How-to-Guide:
SAP Web Dispatcher for Fiori Applications
Document History:

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Authored By</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Kiran Kola</td>
<td>Architect Engineer</td>
</tr>
</tbody>
</table>
**Table of Contents**

1. Business Scenario ................................................. 4
2. Prerequisites .......................................................... 4
3. Fiori Application Architecture .................................... 5
4. SAP Web Dispatcher Installation .................................. 6
5. Web Dispatcher Communication Scenarios ....................... 8
   5.1 HTTP communication ....................................... 8
   5.2 SSL Termination ........................................... 12
   5.3 SSL Re-encryption ......................................... 16
6. Web Administration .................................................. 29
7. Appendix .................................................................. 31
1. Business Scenario

SAP supports the following proxy solutions for Fiori applications:

- SAP Web Dispatcher
- Apache Server (http://scn.sap.com/docs/DOC-62438)

SAP Web Dispatcher:

In this guide, we will illustrate how to set up SAP Web Dispatcher for Fiori applications. Communication scenarios, such as HTTP, SSL Termination, SSL re-encryption, and X.509 client certificate authentication are covered.

2. Prerequisites

All the server names used in this documentation are used to demonstrate end-to-end technical scenarios and for mockup purposes only. Following are the prerequisites and software details:

ECC/Gateway Server:

- Used ECC 6.0 EhP7 SP 7 with HANA DB (Gateway Embedded Approach)
  Host Name: mo-06a18f128.mo.sap.corp
- For Fiori Installation and Configuration, refer http://scn.sap.com/docs/DOC-41598
- For ABAP SSL, refer http://scn.sap.com/docs/DOC-53536

Web Dispatcher

A typical usage of sap web dispatcher is to provide desktop/mobile user access to SAP Gateway servers that are behind the corporate firewall so web dispatcher instance is installed in a DMZ area.

- SAP Web Dispatcher Version: 742, Windows
- SAP Web Dispatcher Server Node: ushplvm1384.phl.sap.corp
- Notepad++ http://notepad-plus-plus.org/

Assumptions:

- For SSL configuration, we used internal SAP CA for signing all the servers and client certificates
- Fiori application with SSL setup (https) is already deployed prior to this setup. However, all the major high level steps are covered in this documentation
- For this to implement, we considered embedded approach (where backend & gateway components on the same system). Same steps applied for central hub approach (where gateway is separated from backend)
- Same configuration steps can be applied for Fiori Client based applications
- SAP Web Dispatcher topics related to Performance factors, Memory requirements, and High Availability is not covered in this paper
3 Fiori Application Architecture

Below diagram is the sample Fiori application Architecture for ECC/Gateway using SAP Web Dispatcher as a proxy/Load balancer solution.

SAP Web dispatcher for Fiori Applications

<table>
<thead>
<tr>
<th>Internal Network</th>
<th>Protocol</th>
<th>Web dispatcher</th>
<th>Internal Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>mo-06a108f128.mso.sap.corp</td>
<td>http</td>
<td>8000</td>
<td>8800</td>
</tr>
<tr>
<td></td>
<td>SSL end-to-end termination</td>
<td>4300</td>
<td>6400</td>
</tr>
<tr>
<td></td>
<td>SSL re-encryption</td>
<td>4300</td>
<td>6400</td>
</tr>
</tbody>
</table>

In the following sections, we provide installation and configuration steps for SAP Web Dispatcher. Communication scenarios such as HTTP, SSL Termination, SSL re-encryption and X.509 client certificate authentication are covered step-by-step.
Installation

4. SAP Web Dispatcher Installation

In this section, SAP Web Dispatcher installation and configuration steps are illustrated:

**Step 1: Download the Software**

Download latest SAP Web Dispatcher from Service Market Place.

**SAP WEB DISPATCHER 7.42 (SUPPORT PACKAGES AND PATCHES)**

- AP, edit
- HP-UX on 64 bit
- Linux on Power edit
- Linux on x86_64 edit
- Linux on zSeries 64 bit
- OS/390
-Solaris on SPARC 64 bit
- Solaris on x86 32 bit
- Windows on x64 64 bit
- Z/OS 64 bit

**Step 2: Extract the Software**

1. Download the latest SAPCAR exe from Service Market Place
2. Create a Folder and have SAPCAR.exe (SAPCAR_315-20010453.EXE) in it along with the web dispatcher SAR file (SAPWEBDISP_SP_100-80000043.SAR)
3. Go to command prompt (search for cmd.exe), Run as Administrator, and then CD to the folder path
4. Type <sapcar.exe file> -xvf <SAP webdispatcher SAR file> as shown below:
Within the folder, you will see following folder structure:

<table>
<thead>
<tr>
<th>Name</th>
<th>Date modified</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGNATURE.SMF</td>
<td>3/6/2015 9:30 AM</td>
<td>SMF File</td>
<td>12 KB</td>
</tr>
<tr>
<td>sapstartsrv.pdb</td>
<td>3/4/2015 2:14 PM</td>
<td>PDB File</td>
<td>23,739 KB</td>
</tr>
<tr>
<td>sapstartsrv.exe</td>
<td>3/4/2015 2:14 PM</td>
<td>Application</td>
<td>11,945 KB</td>
</tr>
<tr>
<td>sapstartsrv.exe.new</td>
<td>3/4/2015 2:14 PM</td>
<td>NEW File</td>
<td>11,945 KB</td>
</tr>
<tr>
<td>librhc32.dll</td>
<td>3/4/2015 2:11 PM</td>
<td>Application extension</td>
<td>4,677 KB</td>
</tr>
<tr>
<td>librhc32.pdb</td>
<td>3/4/2015 2:11 PM</td>
<td>PDB File</td>
<td>7,299 KB</td>
</tr>
<tr>
<td>sapnwrfccm.dll</td>
<td>3/4/2015 2:09 PM</td>
<td>Application extension</td>
<td>4,946 KB</td>
</tr>
<tr>
<td>sapnwrfccm.pdb</td>
<td>3/4/2015 2:09 PM</td>
<td>PDB File</td>
<td>10,219 KB</td>
</tr>
<tr>
<td>sapeventis.dll</td>
<td>3/4/2015 2:04 PM</td>
<td>Application extension</td>
<td>791 KB</td>
</tr>
<tr>
<td>sapeventis.pdb</td>
<td>3/4/2015 2:04 PM</td>
<td>PDB File</td>
<td>171 KB</td>
</tr>
<tr>
<td>sblreg.exe</td>
<td>3/4/2015 1:50 PM</td>
<td>Application</td>
<td>761 KB</td>
</tr>
<tr>
<td>sblreg.pdb</td>
<td>3/4/2015 1:50 PM</td>
<td>PDB File</td>
<td>1,747 KB</td>
</tr>
<tr>
<td>sblreglib.dll</td>
<td>3/4/2015 1:49 PM</td>
<td>Application extension</td>
<td>3,263 KB</td>
</tr>
</tbody>
</table>

Next section illustrates communication scenario of SAP Web Dispatcher.
Communication

5. Web Dispatcher Communication Scenarios

In this section, following communication scenarios for SAP Web Dispatcher are covered:

1. HTTP Communication
2. SSL Termination (with and without X.509 client certificate verification)
3. SSL Re-encryption (with and without X.509 client certificate verification)

5.1 HTTP communication

In this section, SAP Web Dispatcher using HTTP communication is covered:

1. Configure sapwebdisp.pfl
2. Restart SAP Web Dispatcher
3. Verify HTTP Communication
4. Testing Fiori URL using sap Web Dispatcher with HTTP Protocol (unsecured communication – not recommended for productive usage)

1. Configure sapwebdisp.pfl

This is achieved by following below steps- from a) to d):

a) First, get the message server port number and host name information:

```
MSPORT=8100
TCode: SE38
FM: RSM51000_CHANGE_PARAMETER
```

<table>
<thead>
<tr>
<th>ms/http_port</th>
<th>8100</th>
<th>8100</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSHOST</td>
<td>mo-06a18f128.mo.sap.corp</td>
<td></td>
</tr>
<tr>
<td>SID</td>
<td>DV1</td>
<td></td>
</tr>
</tbody>
</table>

b) The bootstrap option is used to start the SAP Web Dispatcher without the profile file. This file is then created automatically.

```
sapwebdisp –bootstrap (don’t copy – type manually)
```

- Provide message server Hostname, MS Port and SID information.
- Leave everything else default
- Provide password for Web Administration
c) Update SAP Web Dispatcher profile

In Step 2, sapwebdisp.pfl file is created. For HTTP communication, update following profile sections as shown below:

