer (use vscan on a SUSE computer). After you run the configure command successfully, enter

```
make; make install
```

Follow the instructions in the ClamAV PDF document to install it (<http://www.clamav.org/doc/latest/>) including the command to add the group “scalix” and user “clamav” or “vscan” to the group, for example

```
groupadd clamav
useradd -g clamav -s /bin/false -c “Clam Antivirus” clamav
```

Note

On some versions of SUSE, simply adding the user to the group file does not give the user the required group rights. If so, change the group for the vscan user to be the group, scalix. To do this, edit the */etc/passwd* file.

Installing Other Anti-Virus Programs

To install other anti-virus software for use with Scalix, follow the manufacturer installation instructions with one exception: Because the service router calls the virus scanner while running as the Scalix user, you must change the permissions on the virus scanner.

To change permissions on the virus scanner

- 1 On the virus scanner, run the following lines:

For McAfee VirusScan for Linux: `chmod a+rx /usr/local/bin/uvscan`

For Trend Micro InterScan VirusWall: `chmod a+rx /opt/trend/isvw6/IScan.BASE/vscan`
where isvw6 varies with installation

Configuring Anti-Virus Protection

Configuration of all anti-virus applications is required for use with Scalix.

Configuring ClamAV for Use with Scalix

If you are working with ClamAV, follow the documentation provided with ClamAV to configure the software with these exceptions:

- On SUSE, the configuration file is sometimes called */etc/clamav.conf*. This name can cause problems because *clamd* cannot parse the configuration file. Instead, name the file *clamd.conf* and place it in the subfolder */etc*. Then edit the file */etc/init.d/clamd* by modifying the start section so that the *clamd* daemon starts with the added parameter “-c */etc/clamd.conf*”.
- Verify that the “User” parameter in the file *clamd.conf* is set to the same user you added to the scalix group above.
- Configure the freshclam software to keep the known virus database up to date. Follow the ClamAV documentation to configure freshclam to run as either a daemon or via a cron job.

Creating a Virus-Scanning Rule Set

For all anti-virus packages, create a virus-scanning rule set.

To do this, create a text file in the directory */var/opt/scalix/<nn>/s/rules* called ALL-ROUTES.VIR, which controls virus protection on the Scalix server. This file contains a message delivery rule set that applies to all routes. If the mapper script detects a virus, the service router refers to these rule sets. They determine whether the message attachment is discarded.

You can use two attributes in virus scanning rules:

- VIRUS-FOUND: Causes the service router to test each message for the presence of viruses.
- VIRUS-UNCLEANED: Causes the service router to test each message for the presence of viruses, and then if needed, remove the infected attachments.

An example of a rule set is:

```
VIRUS-UNCLEANED=1 ACTION=REJECT NDN-INFO=!ndninfo.txt
VIRUS-UNCLEANED=0 VIRUS-FOUND=1 ACTION=ALLOW NOTIFY="A virus was found in
your message. It was successfully cleaned and sent to the recipient. We
recommend that you install or update your virus protection software and
scan your computer for viruses."
```

where...

The first line describes the action the anti-virus software takes if a virus is detected, but the virus cannot be cleaned. In this example, the message is rejected and a non-delivery notification goes to the sender.

The second line describes what action the anti-virus software takes if a virus is detected and the virus can be cleaned out. In this example the rule allows the message to be delivered to the recipient, and a notification is sent to the originating address.

The variables are outlined in the following procedure.

Alert

Each rule must be on a single line and there cannot be any blank lines.

To create the virus scanning rule set

1 Determine whether you want the service router to:

- Repair and deliver the infected message
- Prevent the delivery of infected messages

This choice determines which virus scanning attribute you use in the rule set.

2 Create a text file containing the virus scanning rules you want to use. Each rule is a single line of text:

```
message-attribute=mvalue action-attribute=avalue action-attribute=avalue
...
```

where *message-attribute* is either VIRUS-FOUND or VIRUS-UNCLEANED and *mvalue* is a numerical value specifying the number of viruses detected/uncleaned. Enter 0 to indicate none, or enter 1 to indicate one or more.

action-attribute and *avalue* can be one of the following:

```
ACTION=ALLOW
```

```
ACTION=DISCARD
```

```
ACTION=REJECT
```

```
ACTION=DEFER
```

```
ACTION=RETURN
```

3 Name the file ALL-ROUTES.VIR (all upper case) and save it as a text file to the directory `/var/opt/scalix/<nn>/s/rules`

4 Restart the service router:

```
omoff -s sr
```

```
omon -s sr
```

Note

After starting the service router, a test is done to ensure that the virus scanner can access a Scalix-owned file. If not, the router aborts. Check the event logs (omshowlog) and use the debug logging configured in the file `~/sys/omvscan.cfg`

Configuring Non-Delivery Notification

You can send a non-delivery notification to the address where the infected file originated, but because most viruses come from spoofed addresses, Scalix does not recommend this.

Copying and Modifying the Scan File

For all anti-virus packages, the next step in anti-virus configuration is to copy and modify the scan file (`omvscan`), which provides the necessary information for the anti-virus software to scan all messages sent to Scalix users, even messages sent from one Scalix user to another.

To configure the scan file

- 1 Enable the script in the scan file by copying the `omvscan.map` file from `/opt/scalix/examples/general` to `/var/opt/scalix/<nn>/s/rules`, where it becomes active. For example

```
cp /opt/scalix/examples/general/omvscan.map /var/opt/scalix/<nn>/s/rules
```

where `nn` varies with Scalix installation.

- 2 Change to the directory using the `cd` command and make sure the file is owned by root and has permissions set to 555. For example

```
cd /var/opt/scalix/<nn>/s/rules
chown root omvscan.map
chmod 555 omvscan.map
```

Setting Up the Mapper Script

Next, set up a mapper script. The `omvscan.map` is the virus scanning mapper script that links Scalix and the virus scanning applications.

The `omvscan.map` is enabled when the service router process begins upon startup. The script remains active (enabled) until the service router is shut down. If you configure auxiliary service router processes, each service router process starts its own instance of `omvscan.map`.

To set up the mapper script

- 1 Modify the `/var/opt/scalix/<nn>/s/rules/omvscan.map` file.

Setting up the Mapper Configuration File

Finally, set up the mapper configuration file. The `omvscan.cfg` configuration file defines the anti-virus application to scan messages and defines the various options to be used. This file is located in the `/var/opt/scalix/<nn>/s/sys` folder.

To set up the mapper configuration file

- 1 Modify the `/var/opt/scalix/<nn>/s/sys/omvscan.cfg` file, changing the permissions beforehand.

Specifying File Types to Exclude

Excluding files from scanning can improve router performance. You can exclude them by file type, such as .txt files.

To exclude file types from virus scans

- 1 Determine the numeric codes to use for each file type by looking in the following file:

```
/var/opt/scalix/<nn>/nls/c/filetype
```

where nn varies with Scalix installation.

- 2 Open the following file:

```
/var/opt/scalix/<nn>/s/sys
```

where nn varies with Scalix installation

- 3 Add the following option:

```
SR_VS_IGNORE_ITEM_TYPES
```

creating a colon-separated list of file codes to exclude from virus scanning. For example, setting this parameter to `1166:1167` excludes Scalix distribution lists and text file (.txt) attachments from scanning.

Testing the Anti-Virus Installation

Test the anti-virus installation.

To test your ClamAV installation and configuration

- 1 Turn up audit logging for the service router.

```
omconfaud router 13
```

- 2 Turn up debug logging for the service router.

```
omconf1v1 router 15
```

- 3 Stop/restart the service router

```
omoff -d 0 rtr
omon rtr
```

- 4 If you download the source tarball from the ClamAV site, attach some of the files provided in the test subfolder of your ClamAV installation. Read the ClamAV PDF document for the commands to use for testing.

- 5 Look in the file `~/logs/audit log`, where you see something like:

```
message-filter-info +VIRUS-UNCLEANED=REJECT
```

- 6 If that does not provide the information you need, check the log `~/logs/fatal` for something like:

```
504 anti-virus engine "ClamAV" exhibits unexpected behavior
```

It is likely the clamd user does not have sufficient permissions to access the `~/data` subfolder. If so, see the information provided earlier to insure that the clamd/vscan user is configured properly.

- 7 Once you have confirmed that ClamAV is working properly, reduce log levels to 7.

```
omconfaud router 7
omconflvl router 7
omoff -d 0 rtr omon rtr
```

To test McAfee VirusScan for Linux

- 1 As outlined in the McAfee installation document, create a text file called EICAR.COM with the following content:

```
X50!P%@AP{4\PZX54(P^)7CC)7)$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

and save the file.

- 2 In a terminal window, enter

```
uvscan -v EICAR.COM
```

The file name is case-sensitive, so if you use eicar.com, the file is not found. When successful, lines in the response indicate scanning, for example:

```
Scanning /root/EICAR.COM
Scanning file /root/EICAR.COM
```

- 3 To scan the root user's home folder, enter

```
uvscan -r --secure /root
```

where r means recursive to scan all folders in the directory and --secure means with maximum security. The response includes

```
Found the Exploit-URLspoofer.gen trojan !!!
```

To view a list of scanning options enter

```
uvscan --help
```

To test Trend Micro InterScan VirusWall

- 1 Send the EICAR.COM file, outlined in the previous procedure, in an e-mail.

Updating the Anti-Virus Installation

The application and anti-virus definitions can have updates available.

To update Trend Micro InterScan VirusWall

- 1 Access the Trend Micro management console in a Web browser. The format is

```
http://172.16.1.226:9240
```

where you insert the IP address for the Scalix server for the one shown.

- 2 Click **Update**, then **Manual**. A list of current and available versions displays.
- 3 Click **Update**. The system updates.

Using the Microsoft Outlook Security Model

The Outlook E-mail Security Model provides protection against viruses that arrive in a user's inbox as attachments. See the *Scalix Client Deployment Guide*.

Improving Performance

Although virus scanning adds a performance overhead, because the architecture keeps the scan script running at all times, the overhead rarely creates performance problems. If you want to improve speed, try the following:

- After installation and testing, turn off or lower logging and auditing levels. See “Setting up the Mapper Configuration File” on page 19.
- If you do detect an impact on performance, a binary version of the mapper script (or one written in Perl) can help.
- To prevent the service router from scanning text-only or distribution list message parts, exclude these file types, and others, from virus scanning. See “Specifying File Types to Exclude” on page 20.

Spam Protection

This chapter covers how to integrate SpamAssassin into your Scalix system. If you do not intend to use spam protection at the server level, skip this chapter.

Contents

This chapter includes the following information:

- “Introduction” on page 23
- “Where to Install SpamAssassin” on page 24
- “System Requirements and Installation” on page 24
- “Configuration” on page 25
- “SMTP Authentication and Spam Protection” on page 28
- “Using a DNS Block List” on page 42

Introduction

There are several approaches to protect against unwanted e-mail, or spam, in the Scalix environment:

- Install any anti-spam package on a server in front of the Scalix system (not documented here)
- Install the free, open-source SpamAssassin package on the Scalix system
- Use the anti-spam features included with a supported anti-virus application. For example, Trend Micro InterScan VirusWall includes them, while the McAfee and ClamAV tools do not.
- Blacklist Internet protocol (IP) addresses known to be the source of spam. Blacklisting can be done with a Domain Name System (DNS) or firewall.

There are five steps to integrate anti-spam protection:

- Acquire an anti-spam product
- Install the product on the Scalix server
- Add one option to the smtpd.cfg file

- Configure Sendmail for use with SpamAssassin
- Configure SpamAssassin to start up on boot

These steps are explained in this chapter.

How SpamAssassin Works

When SpamAssassin is installed, client processes communicate with a daemon (spamd) to check whether messages are spam. In most cases, the client hands the daemon a complete message to check. The daemon then returns the message with a series of header lines which indicate how much spam is in the contents.

For more information about SpamAssassin, see the Apache Foundation Web site at <http://spamassassin.apache.org>

SpamAssassin does not remove spam by default. It tags e-mail as spam and adds headers to the e-mail with a rating of the message.

Where to Install SpamAssassin

The traditional route for outside mail to reach Scalix is through the Simple Mail Transfer Protocol (SMTP) relay. So the best place to integrate SpamAssassin is there. This way, it routes messages through Sendmail first to be filtered. The relay can be in a demilitarized zone (DMZ), for example when you have multiple Scalix servers in a local area network (LAN), then a non-Scalix mail server in the DMZ that acts as a gateway for anti-virus and anti-spam functions. Otherwise, you install anti-spam protection on the Scalix server.

System Requirements and Installation

Ensure the computer has the following applications installed (version numbers can vary):

- perl-spamassassin-3.0.4-1.1 (included with SUSE)
- spamassassin-3.0.4-1.1 (included with SUSE)
- spamass-milter-0.3.0-1 (get from Internet)
- sendmail-8.13.3-5 (included with SUSE)
- sendmail-devel-8.13.3-5 (included with SUSE)

To list all .rpm packages installed, run this command:

```
rpm -qa
```

To determine the version number of a specific application (for example diffutils), use this format:

```
rpm -q diffutils
```

Or check in the software management application, for example click **Applications > Add/Remove Software** in Red Hat 5.

Download and install spamassassin, spamass-milter, sendmail-devel, and other RPMs. These RPMs are readily available on the Internet. To install an application on Red Hat, for example, enter yum install and the package name. For example,

```
yum install cyrus-sasl-md5.i386
```

To view which file/version to install, enter

```
yum search sasl|more
```

where sasl is the name of the program that you are searching for.

Or check in the software management application, for example click **Applications > Add/Remove Software** in Red Hat 5.

Configuration

Configuration is required.

Changing the Configuration File

Configure Scalix to filter mail through Sendmail and then SpamAssassin. This involves adding one option to the smtpd.cfg configuration file.

To configure the smtpd.cfg file for SpamAssassin

- 1 Make a copy of the current configuration file:

```
cp /var/opt/scalix/<nn>/s/sys/smtpd.cfg /var/opt/scalix/<nn>/s/sys/
smtpd.cfg.orig
```

- 2 Open the smtpd.cfg file. (This example uses vi.)

```
vi /var/opt/scalix/<nn>/s/sys/smtpd.cfg
```

- 3 Add the following line:

```
SMTPFILTER=TRUE
```

above the line

```
RELAY accept 127.0.0.1
```

- 4 Save the file.

Configuring SendMail

In addition, there are several Sendmail configurations required.

To configure SendMail for use with SpamAssassin

- 1 Back up the sendmail file:

```
cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.orig
```

- 2 Verify that the sendmail.cf file has the following change.

Change OperatorChars from

```
o operatorChars=.:%#!^/[ ]+=
```

to

```
o operatorChars=.:%#!^/[ ]+
```

- Uncomment the following line:

```
#0 InputMailFilters
```

and change it to:

```
0 InputMailFilters=Spamassassin
```

- Immediately below that line, add the following:

```
#Milter options
```

```
#0 Milter.LogLevel
```

```
0 Milter.macros.connect=b, j, _, {daemon_name}, {if_name}, {if_addr}
```

```
0 Milter.macros.helo={tls_version}, {cipher}, {cipher_bits},
{cert_subject}, {cert_issuer}
```

```
0 Milter.macros.envfrom=i, {auth_type}, {auth_authen}, {auth_ssf},
{auth_author}, {mail_mailer}, {mail_host}, {mail_addr}
```

```
0 Milter.macros.envrcpt={rcpt_mailer}, {rcpt_host}, {rcpt_addr}
```

- In the section MAIL FILTER DEFINITIONS, add the following line:

```
Xspamassassin, S=local:/var/run/spamass-milter/spamass-milter.sock,
F=, T=C:15m;S:4m;R:4m;E:10m
```

Adding SpamAssassin to Startup on Boot

You must add spamd to start upon boot and then restart all services.

To add SpamAssassin to startup on boot

- Run the following commands:

```
chkconfig --add spamass-milter
chkconfig --level 345 spamass-milter on
service spamass-milter start
chkconfig --add spamassassin
chkconfig --level 345 spamassassin on
service spamassassin start
```

- Insure all services are now set up to start on bootup:

```
chkconfig --list|grep 'spamassassin\|spamass-milter'
```

- You see output that looks like:

```
spamass-milter 0:off 1:off 2:off 3:off 4:off 5:on 6:off
spamassassin 0:off 1:off 2:off 3:off 4:off 5:on 6:off
```

See the chkconfig MAN page to understand changing run levels.

- Start SpamAssassin:

```
service spamassassin restart
```

- Start spamass-milter:

```
service spamass-milter restart
```

6 Restart Sendmail:

```
service sendmail restart
```

7 Restart the Scalix SMTP Relay:

```
omoff -d 0 smtpd #omon smtpd 6.
```

Confirming SpamAssassin is Working

Before moving on, test that SpamAssassin is working properly.

To confirm that SpamAssassin works

1 Run the following command:

```
tail -f /var/log/maillog
```

2 Successful Spamassassin configuration produces the following type of output in the log file if it is working correctly:

```
scalix.local@MHS>, proto=ESMTP, relay=root@localhost
Nov  3 09:39:56 sca14 sendmail[27547]: jA3Hdueo027547:
from=<Kent.Brake@scalix.com>, size=2089, class=0, nrcpts=1,
msgid=<H00000b60014d0c8.1131039536.hagrid.scalix.local@MHS>,
proto=ESMTP, daemon=MTA, relay=localhost [127.0.0.1]
Nov  3 09:39:56 sca14 spamd[24498]: connection from localhost
[127.0.0.1] at port 59807
Nov  3 09:39:56 sca14 spamd[24498]: info: setuid to root succeeded
Nov  3 09:39:56 sca14 spamd[24498]: Still running as root: user not
specified with -u, not found, or set to root. Fall back to nobody.
Nov  3 09:39:56 sca14 spamd[24498]: processing message
<H00000b60014d0c8.1131039536.hagrid.scalix.local@MHS> for root:65534.
Nov  3 09:39:56 sca14 spamd[24498]: clean message (-1.0/5.0) for
root:65534 in 0.1 seconds, 2338 bytes.
Nov  3 09:39:56 sca14 spamd[24498]: result: . -1 -
ALL_TRUSTED,WEIRD_QUOTING
scantime=0.1,size=2338,mid=<H00000b60014d0c8.1131039536.hagrid.scalix
.local@MHS>,autolearn=failed
```

SMTP Authentication and Spam Protection

Scalix supports SMTP authentication to allow accurate identification of the users of the SMTP service. In addition, Scalix allows you to configure anti-spamming measures to prevent abuse of the Scalix system.

Both of these security measures are implemented as part of the SMTP relay (omsmtpd), and are configured by adding entries to the SMTP relay configuration file:

```
/var/opt/scalix/<nn>/s/sys/smtpd.cfg
```

To configure how the SMTP relay manages incoming connections, you must specify an action that the SMTP relay performs in response to an event for each address or address pattern.

When an event occurs, the SMTP relay checks the relevant entries in the configuration file for matching event/pattern entries. The check is done sequentially, from top to bottom. When it finds the first match, the SMTP relay takes the action specified. If the SMTP relay does not find a match, it processes the message normally.

The default configuration file included with Scalix causes the SMTP relay to accept all relay attempts from hosts in the local domain, and reject all unauthenticated relay attempts from outside the local domain.

The following table lists possible values for the options in the SMTP relay configuration file.

Table 1: Options and Values in SMTP Relay Configurations

Event	Description
SUBMIT	An attempt is made to submit a message from the host specified in <i>pattern</i> . If the <i>action</i> specified is <code>Header</code> , the wording of the header inserted is: X-Scalix-Suspicious-Host: <i>hostname-or-IP-address</i>
ANONYMOUS	An attempt is made to submit a message sent without authentication or after a failed authentication. If the <i>action</i> specified is <code>Header</code> , the wording of the header inserted is: X-Scalix-Anonymous-Message: from <i>hostname at date</i>
AUTH_SUCCESS	An attempt is made to submit a successfully authenticated message. Normally only used with the <code>Accept</code> and <code>Header</code> actions. If the <i>action</i> specified is <code>Header</code> , the wording of the header inserted is: X-Scalix-Authenticated-Sender: <i>email-address; authenticated by hostname at date</i>
AUTH_MISMATCH	An attempt is made to submit a message which was successfully authenticated, but the originator name does not match the authenticated user name. If the <i>action</i> specified is <code>Header</code> , the wording of the header inserted is: X-Scalix-Authentication-Mismatch: Message originated from <i>email-address</i> , but authenticated identity is <i>email-address</i>
RELAY	An attempt is made to relay a message through the SMTP Relay. Normally, you would specify all local hosts in the associated <i>pattern</i> , so that they can all send messages to any external host, and any external host can send messages to the local hosts. A relay attempt from the host on which the SMTP Gateway is running is always accepted. The SMTP Relay always inserts a standard <code>Received:</code> header in the message, so a <code>Header</code> <i>action</i> is not required.

Table 1: Options and Values in SMTP Relay Configurations

Event	Description
ORIGINATOR	An attempt is made to submit a message from a user whose e-mail address matches the <i>pattern</i> specified. Use this event to block mail from known sources of spam. If the <i>action</i> specified is <code>Header</code> , the wording of the header inserted is: X-Scalix-Suspicious-Originator: <i>email-address</i>
RECIPIENT	An attempt is made to submit a message to a user whose e-mail address matches the <i>pattern</i> specified. Use this event to block mail to nonexistent addresses. If the <i>action</i> specified is <code>Header</code> , it will be ignored.
SMTPFILTER	SMTPFILTER has only one state and that's TRUE. So, if you add the following line: SMTPFILTER=TRUE to the <code>~/sys/smtpd.cfg</code> file. It causes the SMTP Relay to hand the incoming message off to Sendmail. As a result, inbound messages will be processed through any installed sendmail filters (mail filters), such as SpamAssassin.
Action	Description
Accept	The message is unconditionally accepted and processed normally.
Defer	The message is deferred, with a 4XX code. Eventually, the sending host will cease transmitting it, and will reject it.
Discard	The message is accepted but then discarded, with no indication to the sending host that it was not delivered.
Header	The message is accepted, but an extra header is inserted. The wording of the header depends on the <i>event</i> .
Reject	The message is rejected, with a 5XX code.
Pattern	Description
Hostname-pattern	The hostname pattern identifies the originating host (or the destination host in the case of the SMTP relay event). It is used for all events except <code>ORIGINATOR</code> and <code>RECIPIENT</code> . Possible values are described in the <code>Hostname Pattern</code> section.
Email-address-pattern	The e-mail address pattern identifies the source or destination e-mail address, used only for the <code>ORIGINATOR</code> and <code>RECIPIENT</code> events. For example: * <code>@*.spam.net</code> matches <code>iama.spammer@lotsof.spam.net</code> . * matches all e-mail addresses.

Note that all actions can be prefixed by the string `Log_` to cause the action to be recorded in the Scalix log file. In addition, SMTP relay information is logged in the Audit Log. See the Audit Log configuration file for more information.

The following information shows the `smtpd.cfg` configuration file. You configure SMTP Authentication and Anti-spam protection in the second part of the `smtpd.cfg` file (format: `event action pattern pattern ...`).

