

END-TO-END SSL SETUP SAP WEB DISPATCHER

Helps you to setup the End-To-End SSL Scenario for SAP Web Dispatcher

by **Prakash Palani**
(Prakash.Palani@basisondemand.com)

Table of Contents

1. Introduction	3
2. High Level Steps	3
3. Installation of Web Dispatcher	4
4. Personal Security Environment.....	9
5. Troubleshooting.....	11
6. References	11

1. Introduction

One of our customers wanted to protect their SRM application which needs to be accessed by internet users, as part of their corporate policy, they were not allowed to directly expose the SAP Web Application Server to internet, and rather they wanted to protect the application by having intermediate application behind firewall/secured subnet. We have proposed Web Dispatcher implementation with multiple options such as End-To-End SSL, SSL Termination, etc., Our customer validated the solution approach and decided to go for End-To-End SSL as that was the corporate security requirement, this article discusses about the steps and approach followed to implement the SAP Web Dispatcher for the earlier mentioned application.

Below diagram describes the simple form of the setup, detailed information on the setup is discussed in the following sections.



2. High Level Steps

@ Web Dispatcher (in this case, web.basisondemand.com)

1. Install Web Dispatcher on web.basisondemand.com server
2. Configure the profile parameters
3. Generate Server and Client PSE
4. Download the Trusted / Self-signed certificate from was.basisondemand.com and import it into web.basisondemand.com

@ Web AS (in this case, was.basisondemand.com)

1. Adjust the Message Server Parameters

3. Installation of Web Dispatcher

Installation of SAP Web Dispatcher is fairly a simple step; in this case, I have not attempted to describe about installing it as a service in windows, and instead described the steps to setup a Web Dispatcher with the combination of few files setup at OS level.

1. Download the Web Dispatcher software from <http://service.sap.com/swdc> -> Browse Our Download Catalog -> Technology Components - > SAP Web Dispatcher - > SAP Web Dispatcher 7.20 (downward compatible)
2. Create a directory g:\usr\sap\WSS\SYS\exe\
3. Uncar the downloaded SAR file into the installation directory
4. Once the SAR file is successfully extracted, execute the command sapwebdisp – bootstrap
5. Bootstrap will create a profile called sapwebdisp.pfl which will be used to setup the parameters needed to setup the End-To-End SSL Scenario

3.1. Parameter Changes @ Web Dispatcher Instance

Once the sapwebdisp.pfl file is generated as a result of sapwebdisp –bootstrap command, then adapt the parameters to setup the End-To-End SSL scenario. There are various parameters to be adapted (other than the parameters listed below), below instructions will help you to understand and setup the bare minimum parameters required for the End-To-End SSL Scenario.

3.1.1. SAPSYSTEM -> This parameter is used to maintain unique instance number for the Web Dispatcher instance, something similar to what we give for an SAP ABAP/Java instance.

Example : SAPSYSTEM = 23

3.1.2. `wdisp/shm_attach_mode` -> This parameter indicates the behavior of the Web Dispatcher when attaching to shared memory, possible values are given below.

<mode>	Meaning
1	The shared memory is cleaned up and the SAP Web Dispatcher terminates. The behavior is the same as with the option <code>-cleanup</code> .
2	The SAP Web Dispatcher connects to the existing shared memory (<code>attach</code>). If this does not exist, the SAP Web Dispatcher ends with an error.
3	Not useful
4	The SAP Web Dispatcher creates a new shared memory. If this exists already, the SAP Web Dispatcher ends with an error.
5	If a shared memory exists already, it is deleted. A new shared memory is then created.
6	The Web Dispatcher attempts to attach itself to an existing shared memory. If a shared memory does not exist, a new one is created. This is also the default value, and the SAP Web Dispatcher behaves like this if options <code>-shm_attach_mode <mode></code> and <code>-cleanup</code> are not used, and parameter <code>wdisp/shm_attach_mode</code> is not explicitly set to another value.
7	As 5

Example : `wdisp/shm_attach_mode = 6` (according to SAP’s general requirement)

3.1.3. `rdisp/mshost` -> This parameter is used to maintain the message server hostname which will be called by the Web Dispatcher / which will receive the requests forwarded by the Web Dispatcher

Example : `rdisp/mshost = 10.1.2.3` (IP Address is the preferred one in End-To-End SSL Scenario)

3.1.4. `ms/https_port` -> This indicates the port in which the message server is listening to, the other alternate parameter is `ms/http_port`, for end-to-end SSL, it is mandatory to configure `ms/https_port` parameter.

