Doc. No. 78-2854-03 Rev. A0

CiscoSecure UNIX Server User Guide Release Notes 1.0(1)

This document provides information about CiscoSecure UNIX Server 1.0(1), a new software release that supports the Solaris platform and provides bug fixes and related improvements over CiscoSecure UNIX Server 1.0. Use this document to complement information contained in the *CiscoSecure UNIX Server User Guide* publication.

In addition to information on new functionality, including how to unpack CiscoSecure UNIX Server software onto a Solaris platform, this document discusses information that became known or available after the user guide was printed.

For related information, including router configuration examples, refer to the *Cisco Security Configuration Guide* publication in your Cisco Internetwork Operating System (Cisco IOS) Release 11.2 documentation set.

This document contains the following sections:

- New Information for CiscoSecure UNIX Server 1.0(1), page 1
- Improvements to the CiscoSecure UNIX Server Graphical User Interface, page 21
- User Guide Corrections, page 27
- Cisco Connection Online, page 31

New Information for CiscoSecure UNIX Server 1.0(1)

This section describes the following new information in CiscoSecure UNIX Server 1.0(1):

- CiscoSecure UNIX Server Software Files, page 2
- Installing CiscoSecure UNIX Server Software 1.0(1) onto Solaris, page 3
- Installing CiscoSecure UNIX Server Software 1.0(1) onto SunOS, page 8
- Storing Passwords, page 10
- Support for Single TCP Connection, page 10

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA

- Displaying Group Membership for Billing and Accounting, page 10
- Authorization Attribute-Value Pairs, page 12
- Support for Cisco IOS Release 11. 2, page 17
- Working with S/Key Authentication, page 17
- Allow/Refuse Functionality Extended to Remote Address, page 20

CiscoSecure UNIX Server Software Files

The CiscoSecure UNIX Server software /bin directory contains the CiscoSecure executable files, and a "samples" directory, that you can use or adjust according to your needs. The /samples directory contains the files specified in Table 1.

The sample control file contains sample software license keys (they have already expired). As such, if you use CiscoSecure UNIX Server software with the sample keys, you will get a "license expired" message. The message indicates how many ports you are licensed to.

Table 1 Samples Directory Files

File	Description
control.file	CiscoSecure master control file.
aa.database	Example AA[1] database, referenced by control.file.
msg_cat.1	Very simple message catalog file, referenced by control.file.
left.cfg	Example router configuration file that uses CiscoSecure UNIX Server software.
right.cfg	Example router configuration file that uses CiscoSecure UNIX Server software.
samples/run_script	CiscoSecure UNIX Server software startup file

1. AA = Authentication and Authorization

Note The sample control file (found in the /samples directory), contains several structures, or interfaces, that remain undocumented. They are reserved for enhancements targeted for later release of CiscoSecure UNIX Server software. These structures remain private and are subject to change.

These files have been used together in a Cisco Systems lab for some simple tests and are a good place to begin your examination of CiscoSecure UNIX Server software. The example router configuration files represent two routers connected to each other by a single serial link. The router, "left," shares an ethernet segment with the CiscoSecure server. While the name of the router is "left," its config file is named "left.cfg."

Note Left and right are merely test names and do not imply location.

Installing CiscoSecure UNIX Server Software 1.0(1) onto Solaris

CiscoSecure UNIX Server software is now supported on the SPARC Solaris platform.

Note Neither SunOS 4.1.2 nor Solaris 2.4 are supported. CiscoSecure UNIX Server software is supported *only* on SPARC Solaris version 2.5, 2.5.1, SunOS 4.1.3, or 4.1.4.

Installing onto a Solaris Server that Does Not Contain a Disk Drive

If you do *not* have a disk drive, you can download CiscoSecure UNIX Server 1.0(1) for Solaris from a web site, as described in this section.

Note To avoid possible conflicts with volume configurations, confirm that your Solaris platform is not running the volume manager (vold) before beginning this procedure.

Note To take the steps described in this section, you must have a valid SmartNet account. If you do not have a SmartNet account, contact your authorized Cisco Systems support representative for instructions on how to get a SmartNet account.

Step 1 Go to the CiscoSecure Software Planner URL as follows:

```
http://www.cisco.com/kobayashi/ciscosecure.html
```

You are prompted for a username and password in order to access Cisco Connection Online (CCO).

Step 2 Using your SmartNet account, log in to CCO, specifying your username and password as prompted.

Step 3 Click **Download CiscoSecure Software.**

You see the CiscoSecure Server Software Images page.

Step 4 Click the button beside CiscoSecure 1.0(1) Solaris, then click **Execute**.

You are prompted to specify the location from which to transfer the software image.

Step 5 Click the location of the CCO server that is closest to your target CiscoSecure server.

You are prompted again for your CCO password.

Step 6 Enter your CCO password.

A file is copied to your home directory.

Step 7 Uncompress and untar the CiscoSecure UNIX Server software image by entering the following command at the UNIX prompt:

```
# zcat /tmp/csu.pkg.tar.z | tar xvf .
```

You see the names of each of the files contained in the software image.

Step 8 Install the CiscoSecure package by entering the following command at the UNIX prompt:

```
# pkgadd -d /tmp
```

The ready-to-use files are added to the /tmp directory and you see output similar to the following:

```
The following package(s) are available:
1Cisco CiscoSecure TACACS+ Server Software
(sun4) Version CSUS-1.0(1.0)

Select the package(s) you wish to process (or 'all' to process all packages).
(default: all) [?,??,q]
```

Step 9 Enter **1.**

You are then prompted to specify a directory in which to install the CiscoSecure UNIX Server for Solaris package file (for most purposes, you can specify /usr/ciscosecure), as follows:

```
Enter path to package base directory [?,q] /usr/ciscosecure
```

You see output similar to the following:

```
Using </usr/ciscosecure> as the package directory.
Cisco Systems, Inc.
## Processing package information.
## Processing system information.
## Verifying disk space requirements.
## checking for conflicts with packages already installed.
## checking for setuid/setgid programs.

Installing Cisco CiscoSecure TACACS+ Server Software as <CSCOcsu>
```

After another series of checking-level messages, you see the following message:

```
Installation of <CSCOcsu> was successful.
```

Step 10 Skip ahead to "Obtaining a Software License Key, page 10"to enable the licensed ports on your network access server.

Installing onto a Solaris Server that Does Contain a Disk Drive

This section contains instructions on how to install CiscoSecure UNIX Server 1.0(1) onto a Solaris platform.

Note To avoid possible conflicts with volume configurations, confirm that your Solaris platform is not running the volume manager (vold) before beginning this procedure.

Take the following steps to unpack the CiscoSecure UNIX Server 1.0(1) files:

Step 1 Find the disk labelled "CiscoSecure UNIX Server for Solaris," in your CiscoSecure UNIX Server software package.

Note To simplify this procedure, you will install software in a /tmp directory. Later, you can move or rename the directory according to your needs. If you need help renaming or moving the directory, refer to the documentation that came with your Solaris software.

Step 2 Run the Solaris utility, dd, to convert and copy the CiscoSecure UNIX Server software image from the floppy disk to a /tmp directory, as follows:

```
# dd if=/dev/diskette of=/tmp/csu.pkg.tar.z
```

When the conversion and copying functions are complete, you see output similar to the following:

```
2880+0 records in
2880+0 records out
```

Step 3 Uncompress and untar the CiscoSecure UNIX Server software image by entering the following command at the UNIX prompt:

```
# zcat /tmp/csu.pkg.tar.z | tar xvf -
```

While the image is expanding, you see a list of its filenames.