```
#####
# SMTP Relay Configuration
# #####
#
```

```

# For details please see Scalix Overview - Security
#
#####

#####

# Relay Configuration
# #####
#
...

# Authentication and Anti-Spamming Measures
# #####
#
# Each line is of the form:
# EVENT ACTION PATTERN PATTERN...
# When an event happens the SMTP Relay checks for a matching event/pattern
# sequentially in this file. When it finds the first match, it takes the
# action specified.
#
# #####
# EVENTS
# #####
#

# AUTH_SUCCESS      An attempt is made to submit a successfully
#                   authenticated message
#
# AUTH_MISMATCH     An attempt is made to submit a successfully
#                   authenticated message but the originator name does not
#                   match the authenticated name
#
# ANONYMOUS         An attempt is made to submit a message sent without
#                   authentication or after failed authentication
#
# SUBMIT            An attempt is made to submit a message from the host
#                   specified in pattern
#
# RELAY             An attempt is made to relay a message through the SMTP
#                   Relay
#
# ORIGINATOR        An attempt is made to submit a message from a user
#                   whose email address matches pattern
#
# RECIPIENT         An attempt is made to submit a message to a user whose
#                   email address matches pattern
#
#
# #####
# ACTIONS
# #####
#

# Accept            The message is unconditionally accepted and processed
#                   normally
#
# Defer             The message is deferred with a 400 code

```

```

# Discard          The message is accepted but then discarded
# Header          The message is accepted, but an extra header is
                  inserted
# Reject          The message is rejected with a 500 code

# If Log_ added to the start of an action, then action is also recorded
# in the SMTP Relay log file.
#
# #####
# PATTERNS
# #####
#
# Hostname Patterns
# - an IP address, eg 123.234.132.231
# - an IP subnet and mask, eg 123.234.200.0/255.255.240.0
# - a hostname, eg bert.loc.co.uk
# - the end of a domain, eg .spammer.net
# - the start of a domain, 123.234.
# - the keyword ALL matches all hosts
# - the keyword LOCAL matches all hosts that do not contain a .
#
# Email Patterns - used by ORIGINATOR and RECIPIENT
# - *@*.spam.net
#
...
# NB Authenticated RELAYS are always allowed
RELAY accept 127.0.0.1
RELAY accept .bert.loc.co.uk
RELAY Log_Reject ALL
#
# extra rules to prevent open relay usage
RECIPIENT Log_Reject *@**@*
RECIPIENT Log_Reject %**
RECIPIENT Log_Reject *!*
RECIPIENT Log_Reject **#@*

```

There can be several configuration entries for the same event, but only one applies to any particular message. For any event, the SMTP relay scans all configuration entries (from top to bottom) and looks for the first match. Any other configuration entries for this event are ignored.

Note the following:

- AUTH_MISMATCH – This event is an attempt to submit a message that was successfully authenticated but the originator name (FROM: in the RFC 822 header) did not match the authenticated user name.
- Header – In this action, an extra header is inserted into the message. The header name and the value are fixed and depend on the event type.
- Defer – In this action, an SMTP 400 code is returned to the sending server. This means that the message remains on the sending server and is repeatedly retried until it times

out and is rejected by the submitting server. This means that the spam message occupies disk space on the sending host, but might cause problems for uninvolved third parties.

- **Reject** – In this action, the message is rejected and an SMTP 500 code is returned to the sending server. For most positively identified SPAM originators and recipients, this is the preferred action because it requires little processing power.
- If debug logging is enabled and any of the action keywords is prefixed with `Log_`, this action will also be recorded in the SMTP relay log file, `~/tmp/smtpd.log`. You enable debug logging by adding the line `debug_log=true` to `smtpd.cfg`.
- Hostname patterns should be used for the ANONYMOUS, AUTH_SUCCESS, AUTH_MISMATCH, RELAY, and SUBMIT events.
- If a hostname cannot be looked up in the DNS, it will not match a domain name pattern or an explicit hostname.
- A subnet and mask separated by a / (for example, `15.145.200.0/255.255.240.00`) will match all IP addresses in the `15.145.200.0` to `15.145.207.255` range. Note that the mask need not correspond to a “real” subnet.
- A string that begins with an @ character is treated as an NIS (YP) netgroup name. A hostname is matched if it is a host member of the specified netgroup.
- A string that begins with a / character is treated as a file name. A hostname or address is matched if it matches any hostname listed in the named file. The file format is zero or more lines with zero or more hostname patterns separated by white space.
- E-mail address patterns should be used for ORIGINATOR and RECIPIENT events.

How To Prevent Message Spoofing From Internal Hosts

The following code shows an example of a person in the intranet using SMTP commands to send a message to the server and appear to be a user they are not. For instance, you can telnet to a Scalix Server named `scalix1`, simulate being the user `bob`, and send a message to `tom`:

```
$telnet scalix1 25
Trying...
Connected to scalix1.pwd.scalix.com.
Escape character is '^]'.
220 scalix1.pwd.scalix.com Hello from the Scalix Lab
ehlo scalix1
250-scalix1.pwd.scalix.com Hello scalix1.pwd.scalix.com [15.145.204.43],
pleased to meet
you
250-AUTH LOGIN PLAIN DIGEST-MD5 ANONYMOUS CRAM-MD5
250-AUTH=LOGIN
250-DSN
250 8BITMIME
mail from: bob@scalix.com
250 bob@scalix.com... Sender ok
rcpt to: tom@scalix1.pwd.scalix.com
```

```
250 ok
data

354 Enter mail, end with "." on a line by itself (relay)
subject: an important meeting
Please attend the meeting taking place in the boardroom at 9am tomorrow
.
250 ok
```

The message received by tom appears to be legitimate:

```
Return-Path: <bob@scalix.com>
Received: from scalix1 (scalix1.pwd.scalix.com 15.145.204.43)
by scalix1.pwd.scalix.com via ESMTP; Tue, 17 Apr 2001 14:21:37 +0100 (BST)
Date: Tue, 17 Apr 2001 14:21:40 +0100
From: bob@scalix.com
Sender: bob@scalix.com
Message-ID: <1642.987513700.scalix1.pwd.scalix.com@MHS>
Subject: an important meeting
MIME-Version: 1.0
Content-Type: text/plain; charset="US-ASCII"
Content-Transfer-Encoding: quoted-printable
Content-Disposition: inline
Please attend the meeting taking place in the boardroom at 9am tomorrow
```

To prevent such messages from being accepted by the SMTP relay, you can configure an ANONYMOUS event which instructs the SMTP Relay what to do when an attempt is made to submit a message without authentication (or which failed authentication).

In the current example, you want to reject all anonymous connections from clients in your domain. However, we have a server set up to relay messages, scalixopp.pwd.scalix.com, and it will try to connect anonymously, so you need to allow for this in your configuration:

```
ANONYMOUS Header scalixopp.pwd.scalix.com
ANONYMOUS Reject .pwd.scalix.com
ANONYMOUS Accept ALL
```

Note that the example asks the SMTP Relay to insert a header in anonymous messages which it relays to users from scalixopp. This header takes the following form (where sender is the message sender and addr is the IP address of the sending host):

```
X-Scalix-Anonymous-Message: from sender at addr
```

(In the current example, the addr will be 15.145.205.23 which is the IP address of scalixopp.pwd.scalix.com.)

Alternatively, you can set up your relaying servers to be within a certain IP range and specify the range using the IP subnet and mask.

When you try using a telnet session to make the message appear to be from "bob@scalix.com", the second configuration line is executed and you are asked to authenticate.

```
$telnet scalix1 25
Trying...
Connected to scalix1.pwd.scalix.com.

Escape character is '^]'.
220 scalix1.pwd.scalix.com Hello from the Scalix Lab
```

```
ehlo scalix1
250-scalix1.pwd.scalix.com Hello scalix1.pwd.scalix.com [15.145.204.43],
pleased to meet you
250-AUTH LOGIN PLAIN DIGEST-MD5 ANONYMOUS CRAM-MD5
250-AUTH=LOGIN
250-DSN
250 8BITMIME
mail from: bob@scalix.com
530 Authentication required
...
```

When you send a message tom on scalix2.pwd.scalix.com, it gets relayed via scalixopp, so the SMTP relay executes the first configuration line and you receive the message with an extra header, showing that this message came from scalixopp (15.145.205.23) and is unauthenticated:

```
Return-Path: <tom@scalix2.pwd.scalix.com>
Received: from scalixopp.pwd.scalix.com (scalixopp.pwd.scalix.com
15.145.205.23)
by scalix1.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTTP; wed, 18 Apr 2001 16:31:04 +0100 (BST)
Received: from scalixopp.pwd.scalix.com (root@localhost)
by scalixopp.pwd.scalix.com (8.9.3 (PHNE_18546)/8.9.3 SMKit7.01 Scalix)
with ESMTTP id QAA19330 for <tom@scalix1.pwd.scalix.com>; wed, 18 Apr 2001
16:31:03 +0100 (BST)
Received: from scalix2.pwd.scalix.com (scalix2.pwd.scalix.com
15.145.204.249)
by scalixopp.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTTP; wed, 18 Apr 2001 16:31:04 +0100 (BST)
Received: from scalix2.pwd.scalix.com (root@localhost)
by scalix2.pwd.scalix.com (8.8.6 (PHNE_17190)/8.8.6) with ESMTTP id
QAA03025
for <tom@scalix1>; wed, 18 Apr 2001 16:31:03 +0100 (BST)
Received: from joyford3 (scalixpwd186.pwd.scalix.com 15.145.205.60)
by scalix2.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTTP; wed, 18 Apr 2001 16:31:03 +0100 (BST)
Date: wed, 18 Apr 2001 16:31:02 +0100
From: "tom" <tom@scalix2.pwd.scalix.com>
Sender: "tom" <tom@scalix2.pwd.scalix.com>
To: tom@scalix1.pwd.scalix.com
Message-ID: <001401c0c81c$938a7ca0$62cc910f@pwd.scalix.com>
Subject: some headers
X-MSMail-Priority: Normal
X-Priority: 3
X-Scalix-Anonymous-Message: from <tom@scalix2.pwd.scalix.com> at
15.145.205.23
X-Mailer: Microsoft Outlook Express 5.50.4133.2400
X-MimeOLE: Produced By Microsoft MimeOLE V5.50.4133.2400
MIME-Version: 1.0
Content-Type: multipart/alternative;
```

```
boundary="-----_NextPart_000_0011_01C0C824.F520F6D0"
```

```
...
```

If you have several hosts set up to relay messages around your intranet, you will need to include them in the ANONYMOUS line to allow them to connect without authenticating.

Remember that for each connection, the SMTP relay reads down through the configuration file from the top and execute the first line that matches. So the selective line:

```
ANONYMOUS Header scalixopp.pwd.scalix.com
```

must come before the more global line:

```
ANONYMOUS Reject .pwd.scalix.com
```

Verifying the Identity of a Sender

You might want the SMTP relay to accept all successfully authenticated messages, but for tracking purposes, you want a header added to messages from your domain (pwd.scalix.com), to confirm the identity of the authenticated sender and the address of the sending client/host.

The event, in this case, is AUTH_SUCCESS, the action is Header and the pattern is .pwd.scalix.com. Therefore, you can add following lines to the configuration file:

```
AUTH_SUCCESS Header .pwd.scalix.com
AUTH_SUCCESS Accept ALL
```

This instructs the SMTP Relay to add an extra header (X-Scalix-Authenticated-Sender:) to authenticated messages from any host ending in .pwd.scalix.com and to accept authenticated messages from any other hosts (LOCAL or not in the pwd.scalix.com domain).

For example, you can add the above code to smtpd.cfg on the server, scalix1, and use Outlook on a separate PC (IP Address = 15.145.205.60) to send a message from kelly on scalix1 to Fred on scalix1. Here is the message received by Fred, showing the added header with kelly's address and the IP address of the connecting machine:

```
Return-Path: <kelly@scalix1.pwd.scalix.com>
Received: from scalix1.pwd.scalix.com (root@localhost)
by scalix1.pwd.scalix.com (8.8.6 (PHNE_17190)/8.8.6) with ESMTMP id
LAA03978
for <fred@scalix1>; wed, 18 Apr 2001 11:21:43 +0100 (BST)
Received: from joyford3 (scalixpwd186.pwd.scalix.com 15.145.205.60)
by scalix1.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTMP; wed, 18 Apr 2001 11:21:43 +0100 (BST)
Date: wed, 18 Apr 2001 11:21:41 +0100
From: kelly@scalix1.pwd.scalix.com
Sender: kelly@scalix1.pwd.scalix.com
To: fred@scalix1.pwd.scalix.com
Message-ID: <002f01c0c7f1$5cb78270$62cc910f@pwd.scalix.com>
Subject: hi
X-MSMail-Priority: Normal
X-Priority: 3
X-Scalix-Authenticated-Sender: <kelly@scalix1.pwd.scalix.com> at
15.145.205.60
X-Mailer: Microsoft Outlook Express 5.50.4133.2400
X-MimeOLE: Produced By Microsoft MimeOLE V5.50.4133.2400
```

```
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary="-----=_NextPart_000_002C_01C0C7F9.BE468290"
...
```

AUTH_SUCCESS should only be used with the Header action to insert the additional header verifying the user's identity.

Handling Messages With Sender/Authenticated User Mismatch

We saw how a spoofer might try to send an unauthenticated message. The following information describes how to prevent the spoofer from connecting to scalix1, authenticating successfully, and sending a fake message to kelly.

To prevent this, use the AUTH_MISMATCH event. It describes an attempt to submit a message that was successfully authenticated, but the originator name (FROM:) did not match the authenticated user name. Such messages should be rejected.

For example, you can add the following lines to the smtpd.cfg file:

```
AUTH_MISMATCH Reject LOCAL
AUTH_MISMATCH Header scalix2.pwd.scalix.com
AUTH_MISMATCH Reject ALL
```

The first line causes all authenticated messages from the local host (no . in the address) to be rejected (with a SMTP 500 response), if the sender does not match the authenticated user.

The second line causes a header to be added to any similarly deficient message from the host, scalix2.pwd.scalix.com. The message is not rejected.

The third line causes any other message with this defect to be rejected.

When you telnet to scalix1, authenticate as tom, and try to send a message as bob:

```
$telnet scalix1 25
Trying...
Connected to scalix1.pwd.scalix.com.
Escape character is '^]'.
220 scalix1.pwd.scalix.com Hello from the Scalix Lab
Connection closed by foreign host.
root@scalix1[] telnet scalix1 25
Trying...
Connected to scalix1.pwd.scalix.com.
Escape character is '^]'.
220 scalix1.pwd.scalix.com Hello from the Scalix Lab
ehlo scalix1
250-scalix1.pwd.scalix.com Hello scalix1.pwd.scalix.com [15.145.204.43],
pleased to meet you
250-AUTH LOGIN PLAIN DIGEST-MD5 ANONYMOUS CRAM-MD5
250-AUTH=LOGIN
250-DSN
250 8BITMIME
auth login
334 VXNlcm5hbWU6
bXIgYWRtaw4=
```

```
334 UGFzc3dvcmQ6
YWRtaW4=
235 Authentication successful
mail from: bob@scalix.com
530 Authentication mismatch
```

Scalix found the mismatch and the connection rejected with a SMTP 530 error code.

When you try sending the message by performing a telnet from scalix2 to scalix1, the following occurs:

```
$telnet scalix1 25
Trying...
Connected to scalix1.pwd.scalix.com.
Escape character is '^]'.
220 scalix1.pwd.scalix.com Hello from the Scalix Lab
ehlo scalix2
250-scalix1.pwd.scalix.com Hello scalix2.pwd.scalix.com [15.145.204.249],
pleased to meet you
250-AUTH LOGIN PLAIN DIGEST-MD5 ANONYMOUS CRAM-MD5
250-AUTH=LOGIN
250-DSN
250 8BITMIME
auth login
334 vXNlcm5hbWU6
bXIgYWRtaW4=
334 UGFzc3dvcmQ6
YWRtaW4=
235 Authentication successful
mail from: bob@scalix.com
250 bob@scalix.com... Sender ok
rcpt to: kelly@scalix1
250 ok
data
354 Enter mail, end with "." on a line by itself (relay)
subject: an important meeting
Please attend the boardroom meeting tomorrow, 9am
bob
.
250 ok
```

The message is accepted, but there is a header highlighting the discrepancy:

```
Return-Path: <bob@scalix.com>
Received: from scalix2 (scalix2.pwd.scalix.com 15.145.204.249)
by scalix1.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTTP; wed, 18 Apr 2001 17:51:59 +0100 (BST)
Date: wed, 18 Apr 2001 17:51:59 +0100
From: bob@scalix.com
Sender: bob@scalix.com
Message-ID: <4752.987612719.scalix1.pwd.scalix.com@MHS>
Subject: an important meeting
X-Scalix-Authentication-Mismatch: originator bob@scalix.com
```

```

authenticated as tom at 15.145.204.249
MIME-Version: 1.0
Content-Type: text/plain;
charset="US-ASCII"
Content-Transfer-Encoding: quoted-printable
Content-Disposition: inline
Please attend the boardroom meeting tomorrow, 9am
bob

```

The IP address is that of scalix2.

Restricting Who Can Use This Server To Relay Mail

It is important to prevent outside hosts using your server to relay large quantities of unwanted mail to other internal and external hosts. You can use configuration entries for RELAY events to tell the SMTP Relay what to do when a host, which matches the pattern, attempts to send a message through the SMTP relay to a Sendmail recipient on another server.

Normally, all local hosts are included so that they are allowed to send messages to any external host. You can also list here any external hosts that are allowed to use this server to relay.

For example, note the two entries in the default file. These are automatically inserted by Scalix, including the domain of this host:

```

RELAY Accept domain
RELAY Reject ALL

```

This default setup means that any attempt to use this server to relay a message will only be successful if the sending server is in the same domain as this server. Because the SMTP relay always inserts a standard Received: header into the message, the Header action does not make sense for a RELAY event and will be ignored.

Note that SMTP relay cannot block relay attempts through the Internet gateway (the message goes into Scalix and is relayed using Scalix routes).

Blocking Mail From Certain Hosts

Configuration entries for SUBMIT events describe what the SMTP relay does when a host matching one of the given patterns attempts to submit a message. You can use lines like the following examples to block/log message submission from hosts known or suspected of sending large amounts of unwanted mail (spam):

```

SUBMIT Reject known.spammer.net
SUBMIT Log_Reject another.spammer.net
SUBMIT Header possible.spammer.net
SUBMIT Accept ALL

```

The SMTP relay looks at the address in the MAIL FROM: line of the message and looks through the SUBMIT lines in the configuration file to see if the address matches any of those specified. If the address is known.spammer.net, an SMTP 500 code is returned and the message is not accepted.

If the address is another.spammer.net, an SMTP 500 code is returned, the message is not accepted and, if debug logging is enabled, this action is logged in the ~/tmp/smtpd.log file.

If the address is possible.spammer.net, the message will be accepted but the following header will be added to the RFC 822 message header to indicate that the message was from a suspect host:

```
X-Scalix-Suspicious-Host: IP address
```

If the message is submitted by any host other than those identified in the lines above, it is accepted.

For example, you determine that a host (scalixopp) might be sending unwanted mail, and you want the SMTP relay to add a header to any message from that host instructing the recipient that the sender is not trusted. Also, you are certain that scalix2 is sending spam and you want to reject connections from scalix2 and have the rejection logged in the SMTP relay's log file.

Add the following lines to scalix1's relay configuration file:

```
SUBMIT Header scalixopp.pwd.scalix.com
SUBMIT Log_Reject scalix2.pwd.scalix.com
```

However, you know that messages from scalix2 are relayed to scalix1 by scalixopp. The following code shows what happens when you send a message from tom on scalix2 to Fred on scalix1:

```
Return-Path: <tom@scalix2.pwd.scalix.com>
Received: from scalixopp.pwd.scalix.com (scalixopp.pwd.scalix.com
15.145.205.23)
by scalix1.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTTP; Thu, 19 Apr 2001 09:27:29 +0100 (BST)
Received: from scalixopp.pwd.scalix.com (root@localhost)
by scalixopp.pwd.scalix.com (8.9.3 (PHNE_18546)/8.9.3 SMKit7.01 Scalix)
with
ESMTTP id JAA20532
for <fred@scalix1.pwd.scalix.com>; Thu, 19 Apr 2001 09:27:28 +0100 (BST)
Received: from scalix2.pwd.scalix.com (scalix2.pwd.scalix.com
15.145.204.249)
by scalixopp.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTTP; Thu, 19 Apr 2001 09:27:28 +0100 (BST)
Received: from scalix2.pwd.scalix.com (root@localhost)
by scalix2.pwd.scalix.com (8.8.6 (PHNE_17190)/8.8.6) with ESMTTP id
JAA03131
for <fred@scalix1>; Thu, 19 Apr 2001 09:27:26 +0100 (BST)
Received: from joyford3 (scalixpwd186.pwd.scalix.com 15.145.205.60)
by scalix2.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTTP; Thu, 19 Apr 2001 09:27:27 +0100 (BST)
Date: Thu, 19 Apr 2001 09:27:26 +0100
From: "tom" <tom@scalix2.pwd.scalix.com>
Sender: "tom" <tom@scalix2.pwd.scalix.com>
To: fred@scalix1.pwd.scalix.com
Message-ID: <001401c0c8aa$90ca33a0$62cc910f@pwd.scalix.com>
Subject: a message
X-MSMail-Priority: Normal

X-Priority: 3
X-Scalix-Suspicious-Host: 15.145.205.23
```

```
X-Mailer: Microsoft Outlook Express 5.50.4133.2400
X-MimeOLE: Produced By Microsoft MimeOLE V5.50.4133.2400
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary="-----_NextPart_000_0011_01C0C8B2.F2455DA0"
...
```

A header is inserted to show that this message was sent (or relayed) to Fred by scalixopp (15.145.205.23), one of the suspect hosts.

If you try sending a message directly from scalix2 to scalix1:

```
$telnet scalix1 25
Trying...
Connected to scalix1.pwd.scalix.com.
Escape character is '^]'.
550 Denied
Connection closed by foreign host.
```

The connection is refused immediately. Note the rejection logged in the smtpd.log file on scalix1.

