HP TCP/IP Services for OpenVMS

Release Notes

July 2006

This document describes the new features and changes introduced with Version 5.6 of the HP TCP/IP Services for OpenVMS software product.

Revision/Update Information:	This is a new document.
Software Version:	HP TCP/IP Services for OpenVMS Version 5.6
Operating Systems:	OpenVMS I64 Version 8.3 OpenVMS I64 Version 8.2.1 OpenVMS Alpha Version 8.3 OpenVMS Alpha Version 8.2

Hewlett-Packard Company
Palo Alto, California

Printed in the US

The HP TCP/IP Services for OpenVMS documentation is available on CD-ROM.

This document was prepared using DECdocument, Version 3.3-1b.

Contents

3 Restrictions and Limitations

4 Corrections

5 Documentation Update

A Implementing NTP Autokeys

Tables

Preface

The HP TCP/IP Services for OpenVMS product is the HP implementation of the TCP/IP protocol suite and Internet services for OpenVMS Alpha and OpenVMS Industry Standard 64 for Integrity Servers (I64) systems. This document describes the latest release of the HP TCP/IP Services for OpenVMS product.

TCP/IP Services provides a comprehensive suite of functions and applications that support industry-standard protocols for heterogeneous network communications and resource sharing.

For installation instructions, see the *HP TCP/IP Services for OpenVMS Installation and Configuration* manual.

The release notes provide version-specific information that supersedes the information in the documentation set. The features, restrictions, and corrections in this version of the software are described in the release notes. Always read the release notes before installing the software.

Intended Audience

These release notes are intended for experienced OpenVMS and UNIX® system managers and assume a working knowledge of OpenVMS system management, TCP/IP networking, TCP/IP terminology, and some familiarity with the TCP/IP Services product.

Document Structure

These release notes are organized into the following chapters:

- Chapter 1 describes new features and special changes to the software that enhances its observed behavior.
- Chapter 2 describes changes to the installation, configuration, and startup procedures, and includes other related information that is not included in the *HP TCP/IP Services for OpenVMS Installation and Configuration* manual.
- Chapter 3 describes information about problems and restrictions, and includes notes describing changes to particular commands or services.
- Chapter 4 describes problems identified in previous versions of TCP/IP Services that have been fixed.
- Chapter 5 describes updates to information in the TCP/IP Services product documentation.

Related Documents

Table 1 lists the documents available with this version of TCP/IP Services.

Table 1 TCP/IP Services Documentation

Manual	Contents
HP TCP/IP Services for OpenVMS Concepts and Planning	This manual provides conceptual information about TCP/IP networking on OpenVMS systems, including general planning issues to consider before configuring your system to use the TCP/IP Services software. This manual also describes the other manuals in the TCP/IP Services documentation set and provides a glossary of terms and acronyms for the TCP/IP Services software product.
HP TCP/IP Services for OpenVMS Release Notes	The release notes provide version-specific information that supersedes the information in the documentation set. The features, restrictions, and corrections in this version of the software are described in the release notes. Always read the release notes before installing the software.
HP TCP/IP Services for OpenVMS Installation and Configuration	This manual explains how to install and configure the TCP/IP Services product.
HP TCP/IP Services for OpenVMS User's Guide	This manual describes how to use the applications available with TCP/IP Services such as remote file operations, e-mail, TELNET, TN3270, and network printing.
HP TCP/IP Services for OpenVMS Management	This manual describes how to configure and manage the TCP/IP Services product.
HP TCP/IP Services for OpenVMS Management Command Reference	This manual describes the TCP/IP Services management commands.
HP TCP/IP Services for OpenVMS Management Command Quick Reference Card	This reference card lists the TCP/IP management commands by component and describes the purpose of each command.
HP TCP/IP Services for OpenVMS UNIX Command Equivalents Reference Card	This reference card contains information about commonly performed network management tasks and their corresponding TCP/IP management and UNIX command formats.
HP TCP/IP Services for OpenVMS ONC RPC Programming	This manual presents an overview of high-level programming using open network computing remote procedure calls (ONC RPC). This manual also describes the RPC programming interface and how to use the RPCGEN protocol compiler to create applications.
HP TCP/IP Services for OpenVMS Guide to SSH	This manual describes how to configure, set up, use, and manage the SSH for OpenVMS software.
HP TCP/IP Services for OpenVMS Sockets API and System Services Programming	This manual describes how to use the Berkeley Sockets API and OpenVMS system services to develop network applications.
HP TCP/IP Services for OpenVMS SNMP Programming and Reference	This manual describes the Simple Network Management Protocol (SNMP) and the SNMP application programming interface (eSNMP). It describes the subagents provided with TCP/IP Services, utilities provided for managing subagents, and how to build your own subagents.

(continued on next page)

Table 1 (Cont.) TCP/IP Services Documentation

Manual	Contents
HP TCP/IP Services for OpenVMS Tuning and Troubleshooting	This manual provides information about how to isolate the causes of network problems and how to tune the TCP/IP Services software for the best performance. It also provides information about using UNIX network management utilities on OpenVMS.
HP TCP/IP Services for OpenVMS Guide to IPv6	This manual describes the IPv6 environment, the roles of systems in this environment, the types and function of the different IPv6 addresses, and how to configure TCP/IP Services to access the IPv6 network.

For additional information about HP OpenVMS products and services, visit the following World Wide Web address:

`http://www.hp.com/go/openvms`

For a comprehensive overview of the TCP/IP protocol suite, refer to the book *Internetworking with TCP/IP: Principles, Protocols, and Architecture*, by Douglas Comer.

Reader's Comments

HP welcomes your comments on this manual. Please send comments to either of the following addresses:

Internet	**openvmsdoc@hp.com**
Postal Mail	Hewlett-Packard Company OSSG Documentation Group, ZKO3-4/U08 110 Spit Brook Rd. Nashua, NH 03062-2698

How to Order Additional Documentation

For information about how to order additional documentation, visit the following World Wide Web address:

`http://www.hp.com/go/openvms/doc/order`

Conventions

In the product documentation, the name TCP/IP Services means any of the following:

- HP TCP/IP Services for OpenVMS Alpha
- HP TCP/IP Services for OpenVMS I64
- HP TCP/IP Services for OpenVMS VAX

In addition, please note that all IP addresses are fictitious.

The following conventions are used in the documentation.

Ctrl/*x*	A sequence such as Ctrl/*x* indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.

PF1 *x*	A sequence such as PF1 *x* indicates that you must first press and release the key labeled PF1 and then press and release another key or a pointing device button.
Return	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.) In the HTML version of this document, this convention appears as brackets, rather than a box.
. . .	A horizontal ellipsis in examples indicates one of the following possibilities: • Additional optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
. . .	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
( )	In command format descriptions, parentheses indicate that you must enclose choices in parentheses if you specify more than one.
[ ]	In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for OpenVMS directory specifications and for a substring specification in an assignment statement.
|	In command format descriptions, vertical bars separate choices within brackets or braces. Within brackets, the choices are optional; within braces, at least one choice is required. Do not type the vertical bars on the command line.
{ }	In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line.
bold type	Bold type represents the introduction of a new term. It also represents the name of an argument, an attribute, or a reason.
italic type	Italic type indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error *number*), in command lines (/PRODUCER=*name*), and in command parameters in text (where *dd* represents the predefined code for the device type).
UPPERCASE TYPE	Uppercase type indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
`Example`	This typeface indicates code examples, command examples, and interactive screen displays. In text, this type also identifies URLs, UNIX commands and pathnames, PC-based commands and folders, and certain elements of the C programming language.
-	A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.

numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

1
New Features and Behavioral Enhancements

This chapter describes new features of TCP/IP Services Version 5.6 as well as behavioral enhancements.

Note

TCP/IP Services Version 5.6 is supported on OpenVMS Alpha and OpenVMS Industry Standard 64 for Integrity Servers (I64) systems only. On VAX systems, use TCP/IP Services Version 5.3.

To use TCP/IP Services Version 5.6, you must upgrade to OpenVMS Version 8.2 or higher.

For information about installing and configuring TCP/IP Services, see the *HP TCP/IP Services for OpenVMS Installation and Configuration* guide.

Table 1–1 lists the new features of TCP/IP Services Version 5.6 and the sections that describe them.

Table 1–1 TCP/IP Services for OpenVMS New Features

Feature	Section	Description
BIND 9 Resolver	1.1	This release includes a new version of the BIND resolver.
DNS/BIND V9.3 Server	1.2	This release includes an updated BIND server codebase.
Integrate Tru64 BL26 Updates	1.3	This release incorporates several critical bug fixes in the Tru64 UNIX-based kernel and management utilities.
NFS Client TCP Support	1.4	The NFS client joins the server in offering the ability to run over TCP.
NFS Server Support for Integrity	1.5	The NFS server is now operational and supported on the OpenVMS I64 platform.
NFS Symbolic Link Support	1.6	The NFS server now recognizes symbolic links and can create them as necessary.
NTP Security Update (SSL)	1.7	New NTP features offer cryptographic security.
SMTP Multiple Domains in a Zone	1.8	SMTP now recognizes more than one domain name for direct local delivery.
SSH Upgrade with Kerberos Support	1.9	Several improvements have been made to SSH.

(continued on next page)

Table 1–1 (Cont.) TCP/IP Services for OpenVMS New Features

Feature	Section	Description
TELNET Upgrade with Kerberos Support	1.10	The TELNET server and client are now supported with the upgraded Kerberos version that ships with OpenVMS V8.3.
TELNET Server Device Limit	1.11	The TELNET server is no longer limited to 9999 sessions for TN devices.
IPv6 Support for LPD and TELNETSYM	1.12	Both LPD and TELNETSYM printing software now allow you to print via the IPv6 transport.
FTP Performance Enhancements for VMS Plus Mode	1.13	The FTP service has been streamlined.
Improved Interface Configuration in TCPIP$CONFIG	1.14	The menu-driven process of defining local interfaces and IP addresses has been significantly reworked to provide better support for failSAFE IP.
Added TSIG-based Authentication Support to the Load Broker	1.15	Added TSIG-based authentication support to the Load Broker.

1.1 BIND 9 Resolver

This release includes a new version of the BIND resolver that brings several API updates including thread-safety for the getaddrinfo( ) and getnameinfo( ) routines. It also brings new features, including the ability to resolve DNS entries via the IPv6 transport. This represents a major upgrade from V5.5 and other recent releases, which provided resolver functionality based on BIND8.

1.2 DNS/BIND V9.3 Server

This release updates the BIND server to Version 9.3.1, which brings several incremental improvements related to security and stability.

1.3 Integrate Tru64 BL26 Updates

Several critical bug fixes in the Tru64 UNIX-based kernel and management utilities were incorporated.

1.4 NFS Client TCP Support

The NFS client joins the server in offering the ability to run over TCP, in addition to the more-traditional UDP mode of operation. This can be useful when mounting filesystems across a Wide Area Network or traversing a firewall.

1.5 NFS Server Support for Integrity

This release includes NFS Server Support for OpenVMS I64 platforms.

1.6 NFS Symbolic Link Support

The NFS server now recognizes symbolic links and can create them as necessary.

1.7 NTP Security Update (SSL)

New NTP features offer cryptographic security, enhancing the protection against an attacker trying to compromise the accuracy of your system clock. For more information, see Appendix A.

1.8 SMTP Multiple Domains in a Zone

During periods of organizational transition such as mergers, it is common for more than one domain name to be in use on a corporate intranet. SMTP will now recognize more than one domain name.

1.9 SSH Upgrade with Kerberos Support

TCP/IP Services for OpenVMS 5.6 introduces SSH support for Kerberos, the popular network authentication protocol from Massachusetts Institute of Technology. SSH password authentication method has been enhanced to support Kerberos. Three new SSH authentication methods based on Kerberos are now supported:

- `gssapi-with-mic`
- `kerberos-2@ssh.com` ("kerberos-2" is used synonymously with "kerberos-2@ssh.com")
- `kerberos-tgt-2@ssh.com` ("kerberos-tgt-2" is used synonymously with "kerberos-tgt-2@ssh.com")

The `kerberos-2@ssh.com` and `kerberos-tgt-2@ssh.com` authentication methods are proprietary, not specified by an IETF draft or RFC, and as such are supported only by the SSH implementations based on software from SSH Communications Inc. Tru64 UNIX support also these two authentication methods.

The `gssapi-with-mic` authentication method is based on an IETF draft (GSSAPI Authentication and Key Exchange for the Secure Shell Protocol). As a public domain specification, it is supported by a broader range of SSH implementations including those based on OpenSSH. TCP/IP Services does not implement the key exchange part of the "GSSAPI Authentication and Key Exchange for the Secure Shell Protocol" draft. It implements only the user authentication portion of this specification.

The SSH server in this version of TCP/IP Services supports Kerberos for OpenVMS Version V2.1 and higher. For more information about Kerberos for OpenVMS, refer to the *HP Open Source Security for OpenVMS, Volume 3: Kerberos* manual.

1.9.1 Forwarding of Credentials

Kerberos provides the ability for applications like SSH to forward credentials from client host to server host, obviating the need for the user to re-enter their Kerberos password each time they use a Kerberized application. For example, with credentials forwarding a user on HOSTA could issue a `kinit` command, connect with SSH from HOSTA to HOSTB and then, once logged into HOSTB, they could connect on to HOSTC without issuing a `kinit` command in their user process on HOSTB. They only entered the `kinit` command on HOSTA and their credentials "followed" them to their session on HOSTB and then on to their session on HOSTC.

The `-f` option on the SSH command indicates that a forwardable TGT is to be produced.

The Kerberized application must also support credentials forwarding. The kerberos-tgt-2 supports credentials being forwarded from the client to the server process.

The kerberos-2 method does not support forwarding of the user's Kerberos credentials to the process on the SSH server host. An application that uses Kerberos from the process on the server side requires the user to enter another kinit command.

The gssapi-with-mic method supports forwarding of the user's Kerberos credentials to the user's process on the SSH server. However, the OpenVMS SSH server does not support this feature. Therefore, when connecting to the OpenVMS SSH server using gssapi-with-mic authentication, the user's Kerberos credentials from the client will not be propagated to the user's process on the server.

Note

Any use of a Kerberized application from the server side process requires the user to issue another kinit command in that process.

For information about how to enable SSH server support for Kerberos, see the *HP TCP/IP Services for OpenVMS Guide to SSH*.

The following example illustrates how to obtain a forwardable TGT.

```
!!! User issues kinit with -f to get a forwardable TGT.
!!! In this example the Kerberos principal user name is lower case and
!!! the realm is uppercase.
SYSA> kinit -f "smith"
Password for smith@SYSA.XYZ.COM:

!!! Connect to system "sysb" forcing use of kerberos-tgt-2 authentication
!!! method.
SYSA> ssh -o"AllowedAuthentications kerberos-tgt-2@ssh.com" smith@sysb
Authentication successful.

 Welcome to HP OpenVMS Industry Standard 64 Evaluation Release V8.2

!!! We've been allowed in. A klist -f (-f for "full") shows that we have a
!!! TGT without having issued a kinit command on SYSB.
SYSB> klist -f
Ticket cache: FILE:WORK10$:[SMITH.KRB.SYSB.TMP]KRB5CC_1480589921
Default principal: smith@SYSA.XYZ.COM

Valid starting     Expires            Service principal
09/22/05 14:18:53  09/23/05 00:17:16  krbtgt/SYSA.XYZ.COM@SYSA.XYZ.COM
        Flags: FfT


Kerberos 4 ticket cache: krb$user:[tmp]k4_tkt_cache33488912
KRB$KLIST: You have no tickets cached

!!! Now use ssh to connect back to sysa but this time use the simpler
!!! kerberos-2 authentication method.
SYSB> ssh -o"AllowedAuthentications kerberos-2@ssh.com" smith@sysa
Authentication successful.

UNAUTHORIZED ACCESS PROHIBITED OpenVMS AXP (TM) Operating System, Version V8.2

!!! We have been allowed in but have no TGT created for us because we
!!! used kerberos-2:
SYSA> klist -f
KRB$KLIST: No credentials cache found (ticket cache FILE:krb$user:[tmp]krb5cc_33488912)
```

```
Kerberos 4 ticket cache: krb$user:[tmp]k4_tkt_cache33488912
KRB$KLIST: You have no tickets cached
```

1.9.2 Password Authentication

In addition, the OpenVMS SSH server provides an optional Kerberos password check. In password authentication mode, the SSH server checks the password against Kerberos before checking it against SYSUAF. If the Kerberos password check passes then the SSH server considers the SSH password authentication successful and the user is allowed in. If not, the password authentication continues on with the SYSUAF check.