### Backend System

```
wdisp/system_0 = SID=DV1, NR=00, MSHOST=mo-06a18f128.mo.sap.corp, MSPORT=8100, SRCSRV=*:*, SRCURL=/sap/opu/odata, SSL_ENCRYPT=0
wdisp/system_1 = SID=DV1, NR=00, MSHOST=mo-06a18f128.mo.sap.corp, MSPORT=8100, SRCSRV=*:*, SRCURL=/sap/public/bc;/sap/bc/, SSL_ENCRYPT=0
```

# Added HANA System for supporting Fiori Analytical type Applications. You can ignore section if this is not applicable for your Environment.
# Used for Multiple systems (for example, Fiori Analytical applications consuming SAP Lumira, Gateway and HANA services)

wdisp/system_conflict_resolution = 1

# SAP Web Dispatcher Ports

icm/server_port_0 = PROT=HTTP,PORT=8088

icm/HTTPS/verify_client = 0

d) Check SAP Web Dispatcher configuration

Check configuration for any issues. Following is the commend:

sapwebdisp pf=sapwebdisp.pf -checkconfig
2. Restart SAP Web Dispatcher

    sapwebdisp pf=sapwebdisp.pfl –t 2

    -t is to enable trace (options 1, 2,3)

3. Verify HTTP Communication

Validate configuration by opening a browser and testing URLs as shown below:


Enter SAP Backend Gateway credentials.

URL should return a page with below information:

![Server reached successfully]

Testing Fiori URL using web dispatcher with HTTP. Port 8088 is the http port.

5.2 SSL Termination

In this section, SAP Web Dispatcher using HTTPS communication is covered. In SSL termination, receives https encrypted data, web dispatcher decrypts the data and forwards unencrypted data to SAP backend.

1. SSL preparation for SAP Web Dispatcher
2. Restart SAP Web Dispatcher
3. Verify Communication
4. Testing Fiori URL using SAP Web Dispatcher Server URL (Secured)

1. SSL Preparation for SAP Web Dispatcher

Step 1: For HTTPS communication, update SAP Web Dispatcher profile sections as shown below:

```
# SAP Web Dispatcher SSL
DIR_INSTANCE=C:\disp\ssl
ssl/server_pse=C:\disp\sec\SAPSSLS.pse
ssl/client_pse=C:\disp\sec\SAPSSLc.pse
# SAP Web Dispatcher Ports
icm/server_port_1 = PROT=HTTPS,PORT=4300
```

Optional SSL Parameters:

```
wdisp/server_info_protocol = https
wdisp/group_info_protocol = https
wdisp/url_map_protocol = https
```

By setting above parameters, application servers, logon groups, and URL prefixes is encrypted with SSL, when it is transferred from the message server to the Web dispatcher.

Testing Fiori application will result in a certificate error as shown below:

![Certificate Error]

Step 2: Create the Web Dispatcher Server PSE and generate a CSR

Before creating the new SAPSSLS.pse, backup the old file. You can create signing request either using SAPGENPSE or OpenSSL.

**Using SAPGENPSE**

The below command will generate the PSE and certificate request (CSR).

```
sapgenpse get_pse -p C:\\disp\sec\SAPSSLS.pse -r sapwebdisp.req " CN=usphlvm1384.phl.sap.corp, OU=SSL Server, O=SAP-AG, C=DE"
```
sapwebdisp.req is created under C:\disp. Now Sign the CSR using internal CA. Once signing is done, Import the CSR response into the server PSE using the following command:

**Using OpenSSL**

**Step 1:** Downloaded OpenSSL from following link:

http://slproweb.com/download/Win64OpenSSL-1_0_0n.exe

**Step 2:** Generate RSA

Go to command line and CD to path where OpenSSL is installed. For example, C:\OpenSSL-Win64\bin

openssl genrsa -des3 -out server.key 2048

**Step 3:** Create CSR file

openssl req -sha256 -out usphlvm1384.phl.sap.corp.csr -new -newkey rsa:2048 -nodes -keyout server.key

**Step 4:** For production environments, the Certificate Signing Request that you generated can be submitted to a CA to create a certificate signed by the CA.

