EPSON

9100-13-GEN-002

Multiple IP (SAN) Certificates for Epson TM Printers

General Information

DISCLAIMER: Epson makes this document available to Epson printer users as general guidance only. Users, and not Epson, are ultimately responsible for their own POS system security, including security certificate practices.

Table of Contents

1. Introduction 3
1.1. What This Document Is (and Isn't) 3
1.2. Requirements 3
1.3. Ubuntu 18.04 Setup 3
1.4. Microsoft Windows ™ Setup 4
2. Procedure 5
2.1. Generate a CA 5
2.2. Generate Signed Printer Certificate (Multiple IP) 6
3. Limitations 9
3.1. Maximum Number of SAN Entries in Certificate 9
4. Appendix 10
4.1. Check Supported Ciphers in OpenSSL 10

1. Introduction

Modern browsers are increasingly sensitive to outdated security certificate practices. A recent breaking change in Chrome and Firefox involves preferring Subject Alternative Name (SAN) over Common Name (CN). Different browsers handle this differently. For example:

- Internet Explorer/Microsoft Edge: use SAN if and only if it exists, otherwise use CN
- Chrome/Firefox/Safari: prefer SAN over CN, deprecate CN

Traditionally, "Intelligent" Epson TM Printers (the -i and -DT series) have come equipped with self-signed certificates. Most modern browsers now throw an exception with that as well: self-signed certificates are now always considered unsafe.
The above browser updates may even interfere with Java ePOS SDK usage. In light of these developments, Epson customers are faced with a need to update the printer certificates in order to continue using them on their corporate networks without security warnings/failures.

1.1. What This Document Is (and Isn't)

- This document demonstrates one of several methods used to generate a signed certificate for use on Epson TM printers.
- It should be treated as a primary guide in creating signed certificates using SAN over CN.
- It is assumed the reader understands the many case-specific customizations possible in details presented herein; this document does not cover all possibilities.
- Red text indicates where users are encouraged to explore other options suitable for their use case (see Appendix for tips).
- This document does not go into specifics of any particular browser.
- This document does not elaborate on certificate deployment methods on systems other than the target Epson TM Printer.

1.2. Requirements

- OpenSSL 1.1.0g or newer
- Ubuntu 18.04, Microsoft Windows 7 or Windows 10

1.3. Ubuntu 18.04 Setup

Ubuntu is a well-known open-source Linux distribution available at no cost. It can be downloaded and installed on most modern desktops and portable computers. Most versions of it come equipped with OpenSSL, the security framework utilized in this document to generate the SSL certificates.

1.3.1. Follow this tutorial to install Ubuntu on a computer:
https://tutorials.ubuntu.com/tutorial/tutorial-install-ubuntu-desktop

1.3.2. Once logged into Ubuntu, launch a shell terminal by either:
i. Pressing Ctrl + Alt + t on the keyboard, *or*
ii. Show Applications (on task bar) -> Search for "Terminal" and run it

1.3.3. Ensure the latest version of OpenSSL is installed on your machine by entering this command in the terminal window:

```
sudo apt install openssl
```

1.3.4. Ensure OpenSSL is working properly using this command:

```
openssl version
```

1.3.5. Ensure the output of the “version” command shows a version 1.1.0g or above.

We are now ready to create our SSL certificates using OpenSSL under Ubuntu.

1.4. Microsoft Windows ™ Setup

Microsoft offers Windows Server ™, which many organizations use for certificate generation and management across their network(s). However, many small businesses may not have or require such an elaborate setup, and may benefit from use of OpenSSL under Microsoft Windows 7 or 10 in order to help safeguard Epson TM Intelligent printers on their private networks.

1.4.1. Users may download and install OpenSSL for Windows from one of the following links:
NOTE: These are utilities hosted by certain third party distributors. The user is free to acquire Windows OpenSSL installers from other distributors as desired. Epson is not responsible for the availability or accuracy of any utilities.

32-bit installer: https://slproweb.com/download/Win32OpenSSL_Light-1_1_1c.exe
64-bit installer: https://slproweb.com/download/Win64OpenSSL_Light-1_1_1c.exe
Distributor website: https://slproweb.com/products/Win32OpenSSL.html

1.4.2. Once installed, open a new Command Prompt (henceforth called terminal window) and navigate to the location where the OpenSSL binaries were installed:

```
cd “C:\Program Files\OpenSSL-Win64\bin”
```

1.4.3. Verify the version is 1.1.0g or newer:

```
openssl.exe version
```

We are now ready to create our SSL certificates using OpenSSL under Microsoft Windows.

