

# OpenSSH Version 8.9-1I01 for VSI OpenVMS

## Release Notes

**Publication Date:** August 2024

**Operating Systems:** VSI OpenVMS x86-64 Version 9.2-1 or higher  
VSI OpenVMS Integrity Version 8.4-2L1 or higher  
VSI OpenVMS Alpha Version 8.4-2L1 or higher

**Kit Names:** VSI-AXPVMS-OPENSSSH-V0809-1I01-1.PCSI  
VSI-I64VMS-OPENSSSH-V0809-1I01-1.PCSI  
VSI-X86VMS-OPENSSSH-V0809-1I01-1.PCSI

## Table of Contents

1. Introduction .....	3
2. Acknowledgements .....	3
3. What's New in This Release .....	3
4. Requirements .....	3
5. Recommended Reading .....	4
6. Installing the Kit .....	4
7. Post-Installation Steps .....	6
8. Migration .....	8
9. Configuration Parameters .....	10
9.1. Server Configuration Parameters .....	10
9.2. Client Configuration Parameters .....	12
10. Logical Names .....	12
11. Rights Identifiers .....	13
12. External Authentication .....	13
13. X11 Port Forwarding .....	13
14. Fixed Issues .....	13
15. Known Problems and Restrictions .....	15

## Warning

The V8.9-1I01 kit supersedes the V8.9-1I kit which has been withdrawn because of an issue that was discovered post-release. If you have installed V8.9-1I, make sure to upgrade to V8.9-1I01.

---

# 1. Introduction

Thank you for your interest in this port of OpenSSH to VSI OpenVMS IA-64, Alpha, and x86-64. The current release of OpenSSH for OpenVMS is based on the OpenSSH 8.9 distribution.

OpenSSH (<https://www.openssh.com/>) is an Open Source (BSD licensed) suite of secure networking utilities based on the Secure Shell (SSH) protocol, which provides a secure channel over a potentially unsecured network. OpenSSH is a complete implementation of the SSH protocol (version 2) for secure remote login, command execution, and file transfer. It includes SSH client and server components, file transfer utilities SCP and SFTP, as well as the tools for key generation, run-time key storage, and a number of other supporting programs.

This port of OpenSSH to VSI OpenVMS IA-64, Alpha, and x86-64 is based on the Portable OpenSSH distribution (see <https://github.com/openssh/openssh-portable>), which is a port of OpenBSD's OpenSSH implementation commonly used on Linux, OS X, and Cygwin.

# 2. Acknowledgements

VMS Software Inc. would like to acknowledge the work of the Portable OpenSSH development team for their ongoing efforts in developing and supporting this software.

# 3. What's New in This Release

For a detailed description of the features and bug fixes included in this release of OpenSSH, please refer to <https://www.openssh.com/txt/release-8.9>.

# 4. Requirements

The kit you are receiving has been compiled and built using the operating system and compiler versions listed below. Note that while you probably will not have any problems installing and using this kit on systems running higher versions of the operating system and/or products listed below, running older versions may cause problems.

- VSI OpenVMS V8.4-2L1 or higher (Integrity, Alpha), V9.2-1 or higher (x86-64)
  - For VSI OpenVMS IA-64 V8.4-2L1, ECO VMS842L1I\_RTL-V0600 or later
  - For VSI OpenVMS IA-64 V8.4-2L3, ECO VMS842L3I\_RTL-V0600 or later
  - For VSI OpenVMS Alpha V8.4-2L1, ECO VMS842L1A\_RTL-V0600 or later
  - For VSI OpenVMS Alpha V8.4-2L2, ECO VMS842L2A\_RTL-V0600 or later

---

## Note

The RTL ECO kits mentioned above require their respective DPML V0200 ECO to be installed first.

---

- VSI TCP/IP

---

## Note

Before you install OpenSSH for VSI OpenVMS, make sure VSI TCP/IP is installed and started.

