Product Support Notice

PSN #	PSN004561u

Original publication date: 25-Aug-15. This is Issue #08, published date: 09-July-18.	Severity/risk level	High	Urgency	Immediately

Name of problem	PSN004561u - Avaya Aura® Application Enablement (AE) Services Server Certificate Change and Expiration Notification.

Products affected

Avaya Aura® Application Enablement (AE) Services Release 5.x, 6.x, 7.x and 8.x (all offer types)

Problem description

Note: If this PSN is not addressed as soon as possible it could lead to service outages!
Note: In configurations with Avaya Aura® Contact Center (AACC), additional steps must be taken. Refer to PSN005110u for additional information.
Note: Ongoing use of default server certificates does not comply with best security practices.

For the AE Services 5.x and 6.x releases, all versions of the default installed server certificate are scheduled to expire no later than January 6, 2018. For any customer using this certificate, once this certificate expires, an AE Services based client using a TLS connection will not be able to communicate with the AE Services server.
Beginning with AE Services 7.0 release, the AE Services server discontinued the use of a default server certificate signed by Avaya. Customers are required to install their own certificates signed by either their own Private Key Infrastructure (PKI) or a third party PKI vendor. If such resources are not available immediately, they may use the temporary AE Services server self-signed certificate. Note that this self-signed certificate is based on SHA2, which may not work with some older clients, and the certificate is valid for only 1 year. It is expected that customers will deploy their own certificates before this certificate expires.

For an upgrade from a previous AE Services 5.x or 6.x release to AE Services 7.x or 8.x, any customer application relying on the old, Avaya provided server certificate for TLS will not be able to connect to the AE Services 7.x or 8.x server. If you have been using these certificates in a production environment, we strongly recommend that you prepare and execute a rollout plan, as soon as possible, to update your client applications and AE Services server with your own certificates. We strongly encourage customers to create this certificate prior to upgrading to the AE Services 7.x or 8.x release.

Note: For any AE Services release, where the installed AE Services server certificate has been replaced with a customer provided certificate, the client/server TLS connection will not be affected by the aforementioned certificate expiration or replacement.

Note: In configurations with Avaya Aura® Contact Center (AACC), additional steps must be taken. Refer to PSN005110u for additional information.

For guidance on how to check AE Services certificate expiration and how to determine whether applications connected to AE Services use a TLS connection refer to this KB Article.

Resolution

Note: In configurations with Avaya Aura® Contact Center (AACC), additional steps must be taken. Refer to PSN005110u for additional information.

Note: Once the certificate is updated on AE Services, the default self-signed Avaya certificate will be overridden and no longer in effect.

Possible customer options to create the new AE Services server certificate:

- Use your own PKI
- Use Avaya Aura's System Manager (SMGR) Trust Management PKI feature **
- Use an Open Source PKI (e.g. EJBCA)*
- Use a third party vendor (e.g. Verisign)*
- Use OpenSSL to create your own Certificate Authority (CA) ***

***** Avaya does not endorse or require the use of this product or vendor. You may use any product or vendor of your choosing.
****** See the System Manager Trust Management section below

******* See the OpenSSL section below

If for some reason none of the above options fit your immediate need, please contact Avaya Services for additional assistance.

System Manager Trust Management

Using Avaya Aura System Manager (SMGR) as a Certificate Authority (CA) to generate signed certificates.

The following steps specify how to use System Manager's Trust Management feature as your PKI for the AE Services server.

Create an "End Entity" for the AE Services server:

1. Using the SMGR web console, navigate to "Security (under Services) > Certificates > Authority > Add End Entity (under RA functions).
2. In the "End Entity Profile" field, click INBOUND_OUTBOUND_TLS.
3. Type a username and password. This password is used to encrypt the P12 trust store file (see below).
4. Complete the fields that you want in your certificate
    a. E-mail address: labmanager@yourcompany.com
    b. CN, Common name: aeshostname.yourcompany.com
    c. OU, Organizational Unit: IT
    d. O, Organization: Your Company Name
    e. L, Locality: Denver
    f. ST, State or Province: CO
    g. C, Country: US
5. In the "Certificate Profile" drop down menu select ID_CLIENT_SERVER
6. In the "CA" drop down menu select "tmdefaultca"
7. In the "Token" drop down menu select "P12 file"
8. Click on "Add" button
9. **Note:** On the top of the page the system displays the message "End Entity <username> added successfully"

Create the AE Services server certificate:

1. Using the SMGR web console, navigate to "Security (under Services) > Certificates >Authority".
2. In the left hand navigation pane near the bottom of the screen, click on "Public Web"
3. On the "Public Web" screen click on "create key store"
4. Enter the user name and password of the "End Entity" and click "OK"
    a. Select the certificate key length (2048 is recommended)
    b. Click on "Enroll"
    c. Save the server certificate to a known location
    Note: This is the signed server certificate you have to import into the AE Services server.