When the Kerberos password check succeeds, the SSH server provides to the user process on the server system a forwardable TGT so that the user need not issue a `kinit` command once logged in. Essentially, the SSH server does a `kinit` on behalf of the user.

This feature is not enabled by default. Use the `TryKerberosPassword` to enable this feature.

Note

The check of the user password against Kerberos is transparent to the SSH client software and is performed entirely on the SSH server. The SSH client software is unaware of how the password is processed by the SSH server. This approach has the advantage of allowing use of Kerberos features from a client host that doesn't have Kerberos configured. The only awareness of Kerberos required on the SSH client side is the knowledge of the user that they may enter their Kerberos password (which may very well be different from the password to their account on the server host) in response to the SSH client's password cue.

Because there is no knowledge on the part of the SSH client software that the SSH server is passing the user password to Kerberos for validation, there is no way for the SSH client user to specify the Kerberos principal name to be used by the SSH server for the Kerberos password check. Therefore the SSH server must compose the Kerberos principal name for the password check using a common sense heuristic. The SSH server uses the target username being logged into on the SSH server system for the username part of the principal and the local Kerberos realm as the principal's realm name. For example, if the SSH server's Kerberos realm was SYSA.XYZ.COM and the user account to be logged into was "smith" then the Kerberos principal used for the Kerberos password check would be smith@SYSA.XYZ.COM.

1.9.3 Logicals Defined by SSH Startup

In order to use the `gssapi-with-mic` authentication method on an OpenVMS host with Kerberos for OpenVMS Version V2.1, the SSH server and client startup procedures define a logical name TCPIP$SSH_KRBRTL_HACK. The presence of this logical tells the SSH client and server to perform steps to circumvent a problem with images that use LIB$FIND_IMAGE_SYMBOL to access both KRB$RTL32.EXE and GSS$RTL32.EXE.

The SSH server and client startup procedures will define TCPIP$SSH_KRBRTL_HACK based on the version of Kerberos running on your system and not whether Kerberos is actually in use on your system or configured to be used by SSH.

If you are running Kerberos for OpenVMS Version V3.0 or higher, the SSH server and client startup procedures will not define this logical, because the steps needed to make GSS$RTL32 work properly with LIB$FIND_IMAGE_SYMBOL are not needed.

1.9.4 Using Kerberos KDC/DNS

To configure Kerberos KDC/DNS, include fully qualified host principals. For example, a host principal for the SSH server host with DNS name myhost.abcd.org in the Kerberos realm ABCD.ORG would be "host/myhost.abcd.org@ABCD.ORG".

For SSH purposes the DNS host name part of the host principal should be fully qualified. The SSH server's checking of the client user's password against Kerberos in password authentication also requires a fully qualified host principal for the SSH server host.

You must define a Kerberos host principal for an SSH client host that is also to serve as an SSH server for the Kerberos-based authentication methods and for the password authentication Kerberos password check.

In addition, to use the gssapi-with-mic authentication method, the first name in the list returned from a TCPIP SHOW HOST/LOCAL command entered on the SSH server for the SSH server must be its fully-qualified canonical name.

For example, say the SSH server host name is myhost.abcd.org. This example illustrates two possible local host database entries for SSH server myhost.abcd.org on myhost.abcd.org. The first example prevents the gssapi-with-mic authentication method from working:

Example 1

```
MYHOST> tcpip show host/local myhost

    LOCAL database

Host address    Host name

10.0.0.1   myhost, myhost.abcd.org, MYHOST, MYHOST.ABCD.ORG
```

The following example shows how to define the host name so that the gssapi-with-mic authentication method works:

Example 2

```
MYHOST> tcpip show host/local myhost

     LOCAL database

Host address    Host name

10.0.0.1   myhost.abcd.org, myhost, MYHOST,MYHOST.ABCD.ORG
```

If your configuration requires a local host database entry as shown in Example 1, then gssapi-with-mic will not work for you.

1.9.5 New Configuration Parameters

This version of SSH recognizes the following new configuration parameters.

- In the server configuration file (SSHD_CONFIG.):
  - TryKerberosPassword
  - GssapiSendError
- In the client configuration file (SSH_CONFIG.):
  - GssapiDelegateCredentials

- In both the client and server configuration files:
  - GssapiSendErrtok

For more information about these configuration parameters, see the *HP TCP/IP Services for OpenVMS Guide to SSH*.

1.10 TELNET Upgrade with Kerberos Support

The TELNET server and client are now supported with the upgraded Kerberos version that ships with OpenVMS V8.3.

1.11 TELNET Server Device Limit

The TELNET server is no longer limited to 9999 sessions or TN devices.

1.12 IPv6 Support for LPD and TELNETSYM

Continuing our work to offer IPv6 support throughout the product, both LPD and TELNETSYM printing software now allow you to print via the IPv6 transport.

1.13 FTP Performance Enhancements for VMS Plus Mode

Streamlining was performed for the FTP service, specifically addressing the case where both server and client are OpenVMS systems.

1.14 Improved Interface Configuration in TCPIP$CONFIG

The menu-driven process of defining local interfaces and IP addresses has been significantly reworked to provide better support for failSAFE IP.

1.15 Added TSIG-based Authentication Support to the Load Broker

The Load Broker can now transact secure dynamic updates with a BIND server.

2

Installation, Configuration, Startup, and Shutdown

This chapter includes notes and changes made to the installation and configuration of TCP/IP Services, as well as startup and shutdown procedures. Use this chapter in conjunction with the *HP TCP/IP Services for OpenVMS Installation and Configuration* manual.

2.1 Installing Over V5.3 Early Adopter's Kits (EAKs)

If you have installed one or more of the following V5.3 EAKs, you must use the PCSI REMOVE command to remove the EAKs before you install TCP/IP Services V5.5:

- SSH for OpenVMS EAK
- failSAFE IP EAK

Note

If you install the current TCP/IP Services version after removing the failSAFE IP EAK, you must run TCPIP$CONFIG.COM to reestablish your target and home interfaces.

2.2 Upgrading from TCP/IP Services Version 4.*x*

Upgrading from versions prior to V5.0 has not been qualified for this release.

2.3 Adding a System to an OpenVMS Cluster

The TCPIP$CONFIG.COM configuration procedure for TCP/IP Services Version 5.6 creates OpenVMS accounts using larger system parameter values than in previous versions. Only new accounts get these larger values. These values are useful on OpenVMS Alpha systems but essential on OpenVMS I64 systems.

To have your OpenVMS I64 system join an OpenVMS Cluster as a TCP/IP host, HP recommends adding the system to the cluster before you configure TCP/IP Services. The guidelines in Section 2.3.1 assume you have followed this recommendation.

If you configure TCP/IP Services before you add the system to a cluster, see Section 2.3.2.

2.3.1 Running a Newly Configured Host on the Cluster

The following recommendations assume you are configuring TCP/IP Services on the system after having added the system to the OpenVMS Cluster.

If TCP/IP Services has previously been installed on the cluster and you encounter problems running a TCP/IP component on the system, modify the cluster System Authorization File (SYSUAF) to raise the parameter values for the account used by the affected component. The minimum recommended values are listed in Table 2–1.

Table 2–1 Minimum Values for SYSUAF Parameters

Parameter	Minimum Value
ASTLM	100
BIOLM	400
BYTLM	108000
DIOLM	50
ENQLM	100
FILLM	100
PGFLQUOTA[1]	50000
TQELM	50
WSEXTENT	4000
WSQUOTA	1024

[1]This parameter's value setting is especially critical.

The IMAP, DHCP, and XDM components can exhibit account parameter problems if the value assigned to PGFLQUOTA or to any of the other listed parameters is too low. Use the OpenVMS AUTHORIZE utility to modify SYSUAF parameters. For more information, see *HP OpenVMS System Management Utilities Reference Manual: A-L*.

2.3.2 Configuring TCP/IP Services Before Adding the System to the Cluster

If you configure TCP/IP Services before you add the system to a cluster, when you add the system to the cluster the owning UIC for each of the TCP/IP service SYS$LOGIN directories (TCPIP$*service-name*, where *service-name* is the name of the service) may be incorrect. Use the OpenVMS AUTHORIZE utility to correct these UICs.

2.3.3 Disabling or Enabling SSH Server

When you use the TCPIP$CONFIG.COM configuration procedure to disable or enable the SSH server, the following prompt is displayed:

```
* Create a new default Server host key? [YES]:
```

Unless you have a specific reason for creating a new default server host key, you should enter "N" at this prompt. If you accept the default, clients with the old key will need to obtain the new key. For more information, see Section 3.10.6.

2.4 SSH Configuration Files Must Be Updated

Note that this section refers to upgrades from a version prior to V5.4 ECO.

The SSH client and server on this version of TCP/IP Services cannot use configuration files from previous versions of SSH.

If the SSH client and server detect systemwide configuration files from an older version of SSH, the client and server will fail to start. The client will display the following warning message, and the server will write the following warning message to the SSH_RUN.LOG file:

```
You may have an old style configuration file. Please follow the
instructions in the release notes to use the new configuration
files.
```

If the SSH client detects a user-specific configuration file from an older version of SSH, the SSH client will display the warning and will allow the user to proceed.

To preserve the modifications made to the SSH server configuration file and the SSH client configuration file, you must edit the templates provided with the new version of SSH, as follows:

1. Extract the template files using the following commands:

   ```
   $ LIBRARY/EXTRACT=SSH2_CONFIG SYS$LIBRARY:TCPIP$TEMPLATES.TLB -
   _$ /OUT=TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]SSH2_CONFIG.
   $ LIBRARY/EXTRACT=SSHD2_CONFIG SYS$LIBRARY:TCPIP$TEMPLATES.TLB -
   _$ /OUT=TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]SSHD2_CONFIG.
   ```

   These commands copy the new template files into the SSH2 configuration directory with a new version number.

2. Copy the modifications made in the old versions of the configuration files to the new versions.

3. Start SSH using the following command:

   ```
   $ @SYS$STARTUP:SSH_STARTUP.COM
   $ @SYS$STARTUP:SSH_CLIENT_STARTUP.COM
   ```

2.5 Troubleshooting SMTP and LPD Shutdown Problems

If SMTP or LPD shutdown generates errors indicating that the queue manager is not running, check your site-specific shutdown command procedure (VMS_SYSHUTDOWN.COM). If this procedure contains the command to stop the queue manager (STOP/QUEUE/MANAGER), make sure this command is after the command that runs the TCPIP$SHUTDOWN.COM command procedure.

Note

You do not have to stop the queue manager explicitly. The queue manager is automatically stopped and started when you restart the system.

3
Restrictions and Limitations

This chapter provides information about problems and restrictions in the current version of TCP/IP Services, and also includes other information specific to a particular command or service, such as changes in command syntax or messages.

3.1 Netstat Utility -z Option No Longer Implemented

In this version of TCP/IP Services for OpenVMS, the -z option to the netstat utility is no longer implemented. It has not been determined whether future versions of TCP/IP Services will restore this functionality.

3.2 Manually Configuring an Interface as DHCP Leads to Startup Problems

Manually configuring an interface to be managed via DHCP may lead to an error, TCPIP-E-DEFINTE, when starting TCP/IP. This causes TCP/IP to not start properly. To work around this problem, shutdown TCP/IP, then on the interface that was manually configured as DHCP, issue the following command: `$ tcpip set config inter` *ifname*`/PRIMARY` Now restart TCP/IP.

3.3 SLIP Restrictions

The serial line IP protocol (SLIP) is not supported in this release.

3.4 Advanced Programming Environment Restrictions and Guidelines

The header files provided in TCPIP$EXAMPLES are provided as part of the advanced TCP/IP programming environment. The following list describes restrictions and guidelines for using them:

- Use of the functions and data structures described in TCPIP$EXAMPLES:RESOLV.H is limited to 32-bit pointers. The underlying implementation will only handle 32-bit pointers. Previously, 64-bit pointers were wrongly accepted, resulting in undefined behavior for the underlying implementation.
- The IP.H and IP6.H header files are incomplete in the OpenVMS environment. They contain `include` directives for header files that are not provided in this version of TCP/IP Services. Refer to the *HP TCP/IP Services for OpenVMS Sockets API and System Services Programming* for more information.

3.5 BIND/DNS Restrictions

BIND Version 9 has the following restrictions:

- Certain DNS server implementations do not support AAAA (IPv6 address) records. When queried for an AAAA (IPv6) record type by the BIND resolver, these name servers will return an NXDOMAIN status, even if an A (IPv4) record exists for the same domain name. These name servers should be returning NOERROR as the status for such a query. This problem can result in delays during host name resolution.

  BIND Version 9.3.1, which is supported with this release of TCP/IP Services, and prior versions of BIND do not exhibit this problem.

- Serving secure zones

  When acting as an authoritative name server, BIND Version 9 includes KEY, SIG, and NXT records in responses as specified in RFC 2535 when the request has the DO flag set in the query.

- Secure resolution

  Basic support for validation of DNSSEC signatures in responses has been implemented but should be considered experimental.

  When acting as a caching name server, BIND Version 9 is capable of performing basic DNSSEC validation of positive as well as nonexistence responses. You can enable this functionality by including a `trusted-keys` clause containing the top-level zone key of the DNSSEC tree in the configuration file.

  Validation of wildcard responses is not currently supported. In particular, a "`name does not exist`" response will validate successfully even if the server does not contain the NXT records to prove the nonexistence of a matching wildcard.

  Proof of insecure status for insecure zones delegated from secure zones works when the zones are completely insecure. Privately secured zones delegated from secure zones will not work in all cases, such as when the privately secured zone is served by the same server as an ancestor (but not parent) zone.

  Handling of the CD bit in queries is now fully implemented. Validation is not attempted for recursive queries if CD is set.

- Secure dynamic update

  Dynamic updating of secure zones has been partially implemented. Affected NXT and SIG records are updated by the server when an update occurs. Use the `update-policy` statement in the zone definition for advanced access control.

- Secure zone transfers

  BIND Version 9 does not implement the zone transfer security mechanisms of RFC 2535 because they are considered inferior to the use of TSIG or SIG(0) to ensure the integrity of zone transfers.

- SSL$LIBCRYPTO_SHR32.EXE requirement

  In this version of TCP/IP Services, the BIND Server and related utilities have been updated to use the OpenSSL shareable image SSL$LIBCRYPTO_SHR32.EXE. There is now a requirement that this shareable image from

OpenSSL V1.2 or higher be installed on the system before starting the BIND Server. It must also be installed before using the following BIND utilities:

```
BIND_CHECKCONF
BIND_CHECKZONE
DIG
DNSSEC_KEYGEN
DNSSEC_SIGNZONE
HOST
NSUPDATE
RNDC_CONFGEN
```

3.6 IPv6 Restrictions

The following sections describe restrictions in the use of IPv6.

3.6.1 Mobile IPv6 Restrictions

Mobile IPv6 is not supported in this release.

3.6.2 IPv6 Requires the BIND Resolver

If you are using IPv6, you must enable the BIND resolver. To enable the BIND resolver, use the TCPIP$CONFIG.COM command procedure. From the `Core environment` menu, select BIND Resolver.

You must specify the BIND server to enable the BIND resolver. If you do not have access to a BIND server, specify the node address 127.0.0.1 as your BIND server.

