VMS Software

VSI TCP/IP Services V6.0-29A

Release Notes

Publication Date: October 2025

Operating Systems: VSI OpenVMS Alpha V8.4-2L1
VSI OpenVMS Alpha V8.4-2L2
VSI OpenVMS IA-64 V8.4-2L1
VSI OpenVMS IA-64 V8.4-2L3
VSI OpenVMS x86-64 V9.2-2 or higher

Software Versions: VSI TCP/IP Services for OpenVMS Alpha V6.0-29A
VSI TCP/IP Services for OpenVMS IA-64 V6.0-29A
VSI TCP/IP Services for OpenVMS x86-64 V6.0-29A

VMS Software, Inc. (VSI)
Boston, Massachusetts, USA

VSI TCP/IP Services V6.0-29A Release Notes

Legal Notice

Table of Contents

1. Introduction

VMS Software, Inc. (VSI) is pleased to introduce VSI TCP/IP Services for OpenVMS Alpha V6.0-29A, VSI TCP/IP Services for OpenVMS Integrity V6.0-29A, and VSI TCP/IP Services for OpenVMS x86-64 V6.0-29A (referred to as VSI TCP/IP Services V6.0 later on in this document).

Important

VSI TCP/IP Services for OpenVMS V6.0-29A contains important security updates. VSI strongly recommends that all users install this version.

VSI TCP/IP Services V6.0 is the VSI implementation of the TCP/IP networking protocol suite and internet services for OpenVMS Alpha, OpenVMS IA-64, and OpenVMS x86-64 systems respectively. VSI TCP/IP Services V6.0 provides a comprehensive suite of functions and applications that support industry-standard protocols for heterogeneous network communications and resource sharing.

If you encounter any issues with VSI TCP/IP Services V6.0, please report them to VSI support.

For detailed information on running the TCPIP$CONFIG configuration procedure, refer to the *VSI TCP/IP Services V6.0 for OpenVMS Installation and Configuration* [https://docs.vmssoftware.com/vsi-tcpip-services-v6-for-openvms-installation-and-configuration/] manual.

2. Prerequisites

VSI TCP/IP Services for OpenVMS Alpha V6.0-29A can be installed on an Alpha system running VSI OpenVMS Alpha V8.4-2L1 or VSI OpenVMS Alpha V8.4-2L2.

VSI TCP/IP Services for OpenVMS IA-64 V6.0-29A can be installed on an IA-64 system running VSI OpenVMS IA-64 V8.4-2L1 or VSI OpenVMS IA-64 V8.4-2L3.

VSI TCP/IP Services for OpenVMS x86-64 V6.0-29A can be installed on an x86-64 system running VSI OpenVMS x86-64 V9.2-2 or higher. VSI recommends that you upgrade to the latest version of OpenVMS x86-64.

VSI SSL3 V3.0-7 or later must be installed on the system on which you are planning to install VSI TCP/IP Services for OpenVMS V6.0-29A.

3. Available Services

The following services are available in VSI TCP/IP Services V6.0:

- BIND[1]
- DHCP Client
- FTP
- FTPS

[1]VSI TCP/IP Services V6.0 uses the BIND 9.11.37 Server. Managing the BIND TCP/IP service is documented in the *VSI TCP/IP Services for OpenVMS Management* [https://docs.vmssoftware.com/vsi-tcpip-services-for-openvms-6-management/] manual.

- Finger
- FailSafe IP
- IMAP (not yet available on x86-64, applies to IA-64 and Alpha only)
- LBROKER
- LPR/LPD
- NFS
- NTP4
- POP
- Remote (R) Commands
- SMTP
- SNMP
- Socket API
- TELNET (except Kerberos authentication)
- XDM

Important

VSI TCP/IP Services V6.0 kits *do not* include an SSH component. However, if you need to use SSH in your environment, VSI recommends that you use the latest available version of VSI OpenSSH.

4. Changes to Automatic Configuration of TCP/IP Services

Automatic configuration of TCP/IP services via TCPIP$STARTUP no longer enables TELNET. If you wish to run TELNET on a freshly installed system, you must configure it manually using TCPIP$CONFIG.

5. Known Issues and Limitations

This section lists the known issues and limitations in VSI TCP/IP Services V6.0.

5.1. Clock Synchronization With NTP Server

An issue has been identified in the current NTP implementation where configuring the minpoll parameter to a value of 8 or higher may cause the local clock to gradually lose synchronization with the NTP server. In some cases, this results in an increased time drift or a broadening of the synchronization interval relative to the server's reference clock. This behavior is under investigation, and users are advised to use lower minpoll values to maintain accurate synchronization.

