HP TCP/IP Services for OpenVMS

Release Notes

June 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 HP OpenVMS Alpha Version 8.3 OpenVMS I64 Version 8.2.1 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

4 Corrections

5 Documentation Update

A Implementing NTP Autokeys

B SSH Kerberos Authentication Methods

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 assumes 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 5.5 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 5.5, 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 or 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.

1.1 BIND 9 Resolver

This release includes a new version of the BIND resolver that brings several API updates and 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 BIND 8.)

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 for direct local delivery.

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-tgt-2@ssh.com`

These authentication methods are described in Appendix B.

For more information about Kerberos, refer to the *HP Open Source Security for OpenVMS, Volume 3: Kerberos* manual.

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.

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 has not been qualified for this release.

Note

In the next version of TCP/IP Services, the capability of upgrading directly from any version of TCP/IP Services prior to 5.0 will be removed. Version 5.5 of TCP/IP Services is the last release that includes this capability.

2.3 Adding a System to an OpenVMS Cluster

The TCPIP$CONFIG.COM configuration procedure for TCP/IP Services Version 5.5 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 3.12.6.

2.4 SSH Configuration Files Must Be Updated

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 The SNMP master agent in TCPIP V5.6 fails to start if IPv6 is not enabled

The following message appears in the log file

```
SYS$SYSDEVICE:[TCPIP$SNMP]TCPIP$SNMP_RUN.LOG:
...
Usage:
$snmpmaster="$manga$dkb500:[sys0.syscommon.][sysexe]tcpip$esnmp_server.exe"
$snmpmaster [-t] [-i] [-p]
```

or

```
$mcr manga$dkb500:[sys0.syscommon.][sysexe]tcpip$esnmp_server.exe[-t][-i][-p]
-t Specifies trace mode.
-i Specifies internet transport allowed for subagent communications.
-p Specifies the port the SNMP master agent listens on for SNMP requests.
TCPIP$SNMP   job terminated at 14-DEC-2005 06:29:38.27
...
```

A partial workaround is to execute the following commands to enable IPv6:

```
$ @sys$startup:tcpip$define_commands
$ ifconfig we0 ipv6
```

After this change SNMP starts up; however, the master agent will still not send out traps. The following message is logged in TCPIP$SNMP_RUN.LOG:

```
20-DEC-2005 08:13:18.86 **ERROR TRAPS.C line 800: Error sending trap: 49
```

3.3 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.

3.4 Restrictions on OpenVMS I64 Platforms

The following restrictions apply to OpenVMS I64 platforms only:

- The output of the TCP/IP management command SHOW VERSION/ALL differs from the output seen on OpenVMS Alpha and VAX systems. (The information is listed in one column, and the image name and location are combined.)

3.5 SLIP Restrictions

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

3.6 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.

3.7 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.

3.8 IPv6 Restrictions

The following sections describe restrictions in the use of IPv6.

3.8.1 Mobile IPv6 Restrictions

Mobile IPv6 is not supported in this release.

3.8.2 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.

3.9 NFS Restrictions on Alpha Platforms

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

3.9.1 NFS Server Problems and Restrictions

For other information specific to the NFS server on OpenVMS I64 systems, see Section 3.4.

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

- Using the `ls` command from a Solaris Version 9 client may hang the OpenVMS server with no error message on either client or server. To avoid this problem, set the `nfs` subsystem attribute `ovms_xqp_plus_enabled` to 7. Refer to the *HP TCP/IP Services for OpenVMS Management* guide for more information about this attribute.
- 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.

3.9.2 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.

3.10 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.

3.11 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.

3.11.1 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
```

3.11.2 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.

3.11.3 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
  ```

3.11.4 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.

3.11.5 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.

3.11.6 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
```

3.11.7 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.

3.11.8 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.

3.11.9 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.

3.12 SSH Problems and Restrictions

This section contains the following information:

- SSH-related security advisories (Section 3.12.1)
- SSH general notes and restrictions (Section 3.12.2)
- UNIX features that are not supported by SSH (Section 3.12.3)
- SSH command syntax notes and restrictions (Section 3.12.4)
- SSH authentication notes and restrictions (Section 3.12.5)
- SSH keys notes and restrictions (Section 3.12.6)
- SSH session restrictions (Section 3.12.7)
- SSH messages notes and restrictions (Section 3.12.8)
- SSH remote command notes and restrictions (Section 3.12.9)
- SSH batch mode restrictions (Section 3.12.10)
- X11 port forwarding restrictions (Section 3.12.11)
- File transfer restrictions (all file sizes) (Section 3.12.12)

- File transfer restrictions (large files) (Section 3.12.13)

Note

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

3.12.1 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.

3.12.2 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.

3.12.3 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.

3.12.4 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]'
  ```

3.12.5 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 lassie
%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.

3.12.6 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.

3.12.7 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.
- You cannot shut down an OpenVMS system from an SSH session, such as by executing the command:

  ```
  $ @SYS$SYSTEM:SHUTDOWN.COM
  ```

  However, the following command can be used instead:

  ```
  $ MCR SYSMAN SHUTDOWN NODE /qualifiers
  ```

  The qualifiers on this command map directly to the options provided by SHUTDOWN.COM.
- 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.