3.7 NFS Restrictions on Alpha Platforms

The following sections describe problems and restrictions with NFS on Alpha platforms.

3.7.1 NFS Server Problems and Restrictions

The following restrictions apply to the NFS server on OpenVMS Alpha systems:

- When performing a mount operation or starting the NFS server with OPCOM enabled, the TCP/IP Services MOUNT server can erroneously display the following message:

  `%TCPIP-E-NFS_BFSCAL, operation MOUNT_POINT failed on file /`*`dev`*`/`*`dir`*

  This message appears even when the MOUNT or NFS startup has successfully completed. In the case of a mount operation, if it has actually succeeded, the following message will also be displayed:

  `%TCPIP-S-NFS_MNTSUC, mounted file system /`*`dev`*`/`*`dir`*

- If the NFS server and the NFS client are in different domains and unqualified host names are used in requests, the lock server (LOCKD) fails to honor the request and leaves the file unlocked.

  When the server attempts to look up a host using its unqualified host name (for example, `johnws`) instead of the fully qualified host name (for example, `johnws.abc com`), and the host is not in the same domain as the server, the request fails.

  To solve this type of problem, you can do one of the following:

  – When you configure the NFS client, specify the fully qualified host name, including the domain name. This ensures that translation will succeed.

– Add an entry to the NFS server's hosts database for the client's unqualified host name. Only that NFS server will be able to translate this host name. This solution will not work if the client obtains its address dynamically from DHCP.

3.7.2 NFS Client Problems and Restrictions

- To get proper timestamps, when the system time is changed for daylight savings time (DST), dismount all DNFS devices. (The TCP/IP management command SHOW MOUNT should show zero mounted devices.) Then remount the devices.
- The NFS client does not properly handle file names with the semicolon character on ODS-5 disk volumes. (For example, `a^;b.dat;5` is a valid file name.) Such file names are truncated at the semicolon.
- The NFS client included with TCP/IP Services uses the NFS Version 2 protocol only.
- With the NFS Version 2 protocol, the value of the file size is limited to 32 bits.
- The ISO Latin-1 character set is supported. The UCS-2 characters are not supported.
- File names, including file extensions, can be no more than 236 characters long.
- Files containing characters not accepted by ODS-5 on the active OpenVMS version or whose name and extension exceeds 236 characters are truncated to zero length. This makes them invisible to OpenVMS and is consistent with prior OpenVMS NFS client behavior.

3.8 NTP Problems and Restrictions

The NTP server has a stratum limit of 15. The server does not synchronize to any time server that reports a stratum of 15 or greater. This may cause problems if you try to synchronize to a server running the UCX NTP server, if that server has been designated as "free running" (with the `local-master` command). For proper operation, the `local-master` designation must be specified with a stratum no greater than 14.

3.9 SNMP Problems and Restrictions

This section describes restrictions to the SNMP component for this release. For more information about using SNMP, refer to the *HP TCP/IP Services for OpenVMS SNMP Programming and Reference* manual.

3.9.1 Incomplete Restart

When the SNMP master agent and subagents fail or are stopped, TCP/IP Services is often able to restart all processes automatically. However, under certain conditions, subagent processes may not restart. When this happens, the display from the DCL command SHOW SYSTEM does not include TCPIP$OS_MIBS and TCPIP$HR_MIB. If this situation occurs, restart SNMP by entering the following commands:

```
$ @SYS$STARTUP:TCPIP$SNMP_SHUTDOWN.COM
$ @SYS$STARTUP:TCPIP$SNMP_STARTUP.COM
```

3.9.2 SNMP IVP Error

On slow systems, the SNMP Installation Verification Procedure can fail because a subagent does not respond to the test query. The error messages look like this:

```
   .
   .
   .
Shutting down the SNMP service... done.

Creating temporary read/write community SNMPIVP_153.

Enabling SET operations.

Starting the SNMP service... done.

SNMPIVP: unexpected text in response to SNMP request:
"- no such name - returned for variable 1"
See file SYS$SYSDEVICE:[TCPIP$SNMP]TCPIP$SNMP_REQUEST.DAT for more
details.
sysContact could not be retrieved.  Status = 0
The SNMP IVP has NOT completed successfully.
SNMP IVP request completed.
Press Return to continue ...
```

You can ignore these types of messages in the IVP.

3.9.3 Using Existing MIB Subagent Modules

If an existing subagent does not execute properly, you may need to relink it against the current version of TCP/IP Services to produce a working image. Some subagents (such as those for HP Insight Management Agents for OpenVMS) also require a minimum version of OpenVMS and a minimum version of TCP/IP Services.

The following restrictions apply:

- In general, only executable images linked against the following versions of the eSNMP shareable image are upward compatible with the current version of TCP/IP Services:
  - UCX$ESNMP_SHR.EXE from TCP/IP Services Version 4.2 ECO 4
  - TCPIP$ESNMP_SHR.EXE from TCP/IP Services Version 5.0A ECO 1

  Images built under versions other than these can be relinked with one of the shareable images, or with TCPIP$ESNMP_SHR.EXE in the current version of TCP/IP Services.
- The underlying eSNMP API changed from DPI in TCP/IP Services Version 5.0 to AgentX in later versions of TCP/IP Services. Therefore, executable images linked against older object library versions of the API (*$ESNMP.OLB) must be relinked against either the new object library or the new shareable image. Linking against the shareable image ensures future upward compatibility and results in smaller image sizes.

  Note

  Although images may run without being relinked, backward compatibility is not guaranteed. Such images can result in inaccurate data or run-time problems.

- This version of TCP/IP Services provides an updated version of the UCX$ESNMP_SHR.EXE shareable image to provide compatibility with subagents linked under TCP/IP Services Version 4.2 ECO 4. Do not delete this file.
- The SNMP server responds correctly to SNMP requests directed to a cluster alias. Note, however, that an unexpected host may be reached when querying from a TCP/IP Services Version 4.*x* system that is a member of a cluster group but is not the current impersonator.
- The SNMP master agent and subagents do not start if the value of the logical name TCPIP$INET_HOST does not yield the IP address of a functional interface on the host when used in a DNS query. This problem does not occur if the server host is configured correctly with a permanent network connection (for example, Ethernet or FDDI). The problem can occur when a host is connected through PPP and the IP address used for the PPP connection does not match the IP address associated with the TCPIP$INET_HOST logical name.
- Under certain conditions observed primarily on OpenVMS VAX systems, the master agent or subagent exits with an error from an internal `select()` socket call. In most circumstances, looping does not occur. If looping occurs, you can control the number of iterations by defining the TCPIP$SNMP_SELECT_ERROR_LIMIT logical name.
- The MIB browser provided with TCP/IP Services (TCPIP$SNMP_REQUEST.EXE) supports `getnext` processing of OIDs that include the 32-bit OpenVMS process ID as a component. However, other MIB browsers may not provide this support.

  For example, the following OIDs and values are supported on OpenVMS:

  ```
  1.3.6.1.2.1.25.4.2.1.1.1321206828 = 1321206828
  1.3.6.1.2.1.25.4.2.1.1.1321206829 = 1321206829
  1.3.6.1.2.1.25.4.2.1.1.1321206830 = 1321206830
  ```

  These examples are from `hrSWRunTable`; the `hrSWRunPerfTable` may be affected as well.
- You can ignore the following warning that appears in the log file if a null OID value (0.0) is retrieved in response to a `Get`, `GetNext`, or `GetBulk` request:

  ```
  o_oid; Null oid or oid->elements, or oid->nelem == 0
  ```

3.9.4 Upgrading SNMP

After upgrading to the current version of TCP/IP Services, you must disable and then enable SNMP using the TCPIP$CONFIG.COM command procedure. When prompted for “this node” or “all nodes,” select the option that reflects the previous configuration.

3.9.5 Communication Controller Data Not Fully Updated

When you upgrade TCP/IP Services and then modify an existing communication controller, programs that use the communication controller might not have access to the updated information.

To ensure that programs like the MIB browser (SNMP_REQUEST) have access to the new data about the communication controller, do the following:

1. Delete the communication controller using the TCP/IP management command DELETE COMMUNICATION_CONTROLLER.

2. Reset the communication controller by running the TCPIP$CONFIG.COM command procedure and exiting.
3. Restart the program (such as SNMP) by entering the following commands:

```
$ @SYS$STARTUP:SNMP_SHUTDOWN.COM
$ @SYS$STARTUP:SNMP_STARTUP.COM
```

4. Use the TCP/IP management command LIST COMMUNICATION_CONTROLLER to display the information.

3.9.6 SNMP MIB Browser Usage

If you use either the `-l` (loop mode) or `-t` (tree mode) flag, you cannot also specify the `-m` (maximum repetitions) flag or the `-n` (nonrepeaters) flag. The latter flags are incompatible with loop mode and tree mode.

Incorrect use of the `-n` and `-m` flags results in the following types of messages:

```
$ snmp_request mynode.co.com public getbulk -v2c -n 20 -m 10 -t 1.3.6.1.2.1
Warning: -n reset to 0 since -l or -t flag is specified.
Warning: -m reset to 1 since -l or -t flag is specified.
1.3.6.1.2.1.1.1.0 = mynode.company.com
```

3.9.7 Duplicate Subagent Identifiers

With this version of TCP/IP Services, two subagents can have the same identifier parameter. Be aware, however, that having two subagents with the same name makes it difficult to determine the cause of problems reported in the log file.

3.9.8 Community Name Restrictions

The following restrictions on community names are imposed by TCPIP$CONFIG.COM:

- Do not specify community names that include a space character.
- A quotation mark (") specified as part of a community name might be handled incorrectly. Check the validity of the name with the SHOW CONFIGURATION SNMP command, and if necessary, correct the name with the SET CONFIGURATION SNMP command.

3.9.9 eSNMP Programming and Subagent Development

The following notes pertain to eSNMP programming and subagent development.

- In the documentation, the terms “extension subagent”, “custom subagent”, and “user-written subagent” refer to any subagent other than the standard subagents for MIB-II and the Host Resources MIB, which are provided as part of the TCP/IP Services product.
- In the [.SNMP] subdirectory of TCPIP$EXAMPLES, files with the .C, .H, .COM, .MY, and .AWK extensions contain additional comments and documentation.
- The TCPIP$SNMP_REQUEST.EXE, TCPIP$SNMP_TRAPSND.EXE, and TCPIP$SNMP_TRAPSND.EXE programs are useful for testing during extension subagent development.
- For information about prototypes and definitions for the routines in the eSNMP API, see the TCPIP$SNMP:ESNMP.H file.

3.9.10 SNMP Installation Verification Program Restriction

The SNMP Installation Verification Program will not run correctly if debug or trace options are turned on for any TCP/IP Services for OpenVMS component.

For example, including the line:

```
options debug
```

in TCPIP$ETC:RESOLV.CONF results in unsuccessful completion status.

The problem also exists if socket tracing is turned on and directed to SYS$OUTPUT with the following command:

```
$ DEFINE TCPIP$SOCKET_TRACE SYS$OUTPUT
```

The additional output produced by these and other debug or trace options can cause problems with the SNMP IVP because it was designed to parse output from a standard configuration only.

Note

To run the SNMP IVP test either run the program directly:

```
$ RUN SYS$SYSROOT:[SYSTEST.TCPIP]TCPIP$SNMPIVP.EXE
```

or execute the TCPIP configuration menu:

```
$ @SYS$MANAGER:TCPIP$CONFIG
```

and then select option "7 - Run tests" and then option "2 - SNMP IVP".

3.10 SSH Problems and Restrictions

This section contains the following information:

- SSH-related security advisories (Section 3.10.1)
- SSH general notes and restrictions (Section 3.10.2)
- UNIX features that are not supported by SSH (Section 3.10.3)
- SSH command syntax notes and restrictions (Section 3.10.4)
- SSH authentication notes and restrictions (Section 3.10.5)
- SSH keys notes and restrictions (Section 3.10.6)
- SSH session restrictions (Section 3.10.7)
- SSH messages notes and restrictions (Section 3.10.8)
- SSH remote command notes and restrictions (Section 3.10.9)
- SSH batch mode restrictions (Section 3.10.10)
- X11 port forwarding restrictions (Section 3.10.12)
- File transfer restrictions (all file sizes) (Section 3.10.13)

- File transfer restrictions (large files) (Section 3.10.14)

Note

References to SSH, SCP, or SFTP commands also imply SSH2, SCP2, and SFTP2, respectively.

3.10.1 SSH-Related Security Advisories

Computer Emergency Readiness Team (CERT®) advisories are issued by the CERT Coordination Center (CERT/CC), a center of Internet security expertise located at the Software Engineering Institute, a federally-funded research and development center operated by Carnegie Mellon University. CERT advisories are a core component of the Technical Cyber Security Alerts document featured by the United States Computer Emergency Readiness Team (US-CERT), which provides timely information about current security issues, vulnerabilities, and exploits.

CERT and HP Software Security Response Team (SSRT) security advisories might be prompted by SSH activity. CERT advisories are documented at the following CERT/CC web site:

`http://www.cert.org/advisories`.

Table 3–1 provides brief interpretations of several SSH-related advisories:

Table 3–1 CERT/SSRT Network Security Advisories

Advisory	Impact on OpenVMS
CERT CA-2003-24	OpenSSH only; OpenVMS is not vulnerable.
CERT CA-2002-36	A worst case consequence of this vulnerability is a denial of service (DoS) for a single connection of one of the following types: • Server process handling a connection from a malicious client • Client process connecting to a malicious server In either case, a malicious remote host cannot gain access to the OpenVMS host (for example, to execute arbitrary code), and the OpenVMS server is still able to receive a new connection.
CERT-2001-35	OpenVMS is not vulnerable. Affects SSH Version 1 only, which is not supported.
CERT CA-1999-15	RSAREF2 library is not used; OpenVMS is not vulnerable.
SSRT3629A/B	OpenVMS is not vulnerable.

3.10.2 SSH General Notes and Restrictions

This section includes general notes and restrictions that are not specific to a particular SSH application.

- The UNIX path `/etc` is interpreted by the OpenVMS SSH server as TCPIP$SSH_DEVICE:[TCPIP$SSH].

- The following images are not included in this release:
  - TCPIP$SSH_SSH-CERTENROLL2.EXE

    This image provides certificate enrollment.
  - TCPIP$SSH_SSH-DUMMY-SHELL.EXE

    This image provides access to systems where only file transfer functionality is permitted.
  - TCPIP$SSH_SSH-PROBE2.EXE

    This image provides the `ssh-probe2` command, which sends a query packet as a UDP datagram to servers and then displays the address and the SSH version number of the servers that respond to the query.

3.10.3 UNIX Features That are Not Supported by SSH

This section describes features that are expected in a UNIX environment but are not supported by SSH for OpenVMS.

- The server configuration parameter `PermitRootLogin` is not supported.
- The client configuration parameter `EnforceSecureRutils` is not supported.
- There is no automatic mapping from the UNIX ROOT account to the OpenVMS SYSTEM account.
- The SSH1 protocol suite is not supported for terminal sessions, remote command execution, and file transfer operations. Parameters unique to SSH1 in the server and client configuration files are ignored.

3.10.4 SSH Command Syntax

This section includes notes and restrictions pertaining to command syntax.

- From a non-OpenVMS client, if you use OpenVMS syntax for names (such as device names), enclose the names in single quotation marks to prevent certain characters from being interpreted as they would be on a UNIX system.

  For example, in the following command, UNIX interprets the dollar sign ($) as a terminator in the device name SYS$SYSDEVICE:[*user*], resulting in SYS:[*user*].

  ```
  # ssh user@vmssystem directory SYS$SYSDEVICE:[user]
  ```

  To avoid this problem, enter the command using the following format:

  ```
  # ssh user@vmssystem directory 'SYS$SYSDEVICE:[user]'
  ```

3.10.5 SSH Authentication

This section includes notes and restrictions pertaining to SSH authentication.