Step 4 Install the CiscoSecure UNIX Server for Solaris package by entering the following command at the UNIX prompt:

```
# pkgadd -d /tmp
```

The ready-to-use files are added to the /tmp directory and you see output similar to the following:

```
The following package(s) are available:
1Cisco CiscoSecure TACACS+ Server Software
(sun4) Version CSUS-1.0(1.0)

Select the package(s) you wish to process (or 'all' to process all packages).
(default: all) [?,??,q]
```

Step 5 Enter **1.**

You are then prompted to specify a directory in which to install the CiscoSecure UNIX Server for Solaris package file (for most purposes, you can specify /usr/ciscosecure), as follows:

```
Enter path to package base directory [?,q] /usr/ciscosecure
```

You see output similar to the following:

```
Using </usr/ciscosecure> as the package directory.
Cisco Systems, Inc.
## Processing package information.
## Processing system information.
## Verifying disk space requirements.
## checking for conflicts with packages already installed.
## checking for setuid/setgid programs.

Installing Cisco CiscoSecure TACACS+ Server Software as <CSCOcsu>
```

After another series of checking-level messages, you see the following message:

```
Installation of <CSCOcsu> was successful.
```

Step 6 Skip ahead to “Obtaining a Software License Key, page 10” to enable the licensed ports on your network access server.

Creating a Startup File on Solaris

This section describes how to create a startup file on Solaris.

Note If you prefer not to create you own startup file, you can use or modify the sample startup file located in /samples/run_script.

Take the following steps to create a startup file on Solaris:

Step 1 Copy the following module to be used as a startup file into the directory /etc/init.d/ciscosecure:

```
#!/sbin/sh
# CiscoSecure control
state=$1
set `who -r`
if [ $8 != "0" ]
then
    exit
fi
pid=`cat /etc/CiscoSecure.pid`
case $state in
'start')
    CSUHOME=/usr/CSU
state=$1
pid=""

#
# It is a bug that the pid file can remain after the
# server exits, and some other process can now be on
# the pid contained in /etc/CiscoSecure.
#
# But lots of Unix commands have the same problem...
#
# We could use 'ps' to attempt to find a running copy of
# CiscoSecure, but the admin could have changed the name.
#
# Not much to do here but cross one's fingers..
#

if [ -w / ]     # only root can write /
then
    if [ -r /etc/CiscoSecure.pid ]
    then
        pid=`cat /etc/CiscoSecure.pid` > /dev/null 2>&1
    fi
else
    echo "not root"
    exit
fi

case $state in
#
# Start-up
#
```

```
'start')
    if [ -x ${CSUHOME}/bin/CiscoSecure -a -f ${CSUHOME}/samples/control.file ]
    then
        cd ${CSUHOME}
        ./bin/CiscoSecure -x -f ./samples/control.file > Logfile 2>&1 &
    fi
    ;;

#
# Stop processing, don't come back.
#
'stop')
    if [ "${pid}" != "" ]
    then
        /usr/bin/kill -INT ${pid} > /dev/null 2>&1
    fi
    ;;

#
# Causes the daemon to re-init.
#
'reload')
    if [ "${pid}" != "" ]
    then
        /usr/bin/kill -HUP ${pid} > /dev/null 2>&1
    fi
    ;;

#
# Ask CiscoSecure to process the 'update' file.
# (Forces changes passwords back into the aa database.
#
'update')
    if [ "${pid}" != "" ]
    then
        /usr/bin/kill -USR1 ${pid} > /dev/null 2>&1
    fi
    ;;

*)
    echo "usage: $0 {start|stop|reload|update}"
    ;;

esac
```

Step 2 Establish links to the startup file so that CiscoSecure UNIX Server software is stopped and started at the appropriate times. By establishing links, you can make the startup file more accessible and you can track changes in version numbers more conveniently. For example, you could specify the latest version of CiscoSecure UNIX Server software by the link “CiscoSecure” rather than entering a full name such as “CiscoSecure UNIX Server 1.0(1b3).”

You can use standard Solaris commands to establish links. If, for example, you want to link the sample startup file to the names /etc/rc2.d/S99csu and /etc/rc0.d/K89csu, then you would use the following commands:

- # **chmod 744 /etc/init.d/ciscosecure**
- # **ln -s /etc/init.d/ciscosecure /etc/rc2.d/S99csu**
- # **ln -s /etc/init.d/ciscosecure /etc/rc0.d/K89csu**

The rc scripts are automatic scripts that the kernel will execute when you enter or exit out of certain run levels of operation. When the kernel enters a certain run level, it will execute the specified script. For more information on establishing links to the CiscoSecure UNIX Server startup file on Solaris, refer to the man page for init (on Solaris).

When installation is complete, the CiscoSecure UNIX Server software control file and AA database must be properly configured before starting the server. (See Chapter 3"Configuring CiscoSecure UNIX Server Software," and Chapter 4 "The AA Database" in the *CiscoSecure UNIX Server User Guide*.)

Installing CiscoSecure UNIX Server Software 1.0(1) onto SunOS

Complete installation information and examples are included in the *CiscoSecure UNIX Server User Guide*. However, for your convenience, the following information is provided to help you install CiscoSecure UNIX Server software, and to start and stop the server using the files contained in the /samples directory. For more information, see Chapter 3 "Configuring CiscoSecure UNIX Server Software" in the *CiscoSecure UNIX Server User Guide*.

Note When you run CiscoSecure UNIX Server software, you are allowed four additional ports beyond the number of ports your license agreement stipulates. For example, if you purchased a license to use 16 ports, CiscoSecure UNIX Server software will indicate that you are licensed to use 20 ports.

CiscoSecure UNIX Server software Version 1.0(1) includes the binary image, CiscoSecure. You can install this program anywhere within the file system. However, for best results, install the CiscoSecure binary image in the directory /usr/local/etc. In order to perform its function, CiscoSecure UNIX Server software must be run with superuser privileges.

Installing onto a SunOS Server that Does Not Contain a Disk Drive

If you do *not* have a disk drive, you can download CiscoSecure UNIX Server 1.0(1) for SunOS from a web site, as described in this section.

Note To take the steps described in this section, you must have a valid SmartNet account. If you do not have a SmartNet account, contact your authorized Cisco Systems support representative for instructions on how to get a SmartNet account.

Step 1 Go to the CiscoSecure Software Planner URL as follows:

```
http://www.cisco.com/kobayashi/ciscosecure.html
```

You are prompted for a username and password in order to access Cisco Connection Online (CCO).

Step 2 Using your SmartNet account, log in to CCO, specifying your username and password as prompted.

Step 3 Click **Download CiscoSecure Software.**

You see the CiscoSecure Server Software Images page.

Step 4 Click the button beside CiscoSecure 1.0(1) SunOS, then click **Execute**.

You are prompted to specify the location from which to transfer the software image.

Step 5 Click the location of the CCO server that is closest to your target CiscoSecure server.

You are prompted again for your CCO password.

Step 6 Enter your CCO password.

A SunOS tar file is copied to your home directory.

Step 7 Unpack the file by entering the command as follows:

```
# tar xvf <filename>
```

Step 8 Edit the /etc/services file to include a definition for the Terminal Access Controller Access Control System Plus (TACACS+) protocol port number, if this is not already present. The protocol port number definition is as follows:

```
tacacs          49/tcp          TACACS+
```

Step 9 Skip ahead to "Obtaining a Software License Key, page 10" to enable the licensed ports on your network access server.

Installing onto a SunOS Server that Does Contain a Disk Drive

Take the following steps to install CiscoSecure UNIX Server software:

Step 1 Become superuser.

Step 2 Select a directory into which to install the CiscoSecure UNIX Server software. Normally, this would be a system directory such as /usr/etc/ciscosecure or /usr/local/etc/ciscosecure. Create this directory if it does not already exist and make it your current directory. For example:

```
% su
Password: <password>
# mkdir /usr/local/etc/ciscosecure
# cd /usr/local/etc/ciscosecure
```

Step 3 Extract the distribution into the selected directory. The installation disks contain a compressed tar file:

- To extract CiscoSecure UNIX Server software onto a SunOS platform, enter the following command:

```
# bar xvZf /dev/rfd0
```

Step 4 Edit the /etc/services file to include a definition for the Terminal Access Controller Access Control System Plus (TACACS+) protocol port number, if this is not already present. The protocol port number definition is as follows:

```
tacacs          49/tcp          TACACS+
```

Step 5 Skip ahead to "Obtaining a Software License Key, page 10" to enable the licensed ports on your network access server.