Example : `ms/https_port = 2443` (this will be described in the section Parameter Changes @ Web AS End)

3.1.5. DIR_INSTANCE - Indicates the home directory which will be used to store the file such as logfile, slog, etc.,

Example : DIR_INSTANCE = G:\usr\sap\WSS

3.1.6. ssl/ssl_lib -> Indicates the path and filename of the cryptography library, this library can be downloaded from <http://service.sap.com/swdc> -> Browse Our Download Catalog - > SAP Cryptographic Software (Download the file and extract it to the directory mentioned in this parameter)

Example : ssl/ssl_lib = g:\usr\sap\WSS\SYS\exe\sapcrypto.dll

3.1.7. ssl/server_pse -> This parameter is used to define the path and filename of the server PSE (Personal Security Environment) file. The same is described in detail under section Personal Security Environment (PSE)

Example : ssl/server_pse = g:\usr\sap\WSS\secudir\sec\SAPSSL.pse

3.1.8. ssl/client_pse - > This indicates the path and filename of the Client PSE, this is also explained in detail under section “Personal Security Environment”

Example : ssl/client_pse = g:\usr\sap\WSS\secudir\sec\SAPSSLC.pse

3.1.9. wdisp/auto_refresh - The period of time after which the route information tables of the SAP Web Dispatcher (server tables, group tables and URL mapping tables) are periodically updated.

Example : wdisp/auto_refresh = 120 (Default Value)

3.1.10. wdisp/max_servers - > This parameter determines the maximum number of entries in the SAP Web Dispatcher’s server table.

Example : wdisp/max_servers = 100

3.1.11. icm/server_port_0 -> One of the most important parameter for Web Dispatcher configuration, this is the parameter used to define the protocol and the listening port of the Web Dispatcher.

Example : icm/server_port_0 = PROT=ROUTER,PORT=60000 (PROT=ROUTER is only used when we have end-to-end SSL scenario, for other scenarios, we either use HTTP/HTTPS)

3.1.12. `icm/server_port_1` -> This is a twin parameter for `icm/server_port_0` when we use the `PROT=ROUTER`, this parameter is used to establish HTTPS communication between Web Dispatcher and Web AS (to exchange the metadata)

Example : `icm/server_port_1 = PROT=HTTPS,PORT=0`

3.1.13. `wdisp/server_info_protocol` -> Indicates the protocol used to exchange the data between the Web Dispatcher and web AS

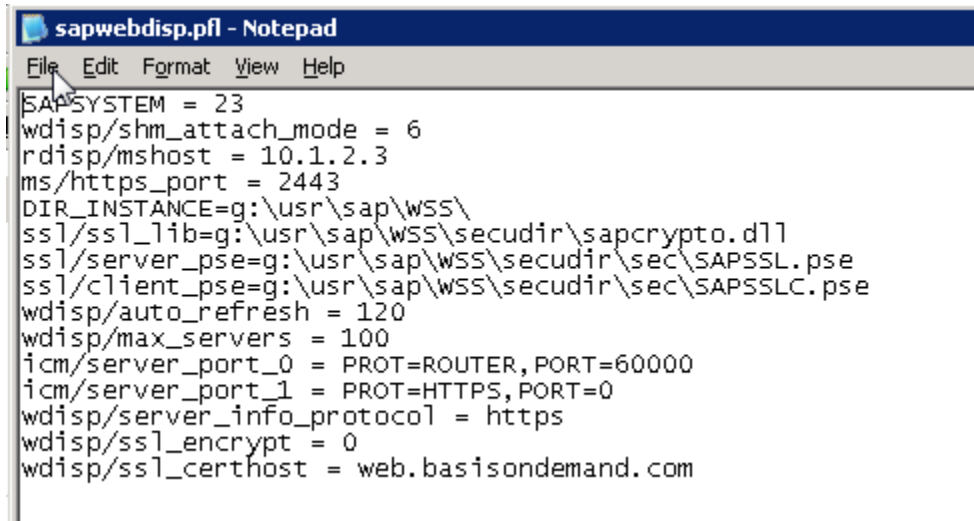
Example : `wdisp/server_info_protocol = HTTPS`

3.1.14. `wdisp/ssl_certhost` -> This is one another important parameter particularly in End-To-End SSL Scenario, this is the parameter used to identify the hostname that is given in the trusted certificate of Web Application Server. If this parameter is not set, then the Web Dispatcher will carry the value mentioned in the `rdisp/mshost` parameter to identify itself against the certificate maintained in the Web AS.