```
Rejected connection from 15.145.204.249
```

where 15.145.204.249 is the IP address of scalix2.

Blocking Mail From Specific Senders

The ORIGINATOR event describes an attempt to send a message from a user whose e-mail address matches a pattern. We can use this event to block mail coming from known spammers.

Here are some sample lines:

```
ORIGINATOR Log_Reject spam@advert.com
ORIGINATOR Discard spam@blast.net
ORIGINATOR Defer spam*@*
ORIGINATOR Accept ALL
```

The first line rejects any message from spam@advert.com and logs the rejection in smtpd.log. The sending host receives a SMTP 500 response. The second line accepts messages from spam@blast.net but immediately discards them.

The third line defers the delivery of any messages from addresses matching spam*@*. The sending hosts receives an SMTP 400 response and the messages is stored on the sending host. The submission is attempted until the sending host rejects the messages.

If you set the header action, the inserted header takes the following form:

```
x-scalix-suspicious-originator: email address
```

To flag any messages as suspicious from users with the scalix2.pwd.scalix.com domain in their address, add the following line to smtpd.cfg on scalix1:

```
ORIGINATOR Header *@scalix2.pwd.scalix.com
```

When you use an Internet client to send a message from tom on scalix2 to Fred on scalix1, the message has the extra header underlined below:

Return-Path: <tom@scalix2.pwd.scalix.com>
Received: from scalixopp.pwd.scalix.com (scalixopp.pwd.scalix.com
15.145.205.23)
by scalix1.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTTP; Thu, 19 Apr 2001 14:30:16 +0100 (BST)
Received: from scalixopp.pwd.scalix.com (root@localhost)
by scalixopp.pwd.scalix.com (8.9.3 (PHNE_18546)/8.9.3 SMKit7.01 scalix)
with
ESMTTP id OAA18566
for <tom@scalix1.pwd.scalix.com>; Thu, 19 Apr 2001 14:30:15 +0100 (BST)
Received: from scalix2.pwd.scalix.com (scalix2.pwd.scalix.com
15.145.204.249)
by scalixopp.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTTP; Thu, 19 Apr 2001 14:30:15 +0100 (BST)
Received: from scalix2.pwd.scalix.com (root@localhost)
by scalix2.pwd.scalix.com (8.8.6 (PHNE_17190)/8.8.6) with ESMTTP id
OAA03265
for <tom@scalix1>; Thu, 19 Apr 2001 14:30:14 +0100 (BST)
Received: from joyford3 (scalixpwd186.pwd.scalix.com 15.145.205.60)
by scalix2.pwd.scalix.com (Scalix SMTP Relay B.07.00.00)
via ESMTTP; Thu, 19 Apr 2001 14:30:15 +0100 (BST)
Date: Thu, 19 Apr 2001 14:30:14 +0100
From: "tom" <tom@scalix2.pwd.scalix.com>
Sender: "tom" <tom@scalix2.pwd.scalix.com>
To: tom@scalix1.pwd.scalix.com
Message-ID: <001401c0c8d4\$ddc0b5b0\$62cc910f@pwd.scalix.com>
Subject: see the headers
X-MSMail-Priority: Normal
X-Priority: 3
X-Scalix-Suspicious-Originator: <tom@scalix2.pwd.scalix.com> at
X-Mailer: Microsoft Outlook Express 5.50.4133.2400
X-MimeOLE: Produced By Microsoft MimeOLE V5.50.4133.2400
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary="-----_NextPart_000_0011_01C0C8DD.3F55A940"

Blocking Mail For Specific Recipients

Use the RECIPIENT event to manage messages sent to addresses that do not exist. This option is useful if an individual is no longer part of an organization and you want to stop mail from being delivered to that account without actually removing (deleting) the account.

You can configure RECIPIENT events as follows:

```
RECIPIENT Log_Reject Unknown.User@pwd.scalix.com
```

```
RECIPIENT Reject User.HasLeftTheCompany@pwd.scalix.com
```

```
RECIPIENT Accept ALL
```

The first line rejects incoming messages for the user Unknown.User@pwd.scalix.com and logs the rejection in smtpd.log.

The second line rejects mail for User.HasLeftTheCompany@pwd.scalix.com.

Using a DNS Block List

If needed, you can create a DNS block list or “blacklist” (DNSBL) in which you identify a list of IP addresses to be avoided. This can be useful as a means to block known spammers.

See also the previous subsections.

To create a DNS block list

- 1 Open the smtpd.cfg file at:

```
/var/opt/scalix/<nn>/s/sys/smtpd.cfg
```

- 2 Add the following lines:

```
# Reject and log submission from addresses listed in bl.spamcop.net:
SUBMIT log_reject DNSBL,bl.spamcop.net,ALL
```

- 3 Restart the smtpd service:

```
omoff -d0 -w smtpd
omon smtpd
```

Authentication

This chapter explains how Scalix's authentication system works and how to integrate Scalix with external authentication systems, such as the Lightweight Directory Access Protocol (LDAP), Windows NT domain, and Kerberos. If you plan to use the native Scalix authentication, you can skip this chapter.

Contents

This chapter includes the following information:

- “Introduction” on page 43
- “Using Pluggable Authentication Modules (PAM)” on page 44
- “Configuring Scalix for LDAP Authentication” on page 49
- “Configuring Scalix for Windows NT Authentication” on page 52
- “Configuring Scalix for Kerberos Authentication” on page 53

Introduction

Many components of Scalix require authentication, and there is more than one authentication mechanism.

Scalix components that require authentication include:

- All User Access Layer (UAL) remote clients, including Scalix Connect for Microsoft Outlook and regular e-mail applications, where UAL is a proprietary Scalix protocol that enables communication between clients and the Scalix server
- Post Office Protocol 3 (POP3)
- Internet Message Access Protocol (IMAP)
- Simple Mail Transfer Protocol (SMTP) relay for SMTP authentication
- LDAP server
- Scalix administrator commands
- Scalix diagnostic commands

There are several ways to perform authentication on a Scalix system. You can rely entirely on Scalix's authentication setup. You can use an existing LDAP server, Microsoft Windows NT

domain, or the Kerberos method, which function along with the Scalix authentication method.

For maximum flexibility, all authentication configurations are based on the Linux standard Pluggable Authentication Modules (PAM). You can use either the Scalix-supplied PAM modules or Linux PAM modules.

Authentication works in Scalix as follows:

- The clients communicate using their own language (UAL, IMAP, or POP3) to request authentication from the Scalix server, where they encounter the PAM modules
- On the server, the PAM modules get the user's information from the directory
- When using the Scalix authentication system, the PAM modules verify the username and password in the directory, then send confirmation back to the client via UAL, IMAP, POP, or SMTPD.

When using external authentication, such as LDAP or Windows NT, it grabs the authentication ID from the directory and sends that to the LDAP server for authentication. The external authentication server sends a success/failure message back to the PAM modules, which pass the message back to the clients via UAL, IMAP, POP, or SMTPD.

Components in Scalix incorporate authentication in different ways, and not every implementation is fully logical. Fortunately there is some level of convergence; most e-mail clients end up using a UAL remote connection type as their point of entry into the server. This is true for Scalix Connect for Microsoft Outlook, Scalix Web Access, IMAP clients such as Mozilla Thunderbird, and even the command-line mail tools omsend, omlogon, and so on. The exception is the POP and SMTP server, which communicates with the message store directly for performance reasons. So the POP service needs its own PAM configuration file, which is quite easy to miss when setting up external authentication.

The Scalix administrator commands also use the PAM infrastructure, which checks if a user executing an administrator command has administrator rights or if their effective user ID equals 0, identifying the user as a root-level user.

Two specific commands, omqdump and omcontain, require another, extra layer of protection. They use the so-called "diag" authentication, which is based on a password known only by those properly initiated through Scalix training.

Using Pluggable Authentication Modules (PAM)

PAM connects applications that require authentication with shared library modules, including those that interface with external authentication mechanisms. This includes authentication mechanisms such as Unix-passwd, SMB/NT Domains, LDAP, Kerberos, and RADIUS. Open network authentication mechanisms, such as RADIUS and TACACS, are also supported by PAM modules, as are a variety of biometric devices, such as fingerprint and iris scanners.

PAM is configurable. Scalix is installed with a collection of PAM modules, but you can configure them for freeware or commercial modules that follow the PAM specification.

Various applications, such as the UNIX login and network services such as HTTP, FTP, or secure shell (SSH), can be set up to use the PAM infrastructure.

PAM Configuration File Syntax

PAM configuration files are rule-based files that follow a common syntax. Two examples are provided, then the syntax is explained in the following sections.

Here is an example for the PAM configuration file called `ual.remote`. The first line is commented out using the `#` symbol.

```
# var/opt/scalix/nn/s/sys/pam.d/ual.remote example
auth required om_auth nullok
account required om_auth
session required om_auth
password required om_auth nullok
```

Here is another example, for the PAM configuration file called `pop3`:

```
# var/opt/scalix/nn/s/sys/pam.d/pop3 example
auth required /lib/security/pam_smb debug nolocal
account required om_auth
session required om_auth
password required /lib/security/pam_smb debug nolocal
```

When modifying the configuration files to set up external systems, remember this syntax:

```
module-type control-flag module-path args
```

Table 1: PAM Syntax

Syntax	Description
module-type	One of the four types (three used by Scalix). <code>auth</code> — Whether or not authentication is required. <code>account</code> — Whether or not a user account name is required. <code>session</code> — Not used but required. <code>password</code> — Whether or not a password is required.
control-flag	Defines how every stacked module contributes to the overall success or failure of the operation. There are four values for this field: <code>required</code> — This module is required for authentication to succeed. <code>optional</code> — The result of this module is used only if there are no other modules returning results. <code>sufficient</code> — If the module succeeds, stop processing modules and return success. <code>requisite</code> — If the module fails, stop processing modules and return failure.
module-path	A relative or absolute path to the PAM shared library module with or without its <code>.so</code> suffix. In Scalix-PAM, path names are interpreted relative to <code>/opt/scalix/lib/security</code> while in standard Linux PAM they are relative to <code>/lib/security</code> . See the next table for the parameters.
args	A list of arguments passed to the PAM module. Every PAM module supports at least a few generic arguments. <code>nullok</code> — Allows empty passwords to be used.

PAM Modules that Scalix Supplies

While any PAM Linux module can be used, Scalix supplies some that differ in their logging behavior (syslog vs. Scalix log), their configuration file location (/etc vs. ~/logs), and the type of username they can accept.

Configuration depends on authentication location. When using native Scalix authentication, leave these modules as they are. When using external authentication, modify them. We explain later how to modify the Scalix PAM modules for use with an external authentication system such as LDAP.

The following list categorizes the Scalix-supplied PAM modules.

Table 2: Scalix PAM Modules

Parameter	Description
Full Authentication Modules	
om_auth	<p>The main Scalix module. Authenticates against the Scalix USERLIST directory. The om_auth PAM module provides authentication functionality against the Scalix internal USERLIST directory in the following way:</p> <ul style="list-style-type: none"> • The authentication function of the module asks the user for a password, encrypts it, and checks it against the UL-PWD attribute in USERLIST. • The account function checks if the password is not expired, the account is not locked, and no other restrictions exist. • The password function enables the user to change the password in USERLIST. It updates the UL-PWD and UL-SASL-PWD attributes. (The latter is needed by SMTP, IMAP, and LDAP authentication when using the more secure SASL authentication.) For SMTP, this is the only supported authentication method. Plain text authentication is not allowed. • The UAL layer canonicalizes the user name to Positional Format. So the username the PAM module works on contains at least the X.400 Personal Name, Mail-node, and Common Name.
om_ldap	A Scalix version of the standard Linux pam_ldap module.
om_krb5	A Scalix version of the standard module for Kerberos 5.
Generic PAM Helper Modules	
pam_permit	Always allow. (Useful if stacking is required.)
pam_deny	Always deny. (Useful if stacking is required.)
pam_if	Conditionally stack.
pam_listfile	Allows if user is part of a textfile-based simple list.
Administrator Scalix PAM Modules	
om_admin	Allows if Scalix user associated with callers UNIX user ID is either root or a Scalix administrator user.
om_unix2om	When this is stacked with om_admin, it converts a UNIX user ID into a Scalix User ID.
om_diag	Support special authentication modes for omqdump/omcontain.

Table 2: Scalix PAM Modules

Parameter	Description
Special Integration PAM Module	
om_om2authid	Convert a Scalix user ID into an authentication identifier. (Useful if stacking is required.)
Unsupported Modules	
pam_radius_auth	Not supported in Scalix PAM modules; use Linux PAM modules instead.
pam_smb	Not supported in Scalix PAM modules; use Linux PAM modules instead. When using a Windows NT domain authentication, see “Using the pam_smb_auth Module” on page 52.
pam_unix	Not supported in Scalix PAM modules; use Linux PAM modules instead.

PAM Modules that the Operating System Supplies

You can use any OS-supplied PAM module with Scalix. When customizing the Scalix PAM configuration file, be sure to enter the absolute path name to the correct module.

The module can need additional configuration; refer to the module’s documentation/MAN page for details.

The Scalix username handed over by the Scalix PAM library must first be converted into the Authentication Identifier. This is done using the special om_om2authid PAM module in the stack.

When using an OS-supplied PAM module, be aware of the following requirements:

- The module name needs to be specified as a fully-qualified path in the PAM configuration files.
- The module will almost certainly require configuration files outside the Scalix directories; therefore it might not be possible to configure multiple instances of the module differently in a multi-instance Scalix environment.
- Scalix PAM modules use a username in positional format, which cannot be interpreted by an OS-supplied PAM module. The username must first be converted into the user’s authid, which is used to *link* to the external authentication provider. The om_om2authid module is stacked before the actual PAM module and does the conversion in the PAM environment.

PAM Module Stacking

If one application has multiple PAM entries, as is the case with an external authentication system, they execute in the order in which they appear in the configuration file. So if one method of authentication fails, it fails over to the other.

For example, if you are using an existing LDAP authentication system that fails, your users can provide their Scalix username and password to access the system. In this scenario, the ual.remote file has these entries:

```
auth sufficient om_ldap
auth sufficient om_auth
auth required pam_deny
```

```
account required om_auth
password required om_auth
session required om_auth
```

where “sufficient” means that the external LDAP authentication is not “required” so it can fail over to the internal Scalix authentication.

In the sample code, the system first tries to authenticate through “om_ldap”. If that succeeds, everything is fine and the user is allowed access. If om_ldap fails, it tries om_auth. If that one succeeds, the user is allowed access. If it fails, the system tries pam_deny, which always fails. The user has two ways to authenticate: through their local password or their LDAP password.

The following examples apply to module stacking for the auth module type only. Note that these configurations can be adapted for the other module types as well.

Try to authenticate using LDAP first and if the user is unknown try Scalix next:

```
auth sufficient om_ldap user_unknown=ignore
auth sufficient om_auth use_first_pass
auth required pam_deny
```

Try to authenticate using a Kerberos password, and if the user is “admin” and Kerberos fails try a second time, authenticating against the Scalix internal password:

```
auth sufficient om_krb5
auth required om_admin
auth required om_auth use_first_pass nullok
```

The user account must exist in Kerberos, even when the password does not match.

Configuring PAM for External Authentication

When using an external authentication system, four configuration files must be modified so that Scalix knows to authenticate against an external source. The configuration files are located in the `var/opt/scalix/<nn>/s/sys/pam.d` folder.

Table 3: PAM Configuration Files

File	Description
ual.remote	Allows Microsoft Outlook and Scalix Web Access users to authenticate against an external authentication server.
omslapdeng	Allows Scalix Web Access personal contacts to be searched.
smtpd.auth	Allows users coming in through SMTPD to authenticate against an external authentication server.
pop3	Allows POP3 users to authenticate against an external server.

To configure PAM for external authentication

- 1 Access the following folder:

```
var/opt/scalix/<nn>/s/sys/pam.d
```
- 2 Modify the ual.remote, omslapdeng, smtpd.auth, pop3 files. Read the next section for help with the ual.remote file.

Configuring the ual.remote File

The om_auth module is used by default for Scalix-based authentication. It can be used for all four module types, meaning auth, account, session, and password. If you do not intend to use external authentication, leave these as is.

The default PAM configuration for ual.remote is as follows:

```
auth    required om_auth nullok
account required om_auth
password required om_auth nullok
```

Commonly-Used Generic Option

The following generic option is commonly used in PAM configuration. It is located in the ual.remote file and commented out by default.

- use_first_pass – Do not prompt the user for a password. Instead, use the password retrieved by a previous module in the stack. Example:

```
auth sufficient om_ldap use_first_pass
```

All modules handling authentication should implement the use_first_pass argument, preventing a secondary module on the stack prompting for a password again and instead taking it from the existing PAM environment created for and modified by the module(s) stacked on top of it.

PAM Integration with LDAP

When integrating with an LDAP server, the PAM configuration for the ual.remote file and other services typically looks like this:

```
auth sufficient om_ldap user_unknown=ignore
auth sufficient om_auth use_first_pass nullok
auth required pam_deny
```

This allows a user to be successfully authenticated through LDAP but still accepts accounts defined locally to Scalix only, for example administrator accounts.

Configuring Scalix for LDAP Authentication

If you choose not to use the PAM authentication that is native to Scalix, the system also supports authentication against a non-Scalix directory, such as OpenLDAP, which is an open-source package available for Red Hat and SUSE Linux. It also provides failover capabilities so that if one directory is not available, a secondary directory is automatically used.

Before integrating with an LDAP authentication server, you must decide whether to use the Scalix PAM LDAP modules (om_ldap) or the regular Linux PAM LDAP modules (pam_ldap). The Scalix-supplied module is more efficiently integrated into the Scalix environment by means of placement of log and configuration files, but it has limitations:

- If a search operation is required, the LDAP server must allow for anonymous access; pre-binding is not supported
- The module does not support LDAP password changes

- When using Transport Layer Security (TLS), the module uses part of the operating system configuration for open LDAP clients. This could interfere with a configuration of the operating system to use LDAP for authentication purposes.

Most Linux vendors deliver newer and more advanced versions of PAM modules for LDAP authentication. The advantages of these versions often outweigh any additional problems. So consider deprecating the `om_ldap` and relying solely on `pam_ldap`.

The rest of this section outlines how to allow users to authenticate with their LDAP passwords against an LDAP-compliant directory when accessing their Scalix mailboxes, using `om_ldap`.

Configuring LDAP Authentication for Clients

To set up communication among the various clients, the PAM modules, and LDAP, you create or configure several files on the Scalix server.

To configure the Scalix server for different clients

- 1 Open the following folder:

```
var/opt/scalix/<nn>/s/sys/pam.d
```

where `nn` varies with Scalix release.

- 2 Inside that directory, find the following files:

- `ual.remote` – For UAL clients, such as Microsoft Outlook and Scalix Web Access
- `smtpd.auth` – For Mozilla Mail, which uses AuthSMTP
- `pop3` – For POP3 clients, such as Eudora or Mozilla Mail, create or modify this file
- `omslapdeng` – For Scalix Web Access personal contacts, create or modify this file

- 3 Add the following lines to each file for which you have active clients:

```
auth sufficient om_ldap
auth sufficient om_auth
auth required pam_deny
account required om_auth
password required om_auth
session required om_auth
```

The first three lines (`auth`) test the username and password that the user supplies. The second line “`auth sufficient om_auth`” provides a secondary opportunity to access the mailbox if invalid credentials are passed to the LDAP source. The second line compares the credentials to the Scalix directory. Remove this line if you do not want to provide this, but if you use Scalix Management Console (SAC) keep this “`auth sufficient om_auth`” second line because Scalix Management Console uses the `omslapdeng` file and needs to authenticate `sxadmin` and `sxqueryadmin` users.

The fourth line (`account`) decides whether or not you have access to the system.

The fifth line (`password`) changes your password if needed.

The sixth line (`session`) currently is not in use but must be present for proper system function.

Final Configuration

The final configuration for LDAP is to the general LDAP configuration file, *om_ldap.conf*.

While the PAM configuration file connects the LDAP PAM module to the Scalix system, the module itself must be configured to know how to connect to the LDAP server. This is done through the *om_ldap.conf* file.

To configure the general LDAP file

- 1 Open the following folder:

```
var/opt/scalix/<nn>/s/sys
```

 where nn varies with Scalix release.
- 2 Create (or modify) the file *om_ldap.conf*
- 3 Add the following lines.