3.12.8 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.

3.12.9 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.

3.12.10 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.
  - Each line of *batchfile*, except the last, must end with a line feed (`<LF>`, ASCII decimal 10).
- 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 3.12.6.

– 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.

3.12.11 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 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 16.20.176.33 is the server node address:

  ```
  $ SET DISPLAY/CREATE/NODE=16.20.176.33/TRANSPORT=TCPIP
  ```

  SSH on OpenVMS only supports 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.

3.12.12 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.

- File transfer is limited to OpenVMS files with the following record formats (as displayed by the DIRECTORY/FULL command):

  For "gets" from an OpenVMS ssh server:

  – STREAM_LF

  – Fixed-length records (any size)

  – Variable length or VFC files

  In addition, the record attributes of the following types of files are preserved for the following types of files on the OpenVMS ssh client side:

  – Fixed-length records (any size)

  – Variable length *except* for files with fortran carriage control

- 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 4.14.

- 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.

3.12.13 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.

3.13 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.

3.14 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 Link Conflicts Occur When Linking to the TCPIP$LIB.OLB Library

Problem:

Link conflicts occur when a program that includes references to the `strdup` or `putenv` function is linked to the TCPIP$LIB.OLB library. The linker produces the %LINK-W-MULDEF warning message, indicating a conflict with functions of the same name in the C RTL library.

Solution:

In earlier versions of TCP/IP Services, the TCPIP$LIB.OLB library included functions that have since been defined in more recent versions of the OpenVMS C RTL library. These TCPIP$LIB.OLB routines, which have the DECC$ prefix, conflict with the routines of the same name in the recent versions of the C RTL library. With this release of TCP/IP Services, the TCPIP$LIB.OLB library has been modified to prevent such conflicts.

4.2 BIND Server Problems Fixed in This Release

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

4.2.1 BIND Slave Refusing Notify Requests

Problem:

A BIND server configured as a slave can refuse notify requests from a master server. The error message written to the TCPIP$BIND_RUN.LOG on the slave includes the text "refused notify from non-master." This problem occurs when the master server has been enabled for IPv6 communication by having the `listen-on-v6` directive specified in the options statement in the TCPIP$BIND.CONF configuration file.

Solution:

This problem is corrected in this release.

4.2.2 The BIND Version 9 Server Process Exits With "Assertion Failure" Error

Problem:

The BIND server process exits with one of the following messages logged in the TCPIP$BIND_RUN.LOG file:

```
REQUIRE((((task) != 0L) && (((const isc__magic_t*)(task))->magic
== (((’T’)<< 24 | (’A’) << 16 | (’S’) << 8 | (’K’)))))) failed
Sun 19 03:00:13 CRITICAL: exiting (due to assertion failure)
%SYSTEM-F-OPCCUS, opcode reserved to customer fault at
PC=FFFFFFFF80A6C924, PS=0000001B

REQUIRE(res->item_out == isc_boolean_true) failed
Fri 19 13:12:04 CRITICAL: exiting (due to assertion failure)
%SYSTEM-F-OPCCUS, opcode reserved to customer fault at
PC=FFFFFFFF80E6C924, PS=0000001B
```

Solution:

This problem is corrected in this release.

4.3 failSAFE IP Problems Fixed in This Release

The following sections describe failSAFE IP problems fixed in this release.

4.3.1 failSAFE IP Phantom Failures

Problem:

Phantom failures can occur on systems where failSAFE IP is configured with a single interface address on multiple interfaces. When LAN traffic is infrequent, failSAFE IP can signal a false error.

Solution:

This problem is corrected in this release. failSAFE IP now generates MAC-level broadcast packets, by default. The new configuration parameter GENERATE_TRAFFIC can be set to force failSAFE IP to generate gratuitous ARP packets. You can include the following new configuration parameters in the TCPIP$FAILSAFE.CONF file:

GENERATE_TRAFFIC

Enables failSAFE IP to periodically generate either MAC-level broadcasts or gratuitous ARP packets. You can also configure failSAFE IP to turn off traffic generation.

Default: `mac` (MAC-level broadcast)
Other options: `arp` (gratuitous ARP packets) or `off`

The following is an example line in the configuration file setting the parameter to generate gratuitous ARP packets:

```
GENERATE_TRAFFIC: ARP
```

MAC_PTY

If MAC-level broadcast traffic is being generated, this parameter allows you to specify the MAC protocol type (a two-byte hexadecimal number, such as 6005).

If MAC_PTY is not specified, the MAC broadcast tries each protocol type until an available one is found.

The following is an example line in the configuration file setting the MAC protocol type as 6005:

```
MAC_PTY: 6005
```

For more information about configuring failSAFE IP, see the *HP TCP/IP Services for OpenVMS Management* guide.

4.3.2 Users Cannot Change the Location for the failSAFE IP Log File

Problem:

failSAFE IP log files are always named:

SYS$SYSDEVICE:[TCPIP$FSAFE]TCPIP$FAILSAFE_*node-name*.LOG

Users cannot specify alternate locations on their systems.

