CHAPTER

36

UNIX Spawner Program

Starting the UNIX Spawner Program

As an alternative method to signing on through the TELNET daemon, the UNIX spawner program allows SAS/CONNECT sessions on UNIX systems without requiring that username and password pairs be passed over the network in clear text mode.

If the local SAS session is running Release 6.09E or a subsequent release or Release 6.11 TS040 or a subsequent release, all data that flow from the local host to the spawner program during sign on are encrypted.

Unlike the TELNET sign-on process, the UNIX spawner program allows sign ons to a UNIX system without scripts. In most cases, you should run the UNIX spawner using the root privilege in order to start the remote SAS processes with the privileges of the user who connects to the spawner.

For connections to a UNIX remote host with the TCP/IP access method, SAS/CONNECT uses the default authentication program to verify the remote host's userid and to verify that the password is correct for the specified userid. A SAS/CONNECT user implicitly invokes the authentication program when making a connection to a UNIX remote host by means of the UNIX spawner program.

The spawner program is stored on the remote host in the !*sasroot*/ **utilities/bin** directory.

Here is the syntax for the command to start the UNIX spawner program:

SASTCPD <-BACKGROUND>
<-HELP>
<-INHERITANCE>
<-NETENCRALG *algorithm*>
<-NETENCRYPT YES | NO>
<-NETENCRKEY *n*>
<-NETMAC | -NONETMAC>
<-NOCLEARTEXT>
<-NOSCRIPT>
<-PATH *filename*>
<-SASCMD *filename*>
<-SERVICE *service-name*>
<-SHELL>

<-USER>

-BACKGROUND
specifies that the UNIX spawner program run as a background process. The default specifies that the spawner program run in the foreground.

-HELP
prints a list of valid parameters.

-INHERITANCE
causes the SAS session that is spawned to inherit the socket that was created when the spawner accepted the initial connection from the local SAS session . This option is useful if your configuration involves a firewall and you want to minimize the number of ports that you define to the firewall for use by SAS/CONNECT.

If you start a spawner with the -INHERITANCE option, you then define the port that the spawner is listening on to the firewall and map it to the server machine's port. This will enable any number of SAS/CONNECT clients to connect through this single port and SIGNON to a remote host on the inside of the firewall. Each client just opens a unique socket on the defined port. This eliminates the need to define an individual port for each client that may need to come in through the firewall. In this configuration you set your REMOTE= value to a two-level name where the first level is the name of the host running the firewall and the second level is the well-known service name of the port that you have enabled for connections.

-NETENCRALG *algorithm*
If you specify more than one algorithm, enclose the algorithm names in parenthesis and use commas to separate the names. If there are embedded blanks in the algorithm name, enclose each algorithm in quotation marks.

Set this option at the remote host and, optionally, at the local host to specify one or more encryption algorithms to use in a SAS/CONNECT session. However, the local host and the remote host must share an encryption algorithm in common. If you specify the option in the remote host session only, the local host attempts to select an algorithm that was specified at the remote host. If you also set the option at the local host and specify an algorithm that is not specified at the remote host, the local host's attempt to connect to that remote host fails when the local host assigns a library.

Valid values for this option are

RC2

RC4

DES

TripleDES

SAS Proprietary.

See the *SAS/CONNECT User's Guide* or the *SAS/SHARE User's Guide* for more information about the -NETENCRALG option.

-NETENCRYPT
Set this option at both the local host and the remote host. At the remote host, this option specifies that encryption is required for each connection from a local host SAS session. At the local host, this option specifies that the local host must connect only to a remote host that supports encryption.

The default for this option is that encryption is used if the NETENCRALG option is set and if both the local host and the remote host are capable of

encryption. If encryption algorithms were specified but either the local host or the remote host is incapable of encryption, then encryption will not be performed.

Encryption may not be supported at the local host or the remote host for the following reasons:

- You are running a release of SAS (prior to Version 7) that does not support encryption.
- Your site has not purchased a SAS/SECURE license for a specific platform.
- You specified encryption algorithms in the local host and the remote host SAS sessions that are incompatible.
- You do not have a cryptographic service provider installed on your UNIX system.

See the *SAS/CONNECT User's Guide* or the *SAS/SHARE User's Guide* for more information about the -NETENCRYPT option.

-NETENCRKEY *n*

You set this option in either the local host or the remote host SAS session. It specifies the key length to be used by the encryption algorithm.

Valid values for this option are

128	specifies 1024-bit RSA and 128-bit RC2 and RC4 key algorithms.
40	specifies 512-bit RSA and 40-bit RC2 and RC4 key algorithms.
0	no value is set. This is the default.

If you require extra security, then set the -NETENCRKEY option to 128. If you prefer to save CPU, then set the -NETENCRKEY option to 40.

By default, if you try to connect a host that is capable of only a 40-bit key length with a host that is capable of both a 40-bit and a 128-bit key length, then the connection is made using the lesser key length. If both hosts are capable of 128-bit key lengths, then a 128-bit key length is used.

See the *SAS/CONNECT User's Guide* or the *SAS/SHARE User's Guide* for more information about the -NETENCRKEY option.