```
host=ldaphost.acme.com
search=subtree
base=dc=acme,dc=com
filter=uid=%s
```

where

- **host** – Specifies your hostname and optionally port number for the LDAP server to use
- **base** – Specifies the search base for the initial search operation as a DN; only set for search values of one or subtree
- **search** – Can be one of none, one, subtree to specify the depth of search; set to none if using a one-level tree to avoid the search operation altogether
- **filter** – LDAP filter string used for the search in one and subtree modes
- **dn** – Set to the DN of the user with search=none
- **tls** – Set to on or off, to enable the LDAP module to negotiate use of TLS; the LDAP server must also be configured to support this

Verifying LDAP Authentication

To determine if the functionality is working properly, choose a user who has a different LDAP password than their Scalix password.

To verify LDAP authentication

- 1 Access Microsoft Outlook.
- 2 In the Scalix login screen, enter the user's LDAP password. Expected behavior is successful login.
- 3 Exit from Microsoft Outlook.
- 4 Access Microsoft Outlook.
- 5 In the Scalix login screen, enter the user's Scalix password. Expected behavior is successful login. (You may to restart the session to get the login window.)
- 6 Exit from Microsoft Outlook.

- 7 Access Microsoft Outlook.
- 8 In the Scalix login screen, enter an incorrect password. Expected behavior is failed login. (You may to restart the session to get the login window.)

Configuring Scalix for Windows NT Authentication

A third option for authenticating is an existing Windows NT 4 Domain authentication system.

Integrating Scalix with Windows NT Authentication

Configure the Domain Name System (DNS) and `/etc/hosts` file, then create a `pam_smb.conf` file.

To integrate Scalix with a Windows NT authentication system

- 1 Put the hostname of your primary and secondary Windows domain controller in your DNS and/or `/etc/hosts` file. As these are NETBIOS names, typing the names in ALL CAPS is important.
- 2 Create the `/etc/pam_smb.conf` configuration file with a text editor. The file must have three lines:

```
NTDOMAINNAME
NETBIOS_NAME_OF_PDC
NETBIOS_NAME_OF_BDC
```

where you insert your domain name, the name of your Primary Domain Controller (PDC), and any backup domain controller (BDC). The domain controllers must be resolvable through the `/etc/hosts` file or DNS and have to resolve into an IP address. If you have only one domain controller, put its name in twice. Examples are

```
XANDROSNT
ASTERIX
OBELIX
```

and

```
XANDROSNT
ASTERIX
ASTERIX
```

Using the `pam_smb_auth` Module

The `pam_smb_auth` PAM module that comes with all supported Linux distributions is a perfect example of the benefits of and issues with OS-supplied PAM modules.

On one hand, the module makes it possible to authenticate against a Windows NT domain controller or other server message block (SMB) system. This is currently not possible with Scalix-supplied PAM modules.

On the other hand, the module provides limited debugging information, configuration file syntax must be followed rigidly, and capitalization and empty lines matter.

To use the `pam_smb_auth` module

- 1 Set up your `ual.remote` Scalix PAM configuration file (not a LINUX PAM configuration file) as follows:

```
auth required om_om2authid
auth sufficient /lib/security/pam_smb_auth debug noLOCAL
auth sufficient om_auth use_first_pass
auth required pam_deny
```

Configuring Scalix for Kerberos Authentication

A fourth method of authentication that works with Scalix is Kerberos, which can take several forms. The approaches to Kerberos authentication and how to implement each one are explained here.

About Kerberos Authentication

Kerberos is a network authentication protocol designed to provide strong authentication for client/server applications by using secret-key cryptography. Kerberos uses authentication tickets to authenticate users and/or services.

A Kerberos client can perform communications with a Kerberos service if both the client and the service authenticate against a Kerberos Key Distribution Center (KDC; the Kerberos server) and obtain a Ticket Granting Ticket (TGT). The client then requests a service ticket for a specific service. When a user logs in to the domain, a request is made for a ticket, and once authenticated, the user can use that ticket for as long as it remains valid. For single-sign-on, a client computer typically stores tickets for services that it has acquired in the past in a temporary file, called the ticket cache. With non single-sign-on Kerberos authentication, the client sends a normal username/password pair to the server. The server, through the `om_krb5` PAM module, acquires a service ticket, which is discarded and not cached.

So a triangular relationship exists between the client and the KDC, the service and the KDC, and the client and the service. A Kerberos principal is either a client identity or a service identity operating in the Kerberos realm.

In the Scalix environment, you can use secure Kerberos communication with the following Scalix services:

- Remote Execution Service, or RES, which is the agent component of Scalix Management Services
- Scalix Management Console
- Scalix UAL service
- Scalix IMAP service

There are two options to use Kerberos:

- **Single-sign-on** – You can configure single sign-on authentication with a KDC on the master domain controller that uses Microsoft Active Directory. This allows Scalix users to automatically authenticate with the Scalix server when they log in to their Windows domain. The user does not need to authenticate when accessing their e-mail application.

- **Non single-sign-on** — A user is required to enter a password to log in to Microsoft Outlook, Scalix Web Access, or IMAP clients, and the password the user enters is their Kerberos password instead of their Scalix password.

Single-Sign-On Kerberos Authentication

Single-sign-on authentication allows Microsoft Outlook users to access their e-mail using the Kerberos security protocol. This authentication mechanism allows users to log in to their local domain in a Microsoft Active Directory environment and access their e-mail without any further authentication.

The Active Directory service is a core component of the Windows operating system. It provides a directory service supporting LDAP, and a Kerberos KDC to authenticate users. It allows organizations to share and manage information about network resources and users and provides a single-sign-on environment that integrates with the standard Windows desktop login. In addition, it acts as a single point of management for Windows-based user accounts, clients, servers, and applications.

Regarding the use of Kerberos with e-mail clients other than Microsoft Outlook, Kerberos single-sign-on for Scalix Web Access is not supported. It is possible to use Kerberos-based sign-on for IMAP, so Thunderbird will work. Thunderbird on Windows currently does not link into Microsoft's built-in Kerberos implementation and requires MIT's kerberos client for Windows to be deployed on the client, which makes it practically unusable, but it works fine on Mac OS/X and Linux.

When Microsoft Outlook is launched, the Scalix Connect for Microsoft Outlook connector takes the user's Kerberos credentials to access the server.

To implement the single-sign-on environment, you must have:

- Active Directory
- Scalix Connect for Microsoft Outlook
- Scalix Server
- The ktpass utility from the Microsoft Developer Network Web site at <http://msdn2.microsoft.com/en-us/default.aspx>. ktpass is also available from the Windows 2000 resource kit and the Windows Server 2003 installation CD under `\Support\Tools\Support.cab`

Creating Keytab Files

The ktpass utility creates the keytab files used by Linux Kerberos-based systems to define KDC hosts and user/service mappings.

Install the ktpass utility on the Windows domain controller server.

The ktpass command (next procedure) generates the keytab files from entries stored in Active Directory.

Configuring Single-Sign-On

Single-sign-on configuration requires you to make configuration changes to DNS and create an Active Directory user. This user, for single-sign-on purposes, is actually the Scalix Service. You must create an Active Directory "user" for the UAL Scalix service. When this is complete, you then convert this user/service into a Kerberos Service Principal. You can also create an Active Directory user for the Scalix IMAP service if single-sign-on users in your network use the Evolution e-mail client.

To configure single-sign-on authentication

- 1 On the domain controller, go to **Start > Programs > Administrative tools > DNS**.
- 2 Create Forward Lookup Zones for your domains and created host records for all Scalix servers in the appropriate zone. For example when your Scalix server is scalix1.yourcompany.com, the forward lookup zone is yourcompany.com and the host field in the host record is scalix1.
- 3 Under **Forward Lookup Zones**, select a Scalix server single-sign-on domain and go to **Action > New Alias**.
- 4 In the **Alias name** field, enter scalix-default-mail.
- 5 In the **Fully qualified name for target host** field, enter the fully-qualified name of the Scalix server with which you are using single-sign-on (for example, scalix-server.acme.net or scalix1.yourcompany.com).
- 6 Click **OK**.
- 7 Select **Reverse Lookup Zones** and make sure you have created zones for your domain subnets.
- 8 In the subnet in which the single-sign-on Scalix server resides, select **Action > New Pointer**.
- 9 Enter the last two or three digits of the Scalix Server IP address and fully-qualified host-name of the Scalix Server (for example, scalixserver.acme.net).
- 10 Click **OK**.
- 11 Close the DNS window.
- 12 Go to **Start > Programs > Administrative Tools > Active Directory Users and Computers**.
- 13 If it does not already exist, right-click the root domain controller and select **New > Organizational Unit** and name the new unit as follows:

scalix services

This creates a separate organizational unit to contain Scalix server data.
- 14 Select the new Scalix Services organizational unit and select **Action > New > User**.
- 15 In the **First Name** field, enter the following:

scalix-ua1

You can also enter the name of the single-sign-on Scalix server in the **Last Name** field. This allows you to identify the keytabs you generate for multiple Scalix servers.
- 16 Do not modify the selection (domain) in the pull-down menu.
- 17 In the **User logon name** field, enter the following:

scalix-ua1
- 18 Click **Next**. The Password window displays.
- 19 Enter and confirm a password for the user. Make sure that the password you enter is sufficiently complex and that
 - The **User must change password at next logon** field is not selected

- The **User cannot change password** field is not selected
 - The **Password never expires** field is selected
- 20 Click **Next**. If Microsoft Exchange is installed on the server, the Exchange Mailbox window displays.
 - 21 Clear the **Create an Exchange Mailbox** field.
 - 22 Click **Next**.
 - 23 Click **Finish**. This completes the creation of an Active Directory user that represents the Scalix UAL Service for the Scalix Server.
 - 24 Open a DOS window and change the directory (cd) to the directory that contains ktpass (typically, *c:\Program Files\Support Tools*).
 - 25 To change the Scalix Service account to a Kerberos Service account and generate a keytab, enter:

```
ktpass -princ scalix-ual/scalixservername.domain@REALM -mapuser
<domain>\scalix-ual -pass password -out path\filename -kvno 3
```

For example:

```
ktpass -princ scalix-ual/scalixserver.acme.net@ACME.NET -mapuser
scalix-ual -pass <password> -out scalix-ual.keytab -kvno 3
```

The -kvno option prevents potential key version mismatches that cause single-sign-on to fail. Setting this value to 3 ensures that keytab version is the same for existing and future users in Active Directory. Please note that the keytab created by ktpass may not work when the -kvno option differs from the one in Active Directory, in which case you can specify the /ptype option as “KRB5_NT_PRINCIPAL” and/or the /crypto option to be “DES-CBC-CRC”.

- 26 If you used the Last name field and entered scalixserver1, enter:

```
ktpass -princ scalix-ual/scalixserver.acme.net@ACME.NET -mapuser
scalix-ual-scalixserver1 -pass password -out scalix-ual-
scalixserver1.keytab
```

Then you see the following information that indicates the keytab was successfully created:

```
Successfully mapped scalix-ual/scalixserver.acme.net to scalix-ual.
Key created.
Output keytab to scalix-ual.keytab:
Keytab version: 0x502
keysize 68 scalix-ual/scalixserver.acme.net@ACME.NET ptype 1
(KRB5_NT_PRINCIPAL)
vno 3 etype 0x1 (DES-CBC-CRC) keylength 8 (0xe6fb762ad01f8a9b)
Account has been set for DES-only encryption.
```

- 27 Copy the keytab to the home directory of the single-sign-on Scalix server. You can use a floppy disk or the scp command to transfer the keytab.
- 28 On the Scalix server, log in using your Linux account and then change to root user.
- 29 Merge the keytab you created with the Kerberos system keytab file. Enter:

```
ommergekeys /path/filename.keytab
```

30 Modify the `/etc/krb5.conf` file. Enter:

```
omkrbconf -r REALM -s servername.domain -d domain
```

where

- `-r` specifies the realm that the Kerberos database controls. For example, if your domain name is `acme.com`, your realm is `ACME` or `ACME.NET`. The realm was specified during installation of Scalix.
- `-s` specifies the fully qualified host name of the Kerberos KDC machine. For single-sign-on, the KDC is the Domain Controller with Active Directory installed. It was also specified during Scalix installation.
- (optional) `-d` specifies the domain name in which the Kerberos realm operates. If you do not specify a value, the domain name is determined from the current domain.

31 In order for single-sign-on to operate, the authentication ID for a Scalix server mailbox must match the domain identity (the ID in Active Directory) for the user. For example, if `jsmith@acme.net` is the User Logon ID for a user in Active Directory, enter the following on the Scalix Server:

```
ommodu -o jsmith --authid jsmith@ACME.NET
```

The REALM information must be uppercase.

32 To view the authid value (`-o`) for a user, enter:

```
omshowu "Joe Smith/mailnode"
```

This user can now use single-sign-on authentication. After the user logs in to the Windows domain, the user no longer must enter username or password information during Microsoft Outlook profile creation or login.

If Active Directory is unavailable at any point after setting up single-sign-on, the Scalix server prompts users for their regular domain password for authentication.

Non Single-Sign-On Kerberos Authentication

This differs from single-sign-on authentication in that the user is required to enter a password to log in to Microsoft Outlook, Scalix Web Access, or IMAP clients, and the password the user enters is their Kerberos password instead of their Scalix password.

To configure non single-sign-on Kerberos authentication

1 Make sure that Kerberos Server, Workstation, and Libraries are installed on a Scalix Server in your network. To verify that the required Kerberos RPMs are installed on the system, enter:

```
rpm -qa | grep krb
```

You can obtain any missing Kerberos RPMs from the Linux operating system installation CDs. See <http://web.mit.edu/kerberos/www/> (Red Hat and Fedora) or <http://www.pdc.kth.se/heimdal/> (SUSE) for more information.

2 Initialize the KDC. Enter:

```
omkrbinstall -r realm -s servername.domain -a username -p password
```

where

- -r specifies the realm that the KDC manages. It was specified during installation of Scalix, for example.
- -s specifies the name of the Scalix Server that hosts the KDC
- -a specifies the principal fully qualified hostname of the KDC administrator, which was also specified during Scalix installation
- -p specifies the KDC administrator password

3 The following prompt displays:

```
Checking MIT Kerberos installation...done. Initializing database
'/var//krb5kdc/principal' for realm 'REALM', master key name
'K/M@REALM' You will be prompted for the database Master Password. It
is important that you NOT FORGET this password.
```

```
Enter KDC database master key:
```

4 Enter a password for the KDC database.

```
Reenter KDC database master key to verify:
```

5 Reenter the password for verification. The following information displays:

```
Success! Kerberos database created, configured and started.
```

6 Create Scalix Service principal keytabs. Enter:

```
omaddprincs -s all -h server.domain -o filename.keytab
```

where

- Specifying all for the -s option creates one keytab file for all Scalix Services (Remote Execution Service, UAL, IMAP, and the Scalix Management Console Service). However, you can also create individual keytabs for specific services. See the omaddprincs man page for more information.
- -h specifies the fully qualified domain hostname of the Scalix Server on which the Scalix Services are installed
- -o specifies the keytab file name

The following information displays when you use -s all to create a keytab:

```
Creating new Scalix principals in Kerberos database and keytab
filename.keytab:
```

```
scalix-ual/server.domain@REALM
imap/server.domain@REALM
ubermanager/server.domain@REALM
res/server.domain@REALM
```

7 If necessary, manually copy the keytab file(s) to the Scalix Servers on which you want to use Kerberos authentication. The location of the keytab file was specified during installation of Scalix (see the screen capture in the *Scalix Installation Guide*).

8 Modify the `var/opt/scalix/nn/s/sys/pam.d/ual.remote` file so that it appears as follows (modify the lines in bold text):

```
# Standard Scalix Authentication
#
# Comment this out if you want to use one of the alternative
```

```
# authentication schemes below.
# auth required om_auth nullok
#
# Kerberos authentication 1
#
# With this scheme we attempt local authentication first and, if that
# fails, we try Kerberos authentication. If we do it the other
# way around we run the risk of the KDC locking a principal account for
# users that are known to both Kerberos and Scalix. See om_krb5(8) for
# more information.
#
# auth sufficient om_krb5 use_first_pass
# auth required pam_deny
# Kerberos authentication 2
```

When you modify and save the `ual.remote` file, client sessions are initiated using `UAL_INIT/UAL_SIGNON`, and include the principal name and password. After the user is found (verified) in the `USERLIST` Directory, the `authid` value from their Directory entry is used to authenticate them through PAM. The `om_krb5` module (with the `authid` value and password) is used to contact the KDC through the Kerberos client libraries.

- 9 For non single-sign-on Kerberos POP access, modify the `~/sys/pam.d/pop3` file so that it appears as follows (modify the lines in bold text):

```
#auth required om_auth
account required om_auth
password required om_auth
auth sufficient om_krb5 use_first_pass
auth required pam_deny
```

- 10 Scalix users will now authenticate against the KDC using their Kerberos password. If users experience problems while logging in to Scalix, make sure they are in the KDC.

```
kinit username
```

- 11 If the user is not in the system, enter:

```
kadmin.local
addprinc -pwd password username
```

- 12 This adds a user principle. To verify that the user was successfully added, enter:

```
listprincs
```

You should see a user principal for the user you created.

Make sure the user's `authid` value is set to `username@DOMAIN.NET`

Using the Domain Password

If you want to use Kerberos authentication and have users enter their Windows (Active Directory) domain password when logging in to Scalix, complete all the steps in “Single-Sign-On Kerberos Authentication”, and then edit the `~/sys/pam.d/ual.remote` file as described in step 8 of “Non Single-Sign-On Kerberos Authentication”.

Behavior to Note

If your system performs Kerberos-based single-sign-on with Scalix Connect for Outlook, note that the behavior of username mapping has changed since the 10.0.0 release:

- For all implementations, the Authentication ID of the user must be modified to convert the existing username to all-lowercase letters. For example, if a users Authentication ID has been “SmithJane@SCALIX.REALM”, it must be converted to “smithjane@SCALIX.REALM”.
- A new omltdapsync option, detailed as follows, is provided to automate this mapping when synchronizing the username from Active Directory or other LDAP-based directories.
- For Active Directory based implementations, where the Active Directory KDC treats the username as case-insensitive, some issues result with capitalizations of usernames during sign-on to be resolved.
- For MIT-Kerberos based Kerberos KDC implementations, where the MIT KDC treats the username as case-sensitive, Kerberos-based single-sign-on will now only work with lowercase usernames on the Kerberos server side. This is consistent with most implementations of MIT Kerberos.

Also, if you make use of Active Directory as the master for omltdapsync agreement type 11, and are using Kerberos for single-sign-on, then the principal username (from Active Directory) will need to be converted to lowercase@UPPERCASE in Scalix. This can be done by editing the following mapping line in the agreement configuration file. Convert this line:

```
userPrincipalName|UL-AUTHID|*,1,256|!TOUPPER=@|
```

to

```
userPrincipalName|UL-AUTHID|*,1,256|!CUSTOM=TO_CANONICAL_PRINCIPAL
```

This should be completed when creating the new synchronization agreement, before initiating omltdapsync. If you are upgrading from an older version of Scalix, the existing synchronization agreement must be edited as just shown. You must then run omltdapsync with the -M option (as detailed in the omltdapsync MAN page). This will force all the existing records in Scalix to be updated.

Securing Scalix

This chapter covers ways to secure a Scalix system, ranging from internal security measures to the use of virtual private networking (VPN), an Apache proxy server, stunnel, and certificates.

Contents

- “Overview” on page 61
- “Internal Security Precautions” on page 62
- “Using a VPN” on page 63
- “Using an Apache Proxy Server” on page 64
- “Using stunnel” on page 68
- “Using Certificates” on page 74
- “Other Forms of Security” on page 75

Overview

Scalix is only as secure as its operating system. Anyone with root permissions has unlimited access.

Scalix servers are typically kept behind a corporate firewall, such as inside an intranet. In these cases, using VPN technology for client access is a useful way to secure remote access, and no additional security provisions are needed for Scalix. You can also use an Apache proxy server or an stunnel server in a DMZ to secure access to the system. If you want to use Secure Sockets Layer (SSL) for Microsoft Outlook, meaning secure e-mail communication, you need to configure stunnel.

SSL can be used for encryption of communication channels for data integrity and privacy (as opposed to authentication). Specifically, SSL is supported for securing IMAP, POP, SMTP, LDAP, HTTP, and UAL protocols, where UAL refers to the Scalix protocol for communication between Scalix Connect for Microsoft Outlook and the Scalix server. (Scalix Connect for Evolution uses IMAP, not UAL.) The Scalix server and e-mail applications need to be configured to use SSL. Configure stunnel for Microsoft Outlook as outlined in this chapter. For the e-mail application, you configure the user’s e-mail account to use SSL.

Tip

To implement even higher security, implement the Kerberos authentication protocol as described in the “Authentication” chapter.

Internal Security Precautions

There are security precautions that you can take.

Data Security

To prevent unauthorized access to data, ensure regular users are not assigned to the *scalix* group because it is a special Linux user group that owns Scalix data. All Scalix data is owned by a user named *scalix* in the group named *scalix*. This user and group are created when you install the software. They are UNIX groups created in */etc/passwd* and */etc/group*, and they are not administrator accounts, so you do not see them when you access Scalix Management Console (SAC). They are used for file ownership.

Individual user data is password protected. Users access their data by being registered with Scalix. All users must enter their passwords before accessing their Scalix data.

Administrator Capabilities

Only administrators or users with root permissions can add, delete, or modify user accounts, or modify the Scalix system.

The `omcheck` command enables the administrator to verify that ownerships and permissions are set correctly for Scalix system files and directories.

Restricted User Access

You can control access to public folders using access control lists.

Individual users can control access to their mail, calendar, and contact folders using delegate permissions.

Message Security

There are instances when a Scalix administrator can read messages addressed to other people. For example, if a non-delivery report comes through the administrator's e-mail box, or when using some Scalix diagnostic tools, the administrator may see the content of an individual message. Messages marked as "Personal", "Private", or "Company Confidential" cannot be read by the administrator.

Monitoring Usage

The Audit Log, which records user activity, can identify unusual usage patterns. In addition, in case of break-ins or inappropriate activity, it can provide evidence of when individual users were on the system.

Virus Protection

Scalix integrates with a few third-party anti-virus programs (Trend Micro, McAfee, and ClamAV). When you activate virus scanning, the service router scans all Scalix message attachments. Depending on how you configure virus scanning, the Scalix server can attempt

to repair infected files, return infected messages to the sender, or discard the message. See the virus protection chapter.

Spam Protection

Scalix allows you to configure anti-spam measures on the SMTP relay to prevent abuse of the Scalix system by external entities. It integrates with SpamAssassin, among other anti-spam programs. See the spam protection chapter.

Microsoft Outlook Security

Microsoft Outlook e-mail security parameters provide protection against software viruses that users can receive in their inbox as an attachment file.

Unless stunnel is configured, allow access to Scalix Connect from outside a network only through a secure VPN.

Using a VPN

The most efficient way to secure communication among various e-mail clients and Scalix is virtual private networking (VPN).

A VPN allows specified users to access your network over the Internet. It enables employees to work from locations other than the office, for example from home. When encryption is used, the connection is secure.

There are three approaches for VPN servers. First, you can use a computer or server to function as a VPN server. With this approach, Xandros Server or other software is installed on the computer. Second, you can use a router or firewall that functions as a VPN server. The VPN-capable routers are located in the home and at the office. The processors used in a router are usually slower than those in computers, so the connections are also slower. Third, you can outsource the service, whereby a service provider sets up a network access server (NAS) and provides software. Your users access the NAS with the software and your local number or toll-free number. This setup is appropriate when you have hundreds of remote, mobile users, but also means that long-distance charges can apply.

A VPN server is typically located in a demilitarized zone (DMZ) behind a firewall, buffering your local area network (LAN) from the Internet. Users require an account on the VPN server and typically log in with that account. To access the VPN, the user creates a new network connection account (for example **Start > Control Panel > Network Connections** on a Windows computer), then connects.

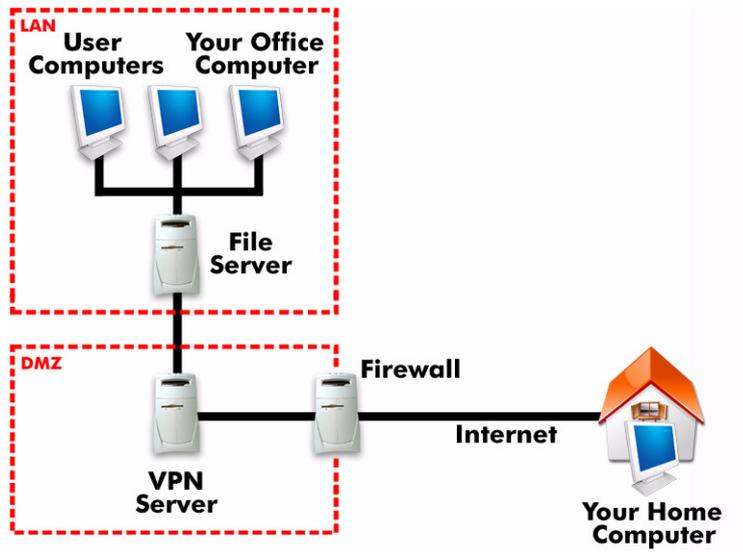


Figure 1: VPN Server Setup

Using an Apache Proxy Server

When the majority of your users use Scalix Web Access/Webmail, another good method to secure client access to an internal Scalix system is through an Apache Web server in the DMZ, which activates HTTPS and serves as a proxy or gateway to the system.

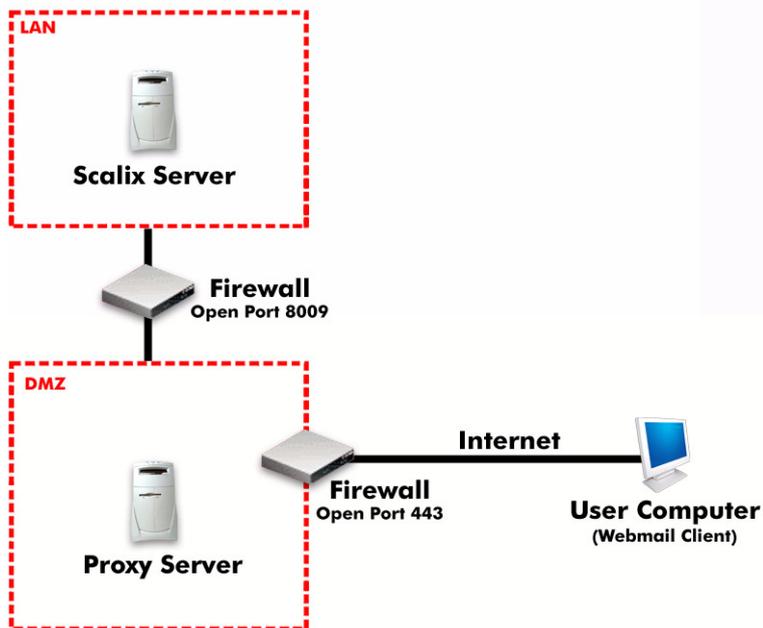


Figure 2: Using a Proxy Server

There are four steps to set up and integrate an Apache proxy server with Scalix:

- Install Scalix as documented in the *Scalix Installation Guide*
- Install and configure the Apache Web server in the DMZ
- Install the Scalix/Apache Tomcat Connector on the Apache Web server to facilitate communication between Tomcat on the Scalix server and Apache in the DMZ
- Copy the scalix-tomcat files from the Scalix server to the Apache server and replace the virtual host entry

Because you already have installed Scalix at this point, and because Apache is bundled with any Linux server, the following instructions begin with Apache configuration.

Configuring the Apache Web Server for use with Scalix

Because Scalix Web Access and Scalix Management Console (SAC) exchange data and credentials with the Scalix Server without encryption, Scalix recommends activating SSL security.

IMPORTANT: Name-Based Virtual Hosts and SSL

It is not possible to run multiple SSL-enabled virtual hosts on a server with only one Internet protocol (IP) address. Users connecting to such a setup receive a warning message stating that the certificate does not match the server name every time they access the URL. A separate IP address or port is necessary for every SSL-enabled domain to achieve communication based on a valid SSL certificate. Despite the warning message, you still get the same level of encryption that you have on any valid SSL site. This means that as long as the warning message is acceptable, communication between the Web server and client is secure. The concept of uniquely knowing the server's identity, which is guaranteed by a valid SSL certificate, is forfeited.

Setting up SSL for SUSE Linux

The process of setting up SSL for SUSE Linux starts by activating mod_ssl by means of Yast.

To create a key and self-signed certificate

- 1 Log in to Scalix as root.
- 2 Start Yast.
- 3 Navigate to **Network Services > HTTP Server**.
- 4 Verify that **Disabled** is selected. (Apache2 will need to be started manually.)
- 5 Select **Modules** and click **Edit**.
- 6 Select **ssl** and click **Toggle Status**.
- 7 Click **OK**, then click **Finish**.
- 8 To create a test SSL certificate, enter these commands:


```
cd /usr/share/doc/packages/apache2
./certificate.sh
```
- 9 Follow the on-screen instructions to build the SSL certificate. The resulting certificate files reside in the directories */etc/apache2/ssl**

Completing the Process

To make a copy of the vhost-ssl.template

- 1 Log in to [name] as root.
- 2 Run these commands:

```
cd /etc/apache2/vhosts.d/
cp vhost-ssl.template vhost.conf
```

You now need to configure Apache to start with SSL by adding a flag directive to the Apache sysconfig file.

To configure Apache to start with SSL

- 1 Log in to Scalix as root.
- 2 Use your preferred editor and open this file:

```
/etc/sysconfig/apache2 +/APACHE_SERVER_FLAGS
```

- 3 Edit this line of code:

```
APACHE_SERVER_FLAGS=""
```

to match this example:

```
APACHE_SERVER_FLAGS="SSL"
```

- 4 Restart Apache:

```
rcapache2 restart
```

Tip

If you have enabled SuSEfirewall2, do not forget to open the firewall for SSL-enabled Apache on port 443. This can be done via YaST by navigating to Security and Users > Firewall > Allowed Services. Add HTTPS Server to the list of Allowed Services.

Setting up SSL for Red Hat Linux

To create a key and self-signed certificate

- 1 Log in to Scalix as root.
- 2 Run the following command to create your key:

```
openssl genrsa 1024 > /etc/httpd/conf/ssl.key/server.key
```

- 3 Run the following command to make sure the permissions are set correctly for the key file:

```
chmod go-rwx /etc/httpd/conf/ssl.key/server.key
umask 77 ; \
/usr/bin/openssl req -new -key /etc/httpd/conf/ssl.key/server.key \
-x509 -days 365 -out /etc/httpd/conf/ssl.crt/server.crt
```

- 4 After entering the passphrase, you are asked for more information. (You do not see a prompt if you created a key without a passphrase.)
- 5 After providing the correct information, a self-signed certificate is created in `/etc/httpd/conf/ssl.crt/server.crt`. Restart the secure server after generating the certificate.

```
service httpd restart
```

- 6 You can get a real certificate with global validity from vendors such as Thawte (www.thawte.com) or Verisign (www.verisign.com). They provide instructions for installing the certificate on Apache.

Installing Scalix Tomcat Connector

Because Tomcat handles communications with the Apache Web server, you install the Scalix Apache/Tomcat Connector component on the Apache server.

To set up an Apache proxy server in a DMZ

- 1 Install Scalix Apache/Tomcat Connector on the Apache server. See the *Scalix Installation Guide*.
- 2 Copy the contents of the following directory from the Scalix Server to the Apache server.
 - For SUSE Linux Enterprise Server 9 or Red Hat Enterprise Linux 4:


```
~/tomcat/jk/instance-$instance.conf
~/tomcat/jk/app-$instance.webmail.conf
~/tomcat/jk/workers.conf
```
 - For SUSE Linux Enterprise Server 10 or Fedora:


```
~/tomcat/ajp/instance-$instance.conf
~/tomcat/ajp/app-$instance.webmail.conf
```
- 3 On the Apache server, edit the following file to replace the VirtualHost entry with the hostname of the external server.
 - For SUSE Linux Enterprise Server 9 or Red Hat Enterprise Linux 4:


```
~/tomcat/connector/jk/instance-$instance.conf
```
 - For SUSE Linux Enterprise Server 10 or Fedora:


```
~/tomcat/connector/ajp/instances-$instance.conf
```
- 4 Restart Apache.
 - For Red Hat Enterprise Linux:


```
service httpd restart
```
 - For SUSE Linux Enterprise Server:


```
/etc init.d/apache2 restart
```

Opening Ports

The default port for SSL and TLS requests on the Web server side is 443. There is no conflict between Apache listening on port 80 and SSL/TLS-enabled Apache listening port 443. In fact, HTTP and HTTPS can be run with the same Apache instance. Usually one virtual host is used to dispatch requests to ports 80 and 443 to separate virtual servers.

To open ports

- 1 Open the following ports in the firewall between the Internet and the DMZ:
 - 443 - HTTPS
 - 80 - HTTPS (only if you want to tolerate the risk)

Alert

Scalix recommends closing port 80 to external traffic. The port must remain open internally as some Scalix services require it.

- 2 Open the following ports in the firewall between the DMZ and the internal network:
 - 8009 - AJP (the Apache/Tomcat connector)

Using stunnel

If VPNs do not work in your circumstances, you can use stunnel as a last resort for securing IMAP, POP3, or SMTP communications.

The stunnel program is a generic open-source SSL encryption wrapper that works on both client and server sides. It can be used to add SSL functionality to commonly used POP3 and IMAP clients, such as Microsoft Outlook Express, Mozilla, or Apple Mail without any changes in the program's code. It supports standard SSL encryption with three levels of authentication.

Use stunnel for SSL for Microsoft Outlook.

The stunnel program protects against interception or manipulation of data by intermediate hosts. If compiled with libwrap support, it also protects against IP source routing, where a host can pretend that an IP packet comes from another, trusted host, as well as DNS spoofing, where an attacker forges name server records.

It does not protect against anything that compromises a host's security. Once an attacker gains root access to a machine, he can subvert stunnel too.

It is supported on UNIX/Linux and Windows, and it can be installed as a Windows Service.

On the server side, it can wrap any application based on port-forwarding, inetd integration, or command execution.

On the client side, it generally uses port-forwarding.

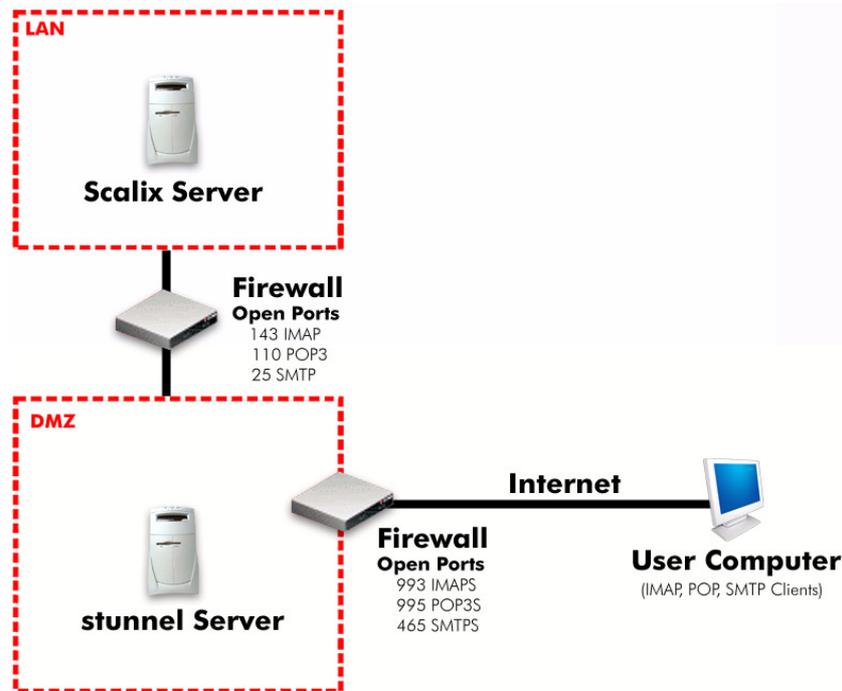


Figure 3: stunnel

Installing stunnel

The stunnel program is available as an RPM that is distributed with Red Hat. It requires OpenSSL for encryption handling.

To install stunnel

- 1 Install the RPM file.
- 2 Generate certificate(s). Do not use the certificate provided with stunnel/Red Hat installation; it is not secure.
- 3 Configure application integration.
- 4 Ensure stunnel is started at boot time.

Setting up stunnel for use with Scalix

stunnel can be set up for use with Scalix.

To set up stunnel

- 1 Access the `/usr/share/doc/stunnel-<versionnumber>/` folder.
- 2 Copy the `stunnel.conf-sample` file to the `/etc/stunnel/stunnel.conf` folder (with that file name).
- 3 Edit `/etc/stunnel/stunnel.conf` file as follows. This example includes the UAL-S service for Microsoft Outlook over SSL.

```

scalixmail:~ # cat /etc/stunnel/stunnel.conf
# Sample stunnel configuration file

# client = yes | no
# client mode (remote service uses SSL)
# default: no (server mode)
client = no

#
# chroot + user (comment out to disable)
#
chroot = /var/lib/stunnel/
setuid = stunnel
setgid = nogroup
# note about the chroot feature and the "exec" keyword to start other
# services...while the init script /etc/init.d/stunnel will copy the
# binaries and libraries into the chroot jail, more files might be needed
# in the jail (configuration files etc.)

pid = /var/run/stunnel.pid

#
# debugging
#
#debug = 7
#output = stunnel.log

#
# Some performance tunings
#
# disable Nagle algorithm (a.k.a. tinygram prevention, see man 7 tcp)
socket = l:TCP_NODELAY=1
socket = r:TCP_NODELAY=1
#compression = rle

# Authentication stuff
#verify = 2
# Don't forget to c_rehash CApath; CApath is located inside chroot jail:
#CApath = /certs
# It's often easier to use CAfile:
#CAfile = /etc/stunnel/certs.pem
# Don't forget to c_rehash CRLpath; CRLpath is located inside chroot
# jail:
#CRLpath = /crls

```

```

# Alternatively you can use CRLfile:
#CRLfile = /etc/stunnel/crls.pem

cert = /etc/stunnel/scalixmail.xandros.com.pem

#
# Examples for service-level configuration:
#

# [pop3s]
# accept = 995
# connect = 110

[imaps]
accept = 993
connect = 143

[ssmtp]
accept = 465
connect = scalixmail.ottawa.xandros.ca:25

[ua1s]
accept = 5767
connect = scalixmail.ottawa.xandros.ca:5729

```

where the name of the Scalix server is scalixmail.ottawa.xandros.ca. This configures the redirection from the secure ports to the non-secure ports. You normally do not open the non-secure ports through the firewall. Because the Scalix server is accepting incoming Internet mail, port 25 is open.

- 4 Create a certificate to encrypt the traffic. You can use the openssl Makefile.

```

cd /usr/share/ssl/certs
make /etc/stunnel/stunnel.pem

```

Enter the requested information but make sure that the Common Name is entered as the hostname that users will be connecting to otherwise they will see certificate errors.

- 5 Ensure that the stunnel certificate contains the correct information. */etc/stunnel/stunnel.conf* has the following setting:

```
cert = /etc/stunnel/stunnel.pem
```

- 6 An init script is necessary to start stunnel on boot:

```

vi /etc/init.d/stunnel
then

```

```

#!/bin/bash
#
# stunnel      This shell script takes care of starting and stop-
#              ping stunnel
#
# chkconfig: 345 80 30
# description:  Secure tunnel

# processname: stunnel
# config: /etc/stunnel/stunnel.conf
# pidfile: /var/run/stunnel/stunnel.pid

# Source function library.
. /etc/rc.d/init.d/functions

# Source stunnel configuration.
if [ -f /etc/sysconfig/stunnel ] ; then
    . /etc/sysconfig/stunnel
fi

RETVAL=0
prog="stunnel"

start() {
    # Start daemons.

    echo -n "Starting $prog: "
    if test -x /usr/sbin/stunnel ; then
        /usr/sbin/stunnel
    fi
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/stunnel
    return $RETVAL
}

stop() {
    # Stop daemons.
    echo -n "Shutting down $prog: "
    killproc stunnel
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/stunnel
}

```

```

    return $RETVAL
}

# See how we were called.
case "$1" in
  start)
    start
    ;;
  stop)
    stop
    ;;
  restart)
    stop
    start
    RETVAL=$?
    ;;
  condrestart)
    if [ -f /var/lock/subsys/stunnel ]; then
      stop
      start
      RETVAL=$?
    fi
    ;;
  status)
    status stunnel
    RETVAL=$?
    ;;
  *)
    echo $"Usage: $0 {start|stop|restart|condrestart|status}"
    exit 1
esac

exit $RETVAL

```

- 7 Now make sure that it will be run at the correct runtime:

```

chmod 755 /etc/init.d/stunnel
chkconfig --add stunnel
chkconfig --level 345 stunnel on
chkconfig --list | grep stunnel
/etc/init.d/stunnel start
/etc/init.d/stunnel status

```

It is up and running in runlevels 3, 4, and 5.

Configuring stunnel for POP3, IMAP, and LDAP

Because POP, IMAP, and LDAP are implemented as a Scalix service or daemon, wrapping based on command line or inetd is not supported. Instead, use port forwarding mode.

This is done by putting a “service” section into stunnel.conf, for example for POP3:

```
[pop3s] # freely chosen service name
accept  = 995 # standard POP3S port number
connect = 110 # standard POP3 port number
```

The drawback is that the standard ports remain available.

Securing SMTP

SMTP can be secured using an stunnel wrapper just as with any other service. The standard port number for SMTPS is 465.

However, normally the same SMTP server (sendmail or Scalix SMTP relay) is used both for incoming traffic from other domains and for Standard Client Mail submission.

As MTA-MTA traffic will never run over SSL, non-secure SMTP still has to be allowed.

As Sendmail supports TLS, both secure and insecure traffic can be handled over the same port in a well-defined way.

Using Certificates

A certificate provides security for e-mail. It is used to authenticate the e-mail server so that the user knows the correct site is being accessed. A certificate can be untrusted or trusted:

- Untrusted – The information of a certificate supplied by the company that created it is not verified by a third party. The user sees a message in a Web browser that the page is not from a trusted source, but they can still accept access. No verification by a third party is done so it is an untrusted certificate. An untrusted certificate is free.
- Trusted – The information within a certificate supplied by the company is verified by a third party, meaning a service provider such as Entrust or Verisign. This certificate is used to verify that the company and site being accessed are legitimate. The certificate is checked when the site is accessed and the user is given the opportunity to accept or reject it. A trusted certificate is purchased, but can also be obtained free from cacert.org. This option is typically used to secure e-mail.

A trusted certificate works as follows. You purchase it from a provider, such as Entrust or VeriSign, for a period, such as one year. You download the certificate from the provider and install it on the Scalix server. When a user accesses a Web page with the certificate applied, the server offers the signed certificate for client verification. The Web browser creates a session that is encrypted by using the public key found in the certificate offered by the server. The Web browser checks the certificate expiry date, that the certificate is issued by a trusted certificate authority, that certificate data has not been altered, and that the certificate is being used for the right site. When the certificate passes the verification stage, the user is prompted to accept the certificate or to reject it. When the server receives information it decrypts the data using its own private key, which is usually stored unencrypted on the server. The Web browser destroys the session once the user leaves the site or closes the application.

For both trusted and untrusted sites, users see a lock in the Web browser in the address bar and at the bottom of the browser, and the address of the site starts with https://

A certificate is also referred to as a Web certificate or Secure Sockets Layer (SSL) certificate.

Using a Certificate with Scalix

Root certificates can be imported for the Connector, where a root certificate is defined as a certificate that can sign other certificates. You can obtain a certificate from cacert.org if you do not want to pay for one from a commercial source. That Web site has lots of information about importing certificates into various browsers and a little about using it with SSL.

The easiest thing to do is to get a server certificate from cacert.org and then append the root certificates (both the Class 1 PKI key and the Class 3 PKI key) from the page on the “Root Certificate” link on the rh-nav of cacert.org’s home page. You can append either the PEM format or the text format certificates to the certificate bundle.

The IMAP server supports plain text login as well as Simple Authentication and Security Layer (SASL) mechanisms supported by the cyrus-sasl-2 library. The LOGIN, PLAIN, CRAM-MD5, DIGEST-MD5, and GSSAPI authentication mechanisms are fully supported; additional SASL mechanisms can be added by creating a symbolic link to the cyrus-sasl-2 module implementing the mechanism from the `/opt/scalix/lib/security` folder on the server.

Other Forms of Security

You can employ other means of securing your system, including “hardening” it or installing SSH tunnel. See the Scalix Wiki at <http://www.scalix.com/wiki>

Advanced Setup and Configuration

The remaining chapters involve more advanced setup and configuration tasks, such as configuring routing between servers, integrating with external directories such as Active Directory or LDAP, and setting up multiserver environments.

Section Contents

This section includes the following chapters:

- “Integrating with Active Directory” on page 78
- “Integrating with an LDAP Directory” on page 92
- “Multiserver Environments” on page 96
- “Localizing Scalix” on page 110
- “Hosting” on page 116

Integrating with Active Directory

This chapter outlines ways to integrate Scalix with Microsoft Active Directory.

Contents

This chapter includes the following information:

- “Integrating with Active Directory” on page 78
- “Installing the Schema Extensions” on page 79
- “Installing the GUI Extensions” on page 80
- “Setting Up Synchronization Agreements” on page 81
- “Using Active Directory to Manage Scalix Mailboxes and Groups” on page 85
- “Scalix Active Directory Extensions” on page 90

Integrating with Active Directory

If you want to manage some or all of your Scalix accounts with Microsoft Active Directory, you can do so after installing a series of Scalix schema and graphical user interface (GUI) extensions.

The tasks to integrate Scalix with Active Directory are:

- Install and run the application known as ScalixForestPrep to add schema extensions to Active Directory
- Install the Scalix Active Directory GUI extensions on every administrative workstation running Active Directory
- Create and test a synchronization agreement between Scalix and Active Directory, then schedule a regularly-occurring synchronization. For this, you use the `omldapsync` command.
- [Optional] Activate authentication among Scalix, Active Directory, and your Kerberos-based security system

Each of these tasks is outlined in the following sections. When done, you can use Active Directory to create and manage Scalix users.

Installing the Schema Extensions

The first step in integrating Scalix and Active Directory is installing the Scalix schema extensions to Active Directory. This extends the Active Directory schema with new Scalix-specific object classes and attributes that allow you to remotely manage your Scalix-based users and groups with Active Directory.

As part of this procedure, you install the ScalixForestPrep application, then run the application to install the extensions.

Any errors are logged into the Event Viewer, providing you with a permanent record in case of Scalix/Active Directory problems that you suspect are related to the extensions.

ScalixForestPrep finds which domain controller is functioning as the Schema Master. It then attempts to apply all extensions to this domain controller.

The extensions are outlined at the end of the chapter in “Scalix Active Directory Extensions” on page 90.

Alert

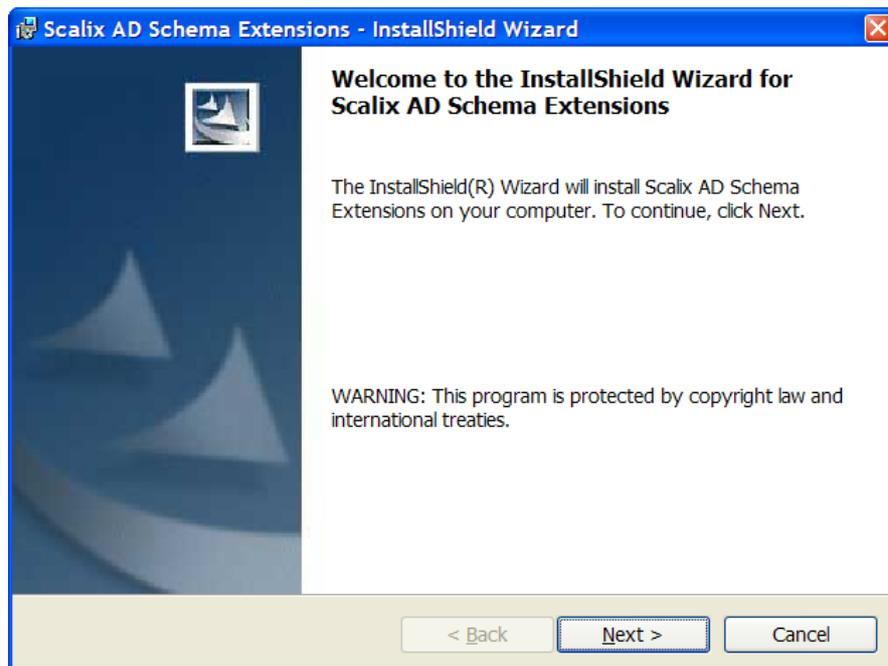
Adding extensions to Active Directory is irreversible.

To install the ScalixForestPrep exe file and Active Directory extensions

- 1 At the Windows computer and using an administrator account with schema administrator rights, log in to the host computer on which the schema master is stored. Or log in to a workstation with access to the schema master host.
- 2 Start the Scalix Active Directory Extensions installer. It is located in the *software/scalix_active_directory_extensions* folder on the Scalix server. Or contact Scalix for it.

Scalix AD Schema Extensions.msi

At the windows computer, clicking the file launches an installation wizard.



- 3 Work through the installation wizard. When installation is complete, ScalixForestPrep is located in the directory:

```
c:\Program Files\Scalix\Administration\
```

- 4 Run the ScalixForestPrep application to install the extensions:

```
ScalixForestPrep.exe --install
```

- 5 When the installation is successful, an “update successful” message appears in the window.
- 6 Exit the terminal window.

Note

You can run the application ScalixForestPrep.exe with several parameters for different results. With --install, it installs the extensions. With --register, it registers the GUI extensions. With --force, it forces the reinstallation of the schema changes. If you run it without any parameters, it only gives you the status.

Alert

The new schema extensions do not become active until the mandatory waiting period expires. The length of that waiting period depends on your setup, but is always at least five minutes. This interval is maintained so that all additions or changes will not upset current processes. Active Directory can take a long time to disseminate the new Scalix extensions through the system. Key factors include the number of domain controllers, the number of Active Directory servers, and connection speeds between network resources. Additionally, the older the Windows OS underneath Active Directory, the slower the full update; Windows 2000-based systems require a complete Active Directory database resynchronization while Windows 2003-based systems take less time to propagate changes. Your Active Directory system can update in minutes—or a weekend.

Installing the GUI Extensions

Now you update all Active Directory workstations with the Scalix Active Directory (ADUC) GUI enhancements, which install several Scalix tabs in the Active Directory Properties window. These options are relevant for users or groups on the Scalix system.

There are several ways to do this:

- Use Active Directory itself (via GPO) to propagate the GUI enhancements
- Distribute the Scalix installer for individual per-station installation, network-accessible file sharing or Web FTP
- Use a third-party utility to script a mass installation that installs the extensions when an administrator logs into the Active Directory server

To install the GUI extensions using the Scalix installer

- 1 For a first-time installation of the Scalix GUI enhancements on an ADUC workstation, log in as a Windows administrator.
- 2 If it is not already present, copy the “Scalix AD GUI Extensions.msi” file to the workstation desktop.
- 3 Start Scalix AD Extensions.msi to launch the installation wizard.

- 4 Work through the wizard.
- 5 Click **Finish** when the process is complete. ADUC is now ready for Scalix account management.

Setting Up Synchronization Agreements

Before the Scalix system can communicate with the Active Directory server, a custom synchronization agreement must be created and configured using the `omldapsync` command. Once this agreement has been tested successfully and run at least once, you can implement a cron job to automatically run synchronization on a regular basis.

Requirements for running synchronization agreements between Scalix and AD are:

- Log in to the Scalix server as root
- Have the domain names of the Active Directory and Scalix servers
- Have the domain name of the server with the Scalix Administration Server installed
- Have the authentication ID and related password for the Scalix administrator
- Have the administrator ID and related password for the Windows/Active Directory server

To prepare and test a new synchronization agreement

- 1 Log in to the Scalix Server as root.
- 2 To run synchronization in “interactive” mode, enter this command at the prompt:

```
omldapsync -i <syncid>
```

where `<syncid>` is a unique name for your Active Directory-Scalix synchronization agreement. Make the name no more than six alphanumeric characters in length; for example, `AD_SX1`.

After you press **Enter**, the synchronization “common tasks menu” appears, followed by a numbered list of interactive setup and administrative tasks.

- 3 When the `omldapsync` menu appears, enter “1” (one) at the prompt, and press **Enter**.
The `omldapsync` command creates the subdirectory for the newly named synchronization agreement along with the `<agreement_name>.cfg` file.
- 4 At the next prompt, you are asked to select the synchronization agreement type. Enter “11” (eleven) at the prompt and press **Enter**.

```
select sync agreement type to create (00):
```

- 5 The first of a series of interactive configuration prompts now appears:

```
INPUT: value for SCALIXHIDEUSERENTRY (scalixHideUserEntry):
```

Press Enter to accept the default value for this prompt and for all of the following value queries, listed below:

```
INPUT: value for SCALIXHIDEUSERENTRY (scalixHideUserEntry):
```

```
INPUT: value for SCALIXMAILBOXCLASS (scalixMailboxClass):
```

```
INPUT: value for SCALIXLIMITMAILBOXSIZE (scalixLimitMailboxSize):
```

INPUT: value for SCALIXLIMITOUTBOUNDMAIL (scalixLimitOutboundMail):

INPUT: value for SCALIXLIMITINBOUNDMAIL (scalixLimitInboundMail):

INPUT: value for SCALIXLIMITNOTIFYUSER (scalixLimitNotifyUser):

INPUT: value for EX_SCALIX_MAILBOX (scalixScalixObject):

INPUT: value for EX_SCALIX_MAILNODE (scalixMailnode):

INPUT: value for EX_SCALIX_MSGLANG (scalixServerLanguage):

INPUT: value for EX_SCALIX_ADMIN (scalixAdministrator):

INPUT: value for EX_SCALIX_MBOXADMIN (scalixMailboxAdministrator):

- 6 When the following prompt appears:

 Edit config file now y/n (n):

type “y” for yes.

- 7 When the following prompt appears:

 Use vi to edit y/n (n):

type “n” to be guided through an interactive session, in which you can efficiently enter the configuration settings. (Another option is to press “y”, and use VI to edit the configuration file manually, which is not documented here.)

The rest of this procedure details the interactive sequence of queries.

- 8 The first configuration prompt (JAVA_HOME) asks for the location of the Java installation on the Scalix server. Enter the full pathway for the Java directory.

- 9 The next prompt (EX_HOST) asks for the remote LDAP server name. Enter the name of your Active Directory server.

- 10 The next prompt (EX_LOGON) asks for the Active Directory administrator account name. The format for your entry is:

 cn=administrator,dc=organization,dc=com

- 11 The next prompt (EX_PASS) asks for the related Active Directory Admin password. Be sure to enter a password, so that the synchronization can be fully automated.

- 12 The next prompt (IM_HOST) asks for the fully qualified domain name (FQDN) of the Scalix server on which the directory is to be stored. If you have one server, enter that domain name. If you have several servers in your Scalix system, enter the FQDN of the server on which Scalix Administration Server is running. The format is

 server_name.domain.com

- 13 The next prompt (IM_CAA_URL) asks for the URL of the Scalix server on which Administration Server is running. If you have one server, enter that URL. The format is

 http://<your_scalix_mailserver_FQDN>:8080/caa/

Be sure to end the address with a slash, as shown.

Note

If you are setting up synchronization on a Scalix server running v10 of Scalix, enter an address without the 8080 port number: http://<your_scalix_mailserver_FQDN>/caa/

- 14 When the next prompt (IM_CAA_KEYSTORE) appears, press **Enter** to accept the default of no entry.
- 15 When the next prompt (IM_CAA_ID) appears, enter the authentication ID for a full Scalix administrator. The authentication ID is separate from the administrator's mailing address or display name.
- 16 When the next prompt (IM_CAA_PASS) appears, type the password associated with the Scalix administrator authentication ID.

Note

Ideally, you will already have verified the usability of this authentication ID and password by logging into Scalix with Scalix Administrative Console using this administrator account.

- 17 When the next prompt appears (EX_BASEn) (with "n" being replaced by a number), enter the container name and its full LDAP suffix, as shown here:

```
EX_BASE1:cn=users,dc=scalix,dc=com
```

If needed, you can list up to nine sequentially numbered containers at this time, if used for Scalix users and groups on Active Directory.

- 18 When the next prompt (EX_SCALIX_MAILNODE) appears, enter the mailnode in this format:

```
EX_SCALIX_MAILNODE=scalixMailNode
```

This query completes your omltdapsync synchronization configuration. Next, proceed through testing and use of the omltdapsync agreement.

- 19 When this prompt appears:

```
Compare old config to new y/n (?):
```

type "y" for yes.

omltdapsync displays a summary of this new configuration on-screen.

- 20 When this prompt appears:

```
Replace old config with new y/n (?):
```

type "y" for yes.

A series of status messages now appear, noting that the updated file was installed.

- 21 When this prompt appears:

```
Attempt to test data extraction now y/n (n):
```

type "y" for yes.

Omltdapsync now initiates a non-destructive test of the synchronization communication parameters. No user data is downloaded from Active Directory to Scalix at this time. A series of status messages appears, as omltdapsync contacts both servers and establishes the connection.

- 22 If the test succeeds, this message appears:

```
[DATE TIME] STATUS: Configuration of [AGREEMENT_NAME] completed
```

If the test fails, you want to edit the configuration file to correct the problem entry, then re-test the data extraction.

The “configuration completed” message is followed by the omldapsync interactive menu.

- 23 Press “2” to start loading all the Active Directory-specific users in a Scalix directory.

After the synchronization is initiated, a series of status messages report the success of various synchronization actions: new users added, users deleted, new limits applied, and so on. Review this list for the “entries failed” counts in each category.

- 24 If the download is unsuccessful, you can see a direction to a log file, a SOAP failure report with details, or a prompt to run an omldap utility to help you fix the problems, after which you restart the users download again.

- 25 When the loading is complete, another series of status messages concludes with:

```
LDAP dir sync export [AGREEMENT_NAME] completed
```

If the synchronization is successful, your Scalix server now hosts a set of users and groups managed by Active Directory.

- 26 Now set up a cron job to run this omldapsync agreement at the regular time intervals of your choosing.

Alert

This newly configured Active Directory/Scalix synchronization is uni-directional; Active Directory records are downloaded to Scalix. This means that you can use Scalix utilities to fully manage Scalix-generated user and group records, but you should only use Active Directory to manage all your Active Directory-generated/controlled records. Changes made with other utilities will be erased in the next synchronization.

Manually Running Synchronization Agreements

To manually run synchronization agreements

- 1 To manually run a synchronization agreement at any time, log in to Scalix, then enter this command:

```
omldapsync -u [AGREEMENT_NAME]
```

- 2 The Active Directory directory downloads to Scalix, and when finished, a series of status messages ends with this line:

```
LDAP dir sync export [AGREEMENT_NAME] completed\
```

Using Active Directory to Manage Scalix Mailboxes and Groups

Once Scalix and Active Directory are integrated, you can manage your Scalix users and groups in the same way you manage Microsoft users except for the following procedures, which are unique to the Scalix system and handled through the Scalix extensions:

- Adding and removing Scalix mailboxes
- Setting mailbox types (Premium and Standard users)
- Assigning mailnodes
- Setting mailbox limits
- Establishing message language
- Granting administrative access
- Hiding user entries

Alert

Deletion of users and groups totally erases all records and user data, such as e-mail.

You can use the Scalix command-line interface to open and change Active-Directory-specific records on the Scalix server, but any changes you make are overwritten in the next Active Directory/Scalix synchronization. Remember, you can use Scalix utilities to fully manage Scalix-generated user and group records, but you should use Active Directory to manage all your Active-Directory-controlled records.

Managing Mailnodes and Email Domains in Active Directory

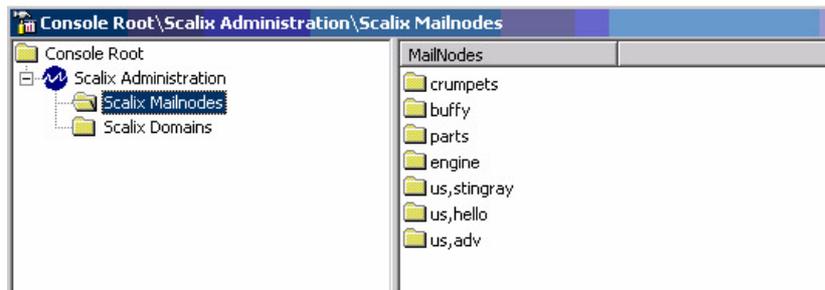
The first step after integrating Active Directory with Scalix is to identify your mailnodes and e-mail domains.

Mailnodes are used to organize your mail user community into manageable groups; for example, by work group, department, or employment status. Each Scalix server is associated with a single mailnode, which is created during installation, but they can be subdivided into multiple subnodes later if needed.

E-mail domains must be entered again now because Scalix has to verify their licensing.

To manage Scalix mailnodes and domains in Active Directory

- 1 Log in to the domain with administrator privileges.
- 2 Launch the Microsoft Management Console and using the Add/Remove Snap-in feature, add the Scalix Management Console snap-in.
- 3 You see a hierarchy on the left with two nodes:
 - Scalix Mailnodes
 - Scalix Domains



- 4 To add a mailnode, you have two options:
 - To add a single mailnode – Right-click **Scalix Mailnodes** and select **Add**. Type the name of the node in the window.
 - To add multiple mailnodes – Right-click **Scalix Mailnodes** and select **Import**. Browse to the location of the file containing the list of mailnodes.
- 5 To add a domain, you have two options:
 - To add a single mailnode – Right-click **Scalix Domains** and select **Add**. Type the name of the node in the window.
 - To add multiple mailnodes – Right-click **Scalix Domains** and select **Import**. Browse to the location of the file containing the list of mailnodes.

Creating New Scalix Mailboxes and Groups within Active Directory

Once you have integrated Active Directory and Scalix, the existing Active Directory “New User” and “New Group” wizards offer additional screens that allow you to create mailboxes for use with Scalix.

To create a new user or group

- 1 Using either the context menu or the menu bar icons, create a new user or group as you normally do.
- 2 Advance through the New User or New Group wizard. When the Scalix screen appears, fill in the fields as outlined here. In most cases, the screens pre-populate with the required information.
 - **Create a Scalix Mailbox** – Checked
 - **Home Mailnode** – Prepopulated
 - **Email Address** – Select whether you want an auto-generated address or to create addresses manually
 - **Mailbox Type** – The choices are Premium, Standard, or Internet. An Internet user does not have a local mailbox. Only Premium users have the following features:
 - Microsoft Outlook and Evolution support
 - Group scheduling functionality, including free/busy lookup in Microsoft Outlook, Scalix Web Access, and Evolution clients
 - Wireless e-mail and PIM
 - Access to public folders

- Personal folder sharing
 - Delegate access
- 3 When satisfied that all is correct, click **Next**.
 - 4 On the summary page, review the Scalix options you selected and click **Finish**.

Adding a Mailbox to an Existing User

If a user account exists in the system but does not have a mailbox, you can add it.

To add a mailbox to an existing user account

- 1 In Active Directory, select the user account.
- 2 Right-click the user account and select **Create Scalix Mailbox**.
- 3 A window opens with the same fields as outlined earlier. Fill in or change those fields as needed.

The screenshot shows a 'New Object - User' dialog box. At the top, it says 'Create in: eng.scalix.local/SX US/Engineering'. Below that is a checked checkbox 'Create a Scalix Mailbox'. Underneath is a text field 'Home Mailnode:' with the value 'us.adv'. A section titled 'Email Address:' contains two radio buttons: 'Use server generated email addresses (Recommended)' (which is selected) and 'Specify email addresses manually'. Below the radio buttons is an empty text box and three buttons: 'Add Email', 'Edit email', and 'Remove'. At the bottom of the dialog is a 'Mailbox Type:' dropdown menu set to 'Scalix Premium user'. At the very bottom are three buttons: '< Back', 'Next >', and 'Cancel'.

- 4 When finished, click **OK**.

Removing a Scalix Mailbox

You can delete a user's or group's Scalix mailbox and all its contents, while retaining the user account or group in Active Directory. For example, you perform this task after migrating the Scalix mailbox to another, separate server.

In the next synchronization, the mailbox and its contents are deleted. The Active Directory account remains for other uses.

To remove a Scalix mailbox

- 1 In Active Directory, right-click the user account and select **Remove Scalix Mailbox**.
- 2 Click **Yes** to verify the deletion.

Modifying a Scalix Mailbox

Once created in Active Directory, you can modify user accounts and groups if needed. The Active Directory Properties window includes two new tabs, **Scalix General** and **Scalix Advanced**, that allow you to modify the following mailbox properties:

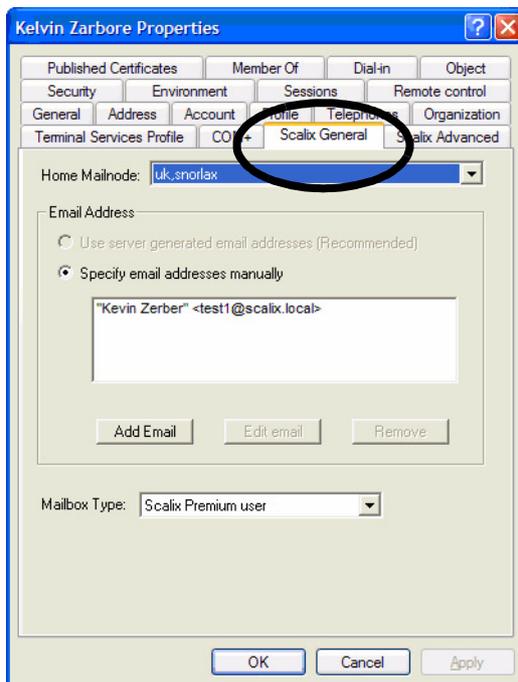
- Mailnode
- E-mail address
- Mailbox type (Premium, Standard, or Internet)
- Server language
- Mailbox size limits and warning settings
- Administrative access
- Hide user

A user's mailbox size is unlimited by default, though you typically set it to be between 100 MB and 1 GB on the server.

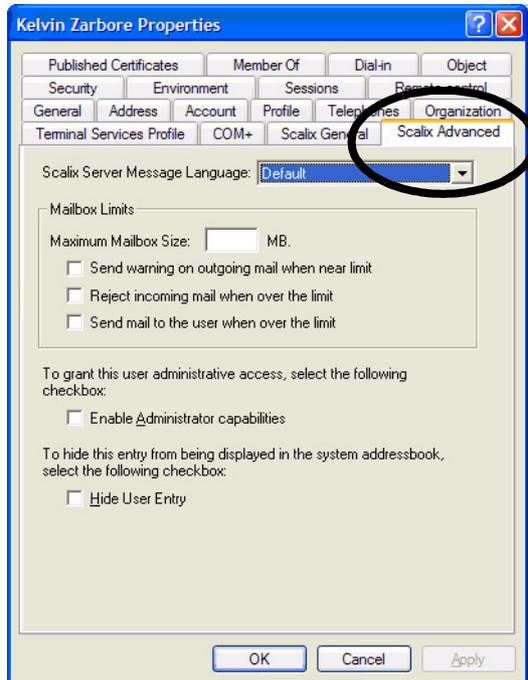
When working with users, both tabs appear in the Properties window. For groups, the Scalix General tab appears.

To modify a user's Scalix attributes

- 1 In Active Directory, right-click the user account and select **Properties**.
- 2 To change the mailnode, address, or mailbox type, click the **Scalix General** tab.



- 3 Modify the fields.
- 4 To change the language, mailbox size limits, administrative access level, or display status, select the **Scalix Advanced** tab.



- 5 In the **Scalix Server Message Language** field, select the option from the drop-down list.
- 6 Under **Mailbox Limits**, type the maximum size (in MB) of the mailbox and check the boxes next to the action you want the server to take if the user nears or exceeds the maximum.
- 7 To grant the user administrator rights, click the check box next to **Enable Administrator Capabilities**.
- 8 To hide this entry from the system address book, click the check box next to **Hide User Entry**.
- 9 Click **OK**.

Scalix Active Directory Extensions

Scalix has the OID root of 1.3.6.1.4.1.19049, and all of the following extensions are appended to it. The first table shows the extensions that match the options in the Scalix Server tab (especially users).

Table 1: Active Directory Extensions and their Definitions

Extension	Definition
[1.1.10] scalixScalixObject	True if this is an object managed by Scalix.
[1.1.11] scalixMailnode	The mailnode that is hosting this object.
[1.1.12] scalixAdministrator	True if this user has general administrator capabilities.
[1.1.13] scalixMailboxAdministrator	True if this user has mailbox administrator capabilities.
[1.1.14] scalixServerLanguage	The language for server-to-client communications.
[1.1.15] scalixEmailAddress	A multivalued list of e-mail addresses for this mailbox.
[1.1.16] scalixLimitMailboxSize	The maximum size of the mailbox in MB -- 0 to use server default, which is unlimited. Typical values are between 100 MB and 1 GB.
[1.1.17] scalixLimitOutboundMail	True if Scalix to warn when near limit on outbound mail.
[1.1.18] scalixLimitInboundMail	TRUE if Scalix to reject inbound mail upon limit reached.
[1.1.19] scalixLimitNotifyUser	TRUE if Scalix to notify user when limit is reached.
[1.1.20] scalixHideUserEntry	TRUE if this directory entry is to be hidden from the CDA.
[1.1.21] scalixMailboxClass	Set to "full" or "limited" to control class, or leave it blank for default.

The following table shows the Scalix object classes that extend the Active Directory OpenLDAP schema.

Table 2: Object Classes and their Definitions

Directory Object Class	Definition
[1.2.10.23] scalixUserClass	Auxiliary class of attributes to extend the User class.
[1.2.11.24] scalixGroupClass	Auxiliary class of attributes to extend the Group class.

Integrating with an LDAP Directory

This chapter outlines ways to integrate Scalix with a Lightweight Directory Access Protocol (LDAP) directory.

Contents

This chapter includes the following information:

- “About the LDAP Server and Directories” on page 92
- “Configuring the LDAP Server” on page 93
- “Starting and Stopping the LDAP Server” on page 94
- “LDAP and Scalix Attribute Type Mappings” on page 94
- “LDAP Commands” on page 94

About the LDAP Server and Directories

This is a server and directories.

Server

The LDAP Server is a Scalix daemon process that provides an interface to enable LDAP clients to store and retrieve data from a Scalix directory without having any information about the operation of Scalix.

The LDAP directory service is based on a client-server model. The LDAP Server provides LDAP clients access to shared Scalix directories that do not have an associated password.

Scalix automatically enables search-only LDAP support. Consequently, there is minimal configuration required to enable LDAP client directory searches. The LDAP Server process (omslapd) starts when Scalix starts and runs until Scalix is shut down.

The LDAP directory is a hierarchical structure comprised of one Scalix directory containing structural information and one or more additional Scalix directories containing user and entity information. Using this structure, the LDAP Server provides a hierarchical view of a Scalix directory, enabling LDAP clients to access directory entries.

An entry is referenced by its Distinguished Name (DN), also known as a directory Distinguished Name (DDN), which is an unambiguous identifier for that entry. The DN is constructed from a Relative Distinguished Name (RDN).

Directories

The LDAP directory service model is based on entries. An entry is a collection of attributes that has a name, called a Distinguished Name (DN). The DN is used to refer to the entry unambiguously. Each of the entry's attributes has a type and one or more values. The types are strings, like "cn" for common name, or "mail" for e-mail address. The values depend on what type of attribute it is. For example, a mail attribute might contain the value JohnDoe@Acme.co.uk

While LDAP has a hierarchical structure, the Scalix directory has a flat structure. The Scalix directory is made up of a series of entries identifying a user (or entity) by attributes. Attributes include O/R Address attributes, personal and employment related attributes, and e-mail address attributes among others. The Scalix directory is a single, flat database. Entries are not grouped into any hierarchal structure.

Configuring the LDAP Server

The behavior of the LDAP Server is controlled by a number of configuration files that allow you to customize the operation of the LDAP Server.

Table 1: LDAP Server Configuration Files

File Name	Description
ldap.attrs	LDAP attribute mapping file. This file defines the mapping between LDAP attribute names and Scalix internal attribute names.
slapd.conf	LDAP Server configuration file. This file sets options that control the runtime behavior of the LDAP Server.
dit.cfg	Directory configuration file. This file specifies the name of the Scalix directory and the default DN suffix used by some of the Scalix directory commands.

To change an LDAP server configuration file

- 1 Access the configuration file located in the following folder:

```
/var/opt/scalix/<nn>/s/sys/
```

where nn varies with Scalix release.

- 2 Open the file in a text editor and change the configuration.
- 3 Restart the LDAP server:

```
omoff -a slapd
omon -a slapd
```

Starting and Stopping the LDAP Server

The LDAP Server process (omslapd) starts when Scalix starts and runs until Scalix is shut down. If required, you can stop the LDAP Server.

To stop the LDAP server

- 1 Enter the following command:

```
omoff -d delay -a slapd
```

where delay indicates the time in seconds to wait before stopping the daemon.

To start the LDAP daemon process

- 1 Enter the following command:

```
omon -a slapd
```

LDAP and Scalix Attribute Type Mappings

There are mappings that are automatically put into the omldapsync command. If you use the defaults, they are fine. If you want to do custom mappings, look on the Wiki or contact customer support.

LDAP Commands

The following table outlines LDAP commands.

Table 2: LDAP Commands

Command	Description
omldapadd	Add one or more entries to an LDAP directory.
omldapdelete	Delete one or more entries from an LDAP directory.
omldapmodify	Modify an LDAP directory entry.
omldapmoddn	Modify the DN of an LDAP entry.
omldapsearch	Search an LDAP directory.

Multiserver Environments

This chapter introduces multiserver environments, including how to set up a high availability system with failover, distribute roles among servers, set up their mail routing, synchronize their directories and public folders, and designate their trust relationships.

Contents

This chapter includes the following information:

- “Distributed Architecture” on page 96
- “Setting Up High Availability” on page 97
- “Routing Mail” on page 102
- “Synchronizing Directories” on page 103
- “Synchronizing Public Folders” on page 106
- “Configuring Outbound Internet Messages” on page 106
- “Server Trust Relationships” on page 107

Distributed Architecture

You can set up Scalix as a distributed system with multiple servers for backup, failover, scalability, geographical, or performance reasons. Before beginning, decide what role each server will take, their relationships, and how they interact. The key decisions you need to make include:

- Which server will host the Scalix Management Console (SAC). This machine also acts as the manager for all other servers.
- How to do directory and public folder synchronization. If the system only has two directories or two sets of public folders, they have an equal import/export relationship and synchronization with each other. But with three or more servers, you must appoint one directory as the master and have all others synchronize with it. This “master” directory can be the same computer as the main one but it does not have to be.
- How you want mail routing done. The possibilities are numerous and include:
 - Each server routes mail to the Internet
 - One server routes to the Internet, and all outbound messages go through it

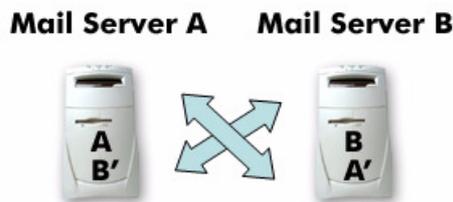
- All servers route to each other
- Which server will act as the gateway into the system. The gateway is the first point of contact for incoming mail.

Setting Up High Availability

With Scalix Enterprise Edition, you can set up a dual-server failover system to ensure high availability. In this scenario, server A fails over to Server B and vice-versa. Each machine has an active physical instance (A or B) and a virtual instance (A' or B') that takes the load if the other fails.

Both servers mount to a single shared storage solution and clustering software provided by Scalix at the time of installation is responsible for relocating the instances among the computers at time of failure. This includes automated unmounting and mounting of the shared storage solution and automated shutdown and startup of the necessary Scalix services.

Each instance needs to be a complete vertical stack, with the Scalix Server, Scalix DB (Postgres database), Scalix Management Agent, Scalix Tomcat, and Scalix Search and Index Service components installed. Each computer needs to have the same components installed, with the exception that Scalix Management Console (which is part of the Scalix Management Services component) is installed on one computer and not the other. Connections from either of the servers to the shared storage must be through direct means, such as a cable, Fibre Channel, or iSCSI. NFS is not recommended or supported at this time.



A fails over to B' and B fails over to A'

Figure 1: Clustered for High Availability

For this setup, you run the installation wizard twice on each computer, naming each of the two instances on Server A as <A> and <B'>, respectively, then on Server B, naming them and <A'>, respectively. The process is similar to the following:

- Physical server with Internet protocol (IP) address 10.17.96.55 = mail1
Physical server with IP address 10.17.96.56 = mail2
Virtual server IP address 10.17.96.57 = virtual1
Virtual server IP address 10.17.96.58 = virtual2

The virtual instances will be accessed by the e-mail clients and referenced during configuration of the Scalix server software.

When installing the operating system, use the virtual hostname and IP address.

- Using the operating system's Domain Name System (DNS) configuration process, add the virtual instances to the DNS just as you would a physical server. The e-mail applications of end users access the virtual IP address and the DNS configuration enables this.
- Install Scalix on both servers. One has Scalix Management Console (which is part of the Scalix Management Services component) installed and the other does not.
- Disable automatic startup of services
- Change the name of the server back to the name of the physical server, still retaining the physical server's IP address
- Install and configure cluster software, which recognizes the physical servers as real servers, and the virtual instances as services. The cluster software can include Red Hat Cluster Suite, SUSE Heartbeat, or Veritas Cluster Server.

Sample Setup Instructions

The instructions are an example specific to Red Hat. If you have a different operating system, use them as guidelines.

Perform the steps on both servers.

To set up a dual-server failover cluster for Scalix

- 1 Build the two servers as if they are simply two servers within the same organization. When installing the operating system, use the virtual hostname and IP address.
- 2 Add the virtual instances to the DNS just as you would a physical server.

To add the virtual instance, launch the Red Hat Network Configuration window to change the name of the server from the name of the physical machine to the name of the virtual instance.

```
system-config-network
```

In the window that launches, click the DNS tab and enter the name of the virtual instance, then save and exit the Network Configuration window.

- 3 Edit the `/etc/hosts` file to enter the IP address of the physical server.

```
vi etc/hosts
ifconfig
```

Add the IP address of the physical server, followed by the virtual domain name and the virtual name. For example:

```
10.17.120.52    virtual1.scalixdemo.com    virtual1
```

Save the new information and exit vi.

- 4 Check to ensure that the physical server is functioning with the virtual IP address:

```
ifconfig
```

- 5 Install Scalix on both servers. One has the Scalix Management Console (Scalix Management Services) installed and the other does not.

Before moving on, check that Scalix is up and running.

```
omstat -s
```

- 6 Disable automatic startup of services. Stop the services first:

```
service ccsd stop
service rgmanager stop
service cman stop
service fenced stop
```

Disable Scalix from restarting upon reboot:

```
chkconfig --level 35 scalix off
chkconfig --level 35 scalix-tomcat off
chkconfig --level 35 scalix-postgres off
```

Alert

If you do not complete this step, you can corrupt your installation.

- 7 Reboot the server:

```
shutdown -r now
```

- 8 Verify that the physical server is now offline and the virtual service is up and running:

```
clustat
```

- 9 Rename the server back to the name of the physical server, still retaining the physical server's IP address. For instructions, see steps 2 and 3 above. In this case, you remove the entry in the `etc/hosts` file that you created in step 3, then save and exit `vi`.

```
vi /etc/hosts
```

- 10 Reboot the server again:

```
shutdown -r now
```

- 11 Restart the cluster:

```
service ccsd start
service rgmanager start
service cman start
service fenced start
```

- 12 Run another check to ensure that the physical server once again is functioning with the physical IP address:

```
ifconfig
```

- 13 Install and configure the cluster software, which recognizes the physical servers as real servers, and the virtual instances as services. Configure the cluster software to include the two physical machines, and then configure the virtual instance of Scalix with your cluster configuration software program. This typically consists of:

- An IP address
- A shared file system
- A script

- 14 On both computers, define the virtual instances to Scalix by editing the `/etc/opt/scalix/instance.cfg` file and changing the `OMNAME` parameter to the actual virtual machine name, then changing the `OMAUTOSTART` parameter to `FALSE`.

- 15 Relocate the information store to a shared directory so that it can switch back and forth between the two clustered servers:

```
cd ~/mail1
mkdir temp
mv /br* ./temp
```

- 16 Relocate the virtual instance to this machine.

```
clusvcadm -r <virtual hostname> -n <physical domain name>
```

- 17 Move everything from temp to the shared disk:

```
mv ./temp/* ./br
```

- 18 Edit the file `/opt/scalix/global/config` file to change the `OMAUTOSTART` value from `true` to `false`:

```
vi /opt/scalix/global/config
```

- 19 Restart the virtual instance:

```
service scalix start <virtual server hostname>
```

- 20 Telnet into the virtual instance to make sure it is running:

```
telnet <virtual server hostname> <port>
```

- 21 Stop Scalix:

```
service scalix stop <hostname>
```

- 22 Copy the configuration files from each machine to the other so that both have the same files:

```
cd /etc/opt/scalix-tomcat/connector/jk
scp *-<hostname 1>.* <hostname 2>:/etc/opt/scalix-tomcat/connector/jk
```

Then repeat, reversing the hostnames.

```
scp mail1:/etc/opt/scalix-tomcat/connector/jk/*-virtual1.*
```

- 23 Edit the `workers.conf` file to include both mailnodes of the virtual hosts. Do this on both physical hosts.

```
cd /etc/opt/scalix-tomcat/connector/jk/workers.conf
vi workers.conf
```

- 24 Go to the following directory to make the shutdown port 8006:

```
cd /var/opt/scalix/<nn>/tomcat/conf/server.xml
vi server.xml
```

Look for the following value:

```
Server port ="8005" shutdown="SHUTDOWN"
```

Change the port number from 8005 to 8006.

While there, also change the short name of the instance to the fully qualified domain name in the three places that it appears.

- 25 Repeat the entire sequence on the other virtual machine.

- 26 Change the server names in the following file to the virtual machine names.

```
cd /etc/httpd/conf.d
vi scalix-web-client.conf
```

At the bottom of the file, notice that the file refers to the physical directory instead of the virtual one. You are going to move these lines to one of the tomcat.conf files.

Select the bottom seven lines and copy then delete, them.

```
Alias      /omhtml/          /var/opt/scalix/n1/s/omhtml/ <Directory
"/var/opt/scalix/n1/s/omhtml">
    AllowOverride None
    Order allow,deny
    Allow from all
    AddDefaultCharset off
</Directory>
```

27 Open the instance-name1.conf file:

```
cd /etc/opt/scalix-tomcat/connector/jk/app-virtual1.*.conf
```

Paste these seven lines in, beginning with the third line. The first nine lines of the modified instance-name1.conf file should look as follows.

```
<VirtualHost virtual1.scalixdemo.com:80>
    Include /etc/opt/scalix-tomcat/connector/jk/app-virtual1.*.conf
Alias      /omhtml/          /var/opt/scalix/n1/s/omhtml/ <Directory "/
var/opt/scalix/n1/s/omhtml">
    AllowOverride None
    Order allow,deny
    Allow from all
    AddDefaultCharset off
</Directory>
```

Perform this step on both physical hosts.

Routing Mail

The first step in putting together a multiple-server environment is setting up mail routing between the different hosts.

There are many routing possibilities, from one extreme (every server routes in to one hub) to the other extreme (every server routes to all others).

On Scalix, routing works through the concept of mailnodes. All servers have mailnodes, and some have more than one. All messages have mailnode notations in their headers. Each server can read the mailnode information and those that are designated for routing, forward messages to their appropriate servers according to their mailnode header notations.

To add the routes between the servers

Note	The following steps use two servers, Server A and Server B as examples. Server A is known as serverA.domain.com and Server B is serverB.domain.com. Substitute your own values for these, for example scalix1.yourcompany.com and scalix2.yourcompany.com. In cases of more than two servers, establish the routing between two first, then repeat the procedure to set up the routing between two others, then so on until all routes are configured.
-------------	--

- 1 Run the following commands on each of the two servers.

On ServerA: `omaddrt -m serverB,mailnode -q SMINTFC -i scalix@serverB.domain.com`

On ServerB: `omaddrt -m serverA,mailnode -q SMINTFC -i scalix@serverA.domain.com`

- 2 On both computers, run the following command to set up the Scalix-to-Scalix transport gateway for sending messages between Scalix servers:

```
omoff -d 0 -w router; omon route
```

- 3 If you are using CNAME DNS records as your hostname (serverA.domain.com is really called something else), you must make some changes to Sendmail and the Scalix SMTP relay configuration before this works. This is because one of the first things Sendmail does is to rewrite outbound addresses to be the A DNS record rather than any CNAME.

To do this, edit the `~/sys/smtpd.cfg` file and set the following:

```
DOMAIN_NAME=real.host.name
LOCAL_NAMES=cname1.domain.com,cname2.domain.com
```

Where `DOMAIN_NAME` is the record name for server A and `LOCAL_NAMES` is a comma-separated list of CNAME record names for the server.

- 4 Stop and restart the SMTP relay:

```
omoff -d0 smtpd; -w;
omom smtpd
```

- 5 If you are using a smart host configuration, configure Sendmail to send directly to the other Scalix servers rather than going through the smart host. This is done by using the mailertable feature of Sendmail.

- 6 In the `/etc/mail` folder, edit the `mailertable` file and add the following line:

```
real.host.name<TAB>esmtpl: [real.host.name]
```

where `<TAB>` is a tab character.

This tells Sendmail that if any message is sent to `@real.host.name`, it is to use the `esmtpl` mailer to send it to `real.host.name`. The square brackets (the `[]` characters) surrounding the host name tell Sendmail not to use DNS to determine MX records.

- 7 To rebuild the mailertable lookup, go to the `/etc/mail` folder and run the following command:

```
make mailertable.db
```

- 8 To ensure that the Scalix rules are added back into the `sendmail.cf` file, run the following command:

```
omsendin
```

- 9 Restart the `sendmail` service.

- 10 Repeat as needed to establish all other routes, substituting server names as required.

Synchronizing Directories

When running multiple directories on different servers, they must be synchronized.

Instructions for synchronizing Lightweight Directory Access Protocol (LDAP) directories are contained in the LDAP integration chapter.

About Directory Synchronization

Synchronization allows you to automatically maintain consistent directory entries across a network. It ensures that whenever you add, modify, or delete an entry at its primary location, the change is applied to other directories throughout the network. Directory synchronization ensures the following:

- Directory entries are always up-to-date throughout the network
- Fewer messages are incorrectly addressed
- A minimal amount of time is required to maintain directories

You can synchronize directories with other Scalix directories or with directories in other mail systems, but you cannot synchronize with directories acting as X.500 directory access points (marked as X500 in the `omlistdirs` command).

A single system can contain more than one directory, and directory synchronization is established between directories (not necessarily between systems).

In a synchronization agreement between two Scalix directories, the importing server requests updates from the exporting server. The exporting server extracts updates from the relevant directory change log and gives the updates to the importing server, where they are applied to the import directory. This process automatically repeats at set intervals, with the importing server always initiating the exchange.

Often, two synchronization agreements are created between a pair of directories so that a bidirectional link is created. In this scenario, each directory acts as both an import and an export directory.

In cases of three or more directories, one directory must be designated as the master and all others synchronize with that one.

The import and export directory must be consistent in terms of:

- The names of the directories to be synchronized
- The addresses of the importing and exporting servers
- The frequency of the updates

For every local import agreement there must be an associated agreement on the exporting system.

Also, you can have to update the routing table on each system to provide routes to the new O/R Addresses that are made available through synchronization.

Note	Synchronization is not possible with directories acting as X.500 Directory access points (marked as X500 in the <code>omlistdirs</code> command).
-------------	---

Scalix recommends that you research and design the optimal network topology to use for synchronization. For a more in-depth look at directory synchronization, see the *Scalix Administration Guide*.

Directory Synchronization Commands

The table lists the commands for Scalix directory synchronization. For information on use, enter the command without arguments to display options or view its manpage.

Table 1: Directory Synchronization Commands

Command	Description
<code>omadds</code>	Add a directory synchronization agreement.
<code>omdels</code>	Delete a directory synchronization agreement.
<code>omlistds</code>	List directory synchronization agreements.
<code>ommodds</code>	Modify a directory synchronization agreement.
<code>omresyncds</code>	Resynchronize a directory.
<code>omshowds</code>	Show details of a directory synchronization agreement.

You create reciprocal export and import commands in order to perform synchronization. For example, you add the `-e` or `-i` option to the `omadds` command.

Creating Directory Synchronization Agreements

The following steps use two servers, Server A and Server B as examples. Server A is known as serverA.domain.com and Server B is serverB.domain.com. Substitute your values for these.

For this to work, you need an import agreement on one side and an export agreement on the other.

To add a directory synchronization agreement

- 1 Set up an import agreement on one server and an export agreement on the other by running the following commands:

On ServerA: `omadds -i -m +DIRSYNC/mailnodeB -t "010101 00:00"`

On ServerB: `omadds -e -m +DIRSYNC/mailnodeA`

where the `-t` option specifies when this agreement comes into effect. The date format is `yymmdd hh:mm`.

Another example is

```
omadds -e -m "+DIRSYNC/boston,factory,mis" -t "081013 23:00"
```

where `e` indicates export, `m` indicates the address of the exporting server, and its mailnode is `boston,factory,mis`. The process starts on 13 October 2008 at 2300 hours and repeats every 15 minutes. This exports data from the local system to the `boston,factory,mis` mailnode.

- 2 Program the synchronization process to make the update requests as soon as the service starts, rather than waiting for a timeout. To do this, add the following line to the file `~/sys/general.cfg` on both machines.