Solution:

This problem is corrected in this release. The new configuration parameter LOGFILE allows users to specify a log file location other than the default.

LOGFILE

Specifies the file specification for the log file created by failSAFE IP. The default is SYS$SYSDEVICE:[TCPIP$FSAFE]TCPIP$FAILSAFE_*node-name*.log. Specify the parameter and location as in the following example:

```
LOGFILE: DEV1:[STATS]FAILSAFE.LOG
```

For more information about configuring failSAFE IP, see the *HP TCP/IP Services for OpenVMS Management* guide.

4.3.3 SHOW INTERFACE Command Does Not Display Pseudointerface Addresses

Problem:

After an interface fails or recovers an alias address, the TCP/IP management command SHOW INTERFACE does not display pseudointerface addresses.

Solution:

This problem is corrected in this release.

4.4 FTP Server Problems Fixed in This Release

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

4.4.1 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.

4.4.2 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.

4.5 FTP Client Problems Fixed in This Release

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

4.5.1 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.

4.6 IMAP Problems Fixed in This Release

The following sections describe IMAP problems fixed in this release.

4.6.1 Mail Message Lost after IMAP Move and Purge

Problem:

If you manually move a message out of a folder and then use IMAP to purge the source folder, the mail is lost.

This problem occurs when you:

1. Select a mail file using the IMAP client.
2. Read the message using OpenVMS Mail and move it to another folder.
3. Enter the Expunge command on the selected folder, using the IMAP client.

The message disappears from the destination folder. If the message was copied to a new folder, the folder ceases to exist.

Solution:

This problem is corrected in this release.

4.6.2 IMAP CLOSE Command Does Not Function Properly

Problem:

When a client logs out by issuing the IMAP CLOSE command, the IMAP server does not delete all the messages marked for deletion.

Solution:

This problem is corrected in this release. When you enter the CLOSE command, the IMAP server deletes all the messages marked for deletion.

4.7 IPv6 Problems Fixed in This Release

The following sections describe IPv6 problems fixed in this release.

4.7.1 TCPIP$IP6_SETUP.COM Problems

This section describes TCPIP$IP6_SETUP.COM problems fixed in this release.

- **Problem:**

  The TCPIP$IP6_SETUP.COM command procedure for configuring IPv6 has the following problems:

  – Attempts to configure a 6to4 tunnel fail.

  – All routes required for 6to4 relay router are not configured.

  – The endpoints for automatic tunnels are not configured correctly.

  – IPv6 over IPv6 manual tunnels cannot be configured.

  – Errors are generated in the IPv6 configuration and initialization files during IPv6 host or router configuration.

  – Manual routes cannot be configured.

  Solution:

The configuration command procedure now enables you to successfully configure 6to4 tunnels, all routes required for a 6to4 relay router, automatic tunnels, IPv6 over IPv6 manual tunnels, and manual routes. (For more information, refer to the *HP TCP/IP Services for OpenVMS Installation and Configuration* guide.)

- **Problem:**

  The TCPIP$IP6_SETUP.COM command procedure requires that TCP/IP Services be started in order to verify specified addresses.

  Solution:

  This problem is corrected in this release. TCP/IP Services no longer needs to be started in order to run TCPIP$IP6_SETUP.COM.

4.7.2 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.

4.8 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.

4.8.4.1 Improper Mount Point Verification

Problem:

The MOUNT service exhibits improper verification of mount points for exported file systems.

Solution:

This problem is corrected in this release.

4.8.4.2 Cannot Mount ODS-5 File System

Problem:

When the TYPELESS_DIRECTORIES option is specified in the ADD EXPORT command, you cannot mount an ODS-5 file system even though the export entry contains a directory specification that does not end in .dir.

Solution:

This problem is corrected in this release.

4.8.4.3 Host Name Verification Occurs During Mount and Causes Failure

Problem:

When a client attempts to mount a file system, host name verification is performed even if the `mountd_option_*` `nfs` subsystem attributes were not set. An error or event message on the client may indicate permission denied. The MOUNT server may produce an OPCOM message indicating that the client host name and IP address are not consistent with the hosts database (TCPIP$HOST) or with DNS/BIND information.

Solution:

This problem is corrected in this release.

4.8.4.4 Misleading Mount Server Error

Problem:

The MOUNT server reports a misleading error message when the mount port is already in use.

If the mount port (port 10) is already in use, the mount server reports the following error:

```
ERROR: bind: address already in use
```

This can be mistaken for a BIND/DNS issue when in fact it is the C RTL call `bind()` that is failing.

Solution:

This problem has been corrected in this release. The message has been changed to:

```
ERROR: bind: mount server port(10) already in use
```

4.9 NTP Problems Fixed in This Release

The following sections describe NTP problems fixed in this release.

4.9.1 On High-Performance Alpha Systems NTP Fails to Adjust System Clock

Problem:

When running on certain high-performance Alpha systems, NTP may be unable to adjust the system clock; therefore, NTP will not be able to provide accurate timekeeping. When this happens, the following error message appears in the NTP log file:

```
%SYSTEM-F-BADLOGIC, internal logic error detected
VMS timekeeping is not working as expected - can't proceed
```

Solution:

This problem is corrected in this release.