---

- VSI SSL3 V3.0-13 or later
- If your system has VSI OpenSSH V8.9-1F or earlier installed, it *must* be uninstalled before installing VSI OpenSSH V8.9-1101. To uninstall the previous version of VSI OpenSSH, perform the following procedure:

1. Enter the command `$ PRODUCT REMOVE OPENSSSH`
2. You will get the following error:

```
%PCSI-I-SPAWNEXE, error executing:
@PCSI$DESTINATION:[OPENSSSH.BIN]SSH$RUN_CLEANUP_PROCEDURE.COM
%PCSI-E-EXERMVFAIL, product supplied EXECUTE REMOVE
procedure failed
-RMS-E-FNF, file not found
%PCSI-E-OPFAILED, operation failed
Terminating is strongly recommended. Do you want to
terminate? [YES]
```

Answer **NO** to the Do you want to terminate? question.

3. Once VSI OpenSSH has been removed, you will get the following message:

```
%PCSIUI-I-COMPWERR, operation completed after explicit continuation
from errors
```

## 5. Recommended Reading

Before installing and using OpenSSH, it is recommended that users read the documentation available at <https://www.openssh.com/manual.html> in order to better understand how to configure and use the software.

## 6. Installing the Kit

---

### Note

Do not use the `/DESTINATION` qualifier with the **PRODUCT INSTALL** command when installing OpenSSH for VSI OpenVMS x86-64 to specify an alternative (non-default) installation location. VSI OpenVMS for x86-64 includes OpenSSH components bundled with the operating system, which imposes specific requirements in terms of location of these components and associated configuration files.

---

This kit is provided as an OpenVMS PCSI kit that can be installed by a suitably privileged user running the following command:

```
$ PRODUCT INSTALL OPENSSSH
```

---

The installation will then proceed as follows. Note that the output may differ slightly from that shown below depending on the platform and other factors.

The following product has been selected:

VSI I64VMS OPENSSH V8.9-1I01                      Layered Product

Do you want to continue? [YES]

Configuration phase starting ...

You will be asked to choose options, if any, for each selected product and for any products that may be installed to satisfy software dependency requirements.

Configuring VSI I64VMS OPENSSH V8.9-1I01: VSI OpenVMS OpenSSH

© (c) Copyright 2024 VMS Software, Inc.

OpenVMS OpenSSH is released under a BSD licence, or a licence more free than that.

\* This product does not have any configuration options.

Execution phase starting ...

The following product will be installed to destination:

VSI I64VMS OPENSSH V8.9-1I01                      DISK\$I64V842L1SYS:[VMS\$COMMON.]

The following product will be removed from destination:

VSI I64VMS OPENSSH V8.9-1H01                      DISK\$I64V842L1SYS:[VMS\$COMMON.]

Portion done: 0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%

User Accounts and User Identification Codes (UICs)

-----

The OpenVMS OpenSSH installation creates two OpenVMS accounts: SSH\$SSH, SSH\$SSHD. The default UIC group number for these new accounts depends on the following:

- o If you are installing the server for the first time, the default is the first unused UIC group number, starting with 3655.
- o If any of these accounts already exists, then the default UIC group number will not be used to change the UIC of any existing accounts.
- o If old account TCPIP\$SSH already exists, then the default UIC group number will be used from TCPIP\$SSH account.

For more information about UIC group numbers, see the OpenVMS System Manager's Manual.

Enter default UIC group number for OpenSSH account

Group: [3655]

Creating OpenVMS account required by OpenSSH

SSH\$SSH account already exists

SSH\$SSHD account already exists

Try to deassign SSH\$ROOT in system table exec

SSH\$ROOT is defined as "\$1\$DGA60:[SYS0.SYSCOMMON.OPENSSH.]"

Setting file protections...

File protections are set

Creating OpenSSH for OpenVMS root definition file SYS\$COMMON:[SYS\$STARTUP]SSH

\$DEFINE\_ROOT.COM...

File created

Save startup files  
Setup OpenSSH logical environment

Generating public/private keys:  
ssh\_host\_dsa\_key. and ssh\_host\_dsa\_key.pub are already present.  
ssh\_host\_ecdsa\_key. and ssh\_host\_ecdsa\_key.pub are already present.  
ssh\_host\_rsa\_key. and ssh\_host\_rsa\_key.pub are already present.  
ssh\_host\_ed25519\_key. and ssh\_host\_ed25519\_key.pub are already present.

Purging installed files...  
Cleaning up temporary files...