- The location of the SHOSTS.EQUIV file has been moved from TCPIP$SSH_DEVICE:[TCPIP$SSH] to TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2].
- If hostbased authentication does not work, the SSH server may have failed to match the host name sent by the client with the one it finds in DNS/BIND. You can check whether this problem exists by comparing the output of the following commands (ignoring differences in case of the output text):
  - On the server host:

```
$ TCPIP
TCPIP> SHOW HOST client-ip-address
```

– On the client host:

```
$ write sys$output -
$_ "''f$trnlnm("TCPIP$INET_HOST")'.''f$trnlnm("TCPIP$INET_DOMAIN")'"
```

If the two strings do not match, you should check the host name and domain configuration on the client host. It may be necessary to reconfigure and restart TCP/IP Services on the client host.

- If the user default directory in the SYSUAF user record is specified with angle brackets (for example, <*user-name*>) instead of square brackets ([*user-name*]), hostkey authentication fails. To solve this problem, change the user record to use square brackets.
- The pairing of user name and UIC in the OpenVMS rights database, as displayed by the AUTHORIZE utility's SHOW /IDENTIFIER command, must match the pairing in the SYSUAF record for that user name. If the pairings do not match, the following message error is displayed when the user attempts to establish an SSH session:

```
$ ssh hosta
%SYSTEM-F-ACCVIO, access violation, reason mask=00, virtual address=000000000000 0000, PC

  Improperly handled condition, image exit forced.
    Signal arguments:   Number = 0000000000000005
                        Name   = 000000000000000C
                                 0000000000000000
                                 0000000000000000
                                 FFFFFFFF811A88E8
                                 000000000000001B

    Register dump:
    R0  = FFFFFFFFFFFFFFFE  R1  = 0000000000495D08  R2  = 000000000001DEE0
    R3  = 00000000004ABE18  R4  = 0000000000000000  R5  = 0000000000000000
    R6  = 0000000000000000  R7  = 0000000000000000  R8  = 0000000000000000
    R9  = 0000000000000000  R10 = 0000000000000000  R11 = 00000000002F7C20
    R12 = 0000000000000000  R13 = 0000000000498708  R14 = 00000000004EDF48
    R15 = 000000007AECFE10  R16 = 0000000000000000  R17 = 0000000000000000
    R18 = 0000000000000000  R19 = 000000007B624258  R20 = 0000000077770000
    R21 = 0000000000000008  R22 = FFFFFFFF77774A00  R23 = 0000000300000000
    R24 = 0000000000000001  R25 = 0000000000000001  R26 = 0000000000118A6C
    R27 = 000000007C062700  R28 = 0000000000000000  R29 = 000000007ADEF290
    SP  = 000000007ADEF290  PC  = FFFFFFFF811A88E8  PS  = 100000000000001B
```

To solve this, use the AUTHORIZE utility to correct the pairing of user name and UIC value in the OpenVMS rights database.

3.10.6 SSH Keys

This section includes notes and restrictions pertaining to SSH keys.

- SSH client users can copy their own customized version of the SSH2_CONFIG. file and modify the value of the variable `StrictHostKeyChecking`. By setting the value of this variable to "no," the user can enable the client to automatically copy the public key (without being prompted for confirmation) from an SSH server when contacting that server for the first time.

  A system manager can tighten security by setting the `StrictHostKeyChecking` variable to "yes" in the systemwide SSH2_CONFIG. file, and forcing users to use only the systemwide version of the file. In this case, to copy the public key from the server, users (and the system manager) must use another mechanism (for example, a privileged user can manually copy the public key).

To enforce this tighter security response, the system manager can perform the following steps:

1. Edit TCPIP$SSH_DEVICE:[TCPIP$SSH]SSH2_CONFIG. to include the following line:

   ```
   StrictHostKeyChecking  yes
   ```

2. Restrict
   user access to TCPIP$SSH_DEVICE:[TCPIP$SSH]SSH2_CONFIG.
   For example:

   ```
   $ SET SECURITY/PROTECTION=(G,W) TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]SSH2_CONFIG.;
   ```

3. Edit the SYS$STARTUP:TCPIP$SSH_CLIENT_STARTUP.COM command procedure to install the SSH server image with the READALL privilege. In the following example, change the existing line to the replacement line, as indicated:

   ```
      .
      .
      .
   $     image = f$edit("sys$system:tcpip$ssh_ssh2.exe","upcase")
   $!    call install_image 'image' ""          <== existing line
   $     call install_image 'image' "readall"   <== replacement
      .
      .
      .
   ```

4. Enable the SSH client, as described in the *HP TCP/IP Services for OpenVMS Guide to SSH*.

Note

Steps 2 and 3 involve modification of system files. Therefore, it may be necessary to repeat the modifications after a future update of TCP/IP Services.

- If you do not specify the key file in the SSH_ADD command, and SSH_ADD finds no INDENTIFICATION. file, it adds only the first private key it finds in the [*username*.SSH2] directory.
- Do not use the SSH_KEYGEN `-e` option (used to edit the comment or passphrase of the key). This option does not work.
- With this release, the default size of keys generated by the SSH_KEYGEN utility is 2048 bits (for earlier releases, the default size was 1024 bits). Consequently, generation of keys takes longer — sometimes five to ten times longer. On slow systems, or during SSH configuration, key generation may seem to be hanging when it is not. No progress indicator is displayed. During SSH configuration, the following messages indicate the keys are being generated:

  ```
  Creating private key file: TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]HOSTKEY
  Creating public key file: TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]HOSTKEY.PUB
  ```

Note

While the keys are being generated, you might notice a delay. This does not indicate a hang.

3.10.7 SSH Sessions

This section includes restrictions pertaining to SSH sessions.

- In an SSH session on the OpenVMS server, the originating client host name and the user name or port identification are not available. For example, in a TELNET session, the OpenVMS DCL command SHOW TERMINAL displays the following information about a UNIX client:

  ```
  Remote Port Info: Host: unixsys.myco.com Port:2728
  ```

  Likewise, information about an OpenVMS client appears as:

  ```
  Remote Port Info: Host: mysys.com Locn:_RTA4:/USER
  ```

  Neither of these lines is displayed in a similar SSH session; however, information for SSH sessions is available in the logical names SYS$REM_ID (username) and SYS$REM_NODE and SYS$REM_NODE_FULLNAME (hostname)

- Starting SSH sessions recursively (for example, starting one SSH session from within an existing SSH session) creates a layer of sessions. Logging out of the innermost session may return to a layer other than the one from which the session was started.

- SSH escape sequences are not fully supported. For example, you may have to enter the `Escape .` (escape character followed by a space and a period) exit sequence twice for it to take effect. On exit, the terminal is left in NOECHO and PASTHRU mode.

- On certain non-OpenVMS clients, after attempting to exit from an SFTP session, you must press Enter an extra time to return to the operating system prompt.

3.10.8 SSH Messages

This section includes notes and restrictions pertaining to SSH session messages.

- Normally, the translation of the system logical name SYS$ANNOUNCE is displayed after authentication is complete. In this version of SSH, no automated mechanism exists for displaying this text as a prelogin banner.

  To provide a prelogin banner from a text file, create the file SSH_BANNER_MESSAGE. containing the text to be displayed before login.

  To enter multiple lines in the banner text, make sure each line ends with an explicit carriage-return character except the last line.

  Save the banner message file in the TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2] directory, with privileges that allow it to be read by the user account [TCPIP$SSH].

  If you do not use the default file name and location for the message banner file, define them using the `BannerMessageFile` option in the TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]SSHD2_CONFIG. file. Specify the location and file name of your banner message file as the argument to the option using one of the following formats:

  ```
  BannerMessageFile   TCPIP$SSH_DEVICE:[TCPIP$SSH]BANNER1.TXT
  BannerMessageFile   /TCPIP$SSH_DEVICE/TCPIP$SSH/BANNER2.TXT
  BannerMessageFile   /etc/banner3.txt
  ```

Note that the argument may be in either OpenVMS or UNIX format and is not case sensitive. (If multiple definitions for the same option are included in the configuration file, the last one listed will take effect.)

- Some SSH informational, warning, and error message codes are truncated in the display. For example:

  ```
  %TCPIP-E-SSH_FC_ERR_NO_S, file doesn't exist
  ```

- Some SSH log and trace output messages, and informational, warning, and error messages display file specifications as UNIX path names.

3.10.9 SSH Remote Commands

This section includes notes and restrictions pertaining to SSH remote commands.

- Command lines for remote command execution through SSH are limited to 153 characters.
- After you execute an SSH remote command, you may need to press the Enter key to get back to the DCL prompt.
- When you execute remote commands on the OpenVMS SSH server, the log file TCPIP$SSH_RCMD.LOG is created in the directory defined by the logical name SYS$LOGIN for your user account. This log file is not purged automatically.
- When you execute remote commands on an OpenVMS SSH client connected to a non-OpenVMS SSH server, output may not be displayed correctly. For example, sequential lines might be offset as if missing a linefeed, as in the following example:

  ```
  $ ssh user@unixhost ls -a
    user's password:
    Authentication successful.
    .
     ..
       .TTauthority
                   .Xauthority
                               .cshrc
                                     .dt
                                        .dtprofile
  ```

  To display the output correctly, use the -t option with the command, as in the following command example:

  ```
  $ ssh -t user@unixhost ls -a
  ```

- Any OpenVMS command that refreshes the display can have unexpected results when executed as a remote SSH command. For example, the following command exhibits this behavior:

  ```
  $ MONITOR PROCESS /TOPCPU
  ```

  Executed locally, this command displays a bar chart that is continuously updated. When executed as a remote command, it displays each update sequentially. In addition, you cannot terminate the command using Ctrl/C.

3.10.10 SSH Batch Mode

This section includes batch mode restrictions.

- Because the SSH, SFTP, and SCP commands are implemented by code ported from UNIX sources, they do not support all of the standard OpenVMS behaviors for SYS$INPUT, SYS$OUTPUT, and SYS$ERROR in command procedures. For example:
  - SYS$INPUT is not the default batch command procedure.
  - Output written to a batch log file or other SYS$OUTPUT file may have an extra <CR> (ASCII decimal 13) or other explicit formatting characters.
  - You can direct SYS$OUTPUT to a file, as in the following example:

    ```
    $ ASSIGN OUT.DAT SYS$OUTPUT
    ```

- When you run these commands from an interactive command procedure, you should use the explicit UNIX batch mode flags, as listed in the following table:

For...	Use...
SSH (remote command execution or port forwarding),	`-o batchmode yes`
SCP,	`"-B"`
SFTP,	`"-B"` {*batchfile*}

- If you use the SSH command in batch mode with an interactive session (that is, not for remote command execution or setting up port forwarding), the batch job hangs.

  If the `"-S"` option is used in an interactive SSH session, or with an SSH command executed interactively in a DCL command procedure, the terminal session hangs. Ctrl/Y and Ctrl/C will not restore the DCL prompt. To release the hung terminal session, you must restart the SSH client and server.

- For the SFTP command, note the following:
  - If the command is used without the `-B` {*batchfile*} option, SFTP uses the following file by default: SYS$LOGIN:TCPIP$SFTP_BATCHFILE.TXT.

- When running in batch mode:
  - The SFTP command displays the final state-of-progress indicator; the SCP command does not.
  - The SSH command will not prompt for a password, password update, or passphrase. If one is required, the batch job fails.
  - The SSH command will not cause a new host key to be saved if the value of `StrictHostkeyChecking` is “no;” SSH will not prompt for one if the value is “ask.”

    For other notes and restrictions pertaining to keys, see Section 3.10.6.
  - If an `ls` command is contained in the SFTP batch input, and the interactive output requires input from the keyboard to continue, then some of the output lines might be omitted from the batch log file.

3.10.11 ls Fails After cd to a Logical Name from a Tru64 UNIX Client

`ls` can fail when using `sftp cd` to a logical name from a Tru64 UNIX client.

For a workaround, try the following:

1. `cd` to the path for the directory in UNIX format, e.g., instead of: `cd tcpip$ssh_home`, use `cd /sys$sysdence/tcpip$ssh`.
2. Perform the `ls` specifiying the logical name in the path, e.g., `ls /tcpip$ssh_home`.

3.10.12 SSH X11 Port Forwarding

This section includes X11 port forwarding restrictions and problems.

- To use X11 forwarding in native mode, the system must be running DECwindows MOTIF Version 1.3 or higher. In addition, the X Authority utility (xauth) is required on the system. The X11 server uses this utility for authenticating host/user connections. For more information on how to use this utility, see the HP DECwindows Motif for OpenVMS documentation.
- To display a remote X11 client application on your X11 server, you must set the display variable on the X11 client to the address of the X11 server the client is connecting to. You can verify that the variable is set correctly on an OpenVMS system by using the following DCL command:

  ```
  $ SHOW LOGICAL DECW$DISPLAY
  ```

  For WSA display devices, use the SHOW DISPLAY command to see the display variable value.

  To set the display variable on an OpenVMS client to point to your server, use the SET DISPLAY command as in the following example, where 127.127.1.1 is the server node address:

  ```
  $ SET DISPLAY/CREATE/NODE=127.127.1.1/TRANSPORT=TCPIP
  ```

  SSH on OpenVMS supports only local and TCP/IP transports. If you are using a local transport, you have to be at the system where the display is to appear, and that system must be running the X11 server. For local transport, use the following command to set the display:

  ```
  $ SET DISPLAY/CREATE/TRANSPORT=LOCAL
  ```

  On UNIX systems, use the following command to set the display variable to point to a server node with address 16.20.176.33 and using the TCP/IP transports:

  ```
  >setenv display 16.20.176.33:0.0
  ```

  To use local transport, use the following UNIX command:

  ```
  >setenv display :0.0
  ```

- To set up a standard port forwarding session for X11 on a remote OpenVMS system, HP recommends that you use remote port forwarding; local port forwarding will not work.

3.10.13 SSH File Transfer (All File Sizes)

This section includes SSH restrictions pertaining to file transfer operations.

- On OpenVMS, setting the ForcePTTYAllocation keyword to "yes" in the SSH2_CONFIG. file can result in failures when performing file copy operations. (In other implementations of SSH, setting the keyword ForcePTTYAllocation to "yes" in the SSH2_CONFIG. file has the same effect as using the -t option to the SSH command.)
- When connected to some servers, the client can detect packet benign file transfer protocol packet-length errors. By default, no message is displayed.

  To display warning messages, type the following:

  ```
  $ DEFINE/SYS NO TCPIP$SSH_TOLERANT_PROTOCOL STATUS
  ```

  using either the "NO" or any string starting with an upper- or lowercase N.

  Following is an example of a warning message:

  ```
  Warning: packet length mismatch: expected 27, got 8; connection to non-standard server?
  ```

  To retain the logical name assignment through each reboot, add the DEFINE command to the appropriate startup command procedure.
- VMS Plus Mode:

  When the client and the server are OpenVMS systems running v5.6, they recognize each other as such and implement TCP/IP Services specific SFTP protocol extensions that allow transfer of files in either direction while preserving the key OpenVMS file attributes: record format and record attributes.

  The TCP/IP Services SCP client uses SFTP as the underlying protocol so VMS Plus mode works with SCP as well.

  VMS Plus mode supports only sequential ogranization files.

  Remember that if a v5.6 system is connected with an older TCP/IP Services system that does not support VMS Plus mode, file attributes will not be preserved. VMS Plus mode can only be used if both sides support it.
- Talking to a system without VMS Plus:

  If one side of the file transfer, client or server, does not support VMS Plus mode for SFTP, file attributes will not be preserved.

  In this mode TCP/IP Servics supports reading of any of the following types of sequential organization files:

  - Stream_LF
  - Variable Length
  - VFC
  - Fortran Carriage Control
  - Fixed Length
  - Undefined

  Note that which side is the server and which is the client is irrelevant. OpenVMS is simply running on the side that is reading the file. You can, for exmaple, use SFTP client from OpenVMS to put a VFC file to UNIX, or you could use the SFTP client on the UNIX system to get the same file from the

OpenVMS system. In either case, the OpenVMS system is reading the file and the Unix file is writing it.