4.9.2 NTP Creates Lowercase File Names on ODS-5 Disks

Problem:

In previous releases of TCP/IP Services, when the NTP server creates files on ODS-5 disks, it gives them lowercase file names. This causes a file-naming inconsistency with non-ODS-5 disks, which assign uppercase names to files.

Solution:

This problem is corrected in this release. All files are created using uppercase file names.

4.10 RCP Problems Fixed in This Release

The following section describes RCP problems fixed in this release.

4.10.1 RCP File Copy Operation Involving Multiple Files or Directories Fails

Problem:

- Attempts to copy files recursively abort prematurely for no apparent reason, or they fail with a read or write error.
- Attempts to copy files might fail with the following error message:

  ```
  %CONV-F-OPENOUT, error opening !AS as output
  ```

  This occurs when using the /RECURSIVE qualifier or wildcards to copy files located in a directory hierarchy that is greater than eight levels deep.

Solution:

These problems are corrected in this release. RCP now supports copy operations involving directory structures greater than eight levels deep. Directory specifications up to 255 levels are now supported.

4.10.2 OpenVMS-to-OpenVMS File Copy Operations Do Not Preserve File Attributes

Problem:

RCP copy operations between OpenVMS systems do not preserve the file attributes (file organization and structure). Files are automatically converted to STREAM_LF format.

Solution:

With this release, RCP allows users to specify the /VMS qualifier to preserve file attributes (UNIX format: use the -v option).

Note

Specify this qualifier only for file copy operations between two OpenVMS systems; otherwise, the operation will fail.

4.10.3 Attempts to Copy Files Larger than 2GBs Fail

Problem:

Attempts to copy files that are greater than 2 gigabytes in size fail.

Solution:

With this release, RCP can copy files larger than 2 GBs. The file size is limited to 4 gigabytes.

4.11 SMTP Problems Fixed in This Release

The following sections describe SMTP problems fixed in this release.

4.11.1 SMTP Receiver Does Not Check Recipient Deliverability

Problem:

The SMTP receiver does not check to see if the recipient email address in the RCPT TO SMTP protocol command is deliverable (for example, that the user account exists on the system). This check is instead deferred to the processing of the mail message in the SMTP queue by the SMTP symbiont process. By this time, the host has taken responsibility for the message and, if there is a problem delivering the message, must bounce the message itself.

This behavior is more problematic when the system receives SPAM. SPAM arrives on the host for a non-existent user and is bounced by the host's symbiont process to the email address in the SPAM's Return-Path: header. The SPAM's Return-Path: header contains an invalid email address, so the bounced SPAM is in turn bounced back to the host's POSTMASTER account. The POSTMASTER account's mail is forwarded to the SYSTEM account, which means that the SYSTEM user must constantly separate these doubly-bounced SPAMs from their valid email.

Solution:

The SMTP receiver has been changed to check to see if the recipient email address in the RCPT TO SMTP protocol command is deliverable. This solves the problem by not letting the SPAM for the unknown user onto the host in the first place.

The `Symbiont-Checks-Deliverability` configuration option allows you to turn this feature on and off. Enter this configuration option in the SMTP configuration file (SMTP.CONFIG).

When this option is set to TRUE, the symbiont checks deliverability of RCPT TO recipients. Setting `Symbiont-Checks-Deliverability` to FALSE (the default) tells the receiver to check the deliverability

4.11.2 SMTP Accepts Mail from Senders Who Should Be Blocked

Problem:

SMTP might accept mail from senders who should be blocked. These are senders listed in the anti-SPAM `Reject-Mail-From` field of the SMTP.CONFIG file. SMTP fails to block such mail when the entries in the `Reject-Mail-From` field exceed the 500-character limit for SMTP.CONFIG fields.

Solution:

This problem is corrected in this release. HP has increased the character limit for fields in the SMTP.CONFIG file from 500 to 10,000.

4.11.3 Two Messages Acquire Same Value in Message-ID Header

Problem:

Any two messages composed in the same one-hundredth of a second will acquire the same value in their Message-ID header. This can cause some mail systems to delete the second of the two messages as a duplicate. Message-IDs should be unique.

Solution:

This problem is corrected in this release. Any two messages created in the same one-hundredth of a second will acquire unique values in their Message-ID headers.

4.11.4 Potential Problems Caused by Multiple Addresses in SMTP To: or Cc: Header

Problem:

Multiple addresses in the To: SMTP mail header that is composed in OpenVMS mail are not separated into multiple lines of text but instead appear on one line. For recipients of such messages on OpenVMS, if the length of this To: line exceeds the OpenVMS mail line length limit of 255 characters, the SMTP symbiont breaks the line into multiple lines when delivering the message, but the lines after the first one are not indented (tabbed in). As a result, the lines will appear as malformed headers. This can cause incorrect behavior with some automated programs that read e-mail. The same problem exists for Cc: lines longer than the OpenVMS mail limit.