Download the SMGR CA certificate that signed the AE Services server certificate:

1. Using the SMGR web console navigate to the "Public Web" page (as described in the above step) and click on "Fetch CA certificates"
2. Click on "Download PEM chain" on the line starting with "CA certificate chain"
3. Save the CA certificate to a known location
Note: This is the CA certificate that needs to be imported into the AE Services erver.

Import the SMGR CA certificate into the AE Services server:

1. Using the AE Services Management Console navigate to "Security > Certificate Management > Trusted Certificates"
2. Click on the "Import" button and upload the SMGR CA certificate you downloaded above. Give it an alias name (e.g. caSMGR)
3. Click the "Apply" button
Note: You need to import this CA before you can import the AE Services server certificate

Import the new AE Services server certificate into the AE Services server:

1. Using the AE Services Management Console navigate to "Security > Certificate Management > Server Certificates"
2. Click on the Import button and upload the new AE Services server certificate you created above. Select an alias (e.g. server) from the drop down menu

3. Click the “Apply” button.
4. Enter the password you used when creating the “End Entity”
5. Click the “Apply” button and then click the “Apply” button again on the next page.
   Note: You need to import the CA before you can import the AE Services server certificate

OpenSSL

Using OpenSSL as a Certificate Authority (CA) to generate signed certificates.

The following steps use a key size, cipher, and a single-level CA, instead of a multi-level CA infrastructure, that may be considered inefficient to support your IT security requirements. It is recommended that you review the OpenSSL commands and make the necessary changes to meet or exceed your certificate security requirements. These commands are provided as is, use at your own risk, with no guarantee that they will protect your network from a possible intrusion.

The OpenSSL package is available on all Linux distributions, Windows (e.g. using cygwin) and it is available for download from the OpenSSL web site.

On a Linux server, use the man command (e.g. man genrsa) to find out additional information on the openssl commands (genrsa, req, x509, ca, and pkcs12).

The following steps, based on Red Hat Enterprise Linux (RHEL), specify how to create a single-level OpenSSL CA.

Create the certificate directory structure:

1. Create a directory to serve as your certificate home directory (e.g. mkdir /certificates)
2. Change directory to the certificates home directory (e.g. cd /certificates)
3. From the Linux command line interface (CLI), execute the following commands
   - mkdir CA
   - mkdir CA/certs
   - mkdir CA/crl
   - mkdir CA/newcerts
   - mkdir CA/private
   - touch CA/index.txt
   - touch CA/private/cakey.pem
   - touch CA/serial

Configure the OpenSSL configuration file:

1. Determine the location of OpenSSL’s default openssl.cnf file. On RHEL, it is at /etc/pki/tls/openssl.cnf. If not, use the find command to locate the file (i.e. **find / -name openssl.cnf**)
2. Copy the openssl.cnf file to the certificates home directory (e.g. **cp /etc/pki/tls/openssl.cnf /certificates**)
3. Change directories to the certificates home directory (e.g. **cd /certificates**)
4. Edit the copied version of the openssl.cnf file with the following changes (e.g. **vi /certificates/openssl.cnf**)
   - Modify the following line in the section [CA_default]
     Change from:
     dir = ../../CA
     Change to:
     dir = ./CA
   - Comment out the 2 appearances of the following line
     Change from:
     nsComment = "OpenSSL Generated Certificate"
     Change to:
     #nsComment = "OpenSSL Generated Certificate"
   - Uncomment the following line and add v3_req to extensions
     Change from:
     # X.509v3 extensions to use:

 # extensions =

Change to:

 # X.509v3 extensions to use:

extensions = v3_req

- Uncomment the following line
  Change from:
  # req_extensions = v3_req # The extensions to add to a certificate request
  Change to:
  req_extensions = v3_req # The extensions to add to a certificate request
- Change the following line in the [v3_req] section
  Change from:
  keyUsage = nonRepudiation, digitalSignature, keyEncipherment
  Change to:
  keyUsage = nonRepudiation, digitalSignature, keyEncipherment, dataEncipherment
- Add the following line to the [v3_req] section
  extendedKeyUsage=serverAuth,clientAuth
- Add the following line to the [usr_cert] section
  Change from:
  # These extensions are added when 'ca' signs a request.
  Change to:
  # These extensions are added when 'ca' signs a request.
  keyUsage = nonRepudiation, digitalSignature, keyEncipherment, dataEncipherment
  extendedKeyUsage=serverAuth,clientAuth
- Change string_mask in the [req] section to the following:
  string_mask = MASK:0x2002
- If you want to use a message digest higher than sha1 (e.g. sha256), change the option "default_md" in the [req] section

5. Save the changes to the openssl.cnf file

Create the CA root key and self-signed certificate:

1. Change directories to the certificates home directory (e.g. **cd /certificates**).
2. Create the CA key by executing the following command:
   openssl genrsa -des3 -out cakey.pem 2048
   You will be asked for a password to encrypt the CA root key, and then you'll be asked for that password again as verification. The key will be encrypted using triple des (i.e. des3). You will be asked for this password when signing the CSR. The size of the key will be 2048 bits. The key will be saved to the file cakey.pem.
3. Create the CA public certificate signed by the CA key by executing the following command:
   openssl req -new -x509 -days 3650 -key cakey.pem -out cacert.pem -config openssl.cnf
   You will be asked for the CA root key password. The public CA certificate will expire in 10 years. In order to change the expiration modify the option "–days". The certificate will be saved in the file cacert.pem. The openssl.cnf will be used to apply some configuration values. You will be asked to enter information that will be incorporated into your certificate request. Leave the Email Address blank.
   For example:

   Country Name (2 letter code) [GB]:US
   State or Province Name (full name) [Berkshire]:CO
   Locality Name (eg, city) [Newbury]:Denver
   Organization Name (eg, company) [My Company Ltd]:Your Company Name
   Organizational Unit Name (eg, section) []:IT
   Common Name (eg, your name or your server's hostname) []:YourCompanyName Root CA
   Email Address []:
4. Move the generated CA root key to the CA/private directory (i.e. **mv cakey.pem ./CA/private/**)
5. Move the generated CA public certificate to the CA directory (i.e. **mv cacert.pem ./CA/**)
6. You can view the contents of the CA public certificate with the following command:
   openssl x509 -in ./CA/cacert.pem -text -noout

Note: Your CA is now configured and ready to issue certificates. This CA can be used to create all your AE Services server certificates and only the CA certificate will need to be imported into your clients trust certificate store.

The following steps specify how to generate a server certificate for each AE Services server.

Create a certificate signing request (CSR) and key for each of your AE Services server:

1. Login to the AE Services Management console of the server for which you need to create the certificate
2. Navigate to "Security > Certificate Management > Server Certificate > Add"
    a. "Certificate Alias": select either "server" or "aeservices" from the drop down menu
    b. "Create Self-Signed Certificate" leave unchecked
    c. "Enrollment Method": from the drop down menu select "Manual"
    d. "Encryption Algorithm": from the drop down menu select "3DES"
    e. "Password": enter a password that will be used to encrypt the private key associated with the certificate. The encrypted private key will be kept on the AE Services server.
    f. "Re-enter Password": Enter the above password for verification
    g. "Key Size": from the drop down menu select "2048"
    h. "Signature Algorithm": from the drop down menu select "sha256". If your client does not support sha256, select sha1.
    i. "Certificate Validity": enter the number of day you want this certificate to be valid for (5 years 1825 days)
    j. "Distinguished Name (DN)": C=US, ST=CO, L=yourCity, O=Your Company Name, OU=yourOrg, CN=aeshostname
        1. Note: The "C" value (US), the ST value (CO) and the O value (Your Company Name) must match the country name, state name and the company name of the CA certificate for the CA certificate to be able to sign the CSR
    k. "Key Usage": holding the Ctrl key down on the keyboard select the "Digital Signature", "Non-repudiation", "Key encipherment", and "Data encipherment" options
    l. "Extended Key Usage": holding the Ctrl key down on the keyboard select the "SSL/TSL Web Server Authentication" and "SSL/TLS Web Client Authentication" option
    m. Leave the fields not addressed above empty.
    n. Click on the "Apply" button
3. From the "Server Certificate Manual Enrollment Request" page copy the CSR certificate in the window starting with "-----BEGIN CERTIFICATE REQUEST-----" and ending with "-----END CERTIFICATE REQUEST----- ". Save the CSR to a file named **myserver.req** in the **/certificate** directory on the server where the CA certificate was created.