Copying some VFC files from OpenVMS to systems not running OpenVMS and then back to OpenVMS may result in a file that the OpenVMS `DIFFERENCES` command shows as different from the original file. This is unpreventable and the file as transferred out and back in is correct in that the `TYPE` and `PRINT` commands display it as expected and the output here is ths same as that for the original file.

Copying Fortran CC files from OpenVMS to systems other than OpenVMS will always result in a file that shows differences from the original. This is because on its transfer from OpenVMS to UNIX the Fortran CC attributes were converted to inline ASCII control character sequences that print the lines as the Fortran CC control bytes require. For example, the Fortran character for overstrike results in a pair of carriage returns for the line thus implementing an overstrike.

- TCP/IP Services supports only sequential file organization, not relative or indexed files

  To transfer these unsupported files you can package the file(s) into an OpenVMS saveset and transfer that or, depending on how many hops over which SFTP/SCP implementations and operating systems, you may need to use more extreme measures. One way that works consistently (provided that you have FTSV installed) is packaging files into a save set, then using SPOOL COMPRESS to make them into an self-extracting VMS image, then using UUENCODE to transform the image into an ASCII text file.

- Not all variants of UNIX path names are supported when referring to files on OpenVMS clients and servers.

- The SCP and SFTP commands from the following Windows clients have been tested and interoperate correctly with the OpenVMS SSH server:

  – PuTTY

  – SSH Communications

  Other versions and other clients may work, depending on protocol implementation and factors such as whether the client can handle OpenVMS-format file specifications.

- When using the SFTP command, pressing Ctrl/C does not display "Cancel" as expected. Also, Ctrl/T does not work as in DCL to display a status line; instead, it switches two adjacent characters, as on UNIX systems. Other problems with character handling have been fixed with this release, as reported in Section 4.19.

- The SFTP `ls` command pauses for an extended time after displaying a page of data and then continues with the next page. This occurs because the ssh server is sending back a complete directory listing, which the client filters; therefore, for directories with many files, the delay is due to the client waiting for listing results from the server. This is typical sftp behavior, and not specific to OpenVMS.

- Using SCP or SFTP command to copy a file back to itself (either in local mode, or by connecting back to the client host) fails with the following error:

  ```
  %TCPIP-E-SSH_FC_ERR_INVA, file record format invalid for copy
  ```

- The SCP command issued from a client using SSH Version 1 will not work with the OpenVMS SSH server. The OpenVMS server does not support SSH Version 1.

3.10.14 SSH Transferring Large Files

This section includes restrictions pertaining to transferring large files:

- The minimum version of DECC$SHR running on your system must be that which was released with OpenVMS Version 8.2.
- You may need to adjust memory parameters (WSDEF, WSQUO, WSEXTENT, and PGFLQUO) to accommodate the memory requirements of the file copy client and server. The exact value depends on system resources and virtual memory configuration. For more information, see Section 2.3. For ssh filecopy, testing has shown that the main parameter to adjust is PGFLQUO.

3.10.15 SSH Server Signals Internal Credentials Cache Error

If an SSH client attempts to use gssapi-with-mic authentication to the TCP/IP Services for OpenVMS SSH server on a server host that is running Kerberos V2.1 and the SSH client user's TGT is forwardable (a kinit -f has been done) and the GssapiDelegateCredentials flag is set then the ssh server will signal the following error in the server log:

```
Internal credentials cache error
```

This error text may appear on the SSH client user's screen, depending on configuration.

This can be worked around in either of the following ways:

1. Upgrade to Kerberos V3.0 on the SSH server host.
2. Use the `kinit` without the `-f` flag on the SSH client.
3. Turn the GssapiDelegateCredentials configuration switch off on the SSH client.

Because forwarding of client credentials with `gssapi-with-mic authentication` to the OpenVMS SSH server is not supported setting GssapiDelegateCredentials is not necessary.

3.10.16 SFTP Generates Audit Warnings with Class Device

This restriction applies only to those using AUDIT with class device as in the following command:

```
$ SET AUDIT/ALARM/ENABLE=ACCESS=ALL/CLASS=DEVICE
```

If the SFTP server generates audit warnings for a logical IO to a mailbox when the SFTP user exits SFTP, perform the following step to prevent this from occurring:

```
$ DEFINE/SYSTEM TCPIP$SSH_SERVER_WAIT_FOR_CHILD 1
```

3.10.17 BIND Resolver Diagnostics Creates an SSH Packet Corruption

When you turn on BIND Resolver Diagnostics using either of the following methods, you can create an SSH packet corruption:

- Define the logical name TCPIP$BIND_RES_OPTIONS to "debug".
- Add the following line to TCPIP$ETC:RESOLV.CONF:

  ```
  options debug
  ```

3.11 TCPDUMP Restrictions

TCPDUMP works the same way on OpenVMS as it does on UNIX systems, with the following restrictions:

- On UNIX systems, tcpdump sets the NIC (Network Interface Controller) into promiscuous mode and everything in the transmission is sent to tcpdump.

  On OpenVMS systems, TCPDUMP only sees the packets destined for and sent from the local host. Therefore, TCPDUMP works in copy-all mode. Because it only sees a copy of the packets that are processed by the TCP/IP kernel, TCPDUMP can only trace natively IP, IPv6, and ARP protocols on Ethernet.

  TCPDUMP can format or filter packets that have been traced from another platform running TCPDUMP in promiscuous mode. In this case it will process other protocols, like DECnet.
- Ethernet is the only supported type of NIC. Other types of NICS (such as ATM, FDDI, Token Ring, SLIP, and PPP) are not supported.
- The -i option is not supported. On UNIX systems, this option specifies the interface that tcpdump is attached to.

  On OpenVMS systems, TCPDUMP obtains packets from the TCP/IP kernel.
- The -p option is not supported. On UNIX systems, this option specifies that tcpdump stops working in promiscuous mode.

  On OpenVMS, TCPDUMP does not work in promiscuous mode. Therefore, this option is set by default.
- If you are using the Ethereal software to dump IPv6 network traffic, use the following command format to write the data in the correct format:

  ```
  $ TCPDUMP -s 1500 -w filename
  ```
- Only one process at a time can issue traces. This restriction applies to both TCPTRACE and TCPDUMP.

3.12 TCP/IP Management Command Restrictions

The following restrictions apply to the TCP/IP management commands:

- TCP/IP Services Version 5.4 introduced failSAFE IP, which obsoletes the IP cluster alias address. Consequently, the following TCP/IP management commands are no longer supported:
  - SET INTERFACE /NOCLUSTER
  - SHOW INTERFACE /CLUSTER

To display interface addresses, including IP cluster alias addresses, use the following TCP/IP management command:

```
TCPIP> ifconfig -a
```

To delete a cluster alias address from the active system, use a command similar to the following:

```
TCPIP> ifconfig ie0 -alias 10.10.10.1
```

The following TCP/IP management commands continue to be supported:

- SET INTERFACE/CLUSTER
- SET CONFIGURATION INTERFACE /CLUSTER
- SET CONFIGURATION INTERFACE /NOCLUSTER
- SHOW CONFIGURATION INTERFACE /CLUSTER

- SET NAME_SERVICE /PATH

  This command requires the SYSNAM privilege. If you enter the command without the appropriate privilege at the process level, the command does not work and you are not notified. If you enter the command at the SYSTEM level, the command does not work and you receive an error message.

- SET SERVICE command

  When you modify parameters to a service, disable and reenable the service for the modifications to take effect.

For more information on TCP/IP Services management commands, refer to the *HP TCP/IP Services for OpenVMS Management Command Reference* guide.

4
Corrections

This chapter describes the problems corrected in this version of TCP/IP Services.

4.1 Advanced Programming Environment Problems Fixed in This Release

The following sections describe programming-related problems fixed in this release.

4.1.1 Socket Routines Limited to 64k Bytes

In previous versions, the socket routines send( ), recv( ), read( ), write( ), sendto( ), and recvrom( ), along with routines (sendmsg( ), recvmsg( ), readv( ), writev( ), etc., were limited to 64k bytes (65535, or FFFF hex). That restriction has been lifted.

The QIO operations IO$_READVBLK and IO$_WRITEVBLK also now accept buffer lengths greater than 64k, with a corresponding change in the format of the IOSB. The size of the IOSB remains unchanged at 8 bytes. However, the second half of the IOSB is now a copy of the returned byte count. The count is still also returned in the second half of the first longword, for compatibility with older applications. If the count equals or exceeds 65535 bytes, that 16-bit count will be returned as 65535, the maximum possible value. Applications designed for TCPIP V5.5 and later are encouraged to reference the second longword of the IOSB in order to determine how many bytes were successfully transferred. In the event of an error return, the UNIX-style errno is still returned in the second half of the first IOSB longword.

4.1.2 Symbol Vector Inappropriately Inserted in the IPC Options File

Problem:

In V5.5, a symbol vector for the routine socketpair was inappropriately inserted in the IPC options file. This caused applications that were linking directly against TCPIP$IPC_SHR to ACCVIO when run on an OpenVMS V8.2 system.

Solution:

This problem has been corrected and allows those previously linked programs to run on recent versions of OpenVMS.

Note

TCP/IP Services does not recommend or support linking directly against the TCPIP$IPC_SHR shareable image.

4.1.3 AF_AAL Defined Twice

Problem:

In previous releases, the file SOCKET.H in TCPIP$EXAMPLES had AF_AAL defined twice, to two different values.

Solution:

This problem has been corrected.

4.2 BIND Server Problems Fixed in This Release

The following sections describe BIND server problems fixed in this release.

4.2.1 BIND Server Not Properly Using the TCPIP$BIND_COMMON Logical Name

Problem:

In previous versions of TCP/IP Services, the BIND Server was not properly using the TCPIP$BIND_COMMON logical name. This logical name is a search list used in the multiple masters BIND server environment. It is designed to detect files first in the sys$specific:[tcpip$bind] directory, then in the BIND common directory. The problem with the logical name caused the files in the sys$specific:[tcpip$bind] directory to be ignored.

Solution:

This problem is corrected in this release; however, the solution requires changes to your configuration.

To modify your configuration, perform the following steps:

1. Shut down the BIND server using the following command:

   ```
   $ @sys$manager:tcpip$bind_shutdown.com
   ```

2. Run the sys$manager:tcpip$bind_cluster_setup.com command procedure. This procedure creates a new common directory that replaces your previous common directory.

3. Copy all of the files in your previous BIND common directory to the new directory: common_device:[tcpip$bind].

4. Edit the directory substatement in the options statement in the BIND configuration file sys$specific:[tcpip$bind]tcpip$bind.conf:

   ```
   options {
    directory "TCPIP$BIND_COMMON:[TCPIP$BIND]";
   };
   ```

5. Start the BIND server using the following command:

   ```
   $ @sys$manager:tcpip$bind_startup.com
   ```

4.2.2 Change to List of BIND Servers in Resolver Configuration Recognized

Problem:

In previous releases, a change to the list of BIND servers in the resolver configuration was not recognized when attempting to set host via DECnet over IP. The customer would have to reboot for changes to take effect.

Solution:

This problem is corrected in this release.

4.2.3 Resolver Clients Not Receiving Responses from the BIND Server

Problem:

In previous releases, some resolver clients did not get responses from the BIND server after a failover event when using the cluster alias.

Solution:

This problem is corrected in this release.

4.2.4 ACCVIO When Using TSIG

Problem:

In previous releases, the NSUPDATE utility could ACCVIO when using TSIG and attempting to delete a CNAME record. The ACCVIO would only occur if some other NSUPDATE command was issued first with a send in between the command.

Solution:

This problem is corrected.

4.3 FTP Server Problems Fixed in This Release

The following sections describe FTP server problems fixed in this release.

4.3.1 FTP Does Not Allow IP Address Specification

Problem:

The FTP server does not allow you to specify an IP address other than that of the connected client, or the specification of a privileged port, in the PORT, LPRT, or EPRT commands. Any such commands are rejected with the following error:

```
500 Illegal {PORT|LPRT|EPRT} command.
```

The FTP server and client prevent data connection "theft" by a third party. For the FTP server, this applies to passive-mode connections from an IP address other than the client's, or from a privileged port. For the FTP client, this applies to active-mode connections from an IP address other than the server's, or from a port other than port 20.

Solution:

If this software change is not acceptable, you can restore the original behavior by defining the following logical names:

Server	Client
TCPIP$FTPD_ALLOW_ADDR_REDIRECT	TCPIP$FTP_ALLOW_ADDR_REDIRECT
TCPIP$FTPD_ALLOW_PORT_REDIRECT	TCPIP$FTP_ALLOW_PORT_REDIRECT

These logical names allow you to relax the IP address and port checks in the FTP server and the FTP client.

4.3.2 DCL DIRECTORY or UNIX ls Command Returns "Illegal Port Command" Error

Problem:

On an FTP client, if you use a password with an embedded space to log into an OpenVMS FTP server, the following error message is returned in response to the DCL command DIRECTORY or the UNIX command `ls`:

```
500 Illegal PORT command.
```

Solution:

This problem is corrected in this release.

4.4 FTP Client Problems Fixed in This Release

The following sections describe FTP client problems fixed in this release.

4.4.1 FTP Client Fails to Delete Interim Files after GET/MGET Commands

Problem:

After an FTP GET or MGET command entered with wildcard characters completes, the temporary TCPIP$FTP_TEMP*nnnnnnnn*.TMD files created by FTP are supposed to be deleted from the SYS$SCRATCH area. However, if no files match the wildcard criteria, FTP fails to delete any of the temporary files. (If at least one file matches the wildcard criteria, FTP successfully deletes any TCPIP$FTP_TEMP*nnnnnnnn*.TMD files created in SYS$SCRATCH.)

Solution:

This problem is corrected in this release.

4.5 IMAP Problems Fixed in This Release

The following sections describe IMAP problems fixed in this release.

4.5.1 TELNET to IMAP SSL Port 993 Hangs and Aborts The Same Results in Server Crash

Problem:

When using IMAP with SSL support, the IMAP client sometimes cannot connect to the server. Events such as the following are signaled in the IMAP server event log:

```
12:41:50 3020041B Session 10: Session::DoRun, one of our exceptions was unprocessed.
12:41:50 3020041B Session 10: Socket::Write, Network Error:0
```

Connection requests to IMAP SSL port 993 should satisfy SSL handshake to complete successfully. Raw telnet cannot perform SSL handshakes and hence hangs. However, exceptions on SSL handshake was server-wide and hence any unhandled exception was fatal to server. **Solution:**

The problem has been rectified by making SSL handshake session specific.

4.5.2 A Message Line Containing More Than 255 Characters Gets Truncated to 255 When Fetched via IMAP

Problem:

In a message, any line containing more than 255 characters (i.e., without intermediate CR/LF) was truncated to 255. This was too short in many cases.
Solution:

With this fix, IMAP now reads message lines up to 2048 characters including CR/LF.

4.5.3 IMAP server crashes intermittently

Problem:

IMAP server crashes intermittently while fetching messages with more than 256 characters per line. IMAP server crashes intermittently while listing empty folders. **Solution:**

This has been rectified in this release. The memory corruptions in various functions, which had caused IMAP to crash, have been fixed.

4.6 IPv6 Problems Fixed in This Release

The following sections describe IPv6 problems fixed in this release.

4.6.1 iptunnel create Command Causes BIND Lookups for IPv4 Addresses

Problem:

When invoking an `iptunnel create` command that specifies IPv4 addresses for the tunnel source or end points, numerous DNS name resolution queries are sent to the name server even though resolution is not needed. These queries could result in a delay.

Solution:

This problem is corrected in this release.

4.7 LPD/LPR and TELNETSYM Problems Fixed in This Release

The following sections describe LPD/LPR and TELNETSYM server problems fixed in this release.

4.7.1 Print Jobs Using Wildcard Proxy from Hosts with No Name to Address Translation Available Are Rejected

Problem:

Print jobs using wildcard proxy from hosts with no name to address translation available should succeed but are rejected.