Solution:

This problem is corrected in this release. When a user composes a mail message, the SMTP software that builds the SMTP To: and Cc: headers ensures that a To: or Cc: header line does not exceed 75 characters. If adding the next recipient address to a header line would cause the line to exceed 75 characters, the SMTP software inserts a line feed and tab into the headers before adding that recipient address.

4.12 SNMP Problems Fixed in This Release

The following sections describe SNMP problems fixed in this release.

4.12.1 TCPIP$CONFIG.COM Refuses SNMP Community Names Containing Special Characters

Problem:

With Versions 5.1 and 5.3 of TCP/IP Services, TCPIP$CONFIG.COM checks for special characters, and disallows community names containing any special character.

Solution:

This release relaxes these restrictions. However, TCPIP$CONFIG.COM does not accept a space in an SNMP community name. In addition, a quotation mark (") specified as part of a community name might not be handled correctly by TCPIP$CONFIG.COM. A message warns the user to check the validity of the name with the SHOW CONFIGURATION SNMP command, and, if necessary, to correct the name with the SET CONFIGURATION SNMP command.

4.13 Sockets API Problems Fixed in This Release

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

4.13.1 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.

4.14 SSH Problems Fixed in This Release

The following sections describe SSH problems fixed in this release.

4.14.1 SSH Server Does Not Allow Password Change

Problem:

The SSH server does not support password change requests for non-VMS clients when account passwords have expired.

Solution:

If the SSH configuration option AllowNonvmsLoginWith ExpiredPwd is set to "yes" and the password has expired, the server sends a request to the client to prompt the user for a new password. The user must change the password, or the account will be locked out, and the next attempt to log in will fail.

However, if the OpenVMS account has the DisForce_Pwd_Change flag set in the SYSUAF, the server allows the user to log in, displaying the following message:

```
WARNING - Your password has expired; update immediately with SET
PASSWORD!
```

The DisForce_Pwd_Change flag must be applied to each OpenVMS account individually.

The default setting for the AllowNonvmsLoginWith ExpiredPwd option has been changed to "yes." If the AllowNonvmsLoginWithExpiredPwd option is set to "no," the server does not allow password authentication for non-OpenVMS clients when the password has expired. The user does not have the option to change the password. For more information, refer to Section 5.2.

4.14.2 Language Tag Support

Problem:

The password change request that is sent to the SSH client can include a language tag. Some clients do not support the language tag.

Solution:

You can control this feature using the DisableLanguageTag configuration option in the SSH server configuration file (SSHD2.CONFIG). By default, OpenVMS password change requests include the language tag. If the client that does not expect the language tab receives it, the client will issue an error message. You can disable sending the language tag by setting the DisableLanguageTag option to "yes" in the SSH server configuration file. This prevents the language tag from being included in any password change request.

4.14.3 Accepting Two Passwords

Problem:

The OpenVMS SSH server does not support a secondary password for password authentication.

Solution:

The SSH server detects when a user has a second password. In this case, OpenVMS prompts for the second password. If one password has expired, the user is prompted to change the password. If both passwords have expired, the user is prompted to change the first one, and then is prompted to change the second one.

In order for the SSH client to accept the OpenVMS prompt for the second password, one or both of the following configuration options must be set to 2:

- In the client configuration file (SSH2_CONFIG): `NumberOfPasswordPrompts`
- In the server configuration file (SSHD2_CONFIG): `PasswordGuesses`

Both configuration files may be stored in TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]. In addition, the user can have a client configuration file in the user-specific SSH directory ([*username*.SSH2]).

Note

Support for multiple passwords is not specified in any SSH-related RFC.

The second password prompt is enabled by forcing an error situation on OpenVMS for the first password; this is handled by the OpenVMS software internally. However, the message displayed after entering the first password depends on the client software. No intrusion record is created if authentication is enabled. However, if either password is entered incorrectly, an intrusion record is created.

Some clients accept the second password request even if both passwords have expired. However, some clients do not accept the second password request; these clients function correctly when only one of the passwords has expired.

4.14.4 Native-Mode X11 Port Forwarding Does Not Work

Problem:

SSH for OpenVMS does not support the native-mode SSH mechanism for implementing X11 port forwarding (using the `-x` or `+x` SSH command options, or the `ForwardX11` keyword in the client configuration file and the `AllowX11Forwarding` keyword in the server configuration file). SSH only supports standard port forwarding, requiring special setup actions to enable the X11 functionality.

Solution:

This problem is corrected in this release.

4.14.5 SFTP Double Echo and Key-Handling Problems

Problem:

Before using SFTP to connect to a remote system, characters typed at the SFTP prompt (SFTP>) are double echoed. In addition, when connected to the remote system, the left and right arrow keys do not work as expected, as well as the Ctrl/X, Ctrl/W, and Ctrl/C sequences (to erase line, refresh line, and exit, respectively).

Solution:

These problems are corrected in this release. However, pressing Ctrl/C does not display "Cancel" as expected.