**Step 5:** Import Certificates

Import Response:

sapgenpse import_own_cert -p C:\disp\sec\SAPSSLS.pse -c sapwebdisp.cer -r sapnet.cer

```
C:\disp> sapgenpse import_own_cert -p C:\disp\sec\SAPSSLS.pse -c sapwebdisp.cer -r sapnet.cer
CA-Response successfully imported into PSE "C:\disp\sec\SAPSSLS.pse"
```

Import Root Certificate:

sapgenpse maintain_pk -a SAPNetCA.cer -p C:\disp\sec\SAPSSLS.pse

Note: Repeat above steps for SAPSSLc.pse. SAPNetCA.cer is the Root certificate of the internal CA

2. **Restart SAP Web Dispatcher**

sapwebdisp pf=sapwebdisp.pfl -t 2

3. **Verify SSL Termination Scenario**

Validate the configuration by opening a browser and testing these URLs:


Enter SAP Backend Gateway credentials.
URL should return a page with this information:


Port 4300 is the HTTPS port.


In SSL termination scenario, incoming connection is decrypted/terminated at SAP web dispatcher and unencrypted data is sent as clear text to backend SAP Gateway (via HTTP). This scenario is considered if network between Web Dispatcher and SAP Gateway is secured. Following example analyze this scenario:

Example for SSL Termination without Client Certificate Validation:

[Thr 3960] ICR parse: <protocol-type> = 'HTTP'
[Thr 3963] ICR parse: <host> = 'mo-06a18f128.mo.sap.corp'
[Thr 3960] ICR parse: <port> = '8800'
[Thr 3960] ICR parse: <description> name='VHOST' val='0'
[Thr 3960] ICR parse: VHOST=0
To further secure communication, Client verification (client certificates) is enabled by adding following parameter:

\[
\text{icm/HTTPS/verify_client = 1 or 2}
\]

Options:

1: The server asks the client to transfer a certificate. If the client does not send a certificate, authentication is carried out by another method, for example, basic authentication (default setting).

2: The client must transfer a valid certificate to the server, otherwise access is denied.

Example for SSL Termination with Client Certificate Validation

5.3 SSL Re-encryption

In this section, SSL Re-encryption with and without Client Certificate Verification are covered.

The `wdisp/ssl_encrypt` determines whether the SAP Web dispatcher encrypts the request again with SSL before forwarding it.

`wdisp/ssl_encrypt = 0` (receives https encrypted data, web dispatcher decrypts the data and forwards unencrypted data to SAP backend)

`wdisp/ssl_encrypt = 1` (receives https encrypted data, web dispatcher decrypts the data, re-encrypt again and forwards encrypted data to SAP backend)

`wdisp/ssl_encrypt = 2` (the SSL is not terminated and request is sent encrypted to SAP backend)

Note: You can also configure the SAP Web Dispatcher for end-to-end SSL, by specifying the protocol ROUTER when you define the `icm/server_port_<xx>` parameter.

Example: `icm/server_port_0 = PROT=ROUTER,PORT=60000`

SSL Re-encryption without Client Certificate Verification:

**Step 1:** Verify profile parameters on the SAP Gateway Backend

Make sure `icm/HTTPS/verify_client = 0` is set.

**Step 2:** Update SAP Web Dispatcher profile

Update following sections of the SAP Web dispatcher profile:

# SAP Web Dispatcher SSL

`wdisp/ssl_encrypt=1`

`icm/HTTPS/verify_client = 0`

# Backend System

`wdisp/system_0 = SID=DV1, NR=00, MSHOST=mo-06a18f128.mo.sap.corp, MSPORT=8100, SRCsrv=*,*, SRCURL=/sap/opu/odata, SSL_ENCRYPT=1`

`wdisp/system_1 = SID=DV1, NR=00, MSHOST=mo-06a18f128.mo.sap.corp, MSPORT=8100, SRCsrv=*,*, SRCURL=/sap/public/bc;/sap/bc/, SSL_ENCRYPT=1`
# HANA (with HTTPS connectivity)

wdisp/system_2 = SID=OHY, EXTSRV=https://mo-btb805bb3.mo.sap.corp:8001, SRCsrv=*:*, SRCURL=/sap/hba/;/sap/hana, SSL_ENCRYPT=0

Troubleshooting 503 Service not available Issue:

Symptom: Your direct Fiori URL to gateway might work but when you try from Web dispatcher you might get 503 error as shown below:

Increase the web dispatcher trace to Level 2 and retry; analyze your trace. There could be peer not trusted error as shown in the below. In the below case, SAP Gateway server host certificate is not trusted.
To fix the above 502 error, export the Server certificate from STRUST (SAP Gateway) and import it to SAP Web Dispatcher as shown in the below example:

```
C:\disp\sapgenpse maintain_pk -a C:\disp\mo-06a18f128.no.sap.corp.cer -p C:\disp\sec\SAPSSL.pse
maintain_pk for PSE "C:\disp\sec\SAPSSL.pse"
Subject : CN=mo-06a18f128.no.sap.corp, OU=SAP Web AS, O=SAP-AG, C=DE
PKList updated (1 entries total, 1 newly added)
```