Successfully finished

In a cluster, on all the nodes that are going to use common  
ssh\$root installation directory as the current node, copy  
the following files to SYS\$STARTUP directory of each node:

```
SYS$STARTUP:SSH$STARTUP.COM  
SYS$STARTUP:SSH$SHUTDOWN.COM  
SYS$STARTUP:SSH$DEFINE_ROOT.COM
```

To automatically start OpenVMS OpenSSH during system startup  
add the following line to the file SYS\$MANAGER:SYSTARTUP\_VMS.COM  
after the TCPIP startup command procedure:

```
$ @SYS$STARTUP:SSH$STARTUP.COM
```

Define symbols for all OpenSSH utilities:

```
$ @SSH$ROOT:[BIN]SSH$DEFINE_COMMANDS.COM
```

To have all symbols defined by the time of login,  
add a caller line of SSH\$ROOT:[BIN]SSH\$DEFINE\_COMMANDS.COM file in either:  
1. in SYS\$MANAGER:SYLOGIN.COM (note that you need system priv's for this)  
2. in SYS\$LOGIN:LOGIN.COM

## 7. Post-Installation Steps

After the installation has successfully completed, follow these steps:

1. Start OpenSSH SSH server by executing the following command:

```
$ @SYS$STARTUP:SSH$STARTUP
```

The name of the TCP/IP service for the OpenSSH SSH server is “SSHD<port>”, where <port> is  
the TCP port number the OpenSSH server is configured to listen on. For example, if OpenSSH is  
configured to listen on TCP port 222, the service name is SSHD222.

2. Include the commands displayed at the end of the installation procedure into the  
SYSTARTUP\_VMS.COM file to ensure that OpenSSH components are correctly started when  
OpenVMS is booted. Details regarding the migration procedure initiated at the end of the installation  
are provided below.
3. To have all necessary symbols defined by the time of login, add the following command to either  
the system-wide login procedure (SYS\$MANAGER:SYLOGIN.COM) or a user's LOGIN.COM  
procedure:

```
$ @SSH$ROOT:[BIN]SSH$DEFINE_COMMANDS.COM
```

## Note

If you plan to allow users with SSH connected sessions to shutdown the system, do not invoke the SSH\$SHUTDOWN.COM from within the site-specific SYSS\$MANAGER:SYSHUTDWN.COM procedure. The SSH shutdown will delete the process executing the shutdown and leave the system in an inconsistent state with logins and SSH disabled, along with some other parts of the system shutdown, but nothing left running to complete the shutdown or reboot.

---

## Note

The command procedure SSH\$ROOT:[BIN]SSH\$DEASSIGN\_COMMANDS.COM can be used to undefine these command symbols.

---

If **SSH\$DEFINE\_COMMANDS.COM** is run with the parameter "ALL" the following additional commands will be defined. These commands are intended primarily for administrative purposes:

- **SSHSTART**

Starts and creates (if necessary) the OpenSSH services. Before running this command, check the file SSH\$ROOT:[ETC]SSHD\_CONFIG to ensure that the SSH server configuration details are correct. If you would like to modify the client configuration, edit the SSH\$ROOT:[ETC]SSH\_CONFIG file *before* starting the services.

- **SSHSTO\*P**

Stops OpenSSH services. If the parameter "ALL" is specified, the service definitions will also be deleted from the TCP/IP configuration.

- **SSHSH\*OW**

Show details of running OpenSSH processes including SSH connections, number of connected clients, etc. Note that each client connection consists of two processes, namely a process named SSHD\_BGxxxxxx (where xxxxxx is the number of the associated BG device) and a user process with a name that either matches the username or begins with the string FTAxxx\_ followed by the username (for example, FTA110\_SMITH). The name of the user process may of course be changed by the user.

- **SSHVER\*SION**

Displays the information about the various OpenSSH programs, including version details and other related data.

---

## Note

When the TCP/IP service is restarted, the OpenSSH server may become unresponsive and stop accepting new connections. To restore the OpenSSH server after a TCP/IP restart, manually restart OpenSSH by executing the following command:

```
$ @SYS$STARTUP:SSH$STARTUP
```

---

## 8. Migration

---

### Note

This section is applicable to IA-64 and Alpha only. No actions will be performed if the migration tool is run on OpenVMS x86-64, and the tool will exit with a message indicating that no existing old TCP/IP Services SSH configuration was found. Similarly, the migration tool does not need to be run on Alpha or Integrity if you have run it previously and are upgrading to a new version of OpenSSH for VSI OpenVMS.