Example : `wdisp/ssl_certhost = web.basisondemand.com` (the certificate which is installed in STRUST (of Web AS) must contain the `CN=web.basisondemand.com`, else the Web Dispatcher will get crashed when there is a mismatch between the values)

Tip : If you have noticed the value given under the `wdisp/ssl_certhost`, it has been marked as Web Dispatcher hostname, the reason for the same is that, when a user calls the Web Dispatcher from browser (i.e. <https://web.basisondemand.com>), it should produce a certificate with the `CN=web.basisondemand.com`, else the user will get a warning saying “not a trusted site”. But in a normal scenario, we generate/get the certificate with hostname (`was.basisondemand.com`) of the web application server as the common name/CN, this should be avoided in case of End-To-End SSL, instead of generating it with Web AS hostname, we should generate the certificate with Web Dispatcher hostname. This way, the certificate with `CN=web.basisondemand.com` will be produced during the runtime (when a user calls from browser), the same will help to avoid the certificate warning in the browser. Another better way of handling it is that you can rely on DNS Alias, which can be used to seamlessly in the URL and in the certificate as well.

3.1.15. Final View of the sapwebdisp.pfl – Below is how the profile will look like once all the above mentioned parameters and values are set.



```
sapwebdisp.pfl - Notepad
File Edit Format View Help
SAPSYSTEM = 23
wdisp/shm_attach_mode = 6
rdisp/mshost = 10.1.2.3
ms/https_port = 2443
DIR_INSTANCE=g:\usr\sap\wss\
ssl/ssl_lib=g:\usr\sap\wss\secudir\sapcrypto.dll
ssl/server_pse=g:\usr\sap\wss\secudir\sec\SAPSSL.pse
ssl/client_pse=g:\usr\sap\wss\secudir\sec\SAPSSLC.pse
wdisp/auto_refresh = 120
wdisp/max_servers = 100
icm/server_port_0 = PROT=ROUTER,PORT=60000
icm/server_port_1 = PROT=HTTPS,PORT=0
wdisp/server_info_protocol = https
wdisp/ssl_encrypt = 0
wdisp/ssl_certhost = web.basisondemand.com
```

3.2. Parameter Changes @ Web Application Server (was.basisondemand.com)

3.2.1. ms/server_port_1 -> This is the parameter used to setup the HTTP/HTTPS protocol and the other relevant parameters for the message server.

Example : ms/server_port_1 = PROT=HTTPS,PORT=2443,TIMEOUT=0,PROCTIMEOUT=0

3.2.2. I don't have much information about the below mentioned parameters, but according to SAP, they are mandatory parameters while setting up HTTPS connection on message server.

```
ms/urlmap_secure = 1
ms/urlprefix_secure = 1
```

3.2.3. ssl/ssl_lib -> As mentioned in the Web Dispatcher parameter changes, this parameter indicates the path and the filename of the cryptography file installed on the web application server.

Example : Ssl/ssl_lib = g:\usr\sap\WAS\SYS\exe\uc\NTAMD64\sapcrypto.dll

3.2.4. In addition to above mentioned parameters, you must activate the below mentioned ICF services as the message server will be used for load balancing.

- sap/public/icf_info/logon_groups
- sap/public/icf_info/icr_groups
- sap/public/icf_info/icr_urlprefix

4. Personal Security Environment

PSE is the environment which is used to store the security information) of a particular instance, it contains :

- Private Key
- Servers Public Key Certificate
- Certificates of trusted CAs (certificate list)

There are various PSEs available as indicated below:

- SNC PSE : Used by the SAP Web AS or ITS for SNC setup
- System PSE : Used by the SAP Web AS for digital signatures
- SSL Server PSE : Used by SAP Web AS / Web Dispatcher for SSL when it is acting as the server which receives the secured connection (HTTPS)
- SSL Client PSE : Used by SAP Web AS / Web Dispatcher for SSL when it is acting as the client which sends the secured connection

In case of Web Dispatcher (End-To-End SSL scenario), we will be using SSL Server PSE and SSL Client PSE, below are the high level steps involved in setting up these PSEs.

1. Set SECUDIR environment variable
2. Generate SSL Server PSE
3. Generate Credentials File for SSL Server PSE
4. Generate SSL Client PSE
5. Update Credentials File for SSL Client PSE
6. Download the certificate from Web Application Server (was.basisondemand.com)
7. Upload the certificate into SSL Client PSE

4.1. Set SECUDIR environment variable

SECUDIR is the environment variable used to identify the path for the ticket and credentials file generated as part of the below configuration.

