HP Secure Web Server for OpenVMS (based on Apache)
Version 2.1-1 Release Notes

September 2006

Version 2.1-1 for OpenVMS Alpha, based on Apache 2.0.52
CPQ-AXPVMS-CSWS-V0201-1-1.PCSI_SFX_AXPEXE

Version 2.1-1 for OpenVMS I64, based on Apache 2.0.52
HP-I64VMS-CSWS-V0201-1-1.PCSI_SFX_I64EXE

Contents

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

Downloading the Kit

The Secure Web Server for OpenVMS kit is available for the Alpha and I64 platforms as a compressed self-extracting file.

You can also download Version 1.3-1, based on Apache 1.3.26, which is an earlier HP-supported, customer release version of the Secure Web Server for OpenVMS Alpha and I64.

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

See the *HP Secure Web Server for OpenVMS Installation and Configuration Guide* for information about expanding and installing the kit.

Secure Web Server Documentation

See the Documentation Page for links to the *Installation and Configuration Guide* and the *SSL User Guide* for Version 2.1-1.

Documentation for Version 1.3-1 and the Version 1.3-1-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.

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">`

Bug Fixes in Version 2.1-1

- Fixed problem using a device name vs. a logical name in a directory specification, for example, /dka0/htdocs vs. /user1/htdocs.
- Fixed ErrorDocument directives to allow '/' in name.
- Fixed problem when 'Status:' header was returned by CGI.
- Fixed problem of case sensitive comparison in configuration file directive.
- Fixed problem of adding a cluster node when suEXEC was enabled.
- Fixed problem of 'Listen' directive on multiple hosts with same port number.
- Fixed problem with partial (byte-range) requests.
- Fixed 'require group' directive to accept rights identified with DYNAMIC attribute.
- Fixed problem reading APACHE$INPUT in CGI when data exceeds 64KB.

New Features in Version 2.1

- Based on Apache 2.0.52 from the Apache Software Foundation

  See the Overview of New Features in Apache 2.0 and the Upgrading to Apache 2.0 from Apache 1.3 from the Apache.org website.

  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.

- Includes suEXEC support and WebDAV support

  SWS Version 2.1 includes support for suEXEC and WebDAV. (These features were not included in V2.0 but were included in previous versions of the Secure Web Server.)

  suEXEC allows you to run CGI and SSI programs under user IDs different from the user ID of the calling web server. WebDAV (Web-based Distributed Authoring and Versioning) allows you to create, move, copy, and delete resources and collections on a remote web server.

  To enable WebDAV support, load the `mod_dav.exe` and `mod_dav_fs.exe` modules by uncommenting these lines in `httpd.conf`, as follows:

```
LoadModule dav_module modules/mod_dav.exe
LoadModule dav_fs_module modules/mod_dav_fs.exe
```

See Known Problems and Restrictions in Version 2.1 for information about a WebDAV database manager type restriction.

- Includes the modules mod_auth_kerberos.exe and mod_auth_ldap.exe (unsupported)

  The modules `mod_auth_kerberos` and `mod_auth_ldap` have not been completely tested are **not supported** in Version 2.1.

  `mod_auth_kerberos` provides Kerberos authentication to the Apache web server. `mod_auth_ldap` allows an LDAP directory to be used to store the database for HTTP authentication.

- Stream-LF restriction removed

  The Secure Web Server Version 2.0 required that all served files must be in Stream_LF format. This restriction has been removed in SWS Version 2.1. **Note:** The EnableMMAP directive must be set to OFF to lift the Stream_LF restriction. In Version 2.1, EnableMMAP is set to OFF by default. (In Version 2.0, the default for EnableMMAP was ON.)

New Features in Version 2.0

- Based on Apache 2.0.47 from the Apache Software Foundation

  Previous versions of the 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.

- IPv6 support

  Version 2.*x* of the Secure Web Server supports IPv4 and IPv6 networking.

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

- Enhanced UNIX compatibility

  Version 2.*x* 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 was redesigned in Version 2.*x*, 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 Installation and Configuration Guide* for more information.