---

As noted above, this release of OpenSSH for VSI OpenVMS includes a migration script (SSH\$ROOT:[BIN]SSH\$MIGRATION.COM) that can be used to convert configuration files and user public/private keys from the format used by VSI TCP/IP Services to the format expected by OpenSSH. After installing OpenSSH for VSI OpenVMS, this tool can be run to establish an initial OpenSSH system configuration that is comparable to that provided by the existing VSI TCP/IP Services.

The migration tool is run at the end of the OpenSSH kit installation. The user is prompted as to whether they wish to perform the migration at this time, the default being NO. In general, it is recommended that the migration be performed manually as a post-installation activity. Note that if OpenSSH is already installed or the VSI TCP/IP SSH service does not exist, the migration tool will *not* run.

Specific features of the migration facility are summarized as follows:

- The migration tool does not modify your old VSI TCP/IP Services files, making it possible to revert if necessary (migration is non-destructive).
- The tool creates a log file in SSH\$ROOT:[VAR] containing the details of all migration activities. The name of the log file is SSH\$MIGRATION\_XXXX.LOG, where XXXX is replaced by the date and time at which the migration was performed.
- The following VSI TCP/IP Services configuration files will be examined and converted. As noted previously, the existing TCP/IP Services configuration files will not be changed.
  - TCPIP\$SSH\_DEVICE:[TCPIP\$SSH.SSH2]SSHD2\_CONFIG
  - TCPIP\$SSH\_DEVICE:[TCPIP\$SSH.SSH2]SSH2\_CONFIG
- Users' public/private keys (RSA, DSA) can be converted optionally. The migration tool assumes that for a given username these files will reside in [.SSH2] under the user's login directory, and the converted keys will be written to [.SSH]. The first key defined in a user's [.SSH2] identification file is renamed to ID\_RSA as the ID\_RSA is one of the filenames used for public keys by default, per an IdentityFile entry in SSH\_CONFIG. If there is an existing [.SSH]AUTHORIZED\_KEYS. file, no conversion will be performed.
- When users run the migration script, they can choose a port that will be used for the OpenSSH server. For the client, the default port is 22. If the specified port is used during the migration, the script will prompt the user for another port.

The following brief notes illustrate how to run the migration tool to perform migration tasks or to revert to your old VSI TCP/IP Services configuration. Note that when running the migration tool, it is recommended that you not be logged into the OpenVMS system via VSI TCP/IP Services SSH.

- The migration tool can be run as follows to create an OpenSSH configuration from existing VSI TCP/IP Services configuration files:



```
$ @SSH$ROOT:[BIN]SSH$MIGRATION.COM
```

- Running the following command will revert the system to using the old VSI TCP/IP Services configuration (assuming the old configuration has not otherwise been removed). This command will delete the OpenSSH SSH service and revert to the VSI TCP/IP Services SSH service.

```
$ @SSH$ROOT:[BIN]SSH$MIGRATION.COM REVERT
```

- Public/private keys can be converted to OpenSSH format for a specified username using the following command.

```
$ @SSH$ROOT:[BIN]SSH$MIGRATION.COM "" <USERNAME>
```

- Conversion of a single key file can be performed as follows:

```
$ @SSH$ROOT:[BIN]SSH$DEFINE_COMMANDS.COM
```

```
$ PIPE SSH_KEYGEN "-I" "-F" [.SSH2]FILENAME > [.SSH]FILENAME
```

```
$ SET FILE/OWNER=<USER UIC> /PROTECTION=(G:"",W:"") [.SSH]FILENAME
```

The following table summarizes the parameter conversions that are performed by the migration tool for the SSHD2\_CONFIG and SSH2\_CONFIG configuration files (for additional details regarding the configuration parameters, see the next section).