5.2. Error Messages After Upgrading From TCP/IP V5.7 With SSH Enabled

After upgrading from TCP/IP V5.7 to TCP/IP V6.0 without first disabling SSH (as detailed in *Section 7, "Upgrading From TCPI/IP Services V5.7"*), users may encounter the following error messages upon starting the stack:

```
%TCPIP-S-STARTDONE, TCPIP$TELNET startup completed
%TCPIP-E-STARTFAIL, failed to start SSH
-TCPIP-E-NOSERVREC, cannot find SSH service database record
%TCPIP-E-STARTFAIL, failed to start SSH_CLIENT
-TCPIP-E-NOSERVREC, cannot find SSH_CLIENT service database record
%TCPIP-S-STARTDONE, TCP/IP Services startup completed at 5-MAY-2025 21:24:34.75
```

These error messages can be safely ignored. Alternatively, you can get rid of them by entering the following commands:

```
$ TCPIP SET CONFIG ENABLE NOSERVICE SSH
$ TCPIP SET CONFIG ENABLE NOSERVICE SSH_CLIENT
```

5.3. TCPTRACE Utility Not Supported on x86-64

The TCPTRACE utility is not currently supported for x86-64 releases of VSI TCP/IP. VSI suggests using the TCPDUMP utility, as it provides comparable functionality and is supported on all architectures (Alpha, IA-64, and x86-64).

5.4. VSI FTP Service Might Not Connect Correctly in Virtual Environments With NAT Enabled

If the FTP service does *not* work after it has been started, switch to passive mode with the following command:

```
FTP> SET PASSIVE ON
Passive is ON
```

In passive mode, the FTP client always initiates a data connection. This is useful in virtual machine environments when there is network address translation (NAT) in your network.

To run this command automatically when you invoke FTP, put it into SYS$LOGIN:FTPINIT.INI. For the full description of the **SET PASSIVE** command, refer to the relevant section in the *VSI TCP/IP Services for OpenVMS User's Guide* [https://docs.vmssoftware.com/vsi-tcpip-services-for-openvms-user-s-guide-60/#d0e6058].

5.5. Running DHCP Client and failSAFE IP Are Not Compatible on the Same NIC

In VSI TCP/IP Services V6.0, the DHCP client and failSAFE IP cannot be activated for the same NIC, because both DHCP client and failSAFE IP manage the address assignment of a NIC. You must select one or the other.

5.6. NTPDATE No Longer Supported

NTPDATE is no longer supported and will be removed from an upcoming release of VSI TCP/IP Services V6.0. To perform the equivalent of NTPDATE, run `NTPD` as follows:

```
$ ntpd :== $tcpip$ntp
$ ntpd "-G" -q
```

5.7. TCPIP$BIND_CONF.TEMPLATE_FORWARD Requires Adjustment in Environments Not Supporting DNSSEC

The following lines in the TCPIP$BIND_CONF.TEMPLATE_FORWARD template file set up the forwarders' addresses and the DNSSEC validation:

```
//Specifies the IP addresses to be used for forwarding.
//The default is the empty list (no forwarding).
forwarders {
    8.8.8.8;
    8.8.4.4;
};

dnssec-validation auto; //Enable DNSSEC validation.
                        //Note dnssec-enable also needs to be set to
                        //yes to be effective. The default is yes.
```

However, if forwarders are changed to DNS servers that do not support DNSSEC or have it disabled, DNS lookup replies will be discarded when the DNSSEC validation fails.

To avoid this, comment out the following line:

```
dnssec-validation auto
```

5.8. NFS Client Issues

5.8.1. File Attribute Issues in Stress Testing

Occasionally, the attributes of a file just written from a client under stress testing are briefly inconsistent with the server's copy, despite having been properly stored on the server.

5.8.2. SS$_BADIRECTORY Error When Deleting a File

The error SS$_BADIRECTORY is occasionally returned when deleting a file.

This error can be safely ignored.

5.8.3. Recommending an Extended Timeout Value

The default timeout value of 1 second is problematic. It is recommended that a larger timeout value is used (5 seconds).

5.8.4. Stale File Reports in OPCOM With NFS Delete Requests

A stale file is occasionally reported by OPCOM, usually with an NFS client delete request. This problem is usually caused by the client performing the GET ATTRIBUTES operation after the deletion.

5.8.5. QIO Operations May Fail With Timeout Error

The QIO operations on the NFS client may fail with an unexpected TIMEOUT error when the operations on the server side take longer than expected.

One such situation may occur when VSI OpenVMS is running on a guest virtual machine, and the NFS client sends server requests too frequently. The solution in this case would be to increase the timeout interval to a larger value, around 5 seconds.

The error may also occur when a large file is created on the server side via a CREATE operation that specifies the file size, and the target disk has the **High-Water Marking** feature enabled. To remedy this, there are two options:

- If the security of the disk data is *not* critical, the **High-Water Marking** feature can be disabled.
- If the security of the disk data *is* critical, the **Erase on Delete** feature should be used instead of **High-Water Marking**.