-NETMAC | -NONETMAC

Set this option to control the use of Message Authentication Codes (MACs) on network communications. A Message Authentication Code is the equivalent of a checksum that is used to ensure that the original message has not been modified.

This option may be set at either the local host or the remote host. The default is -NETMAC.

See the *SAS/CONNECT User's Guide*or the *SAS/SHARE User's Guide* for more information about the -NETMAC option.

-NOCLEARTEXT

prevents a sign on from a local host that does not support username and password encryption. This option prevents local hosts in a SAS session that are running releases prior to 6.09E and 6.11 TS040 from signing on to the spawner program. The default is to accept both encrypted and clear-text userids and passwords. This allows local hosts in a SAS session that are running releases prior to 6.09E and 6.11 TS040 to sign on to the UNIX spawner program.

-NOSCRIPT

prevents sign ons from local hosts that use scripts, and allows sign ons only from local hosts that do not use scripts.

For the TCP/IP access method, the spawner program requires a script file, or it will verify the supplied userid and the password. This option requires that the user set security in the local SAS session prior to sign on. For details about setting

security (for example, by means of the USER= and PASSWORD= options in an appropriate statement), see "Setting Security for SAS/CONNECT and SAS/SHARE" on page 295.

If you use the -NOSCRIPT option, you must also use the -SASCMD option.

-PATH *pathname*
specifies the location of the SAS system directory that contains the dynamic link library (dll) files needed to access the encryption algorithms. The -PATH option is used with -NETENCRALG.

-SASCMD *filename*
specifies the name of an executable file that starts a SAS session when you sign on without a script. If the RLINK fileref is not defined in the local host SAS session, then the user is signing on without a script. In this case, the -SASCMD option must be specified.

Here is an example of the content of an executable file that starts a SAS session:

```
#----------------------------------
# mystartup
#----------------------------------
#!/bin/ksh
. ~/.profile
sas -dmr -noterminal -no\$syntaxcheck -device grlink -comamid tcp
#-----------------------------
```

-SERVICE *service-name*
specifies the name of the service that the UNIX spawner program uses to listen for incoming requests. This value is identical to the *service* value in the REMOTE= option that the user specifies at the local host prior to sign on. Because there is no default, you must specify this value. See "Specifying the Remote Node Name" on page 301 for details.

The service name must be defined identically in the **/etc/services** file on both the local and remote hosts. See "Configuring the SERVICES File" on page 485 for more information about the **/etc/services** file.

-SHELL
allows the SAS session that is invoked by the UNIX spawner program to create a shell. A shell is necessary for the remote host to execute commands.

-USER
allows the UNIX spawner program to run without root privileges. SAS assumes the security status of the user or the administrator who started the spawner program. The default action is to assume the privileges of the user whose username and password are given to the UNIX spawner program by the remote client that is connecting to spawner.

Note: Because some UNIX systems require root privilege in order to validate passwords, this option may not work on all UNIX systems. △

Examples of Starting and Connecting to the UNIX Spawner Program

The following examples illustrate how to start the spawner program and how to connect to it.

Example 1:

The following command starts the spawner program at the remote UNIX host with the *service* **spawner** and allows connections only from local hosts that support username and password encryption.

```
sastcpd -service spawner -nocleartext
```

At a UNIX local host host, the following statements specify a script file named **tcpunix.scr** that makes a connection to the spawner program named **monarch.spawner**. The value **monarch** for REMOTE= is the name of the UNIX node, or it can be a macro variable that contains the Internet address of the UNIX node where the spawner program is running.

```
options comamid=tcp;
options remote=monarch.spawner;
filename rlink '!sasroot\connect\saslink\tcpunix.scr';
signon;
```

Example 2:

From the UNIX node that will be the remote side of a SAS/CONNECT session, the following command starts the spawner program with the service name **spawner**, which supports only sign ons without scripts.

```
sastcpd -service spawner -noscript -sascmd /u/username/mystartup
```

The **mystartup** file starts the remote SAS session. See the -SASCMD option for an example of the content of the **mystartup** executable file.

At an OS/390 local host, the TCP/IP access method is used to connect to the remote host named **rmthost**, which must be either the node name of the OS/390 node or a macro variable that contains the Internet address of the OS/390 node where the spawner program is running. The USER= option to the SIGNON statement prompts the user for a userid and password when connecting to **rmthost** on which the OS/390 spawner named **spawner** runs.

```
options comamid=tcp;
signon rmthost.spawner user=_prompt_;
```

Ending the UNIX Spawner Program

To end the spawner program, enter the interrupt signal, which typically is CTRL-C. If the UNIX spawner is running in the background, kill its process.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 643.

Communications Access Methods for SAS/CONNECT and SAS/SHARE Software, Version 8

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1–58025–479–9

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227–19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, September 1999

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.® indicates USA registration.

IBM®, ACF/VTAM® , AIX® , APPN® , MVS/ESA® , OS/®2® , OS/390® , VM/ESA® , and VTAM® are registered trademarks or trademarks of International Business Machines Corporation. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.