VSI TCP/IP Services	OpenSSH
AccountingAuthentications	VmsAccountingAuthentications
IntrusionAuthentications	VmsIntrusionAuthentications
IntrusionIdentMethod	VmsIntrusionIdentMethods
IntrusionIdentSsh	VmsIntrusionIdentSsh
LogFailAuthentications	VmsLogFailAuthentications
UserLoginLimit	VmsUserLoginLimit
AllowVmsLoginWithExpiredPw no AllowNonVmsLoginWithExpiredPw no	VmsAllowLoginWithExpiredPw no
NumberOfPasswordVerificationPrompts	VmsNumberOfPasswordVerificationPrompts
PrintSysAnnounce	VmsPrintSysAnnounce
PrintSysWelcome	VmsPrintSysWelcome
DisallowSftpServer	VmsDisallowSftpServer
SftpDenyUsers	VmsSftpDenyUsers
MaxConnections	MaxSessions
KeepAlive	TcpKeepAlive
BannerMessageFile	Banner
VerboseMode yes	LogLevel VERBOSE
UserKnownHosts no	IgnoreUserKnownHosts yes
AllowedAuthentications publickey,hostbased,password	HostBasedAuthentication yes PubkeyAuthentication yes PasswordAuthentication yes

VSI TCP/IP Services	OpenSSH
Ciphers AnyStdCipher	Ciphers none
MACs AnyStdMAC	MACs none

## 9. Configuration Parameters

This section describes the unique to OpenVMS configuration parameters to be used in OpenSSH client and server configuration files.

### 9.1. Server Configuration Parameters

The following parameters may be defined in SSH\$ROOT:[ETC]SSHD\_CONFIG to control various aspects of SSH server operation with regard to maximum sessions, authentication, audit logging, and intrusions.

#### VmsUserLoginLimit

This parameter can be used to specify the maximum number of SSH clients that can be logged into the OpenVMS system. The default value is -1 (not limited); the maximum permitted value is 8192.

#### VmsNumberOfPasswordVerificationPrompts

This parameter can be used to specify the maximum number of password change attempts (the number of times that the user will be prompted to verify their new password). The default value is 3.

#### VmsAllowLoginWithExpiredPw

Setting this parameter to `yes` (the default) allows users to change their password if the password has expired and the user is connecting from an OpenVMS system. Permitted values for this parameter are `yes` and `no`.

#### VmsPrintSysAnnounce

Setting this parameter to `yes` (the default) causes the OpenVMS welcome banner associated with the logical name SYSS\$ANNOUNCE to be displayed when logging in. Permitted values for this parameter are `yes` and `no`.

#### VmsPrintSysWelcome

Setting this parameter to `yes` (the default) causes the welcome banner associated with the logical name SYSS\$WELCOME to be displayed when logging in. The permitted values for this parameter are `yes` and `no`.

#### VmsAccountingAuthentications

Generates an accounting record for all authentications via the specified authentication method(s) (`publickey`, `password`, and `hostbased`). The default value for this parameter is `publickey,password,hostbased`.

#### VmsIntrusionAuthentications

Reports users as intruders if they attempt and fail to connect using any one of the specified authentication method or methods. The default value for this parameter is `publickey,password,hostbased`, such that all authentication failures will be reported as intrusions.

### **VmsIntrusionAddServerAddress**

Adds address details to the audit message. For example,  
SSH\_<authentication method>:<client ip-address>:<server ip-address>.  
The default value is no.

### **VmsIntrusionIdentMethods**

Specifying this parameter with a value comprising one or more authentication methods causes intrusion records pertaining to those authentication methods to specify the authentication method in addition to the IP address. Specifically, intrusion records will contain strings of the form SSH\_<authentication method>:<ip-address>. The default value for this parameter is `publickey,password,hostbased`.

### **VmsIntrusionIdentSsh**

If this parameter is specified, only the IP address will be reported in intrusion records; the authentication method will not be included in the record. The default value for this parameter is `publickey,password,hostbased`. If the same values are specified for both `VmsIntrusionIdentMethods` and `VmsIntrusionIdentSsh`, then `VmsIntrusionIdentMethods` takes precedence.

### **VmsLogFailAuthentications**

This parameter can be used to control the reporting of login failures. Default value for this parameter is `publickey,password,hostbased`.

### **VmsDisallowSftpServer**

This parameter can be used to control access to the SFTP server for all users. The default value of this parameter is no. Setting the value to `yes` will deny access to the SFTP server for all users.