SSL Re-encryption with Client Certificate Verification (Two Way SSL Communication):

1. Create X.509 client certificate (SAP Gateway)
2. Adding profile parameter for Client verification and trusted relationship (SAP Gateway)
3. Changing the Logon Procedure
4. Update SAP Web Dispatcher profile
5. Import SAP Gateway certificate into SAP Web Dispatcher trust store
6. Import SAP Web Dispatcher certificate into SAP Gateway trust store using STRUST
7. Restart SAP Web Dispatcher
8. Load.p12 Client Certificate in to the browser
9. Verify SSL Re-encryption communication with Client Certificates

1. Create X.509 client certificate

To digitally identify a particular individual, client certificates can be used. In general, certificates are issued by company’s PKI (X.509 **Public Key Infrastructure**). In our case, to test our scenario we will create end user certificates using OpenSSL.

**Step 1:** Downloaded OpenSSL from following link:

http://slproweb.com/download/Win64OpenSSL-1_0_0n.exe

**Step 2:** Generate RSA

Go to command line and CD to path where OpenSSL is installed. For example, C:\OpenSSL-Win64\bin

openssl genrsa -des3 -out server.key 2048

**Step 3:** Create CSR file

openssl req -sha256 -out SUPUSER.csr -new -newkey rsa:2048 -nodes -keyout server.key

**Step 4:** Sign it using your internal CA

**Step 5:** You will receive signed certificate. Save the SUPUSER.CRT to local drive.

**Step 6:** Convert to crt to pfx format

openssl pkcs12 -export -out SUPUSER.pfx -inkey server.key -in SUPUSER.crt

Note: Enter password if required.
Step 7: Maintain User mapping

TCode: go to SE11,
Database Table: VUSREXTID and hit Display
Click on Contents (Ctl+sht+f10)
External ID Type: DN of certificate (DN)
Click and change and New entries
- External ID should be the Subject DN
- Assign the user (prior to this activity make sure user is already created)
- Check Activated

Important: Subject DN should match with the VUSREXTID entry as shown in the following screens:

2. Adding profile parameters for client verification and establishing trusted relationship

Step 1: TCode: RZ10
Set the AS ABAP profile parameter icm/HTTPS/verify_client to the value 1 (accept certificates) or 2 (require certificates) to support the use of client certificates.

**Step 2**: TCode: RZ10

For X.509-based logon to NW AS using the SAP Web Dispatcher, you need following parameters to create a trusted relationship between the SAP Web dispatcher and ICM

```
icm/HTTPS/trust_client_with_issuer = <Distinguished Name of subject to trust>
icm/HTTPS/trust_client_with_subject = <Distinguished Name of subject to trust>
```

Example:

```
icm/HTTPS/trust_client_with_issuer = EMAIL=maik.mueller@sap-ag.de, CN=SAPNetCA, OU=SAPNet, O=SAP-AG, C=DE
icm/HTTPS/trust_client_with_subject = CN=usphlvm1384.phl.sap.corp, OU=SSL Server, O=SAP-AG, C=DE
```

### 3. Changing the Logon Procedure

**Step 1**: TCode SICF

Navigate to `default_host/sap/bc/ui5_ui5/ui2/`

Service Name: ushell

Change Logon Procedure to: Required with SSL Certificate

![Service Data and Logon Data](image)

**Step 2**: Restart ICM

**Step 3**: Testing SAP Gateway Fiori URL
Note: please refer Point 5 for loading p.12 certificate into browser


When you hit the Gateway URL, a popup for certificate selection is appeared. You select the right end user certificate and hit OK.

User is authenticated with X.509 Certificate as shown below:

Now next step is doing the same via SAP Web Dispatcher.