4.14.6 SSH, SFTP, and SCP Commands Fail or Do Not Work Properly in Batch Mode

Problem:

The SSH, SCP, and SFTP commands fail or work improperly in batch mode.

Solution:

This problem is corrected in this release.

For restrictions pertaining to batch mode, see Section 3.12.10.

4.14.7 RSA Key Types Not Accepted

Problem:

In prior versions of SSH for OpenVMS, RSA keys are accepted for client authentication to the server, but not accepted for server authentication to the client.

Solution:

Starting with this release of TCP/IP Services, both RSA and DSA key types are accepted for client authentication to the server as well as server authentication to the client.

4.15 SSL Problems Fixed in This Release

The following sections describe SSL problems fixed in this release.

4.15.1 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.

4.16 TELNET Problems Fixed in This Release

The following sections describe TELNET problems fixed in this release.

4.16.1 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).

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

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 Installation and Configuration

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.

B

SSH Kerberos Authentication Methods

In addition to the existing authentication methods (password, hostbased, and publickey), this version of TCP/IP Services supports three new authentication methods based on Kerberos:

- `gssapi-with-mic`
- `kerberos-2@ssh.com`
- `kerberos-tgt-2@ssh.com`

Note

Hereafter, “kerberos-2” is used synonymously with “kerberos-2@ssh.com”. The phrase “kerberos-tgt-2” is used synonymously with “kerberos-tgt-2@ssh.com”. Wherever the full string is required, such as in the value for a configuration file field, it is used.

All three Kerberos based SSH authentication methods are used similarly. The SSH client user must have a Kerberos ticket granting ticket (TGT) from Kerberos before connecting to the SSH server. To get a TGT, you issue a Kerberos `kinit` command. Once you have a TGT, you can use any of the three Kerberos-based authentication methods when connecting to a remote SSH server.

B.1 SSH Kerberos Authentication Interoperability

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 supports 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.

B.2 Forwarding of Credentials

Kerberos provides the means for Kerberized applications like SSH to forward Kerberos 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 ever 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.

To enable this feature the user’s initial TGT must be forwardable. On most Kerberos implementations, this simply means that the user must issue their initial kinit command with the -f command line option. The -f option indicates that a forwardable TGT is to be produced.

In addition to the presence of a forwardable TGT, the Kerberized application being used must support credentials forwarding. The different SSH authentication methods support forwarding credentials as follows:

- kerberos-tgt-2

  This method fully supports credentials being forwarded from the client to the server process obviating the need for subsequent kinit commands before use of another Kerberized application.

- kerberos-2

  This method does not support forwarding of the user’s Kerberos credentials to the process on the SSH server host so use of a Kerberized application from the process on the server side requires the user to enter another kinit command.

- gssapi-with-mic

  Although this method supports forwarding of the user’s Kerberos credentials to the user’s process on the SSH server, 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 will require the user to issue another kinit command in that process.

The following examples of the kerberos-tgt-2 and kerberos-2 authentication methods assume the proper Kerberos user and host principals have been configured, that the associated Kerberos keytab entries have been created and that the SSH servers have been configured to accept requests for the two authentication methods. For information about how to enable SSH server support for the kerberos-tgt-2 and kerberos-2 authentication methods, see Section B.4.

The example is annotated with text marked with three leading exclamation points ("!!!").

```
!!! 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
```

B.3 Password Authentication Kerberos Enhancements

In addition to the explicit uses of Kerberos provided with the kerberos-2, kerberos-tgt-2, and gssapi-with-mic authentication methods, 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 the 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 once logged in. Essentially the SSH server has done a kinit on behalf of the user.

Some system managers may not want the SSH server to perform the Kerberos password check as part of its password authentication method processing. Therefore, its use is configurable. See Section B.4 for more information on how to turn it on and off.

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’s 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.

B.4 Configuring SSH/Kerberos

If you want to use any of the Kerberos based functionality, whether it be as an SSH client or server or both, you must install the 32-bit Kerberos RTL image SYS$SHARE:KRB$RTL32.EXE. Additionally, you must install the 32 bit GSSAPI RTL image SYS$SHARE:GSS$RTL32.EXE to use gssapi-with-mic as an SSH client or server. Neither of these images are installed automatically for you by Kerberos startup. To install the images, enter the following commands:

```
$ INSTALL CREATE SYS$SHARE:KRB$RTL32.EXE/OPEN/HEADER_RESIDENT/SHARED
$ INSTALL CREATE SYS$SHARE:GSS$RTL32.EXE/OPEN/HEADER_RESIDENT/SHARED
```

Even if the only Kerberos authentication method that you have configured is `gssapi-with-mic`, you must still install both images above. If you are using one of the other Kerberos features (`kerberos-2@ssh.com`, `kerberos-tgt-2@ssh.com`, or the Kerberos password check in password authentication) then you need only install SYS$SHARE:KRB$RTL32.EXE.

KRB$STARTUP.COM should not be run until TCPIP$STARTUP.COM has been run.

The SSH server in this version of TCP/IP Services supports Kerberos for OpenVMS Version V2.1 and higher. 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 as the steps needed to mak GSS$RTL32 work properly with LIB$FIND_IMAGE_SYMBOL are not needed.