Solution:

This release resolves this problem. Print jobs using wildcard proxy from hosts with no name to address translation available will now succeed.

4.7.2 $PRINT/PARAM=(host=x) would report an access violation (ACCVIO)

Problem:

$PRINT/PARAM=(host=x) would report an access violation (ACCVIO).

Solution:

This problem is corrected in this release.

4.8 NFS Server Problems Fixed in This Release

The following sections describe NFS server problems fixed in this release.

4.8.1 NFS Server Overwrites Files with Case-Sensitive Lookup

With OpenVMS Version 7.3-1 and higher the /CASE_LOOKUP=BLIND qualifier with the SET PROCESS command causes the case of file names to be ignored during lookups, while /CASE_LOOKUP=SENSITIVE causes the case of file names to be considered. However, if case sensitivity is not enabled on the NFS server, and the NFS client attempts to create both of those files, unexpected results can happen. For example the second file might overwrite the first.

With this release of TCP/IP Services, the TCP/IP management command ADD EXPORT has two new options: CASE_BLIND and CASE_SENSITIVE, which control UNIX-like case sensitivity for NFS server file lookups. For example, when case sensitivity is enabled, NFS preserves the case in the file names AaBBc.TXT and AABBC.TXT, regarding them as two different files.

In general, TCP/IP Services clients (not servers) determine whether lookups are case sensitive because they perform lookups in their local directory cache rather than on the server. However, when a file is being created, the server controls whether case sensitivity is in effect. Make sure that the case-sensitivity options for the server and client match; otherwise, unexpected results can occur.

For more information on the CASE_BLIND and CASE_SENSITIVE options, enter the following command:

```
$ TCPIP HELP ADD EXPORT
```

4.8.2 Directories Created by non-VMS Clients Do Not Inherit Version Limit

Problem:

Newly created directories should inherit the version limit attribute from their parent directory. When a directory is created at the request of an OpenVMS NFS client, the attribute is inherited as expected; however, directories created at the request of non-OpenVMS NFS clients do not inherit this attribute. This is a problem particularly for UNIX clients, because UNIX files only have one version, but the version limit of a new directory is set to zero (no limit).

Solution:

This problem is corrected in this release. Directories created for non-OpenVMS clients now inherit the parent directory's version limit attribute.

4.8.3 NFS Server and netstat Do Not Run Properly on Alpha Systems Not Running EV56 or Later Technologies

Problem:

On Alpha systems predating the EV56 processor, the NFS server and the `netstat` utility either experience excessive instruction time or do not run at all.

Solution:

This problem is corrected in this release.

4.8.4 MOUNT Server Problems Fixed in This Release

The following sections describe MOUNT server problems fixed in this release.

4.8.5 Client Unable to Mount Devices

Problem:

In previous releases, if at least two exports were added to the export database, with options specified, a client was unable to mount both of the devices. It would only be able to mount the last export entered.

Solution:

This problem is corrected in this release.

4.9 NTP Problems Fixed in This Release

The following sections describe NTP problems fixed in this release.

4.9.1 NTPDATE Issue If the NTP Service Is Not Defined

Problem:

In previous releases of TCP/IP Services, if the NTP service was not defined in the TCP/IP Configuration database, the `ntpdate` utility would produce an error or ACCVIO.

Solution:

This problem is corrected in this release.

4.9.2 NTP Server Automatically Purges Log Files

Problem:

Previously, the NTP server would automatically purge log files when NTP was started (/keep=5). As long as NTP remained running, another purge was not performed.

Solution:

With this release the NTP server will still purge log files at NTP startup time. In addition it will also automatically purge log files once per day before creating the new daily log file.

4.9.3 NTP Broacast Feature Does Note Work on an IPv6-enabled System

Problem:

In V5.5 the NTP broadcast feature was not working on an IPv6-enabled system.

Solution:

This problem has been corrected.

4.10 LBROKER Problems Fixed in This Release

The following section describes LBROKER problems fixed in this release.

4.10.1 Load Broker Polls Metric Servers Only Twice

Problem:

In previous releases of the TCP/IP software, the load broker would poll the metric servers twice before marking the address for removal from the DNS alias. The documentation stated that metric servers would be polled three times before the address is marked for removal.

Solution:

The software has been corrected to align with the documentation.

4.11 UCP Problems Fixed in This Release

The following section describes UCP problems fixed in this release.

4.11.1 TCPIP SHOW CONFIG NAME Incorrectly Generates Write Audit Alarm

Problem:

`TCPIP SHOW CONFIGURATION NAME` command generates security alarm for `WRITE` operation on `TCPIP$CONFIGURATION.DAT` file.

Solution:

This release fixes this problem. The `WRITE` mode has been removed while accessing the `TCPIP$CONFIGURATION.DAT` when using `TCPIP SHOW CONFIGURATION` commands.

4.11.2 TCPIP SHOW MAIL/ENTRY Failure

Problem:

The `TCPIP SHOW MAIL /ENTRY=`*`entry_number`* fails for every alternative attempts when executed from the same terminal session.

Solution:

This problem has been fixed in this release.

4.11.3 PIPE to tcpip show conf communication fails

Problem:

The `tcpip show configuration communication` command works well when running standalone. However, the command fails when executing the same in a pipe and displays the following error:

```
$pipe tcpip show configuration communication | type sys$input
%TCPIP-E-TCPIPDISPLAY, error displaying information
-TCPIP-F-BUGCHK, TCPIP internal error
-RMS-F-SYS, QIO system service request failed
```

Solution:

This problem has been fixed in this release.

4.11.4 Problems Generating Correct Database Files with the TCPIP CONVERT/UNIX BIND Command

Problem:

In previous releases, there could be problems generating correct database files when using the `TCPIP CONVERT /UNIX BIND` command. This could result in database files that contained unqualified hostnames in the SOA and NS records.

Solution:

This problem has been corrected.

4.11.5 Illegal BIND Resolver Search Lists Defined via the TCPIP SET NAME/PATH Command

Problem:

In previous releases, UCP would allow illegal BIND Resolver search list (paths) to be defined via the TCPIP SET NAME/PATH command.

Solution:

This problem has been corrected.

4.12 RLOGIN Problems Fixed in This Release

The following section describes RLOGIN problems fixed in this release.

4.12.1 System Crash, INCONSTATE for an RLOGIN socket

Problem: System crash with INCONSTATE when logging out of RLOGIN.

Solution:

This problem is only with the Scalable Kernel and fixed in this release.

4.13 RSH Problems Fixed in This Release

The following section describes RSH problems fixed in this release.

4.13.1 RMT Server Does Not Work with Solaris Clients

Problem: OpenVMS RMT server does not work with Solaris RMT clients.

Solution:

This release corrects this problem. Upon the failure, the Solaris client checks that it can still access the server by sending an `S`. The fix is to add code to detect the `S` command.

4.13.2 RSH /Escape_character for the Alpha Client Causes an Access Violation

Problem: RSH /escape_character for the Alpha client cause either an access violation or improper terminal characteristics.

Solution:

This problem is fixed in this release.

4.14 RCP Problems Fixed in This Release

The following section describes RCP problems fixed in this release.

4.14.1 RCP Command Returns Error Status When /LOG Option is Used

Problem: The RCP command returns error status when /LOG option is used though the job completed successfully.

Solution:

This problem is corrected in this release. The RCP command now returns the appropriate status when /log option is used.

4.14.2 RCP Cannot Locate A File in the Current Directory When SET DEFAULTed to a Search List

Problem: RCP cannot locate a file in the current directory when you are SET DEFAULTed to a search list.

Solution:

This release corrects this problem in RCP. RCP can now locate and copy the file in the current directory when you have SET DEFAULTed to a search list.

4.15 SMTP Problems Fixed in This Release

The following sections describe SMTP problems fixed in this release.

4.15.1 Try-A-Records Governs SMTP Symbiont Use of A Records For Relay

Problem:

When attempts to relay outbound mail to the gateway(s) specified in MX records fail, the SMTP symbiont tries to relay outbound mail using A records. This is a hedge against misconfigured MX records. In today's Internet however, hosts pointed to by A records for a domain are often configured to reject mail for the domain when it doesn't come from a known host as a counter measure to protect against SPAM route through. Attempts to relay mail to such a host may be rejected mid-way through the SMTP dialog. This causes the message to be bounced.

Solution:

A new Try-A-Records switch is added to the SMTP.CONFIG file to govern the SMTP symbiont's use of A records for outbound mail relay should attempts to relay to MX gateways fail or should no MX records be present.

The switch can take the values "NEVER", "ALWAYS" or IFNOMX". These values are as follows:

Value	SMTP Symbiont Behavior
NEVER	The SMTP symbiont will never try to relay mail using A records, even if no MX records are found.
ALWAYS	The SMTP symbiont will always try to relay mail using A records. Note that gateways specified in MX records are still tried *first*. A records are used only if attempts to contact MX gateway(s) fail or when no MX records are found.

IFNOMX	The SMTP symbiont will try to relay mail using A records only if no MX records are found. If one or more MX records are found, A records will not be used. The default value of the configuration field for Try-A-records is IFNOMX.

4.15.2 Any Message Header That Unfolds into a Single Line Longer Than 7192 Bytes Causes SFF to Loop Infinitely

Problem:

The SMTP SFF feature (`TCPIP$SMTP_SFF.EXE` image) loops for a mail message that contains a single header longer than 7192 bytes. If such a message is delivered to a recipient who has email forwarded to a `PIPE% MAILSHR` mechanism that uses SFF (such as SpamAssassin), the symbiont will hang, waiting for the looping PIPE% child process.

Solution:

A fix has been implemented in this release.

4.15.3 SMTP Fails to Send Mail with a Record Size Greater than 4093

Problem:

SMTP symbiont had a buffer limit of 4093 characters per record it reads from the control file (CF). Any record that exceeded this limit resulted in `%TCPIP-E-SMTP_CFGETERROR` and deletion of that control file. This eventually resulted in loss of mail.

Solution:

With this release, the read buffer limit of the SMTP symbiont is made flexible to extend itself to the largest record size limit of a message.

4.15.4 Unprivileged User Sending MAIL Results in Security Alarms for Queue CONTROL and READ access

Problem:

When a user sends MAIL, this would be submitted to the SMTP symbiont server queue i.e., TCPIP$SMTP_*nodename*_00. The SMTP has to check for the existence of this queue. Hence when an unprivileged user sends a mail, the security alarms for queue control and read access are being generated.

Solution:

The code has been rectified to suppress these security alarms while searching for the queue.

4.15.5 MAIL to SMTP% Causes Security Alarms

Problem:

When a user without privileges turned on but with `SYSPRV` granted in the SYSUAF as an authorized privileges sends MAIL to %SMTP, this causes security alarms.

Solution:

This fix is provided in this release. The solution was to provide the appropriate privileges.

4.15.6 ACCVIO Due to Improper Parsing

Problem:

Upon starting SMTP or issuing the TCPIP SHOW MX command, there could be an ACCVIO due to improper parsing of the Authority section of the DNS response message.

Solution:

This problem is corrected in this release.

4.15.7 Selecting MX Records to Route Mails Correctly

Problem:

Selection of MX records to route mails to destination did not work according to the preference values. When multiple MX records from the DNS server are given and one of them has a preference value of 32768 or 65535, then the MX record with that value will be used first instead of other MX records with lower preference values.

Solution:

This problem is corrected in this release.

4.16 Startup Problems Corrected in This Release

The following sections describe Startup problems fixed in this release.

4.16.1 Unrecognized Command Verb Errors

Problem:

Previously, users of site specific startup and shutdown command procedures could get `%DCL-W-IVVERB, unrecognized command verb` errors if they attempt to define/use symbols from within those procedures.

Solution:

This problem has been corrected.

4.17 SNMP Problems Fixed in This Release

The following sections describe SNMP problems fixed in this release.

4.17.1 SNMP Poll Time Is Not Configurable

Problem:

The SNMP poll time could not be changed. At times, this would cause the SNMP process to loop consuming high CPU utilization.

Solution:

The default reset/refresh value of SNMP_POLL_TIME value is 30 seconds. This release allows the user to set the desired poll time.

4.18 Sockets API Problems Fixed in This Release

The following sections describe Sockets API problems fixed in this release.

4.18.1 Socket Function getaddrinfo() Hangs

Problem:

Two successive calls to `getaddrinfo()` in the same program cause the second call to hang. This is only true if the `af` parameter is AF_INET6 and the `ai_flags` parameter has not been set to AI_ALL or AI_ADDRCONFIG.

Solution:

This problem is corrected in this release.

4.19 SSH Problems Fixed in This Release

The following sections describe SSH problems fixed in this release.

4.19.1 OpenVMS SSH Does Not Support Mixed Case Passwords

Problem:

OpenVMS SSH does not support mixed case passwords.

Solution:

Mixed passwords supported, assuming the username has the PWDMIX flag set. Note that when converting an account to use mixed case passwords, for access through SSH or other method, you must exercise care in resetting the password. Specifically, beware of the following sequence:

- pwdmix flag not set, password: changeme (Can be entered in any case.)
- set pwdmix flag; now can login only with CHANGEME (all uppercase)
- reset password to be changme (lowercase)
- now can login only with changeme (all lowercase)
- remove pwdmix flag
- Now the user may be unable to login until the password is reset by system administrator.

Note that the problems shown by this example have nothing directly to do with ssh, but are a function of OpenVMS password handling.

4.19.2 Signals Cause Extraneous or Cryptic Messages

Problem:

Signals received by ssh client and server result in the display of extraneous or cryptic messages.

Solution:

Some signals are now silently ignored; others result in standard VMS/DCL output.

4.19.3 CTRL/C Did Not Work During sftp2/scp2 filecopy

Problem:

`CTRL/C` did not work once a filecopy had started in sftp2 or scp2, stopping the ssh/sftp/scp processes was the only way to abort copy on a large file.

Solution:

After entry of CTRL/C additional steps may be needed to restart a filecopy. For sftp, for example, it may be necessary to enter the quit command, and then restart sftp; for scp it may be necessary to enter CTRL/C a second time or a $ STOP at the DCL level.

The target file may remain locked until the client side processes have been fully stopped.

The attributes on an incompletely copied target file may not be correct. In this case a manual deletion of the incomplete target file may be needed once the client side processes have either completed or been stopped.

4.19.4 Usernames with $ Not Supported

Problem:

Usernames with $ not supported.

Solution:

Any valid OpenVMS usernames are now supported.

4.19.5 Problem With Timeout in Locking of X11 xauth Authority File

Problem:

Error in tcpip$ssh_run.log indicates timeout in locking authority file, combined with failure to run X application in the ssh session started at ssh client.

Solution:

The following documentation on a new option, DecwXauthLockAction, is drawn from that included in the file sshd2_config., included in this release:

```
# Valid options are:
# none: no special action (default)
#   This option is also in effect if there is no value specified, or if
#   the variable is commented out.
# break: break lock (xauth -b)
# ignore: ignore lock (xauth -i)
# file: use alternate xauth filename (xauth -f {filename})
#
# DecwXauthLockAction none
```

There is a risk to using the "break" or "ignore" options. The general rule is that whichever user exits last will write a version of the xauth file which includes only the contents at the time it opened the file + any changes that user made. Any changes from other user(s) are lost.

If a user's display station has considerable activity from different users (including applications), then using "ignore" may cause problems. Perhaps in a case of the typical ssh user, the display host is single user, and is that user's dedicated display device; in that case ignore may be reasonable.

If the user is not concerned about having multiple users on a host, then either the "ignore" or "break" values may be appropriate.