4. Update SAP Web Dispatcher profile

Update following sections of the SAP Web Dispatcher profile:

# SAP Web Dispatcher SSL

wdisp/ssl_encrypt = 1
icm/HTTPS/verify_client = 1

# Backend System

wdisp/system_0 = SID=DV1, NR=00, MHOST=mo-06a18f128.mo.sap.corp, MSPORT=8100, SRCRV=*, SRCURL=/sap/opu/odata, SSL_ENCRYPT=1

wdisp/system_1 = SID=DV1, NR=00, MHOST=mo-06a18f128.mo.sap.corp, MSPORT=8100, SRCRV=*, SRCURL=/sap/public/bc;/sap/bc/, SSL_ENCRYPT=1

5. Import SAP Gateway certificate into SAP Web Dispatcher trust store

Hint: Using STRUST, you can export the server certificate

sapgenpse maintain_pk -a mo-06a18f128.mo.sap.corp.cer -p C:\disp\sec\SAPSSL.pse

6. Import SAP Web Dispatcher certificate into SAP Gateway trust store using STRUST

7. Restart SAP Web Dispatcher

sapwebdisp pf=sapwebdisp.pfl –t 2

8. Load .p12 Client Certificate into the browser

For mutual authentication using client certificates, Web Dispatcher needs the private keys to do the signing, and the .p12 file format is the most common for passing around a certificate with its private keys. To test, we need client certificate (.p12 file) which is usually provided by your OS security team who handles Certificate Authority.

Step 1: Load the .p12 client certificate into the personal certificate store. In Chrome, choose Settings > Show Advanced Settings > HTTPS/SSL > Manage certificates as shown below screen:
Step 2: Click Import button:

Step 3: Click Next button:
Step 4: Click browse and select the p.12 file

Step 5: Select All files from dropdown:
Step 6: Select p.12 and hit Next button as shown below:

![Certificate Import Wizard](image)

Step 7: If password exists, provide password and hit next:

![Certificate Import Wizard](image)
9. Verify SSL Re-encryption communication

Validate the configuration by opening a browser and testing these URLs as shown below:


When you hit the web dispatcher URL, a popup for certificate selection is appeared. You select the right end user certificate and hit OK.

URL should return a page with the below information:

Testing Fiori URL using SAP Web Dispatcher with mutual HTTPS connection.

Troubleshooting X.509 Client Certificate communication issues

Symptoms:

- SAP Gateway and SAP Web Dispatcher is configured to use X.509 certificates to authenticate Fiori application (icm/HTTPS/verify_client =1)
- When user hits SAP Web Dispatcher URL, certificate pop-up is displayed; once user certificate is presented, Basic Authentication Fiori Launchpad screen is displayed.

Step 1: Increase your ICM trace to level 2 for retrieving detailed SSL debug information. For more information refer following link:

http://help.sap.com/saphelp_nw70ehp2/helpdata/en/48/3a062c902131c3e10000000a42189d/content.htm

In my case, I see following error in ICM trace with Certificate subject miss match:

```
[Thr 14398547585912] -> SapSslGetPeerInfo(assl_hdl=1371100, acert=7f0749f08ec, acert_len=7f0749f08ec,
[Thr 14398547585912] # asubject_dn=7f0749f08cd, asigner_dn=7f0749f08bd, asigner_len=7f0749f08bd)
[Thr 14398547585912] <- SapSslGetPeerInfo(assl_hdl=1371100)=-SAP_0_X
[Thr 14398547585912] out_subject = "CN=applvm384.ph1.sap.corp, O=SSL Server, O=SAP-AG, C=DE"
[Thr 14398547585912] out_issuer = "EMH4-maik.mueller@ag.de, CN=SAPNetCA, O=SAPNet, O=SAP-AG, C=DE"
[Thr 14398547585912] out_cert_len = 56e
[Thr 14398547585912] out_cipher = "TLS_RSA_WITH_AES_128_CBC_SHA"
[Thr 14398547585912] HrpcSubHandlerIfcDestroy: handle 0: HrpcSubHandler
[Thr 14398547585912] HrpcSubHandlerCall: Call Handler: HrpcModHandler (13fed30/126f010), task=TASK_REQUEST(1), header_len=2608
[Thr 14398547585912] <- SapSslGetPeerInfo(assl_hdl=1371100, acert=7f0749f0878, acert_len=7f0749f0878,
[Thr 14398547585912] # asubject_dn=7f0749f0870, asigner_dn=7f0749f0868, asigner_len=7f0749f0868)
[Thr 14398547585912] <- SapSslGetPeerInfo(assl_hdl=1371100)=-SAP_0_X
[Thr 14398547585912] out_subject = "CN=applvm384.ph1.sap.corp, O=SSL Server, O=SAP-AG, C=DE"
[Thr 14398547585912] out_issuer = "EMH4-maik.mueller@ag.de, CN=SAPNetCA, O=SAPNet, O=SAP-AG, C=DE"
[Thr 14398547585912] out_cert_len = 56e
[Thr 14398547585912] out_cipher = "TLS_RSA_WITH_AES128_CBC_SHA"
[Thr 14398547585912] HttpRedirectRules: Client certificate received: with len=680, subj="CN=applvm384.ph1.sap.corp, O=SSL Server, O=SAPNetCA, O=SAPNet, O=SAP-AG, C=DE"
[Thr 14398547585912] --> GetCertificateInfo: overwriting k3rinfo [issuer.c 606]

