vms Software

OpenSSH for VSI OpenVMS Alpha, I64, and x86-64

January 2023

```
VSI-AXPVMS-OPENSSH-V0809-1D-1.PCSI
VSI-I64VMS-OPENSSH-V0809-1D-1.PCSI
VSI-X86VMS-OPENSSH-V0809-1D-1.PCSI
```

1. Introduction

Thank you for your interest in this port of OpenSSH to VSI OpenVMS I64, 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, file transfer utilities `scp` and `sftp`, as well as tools for key generation, run-time key storage, and a number of other supporting programs.

This port of OpenSSH to VSI OpenVMS x86-64, I64, and Alpha 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 read https://www.openssh.com/txt/release-8.9.

VSI OpenSSH can be installed *only* on VSI versions of OpenVMS. Integrity and Alpha HPE customers now cannot install the OpenSSH PCSI kits.

VSI OpenSSH V8.9-1D now does not support key types ecdsa-sk, ed25519-sk.

VSI OpenSSH V8.9-1D now supports the IPv6 protocol.

The '-V' option has been added to enable the use of pathnames in VMS format.

The shutdown text has been removed from the PCSI text file.

4. Requirements

The kit you are receiving has been compiled and built using the operating system and compiler versions listed below. While it is highly likely that you will have no problems installing and using the kit on systems running higher versions of the operating system or

products listed, we cannot say for sure that you will be so lucky if your system is running older versions.

- VSI OpenVMS Version 9.2 x86-64 or higher; VSI OpenVMS 8.4-2L1 or higher (I64 and Alpha)
- VSI TCP/IP

  Note: VSI TCP/IP *must be installed and started* before installing OpenSSH for VSI OpenVMS.
- VSI SSL3 V3.0-7 or later
- If your system has a previous version of VSI OpenSSH installed, it *must* be uninstalled before installing VSI OpenSSH V8.9-1D. To uninstall the previous version of VSI OpenSSH, perform the following procedure:
  - Enter the command `$ PRODUCT REMOVE OPENSSH`
  - You will get the following error:

    ```
    %PCSI-I-SPAWNEXE, error executing:
    @PCSI$DESTINATION:[OPENSSH.BIN]SSH$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.
  - 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 review 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) location where the software should be installed. VSI OpenVMS for x86-64 includes the OpenSSH components bundled with the operating system, which imposes specific requirements in terms of the location of these components and associated configuration files.

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

```
$ PRODUCT INSTALL OPENSSH
```

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

```
Performing product kit validation of signed kits ...
%PCSI-I-VSIVALPASSED, validation of DSA20:[OPENSSH]VSI-AXPVMS-OPENSSH-V0809-
1D-1.PCSI$COMPRESSED;1 succeeded
```

```
The following product has been selected:
    VSI AXPVMS OPENSSH V8.9-1D             Layered Product

Do you want to continue? [YES] y

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 AXPVMS OPENSSH V8.9-1D: VSI OpenVMS OpenSSH

    ▒ Copyright 2022 VMS Software, Inc.

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

    This installation procedure requires that all the following
    conditions are satisfied:

    1.  This procedure is running on an Alpha processor.
    2.  The system is running OpenVMS V8.4-2L1 or later.
    3.  All required privileges are currently enabled.
    4.  No OpenSSH images are running on this node or anywhere in
        the cluster that make use of common ssh$root installation
        directory.
    5.  Supports migrating SSH Secure Shell OpenVMS (V5.5) 3.2.0

Do you want to continue? [YES] y

* This product does not have any configuration options.

Execution phase starting ...

The following product will be installed to destination:
    VSI AXPVMS OPENSSH V8.9-1D             DISK$SYS_SKOGUL:[SYS0.SYSCOMMON.]

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

   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
SSH$ROOT is defined as "SYS$SYSDEVICE:[SYS0.SYSCOMMON.OPENSSH.]"
```

```
Setting file protections...
  File protections are set

The OpenSSH configuration files were saved in the directory:
  SYS$COMMON:[SYSUPD.SSH$SAFETY]

Should it restore OpenSSH configuration files? [y/n]: [y] y
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.


Do you want to migrate your old SSH settings? [y/n]: [n] y

The OpenSSH migration procedure cannot be executed from an interactive SSH
session.
The current SSH server will be shutdown which will terminate all existing
SSH sessions.

Please login using DECnet, LAT, TELNET, or the Console to run
SSH$MIGRATION.COM.

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

...100%

The following product has been installed:
    VSI AXPVMS OPENSSH V8.9-1D             Layered Product
```

7. Post-installation steps

After the installation has successfully completed, include the commands displayed at the end of the installation procedure into `SYSTARTUP_VMS.COM` and `SYSHUTDWN.COM` to ensure that OpenSSH components are correctly started and stopped when OpenVMS is booted and shutdown. Details regarding the migration procedure initiated at the end of the installation are provided below.