Because of the potential for lost data, users may prefer the "file" option. In this case, each ssh server that starts when the xauth file is locked will write for the user a unique xauth file, to be used only by sessions supported by that instance of the ssh server. The file is located in the user's sys$login, and has a name in the format: DECW$XAUTHORITY.DECW$XAUTH*nnnnnnnn* where *nnnnnnnn* = the 8 digit hex value of the pid of ssh server process (not of the user's interactive session process). On OpenVMS, each ssh session starts a new server process, and so

the xauth file will be used by a user for a single ssh session; hence there will be no conflict with either the default xauth file or xauth files from different ssh server instances. The pid of the server process is used because given the way the base UNIX code works, the file has to be created before the interactive terminal process is created.

One restriction with the "file" option, that does not apply to the "break" or "ignore" options: `$ CREATE/TERM` does not work.

Because the `DECW$DISPLAY` logical is in the job logical name table (so that the terminal process can inherit it with the ssh server process), the `DECW$XAUTH` logical is in that table also, for the same reason.

For more on xauth and interaction of ssh and X11, see the DECwindows/Motif documentation, especially *New Features* guide, and commercially published books on X11.

4.19.6 Cannot Issue a $ CREATE TERM/DETACH from an SSH Session Itself Created Using That Command

Problem:

Cannot cascade `$ CREATE TERM/DETACH` from ssh session (using x11 port forwarding). That is, from window created from original session window, cannot do another `$ CREATE TERM/DETACH`.

Solution:

Cascading is now supported.

4.19.7 SSH Client and Server Startup Fail If the Correct Version of DECwindows Motif Is Not Installed and Started

Problem:

If the host does not have the file `SYS$SHARE:DECW$SETSHODISSHR.EXE` installed, client and server startup fail, even if X11 port forwarding is not requested or used. The following are possible situations:

- DECwindows Motif V1.3 is installed, but not started (executable not available).
-

A pre-V1.3 version is installed (file is not delivered).

Solution:

SSH client and server do not attempt X11 processing if the file is not found or available.

4.19.8 The SFTP Client Does Not Sense the Terminal Page Size Properly

Problem:

The SFTP client does not sense the terminal page size properly. The screen output is forced to the default setting of 23 lines per page.

Solution:

This problem has been corrected.

4.19.9 SSH Filecopy Clients Cannot Use of Group Logical Names on the SFTP Server

Problem:

Users of SCP and SFTP cannot make use of group logical names on the SFTP server. This problem occurs because the group logical name table in the SFTP server process points to the TCPIP$SSH account's group logical name table of the instead of the table of the account that is being used for the file transfer.

Solution:

The SSH file copy now points the group logical name table of the SFTP server process to the table of the group of the account that is being used for the file transfer. For example, when connecting to the account "JONES" in the user group "777", the sftp server process's group logical name table will be set to point to LNM$GROUP_000777.

4.19.10 VMS Text Editor and the DCL SEARCH Command See SSH Server Log File Warning Messages

Problem:

VMS text editor and the DCL SEARCH command see SSH server log file messages like WARNING: Starting image in auxiliary server mode as two separate lines with the text of message on a different line from the "WARNING:" prefix.

The DCL TYPE command functions properly by displaying one line.

This behavior may cause some customer auditing procedures to fail.

Solution:

Write the warning message as a single line.

4.19.11 SSH Client Ignores Any DNS AAAA Records Belonging to the Remote Host

Problem:

The SSH client ignores any DNS AAAA records belonging to the remote host thus effectively disabling connecting via IPv6.

Solution:

Recognize DNS AAAA records.

4.19.12 Publickey Authentication Fails

Problem:

From some clients, e.g., the PuTTY client, when the username entered is other than all lowercase, publickey authentication fails.

Solution:

The SSH server now captures the original filename as sent from the client and uses it in authentication procedure.

4.19.13 Regular Expression Syntax Parsing Not Done

Problem:

Regular expression syntax parsing not being done on for AllowHosts in `sshd2_config`.

Solution:

Regular expressions are now parsed using the "egrep" syntax. Variables to which this appies include `AllowHost`, `DenyHosts`, `AllowUsers`, and `DenyUsers`. See examples in `sshd2_config`. One OpenVMS extension was retained from previous versions: To specify allow all hosts, both of the following work:

```
Standard regular expression egrep syntax:
       AllowHosts  .*

OpenVMS extension:
       AllowHosts  *
```

Note that this format is the only case in which the "*" is accepted in addition to the ".*". In longer expressions, both characters ".*" are still required.

4.19.14 Login Dates Manipulation Sets Off Audit

Problem:

Manipulation of interactive and noninteractive login dates for SSH session sets off audit alrms.

Solution:

The audit alarms are now suppressed for date manipulation.

4.19.15 SFTP Server Causes Auditing Alarms

Problem:

The SFTP server causes auditing alarms in the operator's log.

Solution:

Condition causing the alarms corrected.

4.19.16 SFTP File Transfers Do Not Preserve OpenVMS File Attributes

Problem:

SFTP file transfers do not preserve OpenVMS file attributes even when the SFTP client and server are both running the TCP/IP Services implementation of SFTP. Such file transfers should preserve a file's record format and file attributes and, where applicable the RMS MRS and LRL fields.

Solution:

When both the SFTP client and server are running the TCP/IP Services implementation of SFTP, file transfers preserve the following OpenVMS file attributes:

- Record format
- File attributes
- `mrs` and/or `lrl` (where applicable)

The attributes are preserved regardless of the direction of the file's transfer: to client or to server.

4.19.17 SSH Password Change Sequence Did Not Check for Password in History File

Problem:

SSH password change sequence did not check for password in history file, or for using same new password as the old one.

Solution:

Password history is now checked, unless the username has the DISPWDHIS flag set.

4.19.18 Non-OpenVMS Clients Overwrite Files on OpenVMS Servers

Problem:

Under certain conditions, non-OpenVMS clients would cause existing file on the OpenVMS server to be overwritten on filecopy (instead of creating new version).

Solution:

A new version is created; for scp only the `-k` option is available to force an overwrite.

4.19.19 SSH Client Does Not See Entries in TCPIP$ETC:IPNODES.DAT

Problem:

Entries made in the `TCPIP$ETC:IPNODES.DAT` file are not seen by the SSH client.

Solution:

Condition interfering with reading of the IPNODES.DAT file has been corrected.

4.19.20 Limited Support for ODS-5 File Format

Problem:

SSK file copy clients and servers have limited support for the ODS-5 file format.

Solution:

This problem has been corrected. Note that the following limitations apply to this fix:

- It addresses only problems with sftp ls/get/put and scp copy of files with extended filenames. ODS-5 syntax may work in other situations, but they are neither guaranteed nor supported.
- It is not intended to handle copying of files with extended file names to target directories on ODS-2 volumes. In that case an error of the following type results:

```
tcpip$ssh_scp2.exe: warning: open: ./afile,name.txt (dst):
unspecified failure (server msg: 'syserr: bad file number,
file: ./afile,name.txt')

%TCPIP-E-SSH_FC_ERR_FAIL, undetermined error from sshfilexfer
```

- When wildcards are used in file specification not all files with ODS-5 extended file name may be retrieved. For example: `sftp>get *.*;*` retrieves all files, while `sftp get afil*.txt` does not get fines with ODS-5 characters:

```
sftp> get *.*;*
afile2.txt;1                    |     8B |   0.0 kB/s | TOC: 00:00:01 | 100%
afile^%name.txt;2               |  1008B |   1.0 kB/s | TOC: 00:00:01 | 100%
afile^,name.txt;1               |     6B |   0.0 kB/s | TOC: 00:00:01 | 100%
afile^^^%name.txt;1             |    12B |   0.0 kB/s | TOC: 00:00:01 | 100%
AFILE__NAME.TXT;1               |    20B |   0.0 kB/s | TOC: 00:00:01 | 100%

sftp> get afil*.txt
afile2.txt                      |     8B |   0.0 kB/s | TOC: 00:00:01 | 100%
afile__name.txt                 |    20B |   0.0 kB/s | TOC: 00:00:01 | 100%
```

4.19.21 Fixed SFTP2 Image Exits with Normal Status

Problem:

SFTP2 image exits with status "normal" ever after errors are encountered.

Solution:

The following DCL exit codes are now supported for batch procedure exit. Of these, only TCPIP$_SSH_FC_ERR_TGT_EXISTS is a new code:

```
TCPIP$_SSH_FATAL "non-specific fatal error condition"
TCPIP$_SSH_FC_OK "operation was successful"
TCPIP$_SSH_FX_OK "the operation completed successfully"
TCPIP$_SSH_INFORMATIONAL "ssh informational"
TCPIP$_SSH_ERROR "non-specific error condition"
TCPIP$_SSH_FX_EOF "the operation failed because of trying to read at end of file"
TCPIP$_SSH_FX_NO_SUCH_FILE "the requested file does not exist"
TCPIP$_SSH_FX_PERM_DENIED "insufficient privileges to perform the operation"
TCPIP$_SSH_FX_FAILURE "the requested operation failed"
TCPIP$_SSH_FX_BAD_MESSAGE "a badly formatted message was received; error or incompatibility in the protocol i
TCPIP$_SSH_FX_NO_CONNECTION "connection has not been established (yet)"
TCPIP$_SSH_FX_CONNECTION_LOST "connection to the server was lost, and the operation could not be performed"
TCPIP$_SSH_FX_OP_UNSUPPORTED "operation is unsupported by the fileserver"
TCPIP$_SSH_FX_INVAL_RFMT "record format not supported"
TCPIP$_SSH_FX_OUT_OF_MEMORY "out of dynamic memory"
TCPIP$_SSH_FC_ERROR "error in ssh file transfer operation"
TCPIP$_SSH_FC_ERR_DEST_NOT_DIR "destination is not directory or does not exist"
TCPIP$_SSH_FC_ERR_ELOOP "maximum symlink level exceeded"
TCPIP$_SSH_FC_ERR_CONN_FAILED "connecting to host failed"
TCPIP$_SSH_FC_ERR_CONN_LOST "connection broke for some reason"
TCPIP$_SSH_FC_ERR_NO_SUCH_FILE "file doesn't exist"
TCPIP$_SSH_FC_ERR_PERM_DENIED "no permission to access file"
TCPIP$_SSH_FC_ERR_FAILURE "error in ssh file transfer operation"
TCPIP$_SSH_FC_ERR_PROTO_MSMTCH "file transfer protocol mismatch"
TCPIP$_SSH_FC_ERR_INVAL_RFMT "file record format invalid for copy"
TCPIP$_SSH_FC_ERR_TGT_EXISTS "target file already exists"
```

If multiple errors are encountered, exit status reflects the last error. The following logical names are available to control behavior for the new functionality. To use these names define them to have a value "TRUE" (case insensitive) or any non-zero numeric value.

- TCPIP$SFTP_ALWAYS_EXIT_NORMAL: preserves old behavior of sftp exiting with status $STATUS "%X00000001" (normal), no matter what errors occurred during a session.
- TCPIP$SSH_SFTP_BATCH_ABORT_ON_ERROR: by default an sftp batch procedure continues after any errors except for failure of a cd (change directory) command. This behavior is the same as that for the base UNIX code. Setting this logical enables the OpenVMS-specific behavior of the procedure exiting after the first error is encountered.

4.19.22 SFTP Batch Procedure Files Need Special Format

Problem:

SFTP batch files must be in sream_lf format, or have each line except the last terminated by a linefeed (ASCII 10). **Solution:**

This version detects if a batch file is not in `stream_lf` format, and if it is not, attempts to convert it to `stream_lf` format. The following message is displayed when the process begins, where *batchfile* is the filename of the sftp batch file):

```
Warning: Converting file fail4.cmd to Stream_LF.
```

The following message indicates succesful conversion:

```
Warning: File {batchfile} converted successfully to Stream_LF.
```

while the following type of error indicates a failure (where *n* is an internal VMS error status):

```
openvms_specific/OPENVMS_SPECIFIC.C:1885: Error calling
CONV$PASS_FILES for {batchfile}. STATUS = %NONAME-E-NOMSG,
Message number n
```

If automatic conversion does not succeed, the file can be converted manually by VMS to stream_lf (e.g., by the DCL CONVERT command).

4.19.23 SSH File Transfer Clients and Server Do Not Handle VMS-style Wildcards

Problem:

SSH file transfer clients and server do not handle VMS-style wildcards.

Solution:

Many usages for VMS-style wildcards are now supported. The behavior, where possible, matches that for DCL commands such as `$ COPY` and `$ DIRECTORY`. For example, `ls afile.*` retrieves all versions of a file, while `get afile.*` retrieves only the highest version number. One extension to the standard VMS set is recognition of the `?` in addition to the `%` to match a single character.

4.19.24 Text Display for Usage Does Not Match Documentation

Problem:

Text display for `Usage:` does not match documentation or what is supported or tested.

Solution:

"Usage" text reflects what is implemented, and also matches information in any DCL help files.

4.19.25 Allow Restrictions on Execution of SFTP-server2

Problem:

Allow restrictions on access to SSH filecopy.

Solution:

The following methods are available to restrict users who have ssh access to a server from using scp or sftp for filecopy:

1. Use one of the following options in the `SSHD2_CONFIG.` file:

```
DisallowSftpServer
Default: "no"
"yes" disables sftp-server2 for all users

SftpDenyUsers
Default: empty string
Interprets regular expressions in the same way that
DenyUsers does.
```

Note that SftpDenyUsers is used only if DisallowSftpServer is "no."

2. If neither of the configuration restrictions is used, the server checks for the identifier TCPIP$SSH_FILECOPY_DISALLOWED granted to the current user, in which case access to sftp-server2 is denied.

   To create and grant this identifier, do the following from a privileged account:

```
$ MCR AUTHORIZE
UAF> ADD /IDENTIFIER TCPIP$SSH_FILECOPY_DISALLOWED
%UAF-I-RDBADDMSG, identifier TCPIP$SSH_FILECOPY_DISALLOWED
value %X8001009F added to rights database
UAF> SHOW /IDENTIFIER TCPIP$SSH_FILECOPY_DISALLOWED
Name                            Value           Attributes
TCPIP$SSH_FILECOPY_DISALLOWED   %X8001009F
UAF> GRANT TCPIP$SSH_FILECOPY_DISALLOWED USER1
%UAF-I-GRANTMSG, identifier TCPIP$SSH_FILECOPY_DISALLOWED
granted to USER1
UAF> SHOW USER1

Username: USER1                           Owner:  Default
...
Identifier                       Value          Attributes
TCPIP$SSH_FILECOPY_DISALLOWED    %X8001009F
```

4.19.26 Using SFTP To Pull Fixed Length Files Results In A Corrupted File

Problem:

Using SFTP to pull fixed length files with an odd-numbered record length, e.g., 773 bytes, from an OpenVMS system to a system running an operating system other than OpenVMS results in a corrupted file.

Solution:

This problem has been corrected.

4.19.27 Pasting from Text Editor Loses Characters

Problem:

When a user logs in with SSH and pastes from the paste buffer, characters can be lost. If the user is running a text editor, it receives a "data overrun" error.

Solution:

This problem has been corrected.

4.19.28 sftp ls on Directory with a Large Number of Files Cannot Be Interrupted

Problem:

When doing an ls for a directory or search list with a large number of files, entering q at the prompt "<Press any key for more or q to quit>" results in apparent hang that cannot be interrupted with CTRL/C.

Solution:

Pressing q now returns immediately to the sftp> prompt. Additional improvements for ls displays include the following:

1. The display has no blank lines, but does include the q (or other character) entered after the prompt.
2. To start an SFTP session with continuous display use the "-C" (Continuous display) option, e.g.:

   ```
   $ sftp "-C" yourremote
   ```

   Note that the double quotes are required. Within an SFTP session, use the td (toggle display) command to switch between prompted and continuous display.
3. Long directory listings do not cause the %TCPIP-F-SSH_ALLOC_ERROR error.
4. CTRL/C on continuous listings causes return to the sftp> prompt.

Note

Because global variables are used for this fix, the code is not thread-safe.

In batch mode the default remains to suppress display of the prompt. You cannot force the display of the prompt in batch mode.

If CTRL/C is entered at the "<Press any key...>" prompt, you may need to enter a "q" or a carriage return to return to the sftp> prompt. Note that entering CTRL/C at the sftp> prompt (followed by a carriage return) causes an exit to the DCL level.