Obtaining a Software License Key

After you finish installing CiscoSecure UNIX Server software on your particular platform configuration, you need to obtain a software license key from Cisco in order to enable your licensed ports.

To obtain a software license key take the following steps:

Step 1 Enter the **hostid** command to obtain the host ID of the system host.

To obtain the hostid on a Sun OS platform, enter the following:

```
# hostid
55412315
```

To obtain the hostid on a Solaris platform, enter the following:

```
# /usr/ucb/hostid
55412315
```

Step 2 Fill out the "CiscoSecure Software Key Fax-Back Form," including the host ID of the primary and backup CiscoSecure server systems, and fax it to the number on the form. Your software key will be returned within two business days. (For details, refer to page 6-7 in the section "Licensing" in chapter 6 "Using CiscoSecure UNIX Server Software" of the *CiscoSecure UNIX Server User Guide*.)

Step 3 Edit the control file in /usr/local/etc/ciscosecure. At the beginning of the file, locate "LIST config_license_key" and enter the software key that was returned to you when you completed Step 1.

Step 4 Add the binary image, CiscoSecure, to the /etc/rc.local startup file if it is to be restarted automatically on system reboot.

Storing Passwords

Under Solaris, encrypted passwords are no longer stored in a password file (in other words, /etc/passwd). Instead, passwords are stored in the file /etc/shadow and can be accessed only as root or in cases where read access has been assigned.

The /etc/shadow file is a separate, unreadable file that contains the encrypted passwords on Solaris platforms, and was created to help avoid "dictionary guessing" attacks on passwords.

Support for Single TCP Connection

In CiscoSecure UNIX Server software 1.0(1), the network access server can maintain a longer Transmission Control Protocol (TCP) connection to the TACACS+ daemons. This optimization feature is supported in network access servers running Cisco IOS Release 11.2 or later.

Support for a single TCP connection means that the connection is maintained in an open state for as long as possible, instead of being opened and closed each time a session is negotiated. As such, you can expect that single connection will yield performance improvements.

Displaying Group Membership for Billing and Accounting

CiscoSecure UNIX Server software can now add a field to each accounting record that will indicate the immediate group membership of the corresponding user as listed in an AA database. In this way, accounting organizations can easily know whether to adjust billing information according to the user's group association.

The ability to display group membership for billing and accounting is achieved by specifying the acct_member attribute in the CiscoSecure UNIX Server software control file, and by specifying the acct_member attribute beside each user's name whose group membership you want to be able to display.

Take the following steps to install the acct_member attribute:

Step 1 Copy the libacctmember.so file from the CiscoSecure UNIX Server 1.0(1) disk to the directory where the CiscoSecure binary image resides.

In most cases, the CiscoSecure binary image will reside in a directory named /bin and is found under /usr/local/CiscoSecure or in a similar location.

Step 2 Add a statement (using vi or similar text editor) to the control file as follows:

```
ACCT config_external_acct_symbols = {
  {
  "bin/libacctmember.so",
      "acct_member_fn"
        }
  };
```

The first string should reflect the location of the libacctmember.so shared library. This pathname can be fully qualified (for example, "/usr/local/CiscoSecure/bin/libacctmember.so") or it can be relative to the directory where CiscoSecure UNIX Server software is started.

Step 3 For each member of the AA database where you would like the immediate group membership displayed, you must supply a statement that indicates that "acct_member_fn" should be called as follows:

```
user = test {
    member = foo
    account = acct_member_fn

    password = clear "test" # this is just an example
    [...]                   # and so on, and so on
}
```

The "account = acct_member_fn" statement could be located anywhere above the indicated user in the AA database. For example, the specified AA database file might look as follows:

```
user = test1 {

    member = group1


    password = clear "test"         # this is just an example

    [...]                   # and so on, and so on

}


user = test2 {

    member = testusers

    [...]                   # and so on, and so on

}
```

```
group = testusers {

    member = allusers


}


group = allusers {

    account = acct_member_fn

}
```

In this case, the record “acct_member_fn” would be called for all users who were in the group “allusers” or any subgroup of same (for example, in the previous display, testusers was such a subgroup).

Step 4 Reload the CiscoSecure server by means of the UNIX **kill** command as follows:

```
# kill -HUP `cat /etc/CiscoSecure.pid`
```

When you view your accounting data, you can now identify the group membership of a particular user. For example, the accounting data based on the sample data in this section, would look as follows:

```
cisco.Secure.com    test    tty2    192.207.126.16  stop
server=ciscosecure=21:53:52   date=04/10/96   task_id=26291
service=exec    port=2  service=exec   port=2  elapsed_time=2
member=foo
```

Note that the accounting data now identifies the user “test” as a member of the group “foo.”

Authorization Attribute-Value Pairs

You can skip this section unless you want to bypass the CiscoSecure UNIX Server software graphical user interface (GUI), as the means to set authorization value pairs, or unless you need to view a convenient list of service attributes and the corresponding protocol values.

The authorization attribute-value pairs presented here are primarily for reference and to complement the accounting attribute-value pairs already documented in the publication *CiscoSecure UNIX Server User Guide*. For an example of how authorization attribute-value pairs are used, refer to page 4-6 in the section “The AA Database” File, in Chapter 4, “The AA Database” in the publication *CiscoSecure UNIX Server User Guide*. For related information, refer to the *Cisco Security Configuration Guide* in your Cisco IOS Release 11.2 documentation set.

CiscoSecure supports all four service attributes available to dial-in users, as follows:

```
service=arap
service=shell (for exec startup, and also for command authorizations)
service=ppp
service=slip
```

After the network access server has authorized the user for a specified service, the CiscoSecure UNIX server returns to the network access server a list of attribute-value pairs appropriate for that service. For each service, several attribute-value pairs are generally available depending on the configurability of the service.

Note The attribute value pairs that can be used depend on a given service. However, the CiscoSecure UNIX Server software GUI displays all attribute value pairs even though they might not all be used in a particular context. For example, if you select service=ppp, you see attribute-value pairs for arap even though arap does not run under ppp.

Each of the following attribute-value pairs is accompanied by a notation that identifies the corresponding, supported service.

Note The authorization attribute-value pairs documented here are supported by network access servers running Cisco IOS Release 10.3(3) or greater, except where noted.

- protocol=lcp

  The lower layer of Point-to-Point (PPP), always brought up before Internet Protocol (IP), Internetwork Packet Exchange (IPX), or another protocol capable of running under PPP is brought up.

- protocol=ip

  Used with service=ppp and service=slip to indicate which protocol layer is being authorized.

- protocol=ipx

  Used with service=ppp to indicate which protocol layer is being authorized.

- protocol=atalk

  Used with service=ppp or service=arap.

- protocol=vines

  Used for Virtual Integrated Network Service (VINES) over PPP.

- protocol=unknown

  Used for undefined or unsupported conditions. The use of this pairing should not occur under normal circumstances.

- cmd (EXEC)

  If the value of cmd is NULL; for example, the attribute-value pair is cmd*, then this is an authorization request for starting an EXEC command.

  If cmd has a value other than NULL, this is a command authorization request. It contains the name of the command being authorized, as follows:

  ```
  cmd=telnet
  ```

- cmd-arg (EXEC)

  During command authorization, the name of the command is given by an accompanying “cmd=” attribute-value pair, and each command argument is represented by a cmd-arg attribute-value pair, as follows:

  ```
  cmd-arg=archie.sura.net
  ```

Note The attribute-value pair “cmd-arg” should never appear in a configuration file. “cmd-arg” is used internally by the daemon to construct a string which is then matched against the regular expressions that appear in a cmd clause in the configuration file.

- acl (ARAP, EXEC)

  For AppleTalk Remote Access Protocol (ARAP) this contains an access-list number. For EXEC authorization, acl contains an access-class number, as follows:

  ```
  set acl=2
  ```

- inacl (PPP/IP)

  This attribute-value pair contains an Internet Protocol (IP)-input access-list number for Serial Line Internet Protocol (SLIP) or PPP/IP, as follows:

  ```
  set inacl=2
  ```

  The access list must be preconfigured on the Cisco network access server. Per-user access lists do not currently work with Integrated Services Digital Network (ISDN) interfaces.