2. Procedure

2.1. Generate a CA

Skip Step 1) through Step 3) if you already have a CA setup for your network/organization.

Step 1) Create a private key for the new CA by entering this command into a terminal window:

Ubuntu

```
openssl genrsa -aes256 -out myCA.key 2048
```

Windows

```
openssl.exe genrsa -aes256 -out myCA.key 2048
```

Step 2) Create the (root) CA certificate using the CA private key (signing public key)

Ubuntu

```
openssl req -x509 -new -nodes -key myCA.key -sha256 -days 1825 -out myCA.crt
```

Windows

```
openssl.exe req -x509 -new -nodes -key myCA.key -sha256 -days 1825 -out myCA.crt
```

Step 3) Install the CA certificate on all devices that will access the printer webpage. Operating systems have distinct methods of adding an internal Certificate Authority to the Trusted CA list; the following links can be used as general guidance for Windows, Mac OS and iOS.

Windows (7 and 10)
https://docs.microsoft.com/en-us/skype-sdk/sdn/articles/installing-the-trusted-root-certificate

Mac OS X (various versions)
https://www.digicert.com/ssl-support/p12-import-export-mac-mavericks-server.htm

iOS (11 and above)
- Download the CA certificate file (myCA.crt) to the iOS device (e.g. via an e-mail attachment)
- Click on the .crt file and then “Install” to add your CA certificate to the trusted list (Figure 2.1.a)

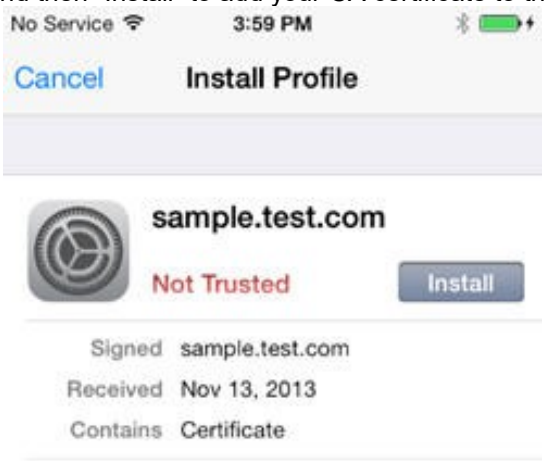

Figure 2.1.a: iOS prompt to add new CA certificate (iOS 10.3 and later)

- Additionally on iOS, you must fully trust the new CA certificate as follows:
https://support.apple.com/en-ca/HT204477

2.2. Generate Signed Printer Certificate (Multiple IP)

Step 4) In a terminal window, create a private key for the printer with the following command:

Ubuntu

```
openssl genrsa -aes256 -out printer.key 2048
```

Windows

```
openssl.exe genrsa -aes256 -out printer.key 2048
```

Step 5) Create a Certificate Signing Request for the printer using the printer's private key:

Ubuntu

```
openssl req -new -key printer.key -out printer.csr
```

Windows

```
openssl.exe -new -key printer.key -out printer.csr
```

Step 6) Create a configuration extension file for openssl to supply the SAN entries:

Ubuntu

```
echo > printer.ext
echo >>printer.ext authorityKeyIdentifier=keyid,issuer
echo >>printer.ext basicConstraints=CA:FALSE
echo >>printer.ext keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
echo >>printer.ext subjectAltName = @alt_names
echo >>printer.ext [alt_names]
echo -e "$ALT_NAMES" >> printer.ext
```

Where, ALT_NAMES is a variable containing all the SAN entries you wish to enlist. This can be multiple DNS entries, multiple IP entries, multiple e-mails, or a combination of the same. For example:

```
ALT_NAMES="
IP.1 = 192.168.192.168
IP.2 = 185.185.185.1
IP.3 = 185.185.185.2
..
..
IP.255 = 185.185.185.254
DNS.1 = labelprint.example.com"
```

Windows

Use your favourite text editor (such as Notepad) and build the extensions file for OpenSSL to supply the SAN entries. Save the file as printer.ext. Here is sample content for printer.ext:

File: printer.ext

```
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonrepudiation, keyEncipherment, dataEncipherment
subjectAltName = @alt_names
```

```
[alt_names]
IP.1 = 192.168.192.168
IP.2 = 185.185.185.1
IP.3 = 185.185.185.2
..
..
IP.255 = 185.185.185.254
DNS.1 = labelprint.example.com
```

Step 7) Create the printer certificate appropriately signed by the CA:

Ubuntu

```
openssl x509 -req -in printer.csr -CA myCA.crt -CAkey myCA.key -CAcreateserial -out printer.crt
-days 1825 -sha256 -extfile printer.ext
```

Windows