6. Resolved Issues

This section describes the issues that were fixed in VSI TCP/IP Services for OpenVMS V6.0-29A. For information about the issues that were fixed in the previous release, refer to *VSI TCP/IP Services for OpenVMS V6.0-28B Release Notes* [https://docs.vmssoftware.com/vsi-tcpip-services-for-openvms-6028b-release-notes/].

6.1. Incorrect Output After Entering a Password

Previously, VSI TCPIP NTPDC failed to insert a newline after password input, causing the subsequent prompt text to be clobbered on the same line. The prompt handling has now been corrected to properly terminate the line after password entry.

6.2. NFS ACP Client Crashes While Double-Queuing Timer Entry

Previously, a VSI TCPIP NFS ACP client process could double-book a timer entry, leading to an internal error and ACP process exit. This, in turn, would cause all NFS mounted disk operations to stall. This issue has been resolved.

6.3. Excessive NTP Server Debug Logging Causing Log File Bloat

NTP server previously generated excessive log data due to debug messages being logged by default, regardless of whether the `-D` or `-d` debug level options were specified. This caused log files to grow significantly over time.

The issue has been resolved. Debug log entries are no longer recorded unless the `-D` or `-d` options are explicitly provided at runtime. By default, NTP server will now suppress all debug-level logging, resulting in smaller log files.

6.4. Performance Issues on Large Systems

Larger systems with multiple NICs that prefer different CPUs which receive NFS requests could previously experience performance issues. The NFS server does not execute NFS server code concurrently. It can concurrently interleave NFS work, but multiple concurrent NFS threads could potentially incur spinlock contention under certain circumstances.

Normally, the NFS server configures itself to streamline thread execution on a preferred CPU, but if INET KVCI CPU scheduling is not enabled (which it the default), then threads schedule on the CPUs that receive the requests. This could lead to threads being scheduled on multiple CPUs concurrently, causing this issue.

To solve this, when the INET KVCI CPU scheduler is disabled, it will now schedule NFS threads on the CPU that currently owns the NFS spinlock. If the spinlock is not owned, it will schedule the thread on the current CPU.

6.5. Memory Management Issues in NFS Client Processes

Previously, the NFS client process could crash due to memory management issues, particularly under low-memory conditions or when an NFS server becomes unavailable. The memory handling logic has been corrected to prevent these failures. This issue has been largely resolved, improving client stability under resource-constrained and server-offline scenarios.

6.6. NFS Client Crashes if Server Goes Offline

Previously, the TCPIP NFS client ACP process could crash or become unresponsive if the server goes offline. This problem has been resolved, and the client now correctly handles server unavailability without process failure.

6.7. NFS Client Causes System Crash

A critical problem in NFS client that was causing the system to crash when encountering a severe fault has been resolved. The updated handling mechanism now ensures the fault is contained within the process, avoiding a system crash.

6.8. An Issue With getaddrinfo

Previously, the TCP/IP service routine `getaddrinfo` crashed when the `Hints` argument was passed as a null pointer. This has been corrected by adding proper validation, ensuring the routine safely handles null input without causing a process crash.

6.9. Issue With Resolving Simple Hostnames

Previously, mounting an NFS share failed if the parameter "host" was not provided as a fully qualified domain name (FQDN). The system now correctly resolves hostnames, allowing NFS mounts to succeed with both short hostnames and FQDNs.

7. Upgrading From TCPI/IP Services V5.7

Before upgrading from TCP/IP Services V5.7 to V6.0, you should make several adjustments to your V5.7 configuration using TCPIP$CONFIG:

- If you are currently using the DHCP server, disable it. This facility is not yet implemented in VSI TCP/IP Services V6.0.
- If you are currently using the DHCP client, disable it. The DHCP client implementation in VSI TCP/IP Services V6.0 differs from that in V5.7. If you plan to enable the DHCP client after upgrading to V6.0, it will utilize the new configuration logic found in TCPIP$CONFIG.

- If you are currently using the SSH client, disable it. The SSH client is now part of the VSI OpenSSH product, and is not included in VSI TCP/IP Services V6.0.
- If you are currently using the SSH server, disable it. The SSH server is now part of the VSI OpenSSH product, and is not included in VSI TCP/IP Services V6.0.

If you had been using the SSH server, you may notice a disabled service definition for SSH in your configuration. If you do not intend to upgrade to the VSI OpenSSH product, you can remove it. Otherwise, consult the release notes for VSI OpenSSH for details on the migration feature included in the product's installation procedure.

A. Security Enhancements for VSI TCP/IP Services V6.0 FTPS

FTPS (FTP over SSL) allows for an encrypted data connection when using FTP. FTPS is run by using either the **FTP /SSL** or **COPY /FTP /SSL** command.