- outacl (PPP/IP, PPP/IPX)

  This attribute-value pair contains an IP or IPX output access-list number for SLIP, PPP/IP, or PPP/IPX connections, as follows:

  ```
  set outacl=4
  ```

  The access list must be preconfigured on the Cisco network access server. Per-user access lists do not currently work with ISDN interfaces. PPP/IPX is supported only in Cisco IOS Release 11.1 and greater.

- addr (SLIP, PPP/IP)

  The IP address the remote host should be assigned when using a SLIP or PPP/IP connection, as follows:

  ```
  set addr=1.2.3.4
  ```

- routing (SLIP, PPP/IP)

  Equivalent to the /routing flag in SLIP and PPP commands. It can have as its value the string “true” or “false.”

- timeout (supported in Cisco IOS Release 11.0 and greater, ARAP, EXEC)

  Sets the time (in minutes) until an ARAP or EXEC session disconnects unconditionally, as follows:

  ```
  set timeout=60
  ```

- autocmd (EXEC)

  During EXEC startup, this specifies an autocommand, like the autocommand option to the username configuration command, as follows:

  ```
  set autocmd="telnet foo.com"
  ```

- noescape (EXEC)

  During EXEC startup, this specifies “noescape,” like the noescape option to the username configuration command. It can have as its value the string “true” or “false,” as follows:

```
set noescape=true
```

- nohangup (EXEC)

  During EXEC startup, this specifies “nohangup,” like the nohangup option to the username configuration command. It can have as its value the string “true” or “false,” as follows:

  ```
  set nohangup=true
  ```

- priv-lvl (EXEC)

  Specifies the current privilege level for command authorizations, a number from 0 to 15 (where 0 specifies the lowest privilege level and 15 specifies the greatest privilege level), as follows:

  ```
  set priv-lvl=5
  ```

Note In Cisco IOS Release 10.3, this attribute was priv_lvl (in other words, it contained an underscore instead of a hyphen).

- zonelist (ARAP)

  An AppleTalk zonelist for ARAP equivalent to the line configuration command **ARAP zonelist**, as follows:

  ```
  set zonelist=5
  ```

- addr-pool (supported in Cisco IOS Release 11.0 and greater, PPP/IP, SLIP)

  This attribute-value pair specifies the name of a local pool from which to get the IP address of the remote host.

Note The attribute-value pair “addr-pool” works in conjunction with local pooling. It specifies the name of a local pool (which needs to be preconfigured on the network access server).

Use the **ip-local pool IOS** configuration command to declare local pools, such as those on the network access server, as follows:

```
ip address-pool local
ip local pool foo 1.0.0.1 1.0.0.10
ip local pool baz 2.0.0.1 2.0.0.20
```

You can indicate from which address pool you want to get this remote node’s address. As shown in the following example, you can use the TACACS+ protocol to return addr-pool=foo or set addr-pool=baz.

```
user = lol {
   service = ppp {
     protocol = lcp {
     }
     protocol = ip {
       set addr-pool=foo
     }
   }
```

```
}
```

- route (supported in Cisco IOS Release 11.1 or greater, PPP/IP, SLIP)

  This attribute-value pair specifies a route to be applied to an interface.

  During network authorization, the “route” attribute can be used to specify a per-user static route, to be installed by means of TACACS+.

  The daemon-side declaration is as follows:

```
service=ppp protocol=ip {
set route = “<dst_addr> <mask> [ <gateway> ]”
        }
```

  This indicates a temporary static route that is to be applied. The parameters <dst_address>, <mask> and [<gateway>] are expected to be in the usual dotted-decimal notation, with meanings that are the same as the familiar **ip route** configuration command on a network access server.

  If gateway is omitted, the peer’s address is taken to be the gateway.

  The route is expunged after the connection terminates.

- callback-rotary (supported in Cisco IOS Release 11.1 and greater, valid for ARAP, EXEC, SLIP or PPP)

  The number of a rotary group (from 0 to 100, inclusive) to use for callback, as follows:

```
set callback-rotary=34
```

- callback-dialstring (supported in Cisco IOS Release 11.1 and greater, valid for ARAP, EXEC, SLIP or PPP) sets the telephone number for a callback, as follows:

```
set callback-dialstring=408-555-1212
```

- callback-line (supported in Cisco IOS Release11.1 or greater, valid for ARAP, EXEC, SLIP or PPP)

  The number of a tty line to use for callback, for example, nocallback-verify (supported in Cisco IOS Release 11.1 or greater, valid for ARAP, EXEC, SLIP or PPP). Indicates that no callback verification is required. The only valid value for this parameter is the digit one, as follows:

```
set nocallback-verify=1
```

- idletime (supported in Cisco IOS Release 11.1 and greater, EXEC)

  Sets a value, in minutes, after which an idle session will be terminated.

Note The attribute, idletime, does *not* work for PPP.

- tunnel-id (supported in Cisco IOS Release 11.2 and greater, PPP/Virtual Public Data Network [VPDN]

  This attribute-value pair specifies the username that will be used to authenticate the tunnel over which the individual user multiplex-ID will be projected. This is analogous to the “NAS name” in the **vpdn outgoing** command.

- ip-addresses (supported in Cisco IOS Release 11.2 and greater, PPP/VPDN)

This is a space separated list of possible IP addresses that can be used for the endpoint of the tunnel.

Support for Cisco IOS Release 11. 2

Cisco IOS Release 11.2 provides a new version of the TACACS+ protocol; this protocol is backward compatible with old daemons (including CiscoSecure UNIX Server 1.0). As such, you can now install Cisco IOS Release 11.2 in your network without having to upgrade your existing CiscoSecure UNIX Server software daemon at the same time.

Cisco IOS Release 11.2 also provides two new attribute-value pairs, tunnel-id and ip-addresses, both of which are supported in CiscoSecure UNIX Server 1.0(1). For more information on attribute-value pairs, see the previous section, “Authorization Attribute-Value Pairs, page 12.”

Working with S/Key Authentication

The S/Key one-time password system from Bellcore provides secure authentication over networks that are subject to eavesdropping. S/Key distinguishes itself from other one-time or multi-use authentication systems by preventing the user’s secret password from ever crossing the network during authentication.

A Scenario of Using S/Key

To help you better understand the benefits of using S/Key with CiscoSecure UNIX Server software, consider the following example of a hypothetical user, Sue, who authenticates to the CiscoSecure network access server by means of the S/Key system.

1. Upon the standard prompt for authentication, Sue identifies herself to the network access server by her login name.

   ```
   User Access Verification
   Username: sue
   s/key 97 fr09072
   Password:
   ```

   The CiscoSecure server observes that Sue needs to supply an S/Key password.

2. The CiscoSecure server then issues a challenge including the sequence number of the one-time password expected and a “seed.” The seed is a special value used by the S/Key algorithm as the starting point for the creation of an S/Key password. This seed will also allow Sue to securely use a single secret password.

   Based on the verification display, the CiscoSecure server instructed the network access server to display the sequence number, 97, and a seed, fr09072, which will be used by a separate program to initiate the encryption process leading to an S/Key password.

   Sue notes the sequence number and seed, then pauses from her interaction with the network access server in order to generate a password. She will generate the password by entering the sequence number and seed, along with her secret password, into an S/Key calculator program.

3. Sue enters 97 and fr09072 into her S/Key calculator program at the UNIX prompt, as shown in the example display. (On UNIX, the S/Key calculator program is called key.)

   ```
   % key 97 fr09072
   Enter secret password: <secret password>
   ```

   The secret password is any string of at least 10 alphanumeric characters generated by Sue, for Sue, and known only by Sue.

4 The secret password triggers the creation of a second password, as follows:

```
CRAG BAKE MOLT JEAN JIBE OFT
```

The one-time S/Key password is always expressed as a sequence of six short English words. Note how the one-time password is generated without any secret information crossing the network.