In order to use any of the OpenSSH utilities it is necessary to run the command procedure SSH$ROOT:[BIN]SSH$DEFINE_COMMANDS.COM, which establishes the commands listed below. Users may run this command procedure from their LOGIN.COM, or running the command procedure can be added to SYLOGIN.COM to define the command for all users.

- SCP
- SFTP
- SSH_ADD
- SSH_AGENT
- SSH_KEYGEN
- SSH_KEYSCAN
- SSH

The command procedure SSH$ROOT:[BIN]SSH$DEASSIGN_COMMANDS.COM can be used if desired to un-define these commands.

Note that 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 and will not generally be used by other users.

- SSHSTART

  Starts and creates (if necessary) OpenSSH services. Before running this command it is important to check the file SSH$ROOT:[ETC]SSHD_CONFIG to ensure that SSH server configuration details are correct. You may also wish to modify the client configuration file SSH$ROOT:[ETC]SSH_CONFIG before starting the services.

- SSHSTO*P

  Stops OpenSSH services. If the parameter "ALL" is specified then 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, and so on. Note that each client connection consists of two processes, namely a process with a name of the form SSHD_BGxxxxx (where xxxxx is the number of the associated BG device), and a user process with a name of the form FTAxxx_USERNAME for the USERNAME in question. The name of the user process may of course be changed by the user.

- SSHVER*SION

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

8. Migration

Note: This section is applicable to I64 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 can be 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 beta 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 migration be performed manually as a post-installation activity. Note that if OpenSSH is already installed or the 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 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$root:[var]` containing details of all migration activities. The name of the log file is of the form `ssh$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 optionally be converted. The migration tool assumes that for a given username these files will reside in `[.ssh2]` under the users login directory, and converted keys will be written to `[.ssh]`. If there is an existing `[.ssh]authorized_keys.` file, no conversion will be performed.

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,f 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. TCP/IP Services configuration files (see following section for additional details regarding configuration parameters).

VSI TCP/IP Services	**OpenSSH**
AccountingAuthentications	VmsAccountingAuthentications
IntrusionAuthentications	VmsIntrusionAuthentications
IntrusionIdentMethod	VmsIntrusionIdentMethods
IntrusionIdentSsh	VmsIntrusionIdentSsh
LogFailAuthentications	VmsLogFailAuthentications
UserLoginLimit	VmsUserLoginLimit
AllowVmsLoginWithExpiredPw no and 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
Ciphers AnyStdCipher	Ciphers none
MACs AnyStdMAC	MACs none

9. 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 of this parameter is 3.

- `VmsAllowVmsLoginWithExpiredPw`

  Setting this parameter to "`yes`" (the default) allows users to change their password if the password has expired and the user is connecting from n 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 `SYS$ANNOUNCE` to be displayed when logging in. The 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 `SYS$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 methods, where the specified authentication methods may be one or more of "`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 then 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.

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.

- OPENSSH$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).

11. Rights identifiers

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

12. Fixed Issues

- The issue that was causing the Change Password dialog to work incorrectly for user accounts with the "Restricted" flag has been fixed.
- The issue that was causing the sshd daemon to crash with the access violation error during external connection (from Windows) has been fixed.
- User passwords no longer get locked when they expire.
- Previously, when a user would try to redirect output from SSH remote command execution, an error would occur. Now, redirecting output works correctly in the Detached, Batch, and Interactive processes.
- The issue that was causing the Change Password dialog to work incorrectly for user accounts that did not allow NETWORK access has been fixed.
- Multiple issues with VSI OpenSSH client hanging have been fixed.
- Several errors in SSH$SSHD_STARTUP.LOG have been fixed.

13. Known problems and restrictions

- Only password, public-key, and host-based authentication methods are currently supported. Additional methods (such as Kerberos) may be added in the future. Additionally, for password authentication, use of secondary passwords is not currently supported. Use of secondary passwords will be provided in a future release.
- On VSI OpenVMS x86-64 only, when using `sftp` or `scp` with the default options you occasionally may see an `ASTFLT` being signalled followed by a stack dump. The `ASTFLT` only happens when the "progress meter" is shown, which is the default. To avoid this problem, it is recommended to use the “quiet” switch, "`-q`", whereupon the "progress meter" is not shown. This problem will be addressed in a future version of VSI OpenVMS x86-64.
- In some cases, OpenSSH terminates the connection after sending the first command. After a restart, OpenSSH works as intended.