### **VmsSftpDenyUsers**

This parameter can be used to specify a list of users to be denied access to the SFTP server. The list of users must be specified as a list of username patterns separated by spaces. By default, no users will be denied access to the SFTP server.

### **VmsSftpDenyGroups**

This parameter can be used to specify a list of user groups to be denied access to the sftp server. The list of groups must be specified as a list of group name patterns separated by spaces. By default, no OpenVMS user groups will be denied access to the SFTP server.

### **XAuthLocation**

This parameter is used to specify the location of the XAuth tool. The default value is `SYS$SYSTEM:DECW$XAUTH.EXE`.

### **X11Forwarding**

The `X11Forwarding` parameter controls whether the SSH Server permits X11 forwarding for connected clients. The value of this parameter can be specified as either `YES` or `NO`:

- Specifying `YES` enables X11 forwarding. This allows SSH Clients to request X11 forwarding on connection.

- Specifying `NO` disables X11 forwarding. This is the default.

## 9.2. Client Configuration Parameters

Client configuration parameters are defined in `SSH$ROOT:[ETC]SSH_CONFIG` to control various aspects of OpenSSH client utilities (SSH, SFTP, and SCP). Additionally, the client configuration parameters may be specified when executing a client utility command using the `-o` command line option.

VSI has created a new client configuration parameter which may be used to disable the SFTP client extension known as the “VMSPlus mode”.

---

### Note

At this time, this new parameter is only effective when specified on the SFTP command line, as shown below. Adding this parameter to the `SSH_CONFIG` file will have no effect.

---

VMS Plus mode is enabled by default in the SFTP client to facilitate transfer of files with various OpenVMS file formats, including OpenVMS BACKUP savesets, between two OpenVMS hosts running OpenSSH v8.9-1G or newer.

Should VMS Plus mode cause issues when communicating with other SSH server implementations, it may be disabled by setting the `NOVMSPLUS` parameter to any positive integer value. For example:

```
$ SFTP "-o NOVMSPLUS 1" USER@HOST
```

## 10. Logical Names

The following logical names may be defined (at any level) to control the exit status of SFTP and other OpenSSH utilities, and to control the behaviour of the SFTP client when errors are encountered during file transfer operations.

### TCPIP\$SSH\_SFTP\_ALWAYS\_EXIT\_NORMAL

If this logical name is defined to `TRUE` or `1`, the OpenSSH SFTP client will exit with an OpenVMS status of `SS$_NORMAL` in all cases. It should be noted that this logical name is applicable to the `sftp` client only (it is not applicable to SCP or any other OpenSSH utilities).

### TCPIP\$SSH\_SFTP\_BATCH\_ABORT\_ON\_ERROR

This logical name can be used to prevent the SFTP client aborting during batch operations involving the transfer of multiple files. If this logical name is defined to `FALSE` or `0`, the SFTP client will continue batch file transfer operations if an error occurs. Details of any errors will be logged.

### OPENS\$SSH\_POSIX\_EXIT\_STATUS

Defining this logical name to `TRUE` or `1` will cause OpenSSH utilities to exit with a POSIX exit status (as would be the case on Linux).

### OPENS\$SSH\_SFTP\_UNIX\_STYLE

The logical name `OPENS$SSH_SFTP_UNIX_STYLE` can be defined to control how VMS file versions are managed in the SFTP tool on the server. When this logical name is set (regardless of its value), the OpenSSH SFTP tool will display and transfer files along with their versions.

---

## Note

By default, the `OPENSSH$SFTP_UNIX_STYLE` is not defined, which means the VMS file versions will be transferred.

---

## 11. Rights Identifiers

The rights identifier `TCPIP$SSH_FILECOPY_DISALLOWED` can be used to prevent users from connecting to the SFTP server.

## 12. External Authentication

---

### Note

This feature is applicable to IA-64 and Alpha only. It will be supported for OpenVMS x86-64 in the future.

---

We are pleased to introduce external authentication support in VSI OpenSSH. All SSH logins now utilize the OpenVMS `SY$ACM(W)` system service for password authentication.

## 13. X11 Port Forwarding

This release of OpenSSH for VSI OpenVMS introduces support for X11 Port Forwarding on the OpenSSH server side.