This second password will be used to authenticate Sue to the CiscoSecure server. Sue now returns to her interaction with the network access server. She enters the S/Key password and is authenticated, as follows:

```
Password: <CRAG BAKE MOLT JEAN JIBE OFT>
```

5 The next time Sue attempts network access, she will be prompted for the one-time password sequence number, 96.

The sequence number is one less than what was used for the previous authentication. In the case of Sue, her last sequence number was 97, so the next required sequence number will be 96. When the sequence number reaches 0, Sue will not be able to log in without reinitializing the S/Key system.

Sue's account could also be configured so that she is required to use S/Key when she enables on the router. In this case, the AA database would be modified to display something like the following:

```
user = sue {
        password = skey
        privilege = skey 15
}
```

In this case, Sue would be required to give a different S/Key password every time she logs in and every time she enables at level 15.

Preparing to Install S/KEY

Take the following steps to prepare for S/Key installation and use:

Step 1 Modify your CiscoSecure UNIX server AA database file to set up each S/Key user. For example, to set up a hypothetical user named Sue, you would modify the CiscoSecure server AA database file as follows:

```
user = sue {
password = skey
}
```

Step 2 Restart the CiscoSecure server by entering the UNIX **kill** command as follows:

```
# kill -HUP `cat /etc/CiscoSecure.pid`
```

Note After modifying the AA database, instruct S/Key users that they will have to run the keyinit program on the CiscoSecure server. The keyinit program initializes the S/Key system for that user. You should also inform users that when they run keyinit, they will be prompted for two passwords. The first is the UNIX login password. The second is the secret password used with S/Key. For security purposes, the UNIX password and the secret password should not be the same. Also note that the secret password must be at least 10 characters.

Installing and Getting Ready to Use S/Key

Take the following steps to install the S/Key system on a CiscoSecure server:

Step 1 Log in to the CiscoSecure web site at the following location and download the S/Key one-time password system prebuilt distribution:

ftp://*userid*@**www.cisco.com/cisco/netmgmt/ciscosecure/sunos**

Step 2 Unpack the skey-cs.tar file, as follows:

tar -xvf skey-cs.tar

Step 3 Run the enclosed INSTALL.S_Key script, as follows.

INSTALL.S_Key

In the next step, each S/Key user will run the keyinit program to initialize the S/Key system for that user. (For the purpose of example, a hypothetical user, Sue, will run the keyinit program to initialize the S/Key system. This process enables Sue to use S/Key authentication.)

Step 4 Have CiscoSecure users who will use S/Key enter the **keyinit** command at the UNIX prompt as follows:

```
% keyinit
Password: <UNIX password>
[Adding sue]
Enter secret password: <secret password>
```

When the keyinit program asks Sue for a secret password, she is free to supply any mix of 10 or more alphanumeric characters.

```
Again secret password: <secret password>


ID sue s/key is 99 fr05065
Next login password: SKI INCA HONE NEE MESS LEAF
```

Now Sue is ready to use S/Key with CiscoSecure UNIX Server software.

S/Key also accounts for previous iterations of keyinit, providing assurance for the user that someone has not altered the system. As a result, the next time that Sue enters the **keyinit** command, she will see a display similar to the following:

```
% keyinit
Password: <Unix password>
[Updating sue]
Old key: fr05064
Enter secret password: <secret password>


Again secret password: <secret password>


ID sue s/key is 99 fr05065
Next login password: SKI INCA HONE NEE MESS LEAF
```

Note When Sue enters her secret password, she is entering a personal identification number in order to generate another password. The secret password could be the same as her UNIX password, or it might be any string of characters. This secret password does not change. Sue, as an S/Key user, must remember her S/Key password in order to generate the second password used for S/Key authentication to the CiscoSecure UNIX server.

Allow/Refuse Functionality Extended to Remote Address

CiscoSecure UNIX Server software can control authorization of services based on several values, including the NAS name, NAS port, and now, the remote address. These fields are supplied by the NAS. For example, the remote address might contain a string representing an X.121 address, the IP address associated with the remote end of a telnet connection, or the calling number (Caller ID) on an interface. These controls are called filters.

Using the CiscoSecure graphical user interface, you can now specify the following by clicking the **Filter** button to enable attributes as follows:

```
allow nas_name port_name rem_addr
refuse nas_name port_name rem_addr
```

The nas_name, port_name, and rem_addr can all be regular expressions. (See the man page for regex for details on writing regular expressions.)

The following two examples show how a CiscoSecure UNIX Server administrator can take advantage of the filter mechanism to control authorization of services:

Example 1:

```
# Let Oscar the Grouch start a shell on any tty lines attached to any NAS
# in the cisco.com domain, and start a shell on any vty,
# as long as he's logging in from the machine trashcan.cisco.com
#
# Once on, Oscar the Grouch can run any command.
#
        user = grouchy {
                password = clear "dratsab"
                service = shell {
                   default cmd = permit
                   default attribute = permit
                   allow ".*\.cisco\.com" "tty.*" ".*"
                   allow ".*\.cisco\.com" "vty.*" "trashcan\.cisco\.com"
                   refuse ".*" ".*" ".*"
                }
        }

# The 'refuse' filter in this example isn't strictly necessary.
# However, if the profile were modified to contain
# 'default service = permit', shell service would be allowed on any
# NAS unless the associated refuse ".*" ".*" *.*" was in-place.
```

Example 2:

```
# Allow Chuck Yager to start a shell on any tty line attached to any NAS
# served by this copy of CiscoSecure.  Further allow access to vtys if
# Chuck is logging in from anywhere in the 198.xxx.yy network, but
# absolutely refuse access to any nas if Charles is logging in from
# champagne.cisco.com (the external NAS). Note that we provide both
# the name and the ip address for champagne, just for assurance in
# case the DNS breaks.
#
#
# Once on, only allow Chuck to use the 'telnet' command.
#
        user = cyager {
                password = clear "gnitekram"
                service = shell {
                   default cmd = telnet
                   default attribute = permit
                   allow ".*" "vty.*" "198\.xxx\.yy\..*"
                   refuse ".*" "vty.*" "171\.xx\.yy\.xx"
                   refuse ".*" "vty.*" "champagne\.cisco\.com"
```

```
                allow ".*" "tty.*" ".*"
            }
        }
```

Filters can also be used to apply caller-id information (if provided by the NAS) to a service authorization. If the server supplies caller-id information, it is supplied in the rem_addr field, as follows:

```
# Allow Jerry to start PPP, but only if he's coming in via ISDN on a
# basic-rate channel, and only if # the caller-id string matches what
# we think it should be, and only on NAS22.cisco.com.

        user = jerry {
                password = chap "was a race car driver"
                service = ppp {
                        protocol = lcp {
                        }
                        protocol = ip {
                        }
                        allow "NAS22\.cisco\.com" "BRI.*" "7145551212"
                }
        }
```

The GUI now acknowledges where a regular expression is required, and provides a default of ".*" for you. (This is a regular expression that matches any string.)

Improvements to the CiscoSecure UNIX Server Graphical User Interface

This section discusses fixes to problems and clarifies difficult points associated with CiscoSecure UNIX Server software version 1.0. Bug-tracking numbers follow descriptions that address specific bugs. The following sections are included:

- Save As Command, page 21
- DES Password Support, page 21
- Retrieving Database Files, page 22
- Validating Data, page 22

Save As Command

In the CiscoSecure UNIX Server 1.0 graphical user interface (GUI), if you initially tried to save a database with an illegal name (such as specifying a directory that did not exist) but subsequently tried to correct the name by using the Save As feature or by loading a new database, you would still receive an error message. Typically, the error message would report that the specified location could not be found. This problem is fixed in CiscoSecure UNIX Server 1.0(1). [CSCdi57881 and CSCdi50395]

DES Password Support

In the CiscoSecure UNIX Server 1.0 GUI, when you specified the password to be Data Encryption Standard (DES) encrypted, the password was stored in clear text instead. In the CiscoSecure UNIX Server 1.0(1) GUI, when you specify DES encryption, a password is taken in clear text, but the CiscoSecure UNIX Server software later encrypts it by running the DES password generator and then inserting the resulting encrypted password into the database. [CSCdi50446]

Note A Challenge Handshake Authentication Protocol (CHAP) password cannot be encrypted. It will always appear as clear text in the AA database because the CHAP protocol requires the clear password and therefore disregards any transformation of it. [CSCdi50384]

Retrieving Database Files

In the CiscoSecure UNIX Server 1.0 GUI, you might have experienced problems retrieving database files in cases where you added a group and the software did not insert closing brackets around the group name. This problem has been corrected in the CiscoSecure UNIX Server 1.0(1) GUI. [CSCdi50426, CSCdi51121, CSCdi50460]

Note CiscoSecure UNIX Server software is supported on Solaris 2.5, SunOS 4.1.3 or 4.1.4. Note that neither SunOS 4.1.2 nor Solaris 2.4 are supported.