```
From the profile, we can see host name is in the CAPS. Now, subject DN is modified and restarted ICM. After retesting the application successfully, following “Accept trusted forward certificates” message is displayed in ICM trace.

Next section illustrates about Web Administration for SAP Web Dispatcher.
Monitoring

6. Web Administration

Web Administration interface is a very useful tool used to administrate, monitor, and set traces in the SAP Web Dispatcher. Web administrator user is automatically created during the Bootstrap. If you want to create a new user, following below link:

https://help.sap.com/saphelp_nw70/helpdata/en/82/9e98d786f040209e6a9e8145153939/content.htm

URL path to access Web Administration: http(s)://host:admin_port/sap/admin


For more information, refer following help link:

In summary, this white paper covers how to set up SAP Web Dispatcher for Fiori applications. Communication scenarios, such as HTTP, SSL Termination, SSL re-encryption, and X.509 client certificate authentication are covered.

Helpful References:

7. Appendix

# Sample Profile for SSL Re-encryption with Client Certificate Verification

# unique instance identifier
SAPSYSTEMNAME = WDP

# unique instance number
SAPSYSTEM = 88

# add default directory settings
DIR_INSTANCE = C:\disp
DIR_EXECUTABLE = $(DIR_INSTANCE)
DIR_PROFILE = $(DIR_INSTANCE)
DIR_HOME = $(DIR_INSTANCE)
Autostart = 1
Restart_Program_00 = local $(DIR_EXECUTABLE)/sapwebdisp$(FT_EXE) pf=$(DIR_PROFILE)/sapwebdisp.pfl

# Backend System
#wdisp/system_0 = SID=DV1, MSHOST=mo-06a18f128.mo.sap.corp, MSPORT=8100, SRCSRV=*, SRCURL=/,
SRCSRV=*, SSL_ENCRYPT=1
wdisp/system_0 = SID=DV1, NR=00, MSHOST=mo-06a18f128.mo.sap.corp, MSPORT=8100, SRCSRV=*, SRCURL=sap/opu/odata, SSL_ENCRYPT=1
wdisp/system_1 = SID=DV1, NR=00, MSHOST=mo-06a18f128.mo.sap.corp, MSPORT=8100, SRCSRV=*, SRCURL=sap/public/bc;/sap/bc/, SSL_ENCRYPT=1
icm/accept_forwarded_cert_via_http = true

# number of parallel connections
icm/max_conn      = 2000
icm/max_sockets   = ($(icm/max_conn) * 2)
icm/req_queue_len = 6000
icm/min_threads   = 10
icm/max_threads   = 500
mpi/total_size_MB = (min(0.06 * $(icm/max_conn) + 50, 2000))
mpi/max_pipes = ($(icm/max_conn))
wdisp/HTTP/max_pooled_con = ($(icm/max_conn))
wdisp/HTTPS/max_pooled_con = ($(icm/max_conn))

# SAP Web Dispatcher Ports
icm/server_port_0 = PROT=HTTP,PORT=8088
icm/server_port_1 = PROT=HTTPS,PORT=4300
wdisp/ssl_cred = SAPSSLC.pse

# SAP Web Dispatcher Web Administration
icm/HTTP/admin_0 = PREFIX=/sap/admin, PORT=8088,DOCROOT=./admin, HOST=usphlvm1384;localhost, AUTHFILE=icmauth.txt

# SAP Web Dispatcher SSL
DIR_INSTANCE=C:\disp\.getvalue
ssl/ssl_lib=C:\disp\sapcrypto.dll
ssl/server_pse=C:\disp\sec\SAPSSL.pse
ssl/client_pse=C:\disp\sec\SAPSSL.pse
icm/HTTPS/verify_client = 1
wdisp/ssl_encrypt=1