B.4.1 SSH Server Configuration

The SSH server fields that accept a list of authentication method names are:

- AccountingAuthentications
- AllowedAuthentications
- IntrusionAuthentications
- IntrusionIdentMethod
- IntrusionIdentSSH
- LogfailAuthentications

The new Kerberos authentication method names are not part of the default lists for any of these SSH configuration fields. They must be explicitly turned on in order to be used. Since the AllowedAuthentications field defines the authentication methods that your server will allow, you must add any desired Kerberos authentication method name to AllowedAuthentications if you want your SSH server to offer it to the client. For example, if you want your SSH server to offer `gssapi-with-mic` and `publickey` authentication, include the following:

```
AllowedAuthentications: gssapi-with-mic, publickey
```

B.4.2 SSH Client Configuration

AllowedAuthentications is the only SSH client configuration file field that accepts a list of authentication method names. None of the Kerberos based authentication methods are enabled for AllowedAuthentications by default. You must enable them in your SSH client configuration file, SSH2_CONFIG. For example, to tell your SSH client to try `gssapi-with-mic` first, then publickey, and then password authentication, the SSH2_CONFIG AllowedAuthentications field, enter the following:

```
AllowedAuthentications: gssapi-with-mic, publickey, password
```

B.4.3 Configuring Kerberos KDC/DNS

To configure Kerberos KDC/DNS, include the following:

- Fully qualified host principals

  You must define a Kerberos host principal for each SSH server host to which a user may want to connect using any of the three Kerberos based authentication methods or the password authentication Kerberos password check. 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, as in the example.

  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 do not need to define a Kerberos host principal for an SSH client host unless the host is also to serve as an SSH server for one of the three Kerberos based authentication methods or the password authentication Kerberos password check. Put another way, the host being connected to requires the host principal, not the host being connected from.

- Fully qualified local host entry for SSH server

The gssapi-with-mic authentication method places an additional configuration requirement for SSH servers running on OpenVMS.

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 entry will prevent the gssapi-with-mic authentication method from working. The second will allow it to work.

Entry 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
```

Entry 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
```

Note that this is a requirement only with the gssapi-with-mic authentication method. Also note that there is nothing essentially wrong with Entry 1 above except that such an entry will prevent the gssapi-with-mic authentication method from working correctly on myhost.abcd.org. If your configuration requires a local host database entry such as Entry 1 above, then gssapi-with-mic will not work for you.

The new configuration file fields supported by the SSH server are:

- TryKerberosPassword

  The new Kerberos password check in password authentication mode is not enabled by default. You must enable it in the SSH server configuration file, SSHD2_CONFIG.

  The SSH server configuration file field TryKerberosPassword tells the SSH server in password authentication mode to validate the user’s password against Kerberos before validating against the SYSUAF. TryKerberosPassword is a boolean field. A "yes" value tells the SSH server to validate the user’s password against Kerberos and a "no" value tells the SSH server not to check Kerberos. The TryKerberosPassword configuration field defaults to "no".

  This feature uses Kerberos functionality in SYS$SHARE:KRB$RTL32.EXE therefore requiring installation of this image as documented above. This feature does not use SYS$SHARE:GSS$RTL32.EXE. Therefore it does not require installation of SYS$SHARE:GSS$RTL32.EXE.

- GssapiSendError

The SSH server configuration file field GssapiSendError tells the SSH server in gssapi-with-mic authentication mode to send a "GSSAPI Error" message (message type SSH_MSG_USERAUTH_GSSAPI_ERROR in the protocol specification) to the client should certain errors occur. This protocol message passes text providing more details of the error and is typically displayed to the client user.

GssapiSendError is a Boolean field. A "yes" value tells the SSH server to send a "GSSAPI Error" message and a "no" value tells it not to send the message. This field defaults to "no".

The new configuration file field supported by the SSH client is:

- GssapiDelegateCredentials

  If the GssapiDelegateCredentials flag is set then the SSH client "delegates" the user's credentials to the SSH server. Note that forwardable credentials and delgating credentials are mutually exclusive. If a user has done kinit with the -f flag, then credentials cannot be delegated.

  GssapiDelegateCredentials is a Boolean. A "yes" value tells the SSH client to delegate credentials to the SSH server and a "no" value tells it not to delegate credentials to the SSH server. This field defaults to "no".

The new configuration file field supported by the SSH server and client is:

- GssapiSendErrtok

  The SSH server and client configuration file field GssapiSendErrtok tells the SSH server or client in gssapi-with-mic authentication mode to send a "GSSAPI Errtok" message (message type SSH_MSG_USERAUTH_GSSAPI_ERRTOK in the protocol specification) to the peer should certain errors occur. This message conveys a gssapi "error token" to the peer and may be decoded on the peer and cause some useful diagnostic information to be displayed.

  GssapiSendErrtok is a Boolean field. A "yes" value tells the SSH client or server to send a "GSSAPI Errtok" message and a "no" value tells it not to send the message. This field defaults to "no".

B.4.4 Solving Provlems with SSH and Kerberos

These common errors occur when using SSH with Kerberos support.

- Expired Kerberos ticket

  Kerberos tickets have expiration dates. If you attempt to use the SSH client to connect using a Kerberos authentication method and your ticket has expired you will see this message:

```
SYSA> ssh -o"AllowedAuthentications kerberos-2@ssh.com" smith@sysb
Kerberos5 authentication failed: Your ticket has expired. Use kinit
```

- Non-forwardable Kerberos ticket with kerberos-tgt-2

  When using the kerberos-tgt-2 authentication method your kinit must include the -f option to tell Kerberos that you want a forwardable TGT. If you have not specified kinit with -f and try to use SSH with kerberos-tgt-2 authentication, you will see this message:

```
SYSA> ssh -o"AllowedAuthentications kerberos-tgt-2@ssh.com" smith@sysb
Kerberos5 TGT forwarding failed: Ticket not forwardable? Try using kinit -f
```

- No Kerberos ticket at all

In the event that you have tried to use Kerberos based SSH authentication without having issued a kinit command to get a Kerberos TGT you will see this message:

```
SYSA> ssh -o"AllowedAuthentications kerberos-tgt-2@ssh.com" smith@sysb
Kerberos5 TGT forwarding failed: You have no ticket. Use kinit -f
```

- gssapi-with-mic credentials being delegated with kinit -f

  If you have tried to use the gssapi-with-mic authentication method with a forwardable TGT, authentication will fail. You may see something like:

  ```
      MYHOST> ssh -o"allowedauthentications gssapi-with-mic" -
  _MYHOST> -o"GssapiDelegateCredentials yes" smith@sysb
      GSSAPI error from server: Miscellaneous failure
                                Internal credentials cache error
      warning: Authentication failed.
  ```

  The text of the error message may be different or may not be displayed at all depending on which SSH server and client are in use and whether the server sends the "GSSAPI Error" message.

  For the TCP/IP Services for OpenVMS SSH server this type of failure will cause the following warning to be signalled in the SSH server log file:

  ```
  Mon 12 06:43:13 WARNING: Miscellaneous failure
  Internal credentials cache error
  ```

Some common system manager errors are related to system-wide or Kerberos configuration.

Misconfiguration can cause problems that are identified by the following error messages.

- Your SSH server log file shows the following error when trying to authenticate using gssapi-with-mic:

  ```
  Mon 12 09:37:37 WARNING: Miscellaneous failure
  Wrong principal in request
  ```

  The problem could be an incorrect local host database entry on the SSH server for the SSH server itself. Make sure the fully-qualified host name is used. Note that this message applies only to gssapi-with-mic authentication.

- Your SSH server log file shows the following error when trying to authenticate using <code-example>(gssapi-with-mic):

  ```
  Mon 19 09:19:37 WARNING: Miscellaneous failure
  No principal in keytab matches desired name
  ```

  This problem can be caused by a missing or incorrect host principal for the SSH server, or the host principal is correct but the keytab entry is missing or incorrect. This is how the problem manifests itself with gssapi-with-mic authentication.

- Your SSH client signals something like:

  ```
  $ ssh -o"AllowedAuthentications kerberos-2@ssh.com" smith@sysb
  Kerberos5 authentication failed: krb5_get_credentials(): KRB5KDC_ERR_S_PRINCIPAL_U
  warning: Authentication failed.
  ```

  This problem can be caused by a missing or incorrect host principal for the SSH server, or the host principal is correct but the keytab entry is missing or incorrect. This is how the problem manifests itself with the kerberos-2 or kerberos-tgt-2 authentication.

- The SSH client or server (in the SSH server log) shows a message like one of these:

  ```
  WARNING: kerberos-2@ssh.com authentication failed since Kerberos wasn’t initialize
  ```

  ```
  WARNING: gssapi-with-mic authentication failed since Kerberos wasn’t initialized.
  ```

  This message indicates that one of the Kerberos shareable images that needs to be installed is not. Refer to the installation instructions to remedy the situation.

Kerberos, while powerful, can be cumbersome to configure and deploy. Often, problems that occur with Kerberos are related to misconfiguration of Kerberos, not a software problem with the applications using it.

In the event that failures in the use of SSH with Kerberos authentication occur it is helpful to test some other Kerberized application, like TELNET, which will often fail in the same way as does SSH, pointing to a problem with Kerberos configuration, rather than to an SSH software problem.

The SSH server and client diagnostics features, when turned on, will show errors and warnings indicating problems with calls to Kerberos library routines. These types of errors are displayed listing the Kerberos routine that failed and the failure return status.

When you are debugging problems with SSH Kerberos authentication, you are likely to repeatedly make changes to Kerberos configuration and then try to connect from a test SSH client. When iteratively testing and making changes, especially when creating host principals and manipulating `keytab` entries and files, it’s a good idea to use `kdestroy` and `kinit` in your test SSH client process before retrying the test. Stale information may be contained in existing tickets in the credentials cache, so changes made in Kerberos configuration elsewhere may appear to have no effect, when a `kdestroy` and `kinit` would show that the changes did, in fact, fix the problem.