Validating Data

In CiscoSecure 1.0, you can typically enter invalid data such as protocols, dates, or time values that do not exist. The data is stored in the AA database but cannot be processed. In CiscoSecure 1.0(1) you are prompted in many cases to enter data in a correct format so that it can be recognized by and acted upon by the CiscoSecure server and the network access server. In other cases, invalid data might still be accepted by the AA database. Be careful to enter only valid attributes when prompted by the GUI. [CSCdi47134]

Console Stalls After Sequence of GUI commands

In CiscoSecure UNIX Server 1.0, if you enter a sequence of GUI commands, you might find the GUI console to stall or hang indefinitely in some conditions. In CiscoSecure UNIX Server 1.0(1), the overall functionality of the GUI is more reliable than its predecessor. [CSCdi50540]

String Dialog Box

In CiscoSecure UNIX Server 1.0, when you click the **User Attributes** button and click the **Cancel** button in the resulting String dialog box, you might view a Tool Command Language (TCL) error message: In CiscoSecure 1.0(1), TCL error messages typically do not display in the GUI. [CSC56081]

Cancelling from Create User Still Stores Name

In CiscoSecure UNIX Server 1.0, if you click the **Cancel** button in the Create User dialog box, the specified username is still recorded by the CiscoSecure server database. Consequently, if you attempt to recreate that username, the CiscoSecure UNIX Server software might disallow the operation because of duplicate names. This problem is resolved in CiscoSecure UNIX Server 1.0(1). [CSCdi50395, CSCdi57312]

Error Message of Unable to Parse File

In CiscoSecure UNIX Server 1.0, when setting user attributes, if you specify “Prohibit service” in combination with a filter option, you see an error message “Unable to Parse File.” In CiscoSecure 1.0(1), however, you can specify “Prohibit service” with any available filter option without generating an error message or causing the GUI to stall or become unresponsive. [CSCdi57913]

Misleading Message of “Protocol - Username too long”

In CiscoSecure UNIX Server 1.0, when the secret key for an access server does not match the one in the control file, the debugging output from the server incorrectly specifies “Protocol - Username too long.” In CiscoSecure UNIX Server 1.0(1), the server now specifies, “Protocol - mismatched encryption.” [CSCdi57951]

PPP Records Two Stop Packets

In CiscoSecure UNIX Server 1.0, when using CiscoSecure UNIX Server software with accounting functionality, and the router is using PPP, you might observe the accounting log to show two stop records for every session. One of the packets, recorded as stop, is actually a “watchdog” packet sent by the network access server during the beginning of the PPP negotiation. This watchdog packet is supposed to signify that the PPP negotiation has proceeded to the point where the IP address was assigned. However, CiscoSecure UNIX Server 1.0 does not recognize this watchdog, and instead writes the record as “stop.” In CiscoSecure UNIX Server 1.0(1), this packet is now correctly identified in the accounting log as a watchdog rather than a stop packet. [CSCdi59124]

Extra Line Deleted When Setting User Attributes

In CiscoSecure UNIX Server 1.0, an extra line is sometimes inadvertently removed by software while you set a user's attributes. In particular, you might observe this behavior when you highlight a service in the console and attempt to press **delete**. In this case, the subsequent line of some other attribute also becomes highlighted and deleted. This problem has been resolved in CiscoSecure UNIX Server 1.0(1). [CSCdi59802, CSCdi59796]

Assigning an Invalid Time Qualifier

In CiscoSecure UNIX Server 1.0, if you enter a time that does not exist, such as 2500, the GUI might hang indefinitely. In CiscoSecure UNIX Server 1.0(1), you see a message that declares the specified time to be invalid and you are prompted to re-enter a value for the time. [CSCdi60542, CSCdi57395, CSCdi58817]

TCL Error Message After Clicking Right Mouse-Button on User Name

In CiscoSecure UNIX Server 1.0, you might receive a TCL error message when you click the right mouse button on a username. In CiscoSecure UNIX Server 1.0(1), TCL error messages typically no longer display in the GUI. [CSCdi61186]

Deleting Users and Groups Without Warning

In CiscoSecure UNIX Server 1.0, you could delete user or group entries, add groups, or change group names, without first being prompted for confirmation. In CiscoSecure UNIX Server 1.0(1), however, when making such permanent database changes, you are first presented with a dialog box in which you must confirm your choices. [CSCdi63364]

TCL Error Message After Specifying No Password

In CiscoSecure UNIX Server 1.0, you might receive a TCL error message when you click **No Password** in the User Attributes window. In CiscoSecure UNIX Server 1.0(1), TCL error messages typically no longer display in the GUI. [CSCdi62437]

Deleting a User

Although this feature is not documented in the user guide, you can delete a user by taking the following steps:

Step 1 Select the name of the group in which the user you want to delete resides.

Step 2 Select **Open User List** from the View menu.

Step 3 Select the name of the user you want to delete.

Step 4 Select **Delete User** from the Users menu.

[CSCdi50445]

Protocol Services

Support for IP, IPX, Link Control Protocol (LCP), ARAP, and EXEC have been added to the list of service attributes in the GUI.

Specifying the Date

CiscoSecure UNIX Server software enables you to specify a date on which authentication or authorization attributes become available or unavailable. The following is an example of the correct format:

```
21 Mar 96
```

This format is required by the CiscoSecure UNIX Server. Although the CiscoSecure UNIX Server GUI does not place any restrictions on the values you enter in the date field, if you enter the date incorrectly, the CiscoSecure UNIX Server will not recognize it.

Extended Encryption Keys

The length of the encryption key shared by the CiscoSecure server and network access server has been extended to 255 characters. In earlier versions of CiscoSecure UNIX Server software, the length of a valid key could not exceed 31 characters.

Editing User Attributes

In CiscoSecure UNIX Server 1.0, if you try to edit user attributes from the Users menu, the Users menu might remain highlighted, but the Edit User Attributes window does not display. The GUI stalls and becomes unresponsive to mouse clicks or keyboard strokes. In CiscoSecure UNIX Server 1.0(1), however, this problem has been resolved. [CSCdi57405]

Syslog Logging Levels

CiscoSecure uses the logging levels DEBUG, INFO, NOTICE, WARNING, ERROR, and ALERT. These logging levels are tied to the operation of CiscoSecure UNIX Server software, not to the interaction of the software to its users. However, much of the output that will help you resolve such problems can be found in the DEBUG level. To examine the output, enable DEBUG logging and turn on the relevant bits in the control file. [CSCdi61830]

Quoted String Dialog Box

In CiscoSecure UNIX Server 1.0, when pressing the **Cancel** button in some of the edit dialog boxes, you might see the error message “Not enough undo info available” and the GUI will pause indefinitely. This problem is resolved in CiscoSecure UNIX Server 1.0(1). [CSCdi57405, CSCdi57059]

Known Problems with the CiscoSecure UNIX Server GUI

This section identifies shortcomings of the CiscoSecure UNIX Server GUI that are expected to be addressed in a subsequent release.