4.20 SSL Problems Fixed in This Release

The following sections describe SSL problems fixed in this release.

4.20.1 After Installing SSL, POP SSL Ceases to Function

Problem:

After installing the SSL V1.2 kit on TCP/IP Services, POP SSL support ceases to function. The POP server will not listen on its SSL port and, consequently, will not service clients coming in through SSL. The TCPIP$POP_RUN.LOG POP server log file contains these lines:

```
POP server will not listen for SSL connections.
SSL$LIBCRYPTO_SHR32_INIT status: %LIB-E-KEYNOTFOU, key not found in tree
```

Solution:

This problem is corrected in this release.

4.21 TELNET Problems Fixed in This Release

The following sections describe TELNET problems fixed in this release.

4.21.1 TELNET Intrusion Detection Inflexibility

Problem:

In certain circumstances, an intrusion (such as an invalid login) by one user can cause the whole system to be locked out, and with multiport servers such as on a terminal server, all ports could be locked out. The workaround has been to set the TCPIP$TELNET_NO_REM_ID logical. However, this allows the intruding user to log in on another port without being locked out.

Solution:

This problem is corrected in this release. The logical name TCPIP$TELNET_TRUST_LOCATION allows you to specify how to handle TELNET intrusion records. When this logical name is defined, any location string specified by the remote client is included in the intrusion record. For example, many terminal servers provide the physical port number, while OpenVMS clients provide the originating user name and terminal line. Including this information in the intrusion records means that only a particular user or port will be locked out, not the entire remote host (and all user ports).

4.22 Miscellaneous Problems Fixed in This Release

The following sections describe miscellaneous problems fixed in this release.

4.22.1 PPP Supports the Scaling Kernel and IA64 Architecture

PPP now supports both the Scaling Kernel and IA64 architecture.

4.22.2 TCPIP SHOW ROUTE/MASK Reports Error

Problem:

TCPIP SHOW ROUTE *dest*/mask did not work as expected in few cases. In cases where mask value was greater than or equal to 24, the response to this command as follows:

```
%TCPIP-E-ROUTEERROR, error accessing routes database(TCPIP$ROUTE)
-TCPIP-W-NORECORD, information not found
This posed problems while checking for the dynamic routes.
```

Solution:

This problem is fixed in this release. The code now considers the CIDR mask specified while matching the given destination address

5
Documentation Update

This chapter describes updates to the information in the TCP/IP Services product documentation.

This information will be supplied in the final release of TCP/IP Services.

5.1 Documentation Updated for This Release

The following manuals are updated for TCP/IP Services Version 5.6. Documentation changes planned for these manuals are indicated.

- *TCP/IP Services for OpenVMS Installation and Configuration*
- *TCP/IP Services for OpenVMS Management Guide*
- *TCP/IP Services for OpenVMS Guide to SSH*

5.2 Documentation Not Being Updated for This Release

The following manuals are not updated for TCP/IP Services Version 5.6. Documentation changes planned for these manuals are indicated.

- *TCP/IP Services for OpenVMS Concepts and Planning*
- *TCP/IP Services for OpenVMS Management Command Reference*
- *TCP/IP Services for OpenVMS Management Command Quick Reference Card*
- *TCP/IP Services for OpenVMS ONC RPC Programming*
- *TCP/IP Services for OpenVMS Sockets API and System Services Programming*
- *TCP/IP Services for OpenVMS Tuning and Troubleshooting*
- *TCP/IP Services for OpenVMS User's Guide*

A

Implementing NTP Autokeys

To set up NTP autokeys, use one of the following procedures:

- For the TC identity scheme, use one of the following methods:
  - Section A.1
  - Section A.2
- For the PC identity scheme, see Section A.3.
- For the IFF scheme, use one of the following methods:
  - Section A.4
  - Section A.5
- For the GQ scheme, see Section A.6.
- For the MV scheme, see Section A.7.

A.1 Default TC Identity Scheme (method 1)

1. Make Alice a stratum 0 server by enabling the lines in TCPIP$NTP.CONF:

   ```
   server 127.127.1.0 prefer
   fudge 127.127.1.0 stratum 0
   ```

2. On both Alice (server) and Bob (client), add two lines to TCPIP$NTP.CONF:

   ```
   keysdir SYS$SPECIFIC:[TCPIP$NTP]
   crypto
   ```

3. On Bob, add the server line for Alice to Bob's TCPIP$NTP.CONF:

   ```
   server alice autokey
   ```

4. On Alice, generate the keys and trusted certificate:

   ```
   ALICE>ntp_keygen -"T"
   ```

5. On Bob, generate the keys and non-trusted certificate:

   ```
   BOB>ntp_keygen
   ```

6. Start NTP on Alice:

   ```
   ALICE>@sys$startup:tcpip$ntp_startup
   ```

7. Wait until Alice is synchronized to itself. `ntpdc -p` should show an asterisk (*) in the leftmost column.

8. Start NTP on Bob:

   ```
   BOB>@sys$startup:tcpip$ntp_startup
   ```

Bob should eventually synch to Alice (this may take up to 10 minutes). ntpdc -p should show an asterisk (*) in the leftmost column.

A.2 Default TC Identity Scheme (method 2)

1. Make Alice a stratum 0 server by enabling the lines in TCPIP$NTP.CONF:

```
server 127.127.1.0 prefer
fudge 127.127.1.0 stratum 0
```

2. On Alice, add two lines to TCPIP$NTP.CONF:

```
keysdir SYS$SPECIFIC:[TCPIP$NTP]
crypto pw littlesecret
```

3. On Bob, add three lines to TCPIP$NTP.CONF:

```
keysdir SYS$SPECIFIC:[TCPIP$NTP]
crypto pw bigsecret
server alice autokey
```

4. On Alice, generate the keys and trusted certificate using passwords:

```
ALICE>ntp_keygen -"T" -p littlesecret -q bigsecret
```

5. On Bob, generate the keys and non-trusted certificate using passwords:

```
BOB>ntp_keygen -q bigsecret
```

6. Start NTP on Alice:

```
ALICE>@sys$startup:tcpip$ntp_startup
```

7. Wait 5 minutes until Alice is synchronized to itself. ntpdc -p should show an asterisk (*) in the leftmost column.

8. Start NTP on Bob:

```
BOB>@sys$startup:tcpip$ntp_startup
```

Bob should eventually synch to Alice (maybe around 10 minutes). ntpdc -p should show an asterisk (*) in the leftmost column.

A.3 PC Identity Scheme

1. Make Alice a stratum 0 server by enabling the lines in TCPIP$NTP.CONF:

```
server 127.127.1.0 prefer
fudge 127.127.1.0 stratum 0
```

2. On both Alice and Bob, add two lines to TCPIP$NTP.CONF:

```
keysdir SYS$SPECIFIC:[TCPIP$NTP]
crypto pw littlesecret
```

3. On Bob, add the server line for Alice to Bob's TCPIP$NTP.CONF:

```
server alice autokey
```

4. On Alice, generate the keys and certificate:

```
ALICE>ntp_keygen -"P" -p littlesecret
```

5. Copy the certificate (tcpip$ntpkey_rsa-md5cert_alice.timestamp) and the key (tcpip$ntpkey_rsakey_alice.timestamp) from Alice to Bob's keysdir.

6. On Bob, create symbolic links to the files:

```
BOB>ntp_keygen -"P" -l tcpip$ntpkey_rsakey_alice.timestamp -
_BOB> tcpip$ntpkey_rsa-md5cert_alice.timestamp
```

7. Start NTP on Alice:

```
ALICE>@sys$startup:tcpip$ntp_startup
```

8. Wait 5 minutes until Alice is synchronized to itself. ntpdc -p should show an asterisk (*) in the leftmost column.

9. Start NTP on Bob:

```
BOB>@sys$startup:tcpip$ntp_startup
```

Bob should eventually synch to Alice (this may take up to 10 minutes). ntpdc -p should show an asterisk (*) in the leftmost column.

A.4 IFF scheme (method 1)

1. Make Alice a stratum 0 server by enabling the lines in TCPIP$NTP.CONF:

```
server 127.127.1.0 prefer
fudge 127.127.1.0 stratum 0
```

2. On both Alice and Bob, add two lines to TCPIP$NTP.CONF:

```
keysdir SYS$SPECIFIC:[TCPIP$NTP]
crypto pw littlesecret
```

3. On Bob, add the server line for Alice to Bob's TCPIP$NTP.CONF:

```
server alice autokey
```

4. On Alice, create the trusted public key and identity scheme parameter file.

   Use a password with at least 4 characters. This example is for the IFF identity scheme:

```
ALICE>ntp_keygen -"T" -"I" -p littlesecret
```

5. On Bob, generate the client parameters using the server password:

```
BOB>ntp_keygen -"H" -p littlesecret
```

6. Copy the tcpip$ntpkey_iffpar_alice.timestamp file from Alice to Bob's keysdir.

7. On Bob, create a symbolic link to the file:

```
BOB>ntp_keygen -"I" -l tcpip$ntpkey_iffpar_alice_tcpip_zko_h.3344261784
```

8. Start NTP on Alice:

```
ALICE>@sys$startup:tcpip$ntp_startup
```

9. Wait 5 minutes until Alice is synchronized to itself. ntpdc -p should show an asterisk (*) in the leftmost column.

10. Start NTP on Bob:

```
BOB>@sys$startup:tcpip$ntp_startup
```

Bob should eventually synch to Alice (this may take up to 10 minutes). ntpdc -p should show an asterisk (*) in the leftmost column.

A.5 Alternate IFF Scheme (method 2)

1. Make Alice a stratum 0 server by enabling the lines in TCPIP$NTP.CONF:

```
server 127.127.1.0 prefer
fudge 127.127.1.0 stratum 0
```

2. On Alice, add two lines to TCPIP$NTP.CONF:

```
keysdir SYS$SPECIFIC:[TCPIP$NTP]
crypto pw littlesecret
```

3. On Bob, add three lines to TCPIP$NTP.CONF:

```
keysdir SYS$SPECIFIC:[TCPIP$NTP]
crypto pw bigsecret
server alice autokey
```

4. On Alice, create the trusted public key and identity scheme parameter file.

   Use a password with at least 4 characters. This example is for the IFF identity scheme:

```
ALICE>ntp_keygen -"T" -"I" -p littlesecret
```

5. On Bob, generate the client parameters using the client password:

```
BOB>ntp_keygen -"H" -p bigsecret
```

6. On Alice, extract the client key specifying the server password and the client password:

```
ALICE>ntp_keygen -e -q littlesecret -p bigsecret
```

   The output will go to the screen.

7. On Bob, create a file with the name specified in the screen output from step 6, the file name after "Writing new IFF key". Paste the output from step 6 into the file. Here is an example of the final file on Bob (the first two line starting with # are just comments):

```
   BOB> typ SYS$SPECIFIC:[TCPIP$NTP]TCPIP$NTPKEY_IFFKEY_ALICE.3344272304
# SYS$SPECIFIC:[TCPIP$NTP]TCPIP$NTPKEY_IFFKEY_ALICE.3344272304
# Thu Dec 22 15:32:10 2005
-----BEGIN DSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-CBC,E03763213C218BDC

O9xAmWUEfJzCYEO6Zgn1KWm67M9NKlc/LzqHH+1K/kWQ/YXudUIf1ugdj+Umpphy
R5UyrpVz8kWms4M/VsPZBvMgP2SIXPyYO5ANz0WlMYbk9Myd8Xfc/6LEhYMEhxeM
Mjo95aUuWq/+YtlEAzrVvWjhQnHvNpHJtQxNw/7L6/ftVOGT0MuB1e9jJoaGo+lp
yBSbhUYmwiyZfJUYvteXfOME/XH3rEx3h8/8k88zL1qACetHxeFmUMIoQq7lUqjg
CeKMAidxgUWlmhixYVcUtvuD0ZNYqQ4jjUFfDrlgfAPmeHNLndehEStcQbB3ItLC
-----END DSA PRIVATE KEY-----
```

8. Create a symbolic link to the client key:

```
BOB>ntp_keygen -"I" -l tcpip$ntpkey_iffkey_alice.3344272304
```

9. Start NTP on Alice:

```
ALICE>@sys$startup:tcpip$ntp_startup
```

10. Wait 5 minutes until Alice is synchronized to itself. `ntpdc -p` should show an asterisk (*) in the leftmost column.

11. Start NTP on Bob:

```
BOB>@sys$startup:tcpip$ntp_startup
```

Bob should eventually synch to Alice (this may take up to 10 minutes). ntpdc -p should show an asterisk (*) in the leftmost column.

A.6 GQ scheme

1. Make Alice a stratum 0 server by enabling the lines in TCPIP$NTP.CONF:

```
server 127.127.1.0 prefer
fudge 127.127.1.0 stratum 0
```

2. On both Alice and Bob, add two lines to TCPIP$NTP.CONF:

```
keysdir SYS$SPECIFIC:[TCPIP$NTP]
crypto pw littlesecret
```

3. On Bob, add the server line for Alice to Bob's TCPIP$NTP.CONF:

```
server alice autokey
```

4. On Alice, generate the GQ parameters:

```
ALICE>ntp_keygen -"T" -"G" -p littlesecret
```

5. On Bob, generate the client parameters using the server password:

```
BOB>ntp_keygen -"H" -p littlesecret
```

6. Copy the GQ group key tcpip$ntpkey_gqpar_alice.timestamp from Alice to Bob's keysdir.

7. On Bob, create a symbolic link to the file, using the -r option to specify the server name:

```
BOB>ntp_keygen -"G" -r alice -l tcpip$ntpkey_gqpar_alice.timestamp
```

8. Start NTP on Alice:

```
ALICE>@sys$startup:tcpip$ntp_startup
```

9. Wait 5 minutes until Alice is synchronized to itself. <code-example>(ntpdc -p) should show an asterisk (*) in the leftmost column.

10. Start NTP on Bob:

```
BOB>@sys$startup:tcpip$ntp_startup
```

Bob should eventually synch to Alice (this may take up to 10 minutes). ntpdc -p should show an asterisk (*) in the leftmost column.

A.7 MV scheme

1. Make Alice a stratum 0 server by enabling the lines in TCPIP$NTP.CONF:

```
server 127.127.1.0 prefer
fudge 127.127.1.0 stratum 0
```

2. On both Alice and Bob, add two lines to TCPIP$NTP.CONF:

```
keysdir SYS$SPECIFIC:[TCPIP$NTP]
crypto pw littlesecret
```

3. On Bob, add the server line for Alice to Bob's TCPIP$NTP.CONF:

```
server alice autokey
```

4. On Alice, generate the MV parameters. The MV parameter generation process produces a server key and a number of client keys. When choosing the number of client keys, avoid factors of 512 and do not exceed 30. The following command will generate 4 keys (N-1, where N is 5):

```
ALICE>ntp_keygen -"T" -"V" 5 -p littlesecret
```

5. On Bob, generate the client parameters using the server password:

```
BOB>ntp_keygen -"H" -p littlesecret
```

6. Copy any one of the MV client keys `tcpip$ntpkey_mvkeyN_alice.timestamp` from Alice to Bob's `keysdir`.

7. On Bob, create a symbolic link to the file. Specify "1" after the -"V" option so it does not complain that the -"V" option requires a value. The "1" will be ignored.

```
BOB>ntp_keygen -"V" 1 -l tcpip$ntpkey_mvkeyN_alice.timestamp
```

8. Start NTP on Alice:

```
ALICE>@sys$startup:tcpip$ntp_startup
```

9. Wait 5 minutes until Alice is synchronized to itself. `ntpdc -p` should show an asterisk (*) in the leftmost column.

10. Start NTP on Bob:

```
BOB>@sys$startup:tcpip$ntp_startup
```

Bob should eventually synch to Alice (this may take up to 10 minutes). `ntpdc -p` should show an asterisk (*) in the leftmost column.