```
DS_CUST_SEND_REQ_NOW=TRUE
DS_CUST_MSGQ_TIMEOUT=2
```

- 3 Restart the `dirsync` service and enable auditing to see the messages transfer between machines:

```
omconfaud dirsync 15
omoff -d 0 -w dirsync ; omon dirsync
```

- 4 To check that mail is flowing correctly, review the messages in the directory `~/logs/audit`.
- 5 Repeat as needed until all directories have synchronization agreements.

Note

It is recommended to manually trigger a re-synchronisation if there occurred many changes in a short time period. For example, if you moved a lot of users from one to another mailnode or if significant PDL administration was performed. To force a re-synchronisation run `omresyncds`, as described later in this chapter.

To view directory synchronization agreements

- 1 Enter `omlistds` and indicate the type of agreements to list. For example

```
omlistds -e
```

lists any export agreements. The options are `-e`, `-i`, and `-n`.

To manually synchronize

- 1 Enter `omresyncds`, the option, and the name of the directory or the agreement number. For example

```
omresyncds -i 3
```

synchronizes agreement number three. The options are `-d` and `-i`.

Tip

See the man page for each command for more information and examples. For example, enter `man omresyncds`

To view logs

- 1 Access the following folder:

```
/var/opt/scalix/<nn>/s/dirchlog
```

Each change log consists of two text files and a lock file. New entries are written to the second text file. When the second text file expires (as specified by the `omaddds` or `omaddss` command), the contents of the second file are copied to the first text file.

- 2 To view the change log for a specific directory, open the following file:

```
/var/opt/scalix/<nn>/s/sys/dir.index
```

The third field in this file displays the name of the directory and the fifth field gives the prefix of the change log file names. The age of the change log is shown in the last field (in seconds).

Synchronizing Public Folders

When using public folders on different servers, the folders must be synchronized.

When you have multiple servers, mail routes need to be set up between them. Each server needs to know which hosts have which mailnodes before synchronizing of public folders can be set up; see “Routing Mail” on page 102.

For instructions to synchronize public folders, see that section in the *Scalix Administration Guide*.

Configuring Outbound Internet Messages

If you want to use one server as a bridgehead to the Internet, configure the other server(s) to route all Internet mail to the first.

To configure Server B to use Server A as a bridgehead

- 1 Run the following commands on Server B:

```
omoff -d 0 router
omdelrt -m internet
omdelrt -m internet,tnef
omaddrt -m internet -q SMINTFC -i scalix@serverA.domain.com
omaddrt -m internet,* -q SMINTFC -i scalix@serverA.domain.com
omon router
```

You need both the `omaddr` commands because you have the standard MIME route and also the TNEF route.

- If you do not want outbound mail to go through another Scalix server, leave your `sendmail.cf` configuration as is. If you have another edge (non-Scalix) server responsible for outbound routing, edit `sendmail.cf` as follows.

Replace

```
DS
```

with

```
DSother.host.name
```

- Restart the Sendmail service. This routes all non-local mail through to the named server.

- If you are using `sendmail.mc` for your configuration, replace:

```
dn1 define(`SMART_HOST', `smtp.your.provider')
```

with

```
define(`SMART_HOST', `other.host.name')
```

and run the following command in the `/etc/mail` directory:

```
make
```

- Restart the Sendmail service.

Alert

If you make any changes to the `sendmail.mc` file and run 'make', you must run the command `omsendin` to ensure that the Scalix rules are added back into the `sendmail.cf` file.

Server Trust Relationships

In multiserver setups, it is essential to establish server trust relationships to enable cross-server delegation, resource booking, and more.

When setting up server trust relationships, all servers must be set up to use a Kerberos server. To do that, you must set up the following Kerberos identities:

- For IMAP:** `imap/<Server 2 FQDN>@KERB.DOMAIN <mailto:<Server 2 FQDN>@KERB.DOMAIN>`
- For UAL:** `ual-scalix/<Server 1 FQDN>@KERB.DOMAIN <mailto:<Server 1 FQDN>@KERB.DOMAIN>`

Where `<Server 1 FQDN>` and `<Server 2 FQDN>` take the form of an FQDN such as `hostname.domain_name.com`. And `KERB.DOMAIN` must be in upper case letters.

To enable cross server booking or delegation in the opposite direction, simply reverse these directions.

To set up the Kerberos identities for server trust relationships

- 1 Add one of the following Kerberos principles to the keytab file on the first server.

For IMAP: `imap/<Server 2 FQDN>@KERB.DOMAIN <mailto:<Server 2 FQDN>@KERB.DOMAIN>`

For UAL: `<Server 2 FQDN> ual-scalix/<Server 1 FQDN>@KERB.DOMAIN <mailto:<Server 1 FQDN>@KERB.DOMAIN>`

- 2 Set up the `~/sys/trust` file on the first server to contain the following line:

```
imap/<Server 2 FQDN>@KERB.DOMAIN <mailto:<Server 2 FQDN>@KERB.DOMAIN>
ASUSER
```

- 3 Then telnet into the second server and perform the following steps.

Telnet to server 2:

```
<Server 2 FQDN> 143
```

You see a system response that looks something like this:

```
* OK Scalix IMAP server 9.3.0.10-alpha ready on two.example.com
```

Type the following:

```
1 login delegate pass
```

You see a system response that looks something like this:

```
1 OK LOGIN completed, now connected to two.example.com
```

Type the following:

```
2 namespace
```

You see the following system response:

```
* NAMESPACE ((" "/"")) (("Other Users/" "/")) (("Public Folders/" "/"
"))
```

```
2 OK NAMESPACE completed
```

Type the following:

```
3 select "Other Users/principal@<Server 1 FQDN> <mailto:princi-
pal@<Server 1 FQDN>/INBOX"
```

You see the following system response:

```
* 7 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1] UIDVALIDITY value
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft $MdnSent)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft
$MdnSent)] flags will stay set
3 OK [READ-WRITE] SELECT completed
```


Localizing Scalix

This chapter explains how to use Scalix in other languages.

Contents

This chapter includes the following information:

- “Introduction” on page 110
- “Localizing Microsoft Outlook” on page 111
- “Localizing Scalix Web Access” on page 112
- “Localizing the Scalix Search and Index Service” on page 113
- “Localizing for Japanese Language Characters” on page 115

Introduction

Scalix runs one language at a time per installation. You installed the English or German version. Using UTF8 character encoding, Scalix provides full multibyte language support, coupled with an open-source localization kit for channel partners and customers to facilitate international deployments. Contact Scalix.

The language used in Microsoft Outlook can be changed, as can the language in Scalix Web Access. Both are documented here.

When Scalix was installed, the language to use was chosen for searches, for example English or German. This was for the Scalix Search and Index Service, and it can be changed (documented here).

There is also a default option for user language. It merely specifies the language that the user speaks and has no effect on the user interface when they use Scalix Web Access. This language setting is found under **Settings > Administration** in Scalix Management Console (SAC).

Localizing Microsoft Outlook

MAPI is a programming interface from Microsoft that enables applications, such as Microsoft Outlook, to send and receive e-mail. You can localize the MAPI connector for any language that Microsoft Outlook supports. You need the following tool:

- Microsoft resource compiler Visual Studio 6.0, including the compiler RC.EXE and the linker LINK.EXE. For more on the compiler, see http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tools/tools/resource_compiler.asp

To localize the MAPI connector

- 1 Localize the English strings in the `sclxres\resen.rc` file and in the `sclxres\en` subfolder. To locate these files, search the Windows computer for the `sclxres` folder.

Make sure you can save the file with the proper font files.

- 2 From the `sclxres` folder, run the resource compiler. The command is:

```
rc /l<LCID Culture ID> /fo "<file name>\sclxres.res" /i
"..\shared\include" /i "..\common" /d "THIS_LANG_<NAME OF LANGUAGE>"
sclxres.rc
```

where

- `rc` – The compiler command
- `<LCID Culture ID>` – Can be found at <http://www.microsoft.com/globaldev/nlsweb/default.msp?OS=Windows%20Vista>. This code number must be preceded by an "l" (a lower-case l, not an i) and must match the "THIS-LANG..." option
- `l` – Specifies default language for compilation. For example, `-l409` is equivalent to including the following statement at the top of the resource script file: `LANGUAGE LANG_ENGLISH,SUBLANG_ENGLISH_US`
- `fo` – Renames the source file so that it comes out as a `.res` file
- `i` – The "include" command so takes a directory as its variable
- `d` – This variable changes according to the culture ID specified earlier

In this step you are compiling the `sclxres\resen.rc` file, which includes all subfiles and subfolders, and creates a new file named `sclxres.res` to the same directory.

- 3 When the compiler completes, you see a new file named `sclxres.res` in the `sclxres` folder.
- 4 Using the resource compiler's LINK.EXE tool, link the `.res` file to produce a `dll` in the same directory that will be called `sclxres.dll`. The command is:

```
link /nologo /dll /pdb:none /machine:I386 /nodefaultlib /
implib:"sclxres.lib" /NOENTRY sclxres.res
```

- 5 Copy the new file, `sclxres.dll`, to the MAPI connector's installation directory (the default installation directory is `C:\Program Files\Scalix\Connect`).

Note

For more on the link command, see http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vccore98/html/_core_linker_reference.asp

Localizing Scalix Web Access

You can change the language of the Scalix Web Access interface two ways. First, you can translate the files into languages other than those versions provided by Scalix. This involves translating two XML files, putting them in the appropriate location, and restarting Tomcat. Second, the user can select a supported language, such as English or German, within Scalix Web Access and without modifying the two XML files. Both procedures are provided here.

To localize Scalix Web Access

- 1 Manually translate contents of files `strings_en.xml` and `strings_en_US.xml` to create the files `strings_xx.xml` and `strings_xx_XX.xml`. The files are located at:

```
/var/opt/scalix/<nn>/tomcat/webapps/webmail/WEB-INF/data/
```

where `nn` varies with Scalix installation. For example, for Netherlands Dutch, create files named `strings_nl.xml` and `strings_nl_NL.xml` and manually change the contents of the files.

- 2 Stop Tomcat by running the shutdown script found at:

```
/etc/init.d/scalix-tomcat stop
```

- 3 Delete the Tomcat cache if needed by deleting the contents of the following folder:

```
/var/opt/scalix/<nn>/tomcat/work/Catalina/
```

- 4 Start Tomcat:

```
/etc/init.d/scalix-tomcat start
```

- 5 Restart the computer, or restart Scalix using the following command:

```
/etc/init.d/scalix
```

Note

Unless your localization files are part of a Scalix distribution, you need to back up the changed files before the next upgrade. If you do not, your files will be lost.

To change the language in Scalix Web Access

- 1 In Scalix Web Access, click **Tools > Options**.
- 2 In the **General** category, select the user interface language to use from the **Language** drop-down list. Options include English and German.



- 3 Click **OK**.
- 4 Log out then in again. The interface reflects the language chosen.

Localizing the Scalix Search and Index Service

The Scalix Search and Index Service can be configured to process text for any language. To work with different languages, it uses stemming rules for that specific language, which break down words by removing suffixes and endings. For example, a search for the English word "singing" matches the word "sing".

To change the language assumed when indexing text or building search queries

- 1 Stop Tomcat.

```
/etc/init.d/scalix-tomcat stop
```

- 2 On the server where Scalix Search and Index Service is installed, open the following file:

```
/etc/opt/scalix/sis/sis.properties
```

- 3 Edit the property, `sis.language`, to one of the following:

- Danish
- Dutch
- English
- Finnish
- French
- German
- Italian
- Norwegian
- Portuguese
- Russian
- Spanish
- Swedish

(English is default)

- 4 In the `/var/opt/scalix-sis` file, remove existing indexes and rebuild them using the command `sxmkindex`.
- 5 All subsequent index and search operations will use the stemming rules for the given language.

Note

This procedure uses the Snowball Analyzer written by Martin Porter. For more information, see the Lucene API at: <http://lucene.apache.org/java/docs/api/net/sf/snowball/ext/package-summary.html>

Using Custom Analyzers

For additional languages, you can use other analyzers, all of which have built-in stemmers.

To use these additional stemmers

- 1 In the file `sis.properties`, specify a custom class of your own.

```
index.content.analyzer.class=com.scalix.index.message.MyRussianAnalyzer
```

- 2 The list of potential analyzers includes:

- BrazilianAnalyzer
- ChineseAnalyzer
- CJKAnalyzer
- CzechAnalyzer
- DutchAnalyzer
- FrenchAnalyzer
- GermanAnalyzer
- GreekAnalyzer

- 3 The class must be implemented like this:

```
import org.apache.lucene.analysis.Analyzer;
import org.apache.lucene.analysis.ru.RussianAnalyzer;
import com.scalix.index.message.MessageAnalyzer;
public class MyRussianAnalyser extends MessageAnalyzer {
    public Analyzer getAnalyzer() {
        return new RussianAnalyzer();
    }
}
```

- 4 Compile the class and put it in the following directory so that it can be picked up by the Web Apps classloader.

```
~/tomcat/webapps/sis/WEB-INF/classes
```

Localizing for Japanese Language Characters

When using Japanese characters with Scalix, you can change the preferred character standard for representing rich multi-byte text in messages from UTF-8 to ISO-2022-JP. More Japanese Internet mail clients and systems understand ISO-2022-JP than UTF-8.

To change the default character standard

- 1 At the Scalix computer, open the following file in a text editor:

```
/var/opt/scalix/<nn>/s/sys/general.cfg
```

where nn varies with Scalix installation.

- 2 Add the following lines (or modify them if they are already present):

- For IMAP and POP3 clients:

```
BRW_MIME_TEXTFILE_CHARSETS="US-ASCII, ISO-8859-1, ISO-2022-JP, UTF-8"
```

- For the outgoing Internet gateway:

```
UXO_MIME_TEXTFILE_CHARSETS="US-ASCII, ISO-8859-1, ISO-2022-JP, UTF-8"
```

Different settings may be relevant for the domains of different recipient organizations. Therefore, you can add different values for UXO_MIME_TEXTFILE_CHARSETS in per-domain configuration files located in

```
/var/opt/scalix/<nn>/s/sys/domain.cfg/<domainname>
```

Create these files if they do not exist already.

- 3 Restart the relevant Scalix services:

```
omoff -d0 unix mime imap
```

```
omon unix mime imap
```

Once a message has been cached for use by IMAP, POP3, and Scalix Web Access clients, it is not typically re-constructed to reflect changes to BRW_MIME_TEXTFILE_CHARSETS unless the cache is manually refreshed (next procedure).

To refresh the cache

- 1 In the general.cfg file, locate the following setting and change it to a small number, such as 1, so that when checking its cache, the MIME construction routines can see that the version has changed, and so regenerate.

```
MSLDATA_SUBVERSION=1
```

- 2 Run the following command to traverse each user's message store and prime the cache:

```
omtidyallu -M
```

Hosting

This chapter explains how to use a Scalix server to support the mailboxes of multiple companies.

Contents

This chapter includes the following information:

- “Introduction” on page 116
- “Restrictions” on page 118
- “Managing Hosting” on page 119
- “Managing User Accounts” on page 120
- “Moving a Company” on page 121

Introduction

A hosting features allows a single Scalix server to support the mailboxes of multiple companies. It requires a MAILNODE_HOSTING license and is enabled in Scalix Management Console (SAC) or by command line.

Each company has:

- Its own domain
- Its own set of user mailboxes
- Its own view of the directory
- Its own public folder area

For example:

```
mailnode – acme  
domain – acme.com  
public folder – ACME Shared Info
```

The mailboxes, directories, and public folder areas are not visible to other hosted companies, while Scalix administrators have an unrestricted view of the system.

Three plug-ins are provided: sxhostcfg, sxhostadd, and sxhostdel.

Hosted Company Linked to Mailnode (OU1)

Each hosted company is associated with a different mailnode (OU1).

Associating each hosted company with a different mailnode allows the users of those companies to see a filtered view of the system of only the data associated with their company (mailnode).

The primary mailnode is reserved for administration users (who do not have a restricted view of the system). Administration of hosted companies is via Scalix Management Console extensions and plug-ins.

Full Scalix Functionality for Users Within Hosted Company

Full support of Scalix features is provided for users of a hosted company for Microsoft Outlook and Scalix Web Access clients. There is full feature support, for example calendaring, among users with the same hosted company.

Each Company Has Its Own Domain Name

The hosted company mailnode (OU1) is associated with a domain name appropriate to the hosted company. This is the externally visible domain name and is used to construct the Internet address of the mailboxes of the hosted company.

Each Company Has Its Own Directory View

Each hosted company has its own view of the Scalix system directory. Any entries in the directory that have the same mailnode (OU1) of the hosted company are visible to the users of the company. By default these entries will be all users of the hosted company.

Directory entries can also be added for non-company recipients. These non-company recipients must be mail addresses external to the system. Additionally, these external recipients can be configured as Internet (MIME) users, or “rich text” (TNEF) users.

Each Company Has Its Own Public Folder View

Each hosted company has its own view of the Scalix public folders (bulletin board area).

When a hosted company is added, a public folder for that company is added as a top-level public folder. The permissions set on this public folder ensure that this is only folder visible to the hosted company.

By default any user of a hosted company can create a public subfolder under their top-level public folder and add items to any public folder within their view.

Restrictions

There are some limitations.

A Hosted Company Cannot Span Multiple Servers

A hosted company must be configured entirely on a single Scalix server. There is no support for the mailboxes of a individual hosted company being split over two or more Scalix servers.

No Directory Synchronization between Multiple Servers

Directory synchronization between Scalix servers running mailbox hosting is not supported.

No Global Cross-Server Single-Console SAC Management

Each Scalix server running mailnode hosting must be administered by Scalix Management Server (SAC) running on that server. The ability to manage a Scalix hosting server from SAC running on a different Scalix server is not supported.

Third-Party Client LDAP Access Requires Authenticated Bind

When mailnode hosting is enabled for a Scalix server, LDAP access to the Scalix directory requires an authenticated bind. So third-party IMAP clients need to be configured to perform an authenticated bind on behalf of the mail user (username, password) if directory access is required.

Internet Directory Entries Cannot be Shared by Hosted Companies

The Internet address of a user must be unique in the Scalix directory. This has two consequences:

- If an external Internet user has been added for one hosted company it cannot be added again for a second hosted company (because the Internet address has been used in the first external Internet user entry).
- An Internet user account cannot be added for a user in a different hosted company on the same Scalix server (because the Internet address is already present associated with that user).

There are workarounds for this restriction:

- Use personal Contacts
- Use a company Contacts Public Folder
- Use Scalix Management Console (or command line) to “wrap” the entry in a company-visible public distribution list (PDL)

Upgrading Existing Scalix Servers is Not Supported

Although it is possible to enable mailnode hosting on a server that has been upgraded to the hosting release, this is not recommended because existing mailboxes will not have the correct hosting environment setup.

Managing Hosting

You can enable hosting, add hosted companies, and delete hosted companies using the `sxhostcfg`, `sxhostadd`, and `sxhostdel` plug-ins. Each of these commands has a MAN page with information.

Turning Hosting On and Off

Hosting is enabled on the Scalix server by running the `sxhostcfg` script, which is available on any Scalix server that has the `MAILNODE_HOSTING` license installed. It is run at the command line and enables use in Scalix Management Console.

The `--on` switch does the following:

- Checks that an appropriate `MAILNODE_HOSTING` license is present on the server
- Adjusts the folder permissions for the public folders (bulletin board area)
- Deploys the `sxhostadd` and `sxhostdel` plug-ins for use with Scalix Management Console
- Configures `ldapmapper` to user authenticated bind (`user=sxqueryadmin`)

There is also a `--restart` switch to ensure that all Scalix processes pick up the new configuration.

The `--off` switch disables hosting and undoes all of the changes outlined. Turning off hosting allows any user to see the company directory and also the contents of all public folders.

You enable hosting and restart Scalix by command line, then you can use Scalix Management Console to add and delete hosted companies.

To turn hosting on

1 Enter the following command:

```
sxhostcfg --on
```

An error indicates when the `MAILNODE_HOSTING` license is not found.

To restart Scalix

1 Enter the following command:

```
sxhostcfg --restart
```

To turn hosting off

1 Enter the following command:

```
sxhostcfg --off
```

Alert

Turning off hosting allows any user to see the company directory and also the contents of all public folders.

Adding and Deleting a Hosted Company

You can add and deleted hosted companies using the sxhostadd and sxhostdel plug-ins. These can be run in Scalix Management Console or by command line.

To add a hosted company in Scalix Management Console

- 1 Log in to Scalix Management Console (SAC) with an administrator account, for example sxadmin.
- 2 Contact Scalix technical support.

To add a hosted company by command line

- 1 Contact Scalix technical support.

To delete a hosted company in Scalix Management Console

- 1 Contact Scalix technical support.

To delete a hosted company by command line

- 1 Contact Scalix technical support.

Managing User Accounts

When hosting is used, the user account window in Scalix Management Console is modified so that you can select the mailnode because each hosted company has a different mailnode.

Regular user accounts and Internet-only user accounts are possible, referred to as company users and Internet users.

To add a company user account

- 1 Log in to Scalix Management Console (SAC) with an administrator account, for example sxadmin.
- 2 Click the **Users** icon on the toolbar.
- 3 Click **Create User(s)**. The company mailnode is selected from a drop-down list when creating the account.

To add an Internet user account

- 1 Using the previous procedure, select the **mime** or **tnef** option from the mailnode drop-down list when creating the user account. The former produces a MIME format message for the user added, and the latter produces Microsoft Outlook rich text format. The rich text format retains Microsoft Outlook attributes, flags, categories, calendaring information, task information, and so on, and is used when the recipient is another Microsoft Outlook user.

Moving a Company

You can move a company to a different Scalix server.

To move a company to a different server

- 1 Move each user. Enter the following command:

```
sxmbosexp --u
```

for each user, for example

```
sxmbosexp --u "Jane Rogers"
```

- 2 Move the public folders using the following command:

```
sxmbosexp --p --f --s
```