Clicking Cancel Causes GUI to Stall

In CiscoSecure UNIX Server 1.0(1), you might experience the GUI to stall or hang after you click the **Cancel** button of many of the dialog boxes. If you encounter this problem, restart the GUI and try the operation again. [CSCdi63554]

Assigning the Same Name to a Group and User

In CiscoSecure UNIX Server 1.0(1), the GUI does not acknowledge the difference between a group and user with the same name. This will be fixed in a subsequent release of CiscoSecure UNIX Server software. [CSCdi70114]

Multiple Expiration Dates

In CiscoSecure UNIX Server 1.0(1), the GUI enables you to specify dates when security services become available or unavailable. If you inadvertently enter two or more expiration dates, the GUI will record them; however, the actual date on which the specified service will expire is unpredictable. Furthermore, if you save the database containing multiple-conflicting expiration dates, you might not be able to reload the database. To avoid uncertainty in when a particular service will expire, make sure that you enter only one expiration date. [CSCdi47149]

Unclear Error Messages with Clear Password

In some cases, you might see a TCL script error message similar to the following when establishing a clear password.

```
Error in Tcl Script
Error: can’t read
“TacFMenu(NULL,function)”:
no such element in array
```

In such cases, clicking on any other buttons in the GUI will yield other TCL script errors. To recover from this condition, you must cancel from the User Create window then edit the user attributes. This problem will be fixed in a subsequent version. [CSCdi56081, CSCdi62437, CSCdi50457]

Setting Passwords that do not Expire

In some cases, you might want to set a password so that it does not expire. Although not immediately apparent, you can set expiration data in the AA database so that different expiration periods can be assigned to each password at the user level.

Take the following steps to set a password declaration which does not carry a qualification time to expire:

Step 1 From the GUI, create a user whose password you do not want to expire.

You see the User Attributes window.

Step 2 Click the **Password** button.

Step 3 Click **Clear** to display the Quoted String dialog box.

Step 4 Enter a password such as Pass2, then click the **OK** button.

The GUI next displays buttons (From and Until) to specify time qualification.

Step 5 Rather than specify a time qualification, click the **Finished** button at the bottom of the column of user attribute buttons.

Requiring Users to Modify Their Passwords

In some cases you might want the ability to require that users modify their passwords upon initial login to CiscoSecure UNIX Server. This ensures that only the user knows his or her own password. You can set up an initial password when a new account is first created, then the first time the user authenticates, he or she must change the password.

Note CiscoSecure requires a privilege level of 1 to allow a password to be changed. in each user's control file. By default, users log in with a privilege level of 1 so they are able to change their own passwords. When users change their own passwords, they must supply as few as 6 and as many as 13 characters. Of those characters, at least one number and one letter are required. However, as a CiscoSecure administrator, you can assign user passwords of any length or even forego a password requirement all together. As a CiscoSecure administrator, you can also prevent users from changing their own passwords by assigning them a password variable level that is greater than the user's. To assign a new password variable, modify the user's config_priv_level_for_own_CHPASS control file. (See Table 3-1, Variables in Software Control Files, found in the chapter, "Configuring CiscoSecure UNIX Server Software" in the CiscoSecure UNIX Server User Guide publication.)

Take the following steps to set up CiscoSecure UNIX Server software to require that users change their password when they authenticate to the network access server for the first time:

Step 1 Create a user from the GUI.

Step 2 Click the **Password** button.

Step 3 Click **Clear** to display the Quoted String dialog box.

Step 4 Enter a password such as Pass2, then click the **OK** button. (Remember to inform the user of this password.)

The GUI displays buttons (From and Until) to specify time qualification.

Step 5 Specify a qualification period, or date, that has already expired.

Note Because the expiration date for the validity of that password has already expired, the user will be required to enter a new one upon initial authentication.

User Guide Corrections

This section addresses errors in the *CiscoSecure UNIX Server User Guide* publication.

Restricting the Enable Command

In Appendix B, "CiscoSecure UNIX Server File Formats and Syntax," page B-6, the example that shows how the enable command can be restricted by means of aaa authorization is incorrect. The enable command is a function of the authentication process, not authorization.

To prevent the example user, joy, from enabling, the profile should be changed as follows:

```
user = joy {
    member = staff
    privilege = des "*" 2
    password = clear "My ClearText Password"
    service = ppp {
        default protocol = permit
        prohibit protocol = ipx
    }
    service = shell {
        default cmd = permit
        prohibit cmd = enable
    }
}
```

In the preceding example, the "*" creates a password that will never be matched by the CiscoSecure server and therefore will always be invalid. Furthermore, the example puts a privilege statement at level 2, thus preventing the user's password from being found at any preceding level.

Another way to prevent the example user from enabling, is simply to remove the "member = staff" statement from user's profile.

Pre and Post Functions

In Chapter 1, "Overview of CiscoSecure UNIX Server Software," page 1-6, preprocess and postprocess authorization functions are mentioned as supported features. While these features are in place, they remain undocumented because they are reserved for enhancements targeted for a later release of CiscoSecure UNIX Server. These structures remain private and are subject to change.

Similarly, on page 1-7, preaccounting and postaccounting functions are also mentioned as supported functions. Only preaccounting processing is supported.

The capability for postaccounting is in place but remains undocumented because the capability is reserved for enhancements targeted for a later release of CiscoSecure UNIX Server software.

Protocol Port Number Definition

In Chapter 2, "Getting Started With CiscoSecure UNIX Server Software," page 2-3, the example protocol port number definition for the TACACS+ service shown in Step 4 is shown incorrectly. It should read as follows:

```
tacacs 49/tcp TACACS+
```

Password in NAS Configuration Record

In Chapter 3, “Configuring CiscoSecure UNIX Server Software,” on page 3-8 you see a sample configuration file structure called NAS config_nas_config. The last entry in the control file structure is a definition as follows:

```
5 /* password expiration period, in days */
```

However, the samples directory installed as part of the release does not have this field in its NAS config_nas_config structure.

Disregard this entry (shown in the previous example).

Early versions of CiscoSecure UNIX Server software provided different expiration and warning periods for different network access servers. However, this proposed feature was removed in the 1.0 release. Note that a reference to this feature appears in the user guide.

Incorrect Display of kill Command

In Chapter 6, “Using CiscoSecure UNIX Server Software,” page 6-3, the example **kill** command for SIGUSR1 is shown incorrectly in that it uses a single quote instead of the back quote character. It should read as follows:

```
kill -SIGUSR1 `cat /etc/CiscoSecure.pid‘
```

Sample /etc/syslog.conf Entry

In Chapter 6, “Using CiscoSecure UNIX Server Software,” page 6-6, the sample /etc/syslog.conf entry is shown incorrectly. It should read as follows:

```
local0.debug    var/log/csuslog
```

The blank space between the two parameters must be created by pressing the **tab** key. Spaces entered by pressing the space bar will cause errors in the sample entry.

AA Database Grammar

In Appendix B, “CiscoSecure UNIX Server File Formats and Syntax,” page B-2, the AA Database grammar is incorrect. The correct grammar and statement of clarification follow:

```
  string = STRING | QSTRING
  password: FILESPEC string
| SYSTEM
| NO_PASSWORD
| DES QSTRING
| CLEAR QSTRING
| ARAP QSTRING
| PAP QSTRING
| STRING
| EXTERNAL STRING string
| Shadow for Solaris
```

Note 'STRING' is an unquoted string, QSTRING is a quoted string, and “string” is either STRING or QSTRING.

Accounting Record Format

In Chapter 5, “CiscoSecure UNIX Server Accounting,” on page 5-2, the accounting record format is incorrect. The correct format of the accounting record is structured as follows:

```
char    nas_name[]/* NAS name */
char    user_name[]/* username */
char    port_name[]/* port the connection is on */
char    remote_address[]/* where the user connected from */
char    record_type[]/* (start, update, stop etc) */
char    server[]/* hostname of the server, as an AV pair */
char    time[]/* time of this record, as an AV pair */
char    date[]/* date of this record, as an AV pair */
char    attribute_value_pairs[]/* there are an arbitrary number of these */
```

[CSCdi50422]

Storing Messages in Syslog

In Chapter 6, “Using CiscoSecure UNIX Server Software,” on page 6-6 the instruction on how to maintain a centralized database of messages is somewhat incomplete. Note that in order for information to go into the file, you must create the file /var/log/csuslog before syslog will store messages into it.