- New Apache modules

  Version 2.*x* 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 Installation and Configuration Guide*.

- New and obsolete logical names

  Version 2.*x* of the Secure Web Server contains new system, process, and user-defined logical names. Some V1.3 logical names are now obsolete. See the Logical Names section in the *HP Secure Web Server for OpenVMS Installation and Configuration Guide* for more information.

Changed Features in Version 2.1

- Changes required in httpd.conf when upgrading from V1.3-1 to V2.1

  In SWS Version 2.1, many loadable modules are no longer loaded by default. You must uncomment the modules in `httpd.conf` to load them. (See the file `httpd-vms.conf` for other modules you may want to load.)

  For example, to load these modules, uncomment the following lines in `httpd.conf`:

```
LoadModule osuscript_module modules/mod_osuscript.exe
LoadModule dav_module modules/mod_dav.exe
LoadModule dav_fs_module modules/mod_dav_fs.exe
```

  Note: In Version 2.1, the file `mod_ssl.conf` is named **`ssl.conf`**.

- New directives

  Following are the **new** directives in Version 2.*x*:

```
AcceptMutex vmsdlm
VMSServerTag SWS
VMSServerStartup "/apache$root/000000/apache$startup.com"
VMSServerShutdown "/apache$root/000000/apache$shutdown.com"
EnableMMAP on/off
EnableSendFile on/off
```

  Note: In Version 2.1, EnableMMAP is set to OFF by default, In V2.0, the default was ON.

- Obsolete directives

  Following are the **obsolete** directives in Version 2.*x*:

```
ServerType
Port
```

- Changed server process naming scheme

  In Version 2.*x*, SWS uses a new server process naming scheme where *xxx* is defined by the `VMSServerTag` directive in `httpd.conf`. For example:

```
APACHE$xxx
APACHE$xxx0000
APACHE$xxx0001
```

The old naming scheme (in SWS V1.3-1 and earlier) defined *xx* by `SERVER_TAG` in `SYS$MANAGER:APACHE$CONFIG.DAT` via `APACHE$CONFIG.COM`. For example:

```
APACHE$xx
APACHE$xx000
APACHE$xx001
```

- Changed site-specific startup and shutdown procedure definitions

  In Version 2.*x*, new startup and shutdown procedure definitions are defined by the `VMSServerStartup` and `VMSServerShutdown` directives in `httpd.conf`.

  The old definitions (in SWS V1.3-1 and earlier) were defined by the `SERVER_STARTUP` and `SERVER_SHUTDOWN` directives in `SYS$MANAGER:APACHE$CONFIG.DAT` via `APACHE$CONFIG.COM`.

Changed Features in Version 2.0

- Flush | New Obsolete

  The following command is no longer supported in Version 2.*x* 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.*x* 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, and it 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 an OpenVMS 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.*x* 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 2.1 and higher

- Do not use Secure Web Server V2.1 and higher with older SWS optional kits

  Do not attempt to use SWS Version 2.1 with the following optional kits. Using these kits together causes a process crash. (These kits, in addition to the newer optional kits, are currently available for download from http://h71000.www7.hp.com/openvms/products/ips/apache/csws.html).

  PERL for OpenVMS 5.6.1 and 5.6.1-1A1
  CSWS_PERL V2.0, 1.1, and 1.1-1
  CSWS_PHP V1.2-1 and 1.1
  CSWS_JAVA V2.1

SWS Version 2.1 and higher **works properly** with the following optional kits:

PERL for OpenVMS 5.8.6
CSWS_PERL V2.1
CSWS_PHP V1.3
CSWS_JAVA V3.0

- Installing SWS V2.1 and higher on ODS-2 volume corrupts previous CSWS V1.3 installation

  You must install the Version 2.1 kit on an **ODS-5** target volume. If you attempt to install this kit on an ODS-2 volume, the installation will fail as follows:

```
Destination disk is not ODS-5, aborting installation ...
%PCSI-I-PRCOUTPUT, output from subprocess follows ...
%SYSTEM-F-ABORT, abort
```

  If you had an existing CSWS V1.3 installation, the failed operation will leave it in a corrupt state.

