Secure Web Server for OpenVMS Release Notes

December 2003

Version 2.0, based on Apache 2.0.47
CPQ-AXPVMS-CSWS-V0200--1.PCSI-DCX-AXPEXE

Contents

HP is pleased to provide you with a new HP-supported, customer release version of *HP Secure Web Server for OpenVMS Alpha (based on Apache)*. The Secure Web Server includes Secure Sockets Layer (SSL) through mod_ssl and OpenSSL.

What's New

The Secure Web Server Version 2.0 is based on Apache 2.0.47 and contains several new features, including IPv6 support, enhanced UNIX compatibility, easier configuration and management of your server environment, new Apache modules not previously included in the Secure Web Server, and new logical names.

You can also download Version 1.3, based on Apache 1.3.26, which is an earlier HP-supported, customer release version of the Secure Web Server for OpenVMS Alpha.

Downloading the Kit

The Secure Web Server for OpenVMS Alpha kit is available for the Alpha platform as a compressed self-extracting file.

Please fill out and submit the Secure Web Server for OpenVMS Alpha registration form to download the kit.

See the *HP Secure Web Server for OpenVMS Alpha Installation and Configuration Guide* for information about expanding and installing the kit after you have downloaded it.

Secure Web Server Documentation

See the Documentation Page for links to the *Installation and Configuration Guide* and the *SSL User Guide* for Version 2.0.

Documentation for Version 1.3 and the Version 1.3-compatible optional kits (CSWS_PERL, CSWS_JAVA, and CSWS_PHP) is also available from the Secure Web Server Documentation Page.

Apache Server Documentation

Refer to the Apache HTTP Server documentation for information about the Apache server after you have completed the installation.

You can also view the online Apache server documentation on your web site at

```
http://your.domain/manual
```

Note: To view some of the Apache server documentation on your web site, you must enable MultiViews under `<Directory "/apache$common/htdocs">`

New Features in Version 2.0

- Based on Apache 2.0.47 from the Apache Software Foundation

  Previous versions of the Compaq Secure Web Server were based on the Apache 1.3 series, including 1.3.26, 1.3.20, 1.3.14, and 1.3.12.

  Although the Apache 2.0 stream contains support for running the server in a hybrid multiprocess, multithreaded mode, the Version 2.0 kit is built on a **process-based model**. A threads-based version of the Secure Web Server is under investigation and may be included in a subsequent release.

- IPv6 support

  Version 2.0 of the Secure Web Server supports IPv6 networking, if your OpenVMS system supports and is configured for IPv6. Otherwise, Version 2.0 supports IPv4 networking.

  IPv6-supported browsers include Mozilla for OpenVMS and the HP Secure Web Browser for OpenVMS (based on Mozilla).

- Enhanced UNIX compatibility

  Version 2.0 of the Secure Web Server supports file negotiation using ODS-5 naming. You can also specify any UNIX directive on the command line.

- New server configuration features

  Server configuration has been redesigned in Version 2.0, and includes a configuration menu that gives you options for configuring and managing your server environment.

  See the Configure the Secure Web Server section in the *HP Secure Web Server for OpenVMS Alpha Installation and Configuration Guide* for more information.

- New Apache modules

  Version 2.0 of the Secure Web Server includes Apache modules that were not supported in previous versions of the Secure Web Server. These modules are listed in Apache Modules section in the *HP Secure Web Server for OpenVMS Alpha Installation and Configuration Guide*.

- New and obsolete logical names

  Version 2.0 of the Secure Web Server contains new system, process, and user-defined logical names. Some Version 1.3 logical names are now obsolete. See the Logical Names section in the *HP Secure Web Server for OpenVMS Alpha Installation and Configuration Guide* for more information.

Changed Features in Version 2.0

- Flush | New Obsolete

  The following command is no longer supported in Version 2.0 of the Secure Web Server:

  ```
  $ @APACHE$CONFIG FLUSH | NEW
  ```

  Use the following DCL command instead:

  ```
  $ httpd -k flush | new
  ```

  'flush' forces a data flush to the log files.
  'new' creates new versions of the log files.

- Replacement for APACHE$FIXBG() Routine

  In Version 2.0 of the Secure Web Server, the `apache$fixbg()` routine has been replaced by the new `apache$$setsocketopt()` routine.

  The `apache$$setsockopt()` entry point resides in `APACHE$APR_SHRP.EXE`.

  The function prototype is as follows:

  ```
  int apache$$setsockopt (
        short int   SockChan,
        int         OptName,
        void        *OptVal,
        int         OptLen)
  ```

  `OptName` is one of the following values:

  1 Set device CCL bit (specified by OptVal: -1 [toggle], 0 [clear], 1 [set])
  2 Set device buffer size (specified by OptVal: 1-65535)
  3 Set device shareable (specified by OptVal: 0 [clear] or 1 [set])

OptVal is a pointer to the buffer containing the value to be set. Can be a byte, word, or longword. The size is determined by OptLen.