4.2. Generate SSL Server PSE

Syntax : sapgenpse get_pse -p <PSEFile> -x <PIN> -r <CertificateRequest>
“CN=web.basisondemand.com, OU=SAP Web AS, O=Basisondemand Community, C=IN”

Tip : <PIN> can be chosen according to your requirement, it is like a password to operate with the particular PSE. The usage of PIN is explained in the next section 4.3.

```
sapgenpse get_pse -p SAPSSL.pse -x 1234 -r webbasisondemand.req  
“CN=web.basisondemand.com, OU=SAP Web AS, O=Basisondemand Community, C=IN”
```

This command will generate SAPSSL.pse, make sure to store this file under the directory indicated in parameter ssl/server_pse (in this case g:\usr\sap\WSS\secudir\sec\SAPSSL.pse)

4.3. Generate Credentials File for SSL Server PSE

Syntax : sapgenpse seclogin -x <PIN> -p <PSEFile> -O <ServiceUserUsedByWeb Dispatcher>
sapgenpse seclogin -x 1234 -p SAPSSL.pse -O BOD\PrakashPalani
Tips : -O (is case sensitive)

This command will generate cred_v2 file under SECUDIR path, make sure to store the PIN somewhere in the diary, else you will not be able to make any change to PSE without this PIN.

4.4. Generate SSL Client PSE

Syntax : sapgenpse get_pse -p <PSEFile> -x <PIN> -r <CertificateRequest>
“CN=web.basisondemand.com, OU=SAP Web AS, O=Basisondemand Community, C=IN”

```
sapgenpse get_pse -p SAPSSLC.pse -x 1234 -r webbasisondemand.req  
“CN=web.basisondemand.com, OU=SAP Web AS, O=Basisondemand Community, C=IN”
```

This command will generate SAPSSLC.pse, make sure to store this file under the directory indicated in parameter ssl/server_pse (in this case g:\usr\sap\WSS\secudir\sec\SAPSSLC.pse)

Update Credentials File for SSL Client PSE

Syntax : sapgenpse seclogin -x <PIN> -p <PSEFile> -O <ServiceUserUsedByWeb Dispatcher>
sapgenpse seclogin -x 1234 -p SAPSSLC.pse -O BOD\PrakashPalani

This command will generate/update cred_v2 file (for SAPSSLC.pse) under SECUDIR path, make sure to store the PIN somewhere in the diary, else you will not be able to make any change to PSE without this PIN.

4.5. Download the certificate from Web Application Server

Use transaction STRUST -> SSL Server Standard -> Download the Own Certificate (of Web AS)
Download the certificate locally with .cer extension.

4.6. Upload the certificate into SSL Client PSE

As Web Dispatcher acts as a SSL Client in this scenario, it is very important to import the downloaded certificate file into SSL Client PSE using the below command, else the Web Dispatcher will crash as it will not be able to establish secured connection with the message server (of was.basisondemand.com)

Syntax : sapgenpse maintain_pk -a <certificatefilename> -p <PSEfilename> -x <PIN>

```
sapgenpse.exe maintain_pk -a bod.cer -p SAPSSLC.pse -x 1234
```

Now you can test the Web Dispatcher with the url <https://web.basisondemand.com:60000>, this brings an end to this article, I have attempted to give as much information as possible to make your End-To-End SSL Web Dispatcher setup a successful one. If you have any other suggestions/questions/comments, please write to me at Prakash.palani@basisondemand.com.

5. Troubleshooting

You may use the below commands to analyze the issue that you may encounter during the Web Dispatcher startup.

Configuration Check - sapwebdisp pf=<profil> -checkconfig
Increase Trace Level – sapwebdisp pf=<profile> -t <tracelevel>

Trace Level :

- 1: Error trace
- 2: Complete process, short data trace
- 3: Complete process, complete data trace

6. References

http://help.sap.com/saphelp_nw04/helpdata/en/de/89023c59698908e10000000a11402f/content.htm
http://help.sap.com/saphelp_nwes70/helpdata/en/28/75153a1a5b4c2de10000000a114084/content.htm

Note 1026191 - Limitations on the HTTP router protocol in Web Dispatcher ([https://websmp230.sap-ag.de/sap/bd1lbiZjPTAwMQ=\)/bc/bsp/spn/sapnotes/index2.htm?numm=1026191](https://websmp230.sap-ag.de/sap/bd1lbiZjPTAwMQ=)/bc/bsp/spn/sapnotes/index2.htm?numm=1026191))
http://help.sap.com/saphelp_nw70/helpdata/en/ea/c30829166944edbc74b3c805bf529f/content.htm