Sign the AE Service server CSR:

Using the server where the CA certificate was created:

1. Change directories to the certificates home directory (e.g. **cd /certificates**).
2. Create a serial number for the certificate by executing the following command:
   tr -c -d 0-9 < /dev/urandom | head -c 10 > ./CA/serial
3. Sign the CSR by executing the following command
   openssl ca -config openssl.cnf -days 730 -out myserver.crt -infiles myserver.req
   The openssl.cnf file will be used to apply some configuration options. The signed public certificate will expire in 2 years. In order to change the expiration modify the option "–days". The certificate will be saved in the file myserver.crt. The CSR is in the file myserver.req. You will be asked for the CA root key password to sign the CSR and confirmation to sign and commit the request.
4. You can view the contents of the newly signed public server certificate with the following command:
   openssl x509 -in ./ myserver.crt -text –noout

Import the Certificate:

1. Import the public CA certificate into the AE Services trust certificate store using the AE Services Management Console.
    a. Using the AE Services Management Console navigate to "Security > Certificate Management > Trusted Certificates"
    b. Click on the Import button and upload the OpenSSL generated CA certificate you created above. Give it an alias name (e.g. myCA)
    c. Click the Apply button
    Note: The CA certificate must be imported before the signed server certificate can be imported into the AE Services server in the next step.

2. Import the **myserver.crt** file created in the previous step into the AE Services server using the AE Services Management Console.
    a. Using the AE Services Management Console navigate to “Security > Certificate Management > Server Certificates”
    b. Click on the Import button and upload the new AE Services server certificate (**myserver.crt**) you created above. Select the alias you chose when creating the CSR (e.g. either “server” or “aeservices”) from the dropdown menu
    c. Click the Apply button.
    Note: Make sure, the AE Services server and the server where the CA was created are properly time synchronized to a NTP server.

3. Import the public CA certificate into the AE Services related clients trust certificate stores.

Note: You can repeat the CSR creation and subsequent steps for different AE Services servers and use the same CA to sign each of the CSRs. This way all the AE Services clients only need to know one CA certificate, that signed all the AE Services server certificates.

Note: In configurations with Avaya Aura® Contact Center (AACC), additional steps must be taken. Refer to PSN005110u for additional information.

Workaround or alternative remediation

n/a

Remarks

Issue 8 – added AE Services 8.0 reference

Patch Notes

The information in this section concerns the patch, if any, recommended in the Resolution above.

Backup before applying the patch

n/a

Download

n/a

Patch install instructions

n/a

Service-interrupting?

Yes

Verification

n/a

Failure

n/a

Patch uninstall instructions

n/a

Security Notes

The information in this section concerns the security risk, if any, represented by the topic of this PSN.

Security risks

n/a

Avaya Security Vulnerability Classification

Not Susceptible

Mitigation

n/a

If you require further information or assistance please contact your Authorized Service Provider, or visit support.avaya.com. There you can access more product information, chat with an Agent, or open an online Service Request. Support is provided per your warranty or service contract terms unless otherwise specified in the Avaya support Terms of Use.

Disclaimer: ALL INFORMATION IS BELIEVED TO BE CORRECT AT THE TIME OF PUBLICATION AND IS PROVIDED "AS IS". AVAYA INC., ON BEHALF OF ITSELF AND ITS SUBSIDIARIES AND AFFILIATES (HEREINAFTER COLLECTIVELY REFERRED TO AS "AVAYA"), DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND FURTHERMORE, AVAYA MAKES NO REPRESENTATIONS OR WARRANTIES THAT THE STEPS RECOMMENDED WILL ELIMINATE SECURITY OR VIRUS THREATS TO CUSTOMERS' SYSTEMS. IN NO EVENT SHALL AVAYA BE LIABLE FOR ANY DAMAGES WHATSOEVER ARISING OUT OF OR IN CONNECTION WITH THE INFORMATION OR RECOMMENDED ACTIONS PROVIDED HEREIN, INCLUDING DIRECT, INDIRECT, CONSEQUENTIAL DAMAGES, LOSS OF BUSINESS PROFITS OR SPECIAL DAMAGES, EVEN IF AVAYA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THE INFORMATION PROVIDED HERE DOES NOT AFFECT THE SUPPORT AGREEMENTS IN PLACE FOR AVAYA PRODUCTS. SUPPORT FOR AVAYA PRODUCTS CONTINUES TO BE EXECUTED AS PER EXISTING AGREEMENTS WITH AVAYA.