OptLen is the size in bytes of the buffer specified by OptVal.

The return status is a VMS status reflecting the result of the requested operation.

This routine requires that the caller possess CMKRNL privilege or have either the APACHE$APR_ALL or APACHE$APR_SETSOCKOPT rights identifier.

- Replacement for APACHE$FIXBG.EXE Image

  In Version 2.0 of the Secure Web Server, the APACHE$FIXBG.EXE image has been replaced by the new executable utility [APACHE]APACHE$SET_CCL.EXE, defined as follows:

```
$ SET_CCL := $APACHE$COMMON:[000000]APACHE$SET_CCL.EXE
```

  Use one of the following commands to execute APACHE$SET_CCL.EXE:

```
$ SET_CCL -S 0 <device_name>      ! to force CCL bit = 0
$ SET_CCL -S 1 <device_name>      ! to force CCL bit = 1
$ SET_CCL -S -1 <device_name>     ! to flip the CCL bit
```

  The <device_name> is optional. The default is SYS$OUTPUT.

  There are two new symbols for APACHE$SET_CCL.EXE provided in APACHE$SYMBOLS.COM:

```
APACHE$FLIP_CCL
APACHE$SET_CCL
```

  The usage for this image can be displayed by entering:

```
APACHE$SET_CCL -?
```

Known Problems and Restrictions in Version 2.0

- Unsupported features

  The following features are not yet supported in the Secure Web Server Version 2.0:

  suEXEC
  MOD_DAV

  Enabling these features may cause unpredictable results.

  Support for SUEXEC and MOD_DAV will be added in a future release.

- Language variant filename restriction

Beginning with the Secure Web Server Version 2.0, you must specify language variants on OpenVMS systems in the same way as you do on UNIX systems, using multiple dots in the filename. For example, the French variant of a filename is *filename.html.fr*.

In previous versions of the Secure Web Server, you would use an underscore instead of a dot before the language extension (for example, *filename.html_fr*).

- Command procedure converts files to Stream_LF

  The Secure Web Server Version 2.0 kit requires that all served files must be in Stream_LF format. The Version 2.0 kit includes the APACHE$CONVERT_STREAMLF command procedure that recursively searches down a directory tree for sequential files and converts them to Stream_LF. The command procedure excludes some sequential files; in particular, it ignores directory files, executable files (such as command procedures, OpenVMS images, CGI, PHP, and Perl scripts), object files, indexed files, and relative files.

  The APACHE$CONVERT_STREAMLF command procedure converts all sequential files (with the exceptions listed above) to Stream_LF format, including sequential files currently in Stream format. After you run the procedure, be sure to check the SYS$SCRATCH:CONVERT_DIR.LOG file for files that should not be in Stream_LF format, and delete the newest version of those files.

  See the Converting Files to Stream_LF section in the *HP Secure Web Server for OpenVMS Alpha Installation and Configuration Guide* for more information.

- Preserving existing DAV environment

  If you have an existing DAV environment from a previous version of the Secure Web Server, and you want to preserve that environment in your installation of the Secure Web Server Version 2.0 kit, set the logical APACHE$DAV_DM_TYPE to VDBM. The default is SDBM.

  See the MOD_DAV (Distributed Authoring and Versioning) Support section in the *HP Secure Web Server for OpenVMS Alpha Installation and Configuration Guide* for more information.

- OpenSSL verify command with a qualifier of "-CApath" fails

  This problem occurs if you are running HP SSL Version 1.1-A with OpenVMS Version 7.2-2, 7.3, 7.3-1, and 7.3-2.

  A certificate fails to authenticate because the OpenSSL verify command cannot parse a CApath value as a directory specification. The command does not fail if the CApath value is a logical name that points to a directory, or the directory is specified as a UNIX path.

  An example of a certificate authentication failure is as follows:

  ```
  $ openssl verify -verbose "-CApath" ssl$root:[mycerts] -
  purpose any xxx.crt
  ```

```
xxx.crt: /C=US/ST=New Hampshire/L=Nashua/O=Hewlett-Packard
Company
/OU=OpenVMS Engineering/CN=OpenVMS Bootcamp Client
/Email=webmaster@mynode.zko.dec.com
error 20 at 0 depth lookup:unable to get local issuer
certificate
538972898:error:0B086079:x509 certificate
routines:X509_STORE_CTX_purpose_inherit:unknown purpose
id:X509_VFY:814:
$
```

There are two workarounds for this problem. The first workaround is to specify the -CApath directory by defining a logical name. For example:

```
$ DEFINE/SYSTEM ssl$certs SSL$ROOT:[MYCERTS]
$ openssl verify -verbose "-CApath" ssl$certs -purpose any
xxx.crt
xxx.crt: OK
```

The second workaround is to use a UNIX path specification. For example:

```
$ openssl verify -verbose "-CApath" /ssl$root/mycerts -
purpose any xxx.crt
xxx.crt: OK
$
```

-- End of file --