A.1. Changes in Connection Behavior

With TCP/IP Services V5.7 and prior versions, if you use FTPS and the FTP server is not set up to run SSL by not having the proper certificate, the following messages will be displayed, and the connection will continue in plain text:

```
TCPIP$_FTP_SSLERR, SSL not enabled on server
TCPIP$_FTP_SSLERR, Session will continue in plain text
```

See the following example:

```
$ ftp /ssl node1
220 node1.example.com FTP Server (Version 5.7) Ready.
Connected to node1.
500 AUTH command unsuccessful.
TCPIP$_FTP_SSLERR, SSL not enabled on server
TCPIP$_FTP_SSLERR, Session will continue in plain text
Name (node1:username):

$ copy /ftp /ssl /log node2"username password"::file.txt *.*
TCPIP$_FTP_SSLERR, SSL not enabled on server
TCPIP$_FTP_SSLERR, Session will continue in plain text

%TCPIP-S-FTP_COPIED, NODE2.EXAMPLE.COM"username
password"::file.txt copied to DISK:[USERNAME]FILE.TXT;7
(968408 bytes)
```

With VSI TCP/IP Services V6.0, if you use FTPS and the FTP server is not set up to run SSL, the connection will be terminated. See the following examples:

```
$ ftp /ssl node1
220 node1.example.com FTP Server (Version 5.7) Ready.
Connected to node1.
500 AUTH command unsuccessful.
%TCPIP-E-SSLERR, SSL not enabled on server

$ copy /ftp /ssl /log node2"username password"::file.txt *.*
```

```
%TCPIP-E-SSLERR, SSL not enabled on server
```

You must either connect to an SSL-enabled FTP server or reissue the command without the **/SSL** qualifier.

A.2. Changes in Certificate Verification

VSI TCP/IP Services V5.7 and prior versions only check for certificate integrity and do not perform the full server certificate verification. Blindly using a self-signed certificate is not a secure practice.

In the following example, VSI TCP/IP Services V5.7 allows the connection to the FTP server without notifying about the self-signed certificate used by the server:

```
$ ftp /ssl node3
220 node3.example.com FTP Server (Version 5.7) Ready.
Connected to node3.
234 AUTH command successful.
200 PBSZ command successful.
200 PROT command successful.
Name (node3:username):

$ copy /ftp /ssl /log node3"username password"::file.txt *.*
%TCPIP-S-FTP_COPIED, node3"username password"::FILE.TXT;18 copied
to DISK$WORK:[USERNAME]FILE.TXT;19 (1476 bytes)
```

VSI TCP/IP Services V6.0 includes a check for a self-signed or expired server certificate and outputs the appropriate message if such certificates are encountered. You can use a self-signed certificate if you trust the certificate and accept to use it.

The following example shows the connection to the FTP server with a self-signed certificate using VSI TCP/IP Services V6.0:

```
$ ftp /ssl node4
220 node4.example.com FTP Server (Version 6.0) Ready.
Connected to node4.
234 AUTH command successful.
200 PBSZ command successful.
200 PROT command successful.

%TCPIP-F-SSLERR, self signed certificate

       Country: US
         State: MA
      Locality: Boston
  Organization: Certificate Company
          Name: company.com
        E-Mail: first.last@company.com
    Valid from: 30-Apr-2021 22:57
       Expires: 30-Apr-2022 22:57

If you trust the certificate, re-issue the command with the /TRUST
 qualifier.

$ copy /ftp /ssl node3"username password"::file.txt *.*
%TCPIP-F-SSLERR, self signed certificate

       Country: US
         State: MA
```

```
      Locality: Boston
  Organization: Certificate Company
          Name: company.com
        E-Mail: first.last@company.com
    Valid from: 30-Apr-2021 22:57
       Expires: 30-Apr-2022 22:57

If you trust the certificate, re-issue the command with the /TRUST
 qualifier.
```

Add the **/TRUST** qualifier to the command to proceed with the FTPS connection as in the following example:

```
$ ftp /ssl /trust node4
220 node4.example.com FTP Server (Version 6.0) Ready.
Connected to node4.
234 AUTH command successful.
200 PBSZ command successful.
200 PROT command successful.
%TCPIP-I-SSLERR, self signed certificate
%TCPIP-I-SSLERR, TRUST specified; FTP/SSL continuing...
Name (node4:username):

$ copy /ftp /ssl /log /trust node4"username password"::file.txt *.*
%TCPIP-I-SSLERR, self signed certificate
%TCPIP-I-SSLERR, TRUST specified; FTP/SSL continuing...

%TCPIP-S-FTP_COPIED, node4"username password"::FILE.TXT;18 copied to
 DISK:FILE.TXT;22 (1476 bytes)
```