X11 Port Forwarding via OpenSSH enables users to connect to an SSH server on the VSI OpenVMS host and run X11 client programs, which will appear on their local display.

X11 Port Forwarding must be enabled on both the SSH client and server. On the client side, this is typically done with the `-X` or `-Y` option when initiating an SSH connection.

The server must also have X11 Port Forwarding enabled in its SSH configuration file, `ssh$root:[etc]sshd_config`, by setting the option **X11Forwarding** to **Yes**. For further details see Section 9.1, “Server Configuration Parameters”.

---

### Note

Ensure that DECwindows Motif Version 1.8 is installed for proper X11 Port Forwarding on x86-64 VSI OpenVMS.

---

### Note

This version of OpenSSH for VSI OpenVMS does not support X11 Port Forwarding on the client side, which means you cannot use this feature to run graphical applications from a remote server on your local machine.

---

## 14. Fixed Issues

The following issues have been fixed in this release:

---

- An issue causing the CXX compiler to fail in locating included files.
- An issue causing SFTP **get** command failure from Non-VMS systems for non-Stream-LF binary files.
- An issue causing SCP failure to transfer files.
- An issue causing channel shutdown failure.
- An issue causing security alarms not to be generated for incorrect passwords.
- An issue causing failure error code reporting as a result of successful login.
- An issue causing SYSTEM-F-ACCVIO as a result of SSH to localhost.
- An issue causing SYSTEM-F-ACCVIO when SYS\$ANNOUNCE or SYS\$WELCOME is not defined.
- An issue causing the OpenSSH server to display an incorrect file name after a successful **put** command using the Windows PSFTP client.
- An issue causing the sftp\_opendir() failure with a logical name without leading \ when using the Windows libssh client.
- An issue causing public key authentication failure when the target user is defined as a logical name in SYSUAF.
- An issue causing the OpenSSH server to fail to work with wildcard using the Windows PSFTP client.
- An issue in the OpenSSH installation process that leads to the deletion of user files unless the user selects the option to retain them.
- An issue causing failure in directory listing when using CURL as a client system.
- An issue causing failure when uploading and downloading files with file names containing logical names using a libssh-based client.
- An issue causing a readdir() failure to list files with extended file names when using a libssh-based client.
- An issue causing the **cd** command failure with file name using looted logical names when using the Windows PSFTP client.
- An issue causing **cd**, **ls**, **put**, and **get** command failures with file names containing logical names when using the Windows PSFTP client.
- An issue causing ACCVIO errors while uploading a file with spaces in the name using the Windows PSFTP client.
- An issue causing directory listing failure to show VMS file versions.
- An issue causing **cd** commands to fail with VMS style file names when using the Windows PSFTP client.
- An issue causing readdir() to show only one file whereas the directory contains more than one file when using a libssh2-based client.

- An issue causing `readdir()` to get a NULL value for file attributes when using a libssh-based client.
- An issue causing the `ls` command failure with explicit file names on the server when using the TCP/IP V5.7 ECO 5 kit.
- An issue causing the `put` command failure with special symbols in the file name when using the Windows PSFTP client.
- An issue causing `sftp_write()` failure with a buffer of around 262138 bytes when using a libssh-based client.
- An issue causing the `rename` command failure on the server when using the TCP/IP V5.7 ECO 5 kit.
- An issue causing failure when downloading fixed-length files from the VMS system to Windows.
- An issue causing record format files to be uploaded without a carriage return from the VMS system to a non-VMS system.

## 15. Known Problems and Restrictions

- Authentication using a public key may fail with the following SSHD server message:

```
Authentication refused: realpath /<disk>/<user_directory>/ssh/  
authorized_keys failed: file name too long
```

To resolve this issue, edit the `SSH$ROOT:[ETC]SSHD_CONFIG` file: uncomment the following line and set the value of the `StrictModes` parameter to `no`.

```
StrictModes no
```

- Restarting TCP/IP Services breaks the OpenSSH Server (OpenSSH Client is not affected). To fix that, you need to restart OpenSSH Server manually.
- Only the password, public-key, and host-based authentication methods are currently supported. Additional methods (such as Kerberos) may be added in the future.
- The use of secondary passwords (for password authentication) is not currently supported, but will be provided in a future release.