Message Catalogs

In Appendix B, “CiscoSecure UNIX Server File Formats and Syntax,” on page 3, the message catalog is incorrect. The following list identifies the correct default message IDs, message names, and message strings used by CiscoSecure UNIX Server software:

```
0 AUTHEN_CLIENT_LOGIN_PROMPT    “\nUser Access Verification\n”
1 AUTHEN_CLIENT_USERNAME_PROMPT       “Username: “
2 AUTHEN_CLIENT_PASSWORD_PROMPT       “Password: “
3 AUTHEN_CLIENT_SIGN_ON_MESSAGE       “”
4 AUTHEN_CLIENT_CHANGEPASS_INTRO  “Change password sequence”
5 AUTHEN_CLIENT_PASSWORDS_IDENTICAL  “Error - passwords the same”
6 AUTHEN_CLIENT_PASSWORD_EXPIRED  “Your password has expired”
7 AUTHEN_CLIENT_TOO_MANY_TRIES_FOR_USERNAME  “Too many tries for username”
8 AUTHEN_CLIENT_TOO_MANY_TRIES_FOR_PASSWORD  “Too many tries for password”
9 AUTHEN_CLIENT_NEW_PASSWORD1   “New password: “
10 AUTHEN_CLIENT_NEW_PASSWORD2   “New password again: “
11 AUTHEN_CLIENT_PASSWORDS_DIFFERENT  “The passwords are different”
12 AUTHEN_CLIENT_BAD_PASSWORD   “Bad password”
13 AUTHEN_CLIENT_CANT_CHANGE_PASSWORD  “You cannot change your password”
14 AUTHEN_CLIENT_ACCOUNT_EXPIRY_WARNING  “Your account will expire in %d days”
15 AUTHEN_CLIENT_PASSWORD_EXPIRY_WARNING   “Your password will expire in %d days”
16 AUTHEN_CLIENT_NEW_PASSWORD_CRITERIA “A password must be between six and thirteen
characters, containing at least one alphabetic and numeric character.”

18 AUTHEN_USER_NOT_FOUND        “Authentication - User not found”
19 AUTHEN_BAD_METHOD_FOR_USER   “Authentication - Bad method for user”
20 AUTHEN_BAD_TYPE              “Authentication - Bad type”
21 AUTHEN_NO_USERNAME           “Authentication - No username specified”
22 AUTHEN_INSUFFICIENT_PRIVILEGE  “Authentication - Insufficient privilege”
23 AUTHEN_UNEXPECTED_DATA       “Authentication - Unexpected data”
24 AUTHEN_UNEXPECTED_RESERVED_DATA  “Authentication - Unexpected reserved data”
25 AUTHEN_INCORRECT_PASSWORD    “Authentication - Incorrect password”
26 AUTHEN_ABORTED_SEQUENCE      “Authentication - Aborted sequence”
27 AUTHEN_FILEHANDLING_ERROR    “Authentication - File handling error”
28 AUTHEN_UNKNOWN_PASSWORD_TYPE  “Authen - Unknown password type”
29 AUTHEN_USER_NOT_IN_FILE      “Authentication - User not in file”
30 AUTHEN_ERROR_IN_EXTERNAL_FN,”Authentication - Error in external function”
```

```
31 AUTHEN_BAD_SERVICE           “Authentication - Bad Service”
32 AUTHEN_BAD_ACTION            “Authentication - Bad Action”
33 AUTHEN_SENDPASS_OK           “Authentication - SENDPASS (ok)”
34 AUTHEN_SENDPASS_FAIL         “Authentication - SENDPASS (fail)”

35 PROTOCOL_USERNAME_TOO_LONG   “Protocol - Username too long”
36 PROTOCOL_NASNAME_TOO_LONG    “Protocol - NAS name too long”
37 PROTOCOL_NASPORT_TOO_LONG    “Protocol - NAS port name too long”
38 PROTOCOL_NACADDR_TOO_LONG    “Protocol - NAC address too long”
39 PROTOCOL_BAD_PRIVILEGE       “Protocol - Invalid privilege field”
40 PROTOCOL_ACTIVE_SESSION      “Protocol - Session id in use”
41 PROTOCOL_NO_SESSION          “Protocol - No session found”
42 PROTOCOL_INCORRECT_TYPE      “Protocol - Incorrect type”
43 PROTOCOL_INCORRECT_SESSION   “Protocol - Incorrect session”
44 PROTOCOL_INCORRECT_SEQUENCE  “Protocol - Incorrect sequence”
45 PROTOCOL_INCORRECT_VERSION   “Protocol - Incorrect version”
46 PROTOCOL_GARBLED             “Protocol - Garbled message”
47 PROTOCOL_READ_TIMEOUT        “Protocol - Read timeout”
48 PROTOCOL_CONNECTION_CLOSED   “Protocol - Connection closed”
49 PROTOCOL_BAD_TYPE            “Protocol - Bad type”
50 PROTOCOL_MAX_USERS_EXCEEDED  “Maximum number of users exceeded”
51 PROTOCOL_ENCRYPTION_MISMATCH  “Mismatched encryption”

52 AUTHOR_NO_SERVICE            “Authorization - No service specified”
53 AUTHOR_FAILED_MANDATORY_ARG  “Authorization - Failed mandatory argument”
54 AUTHOR_FAILED_COMMAND_LINE   “Authorization - Failed command line”
55 AUTHOR_FAILED_SERVICE        “Authorization - Failed service”
56 AUTHOR_FAILED_TIME           “Authorization - Failed time qualification”
57 AUTHOR_BAD_ARGUMENT          “Authorization - Bad argument”
58 AUTHOR_NO_COMMAND            “Authorization - No command specified”
59 AUTHOR_FAILED_CMD            “Authorization - Failed command”
60 AUTHOR_NO_PROTOCOL           “Authorization - No protocol”
61 AUTHOR_UNKNOWN_USER          “Authorization - Unknown user”
62 AUTHOR_INVALID_NAS_OR_PORT   “Authorization - Unauthorized NAS or PORT”
63 AUTHOR_COMMAND_AUTHORIZED    “Authorization - Command authorized”
```

Cisco Connection Online

Cisco Connection Online (CCO), formerly Cisco Information Online (CIO), is Cisco Systems' primary, real-time support channel. Maintenance customers and partners can self-register on CCO to obtain additional content and services.

Available 24 hours a day, 7 days a week, CCO provides a wealth of standard and value-added services to Cisco's customers and business partners. CCO services include product information, software updates, release notes, technical tips, the Bug Navigator, configuration notes, brochures, descriptions of service offerings, and download access to public and authorized files.

CCO serves a wide variety of users through two interfaces that are updated and enhanced simultaneously—a character-based version and a multimedia version that resides on the World Wide Web (WWW). The character-based CCO supports Zmodem, Kermit, Xmodem, FTP, Internet e-mail, and fax download options, and is excellent for quick access to information over lower bandwidths. The WWW version of CCO provides richly formatted documents with photographs, figures, graphics, and video, as well as hyperlinks to related information.

You can access CCO in the following ways:

- WWW: `http://www.cisco.com`.
- Telnet: `cco.cisco.com`.
- Modem: From North America, 408 526-8070; from Europe, 33 1 64 46 40 82. Use the following terminal settings: VT100 emulation; databits: 8; parity: none; stop bits: 1; and baud rates up to 14.4 kbps.

For a copy of CCO's Frequently Asked Questions (FAQ), contact `cco-help@cisco.com`. For additional information, contact `cco-team@cisco.com`.

Note If you are a network administrator and need personal technical assistance with a Cisco product that is under warranty or covered by a maintenance contract, contact Cisco's Technical Assistance Center (TAC) at 800 553-2447, 408 526-7209, or `tac@cisco.com`. To obtain general information about Cisco Systems, Cisco products, or upgrades, contact 800 553-6387, 408 526-7208, or `cs-rep@cisco.com`.

This document is to be used in conjunction with the *CiscoSecure UNIX Server User Guide* publication.