- PDF files are corrupted when downloaded with Microsoft Internet Explorer

  When you download a PDF file from the Secure Web Server V2.1 and higher using Microsoft Internet Explorer, the PDF files are corrupted.

  This problem occurs because SWS does not process range headers (partial content requests) correctly in certain cases. Internet Explorer uses range headers to fetch pieces of PDF files which results in corrupt PDF content.

  To work around this problem, configure Apache to indicate to the client that range headers are not supported. Edit your `httpd.conf` file and add the following directive to any directory that contains PDF files:

```
Header unset Accept-Ranges
```

  Note: `mod_headers` must be loaded to use this directive. If `mod_headers` is not enabled, you can enable it by including the following directive in `httpd.conf`:

```
LoadModule headers_module     modules/mod_headers.exe
```

  This problem will be corrected in a future release.

- Language variant filename restriction

  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*).

- WebDAV database manager type restriction

  WebDAV support requires the VDBM database manager type. VDBM is the default.

To change the database manager type, set the logical name APACHE$DAV_DBM_TYPE so that it is visible to Apache, such as in APACHE$COMMON:[000000]LOGIN.COM. Note that SDBM and GDBM are not supported in this release.

- SSLSessionCache DBM file error

  Using either SDBM or GDBM database manager types causes the following failure to create the SSL session cache:

  ```
  [Thu Apr 14 16:25:12 2005] [error] (2)no such file or
  directory: Cannot create SSLSessionCache DBM file
  `/apache$root/000000/logs/ssl_scache'
  ```

  You must use the VDBM database manager type (the default). To change the database manager type, set the logical name APACHE$SSL_DBM_TYPE so that it is visible to Apache, such as in APACHE$COMMON:[000000]LOGIN.COM. Note that SDBM and GDBM are not supported in this release.

- Cannot add node to SWS in a cluster environment if suEXEC is enabled first

  Choosing Option 10 in the SWS configuration menu (Add a node to CSWS in a cluster environment) fails when adding a node if you enabled suEXEC during the initial configuration of SWS or by using Option 4 (Manage suEXEC users).

  As a temporary workaround, use Option 4 to disable suEXEC, use Option 10 to add the node, then use Option 4 re-enable suEXEC.

- Problems with APACHE$MENU.COM "Create an Apache instance" Option 2

  Option 2 in APACHE$MENU.COM, called Create an Apache instance, fails under the following circumstances:

  1. Specifying a nonexistant target directory fails with the following error where directory [.FOO] does not exist.

  ```
  Root Location: dev:[APACHE.SPECIFIC.FOO]
  %SYSTEM-W-NOSUCHFILE, no such file \_DKA0:[APACHE.SPECIFIC]FOO.DIR\
  %DCL-W-UNDSYM, undefined symbol - check validity and spelling \INDID\
  %DCL-W-UNDSYM, undefined symbol - check validity and spelling \INDID\
  ```

  2. Creating an instance under a name other than APACHE$WWW fails with the following error:

  ```
  [Tue Apr 19 11:20:23 2005] [error] (13)permission denied: Unable
  to create input file dev:[directory.[000000]APACHE$xyz.COM
  ```

- Cached files not served

  The proxy cache feature does not serve the cached files in SWS Version 2.1 and higher.

- AuthUserFile/AuthGroupFile Override AuthOpenVMSUser/AuthOpenVMSGroup

  If the `mod_auth` directives `AuthUserFile` and `AuthGroupFile` are combined with the `mod_auth_openvms` directives `AuthOpenVMSUser` and `AuthOpenVMSGroup`, the `mod_auth` directives override `mod_auth_openvms`.

  In practice, these sets of directives are not commonly used together.

- Require user directive must specify uppercase username with mod_auth_openvms

  The `require user` directive must specify usernames in uppercase when used with `mod_auth_openvms`.

- ApacheBench tool failure (I64 only)

  On OpenVMS I64, the ApacheBench tool fails as follows:

```
Test aborted after 10 failures
apr_connect(): socket is already connected
(56) %SYSTEM-W-BADESCAPE, syntax
error in escape sequence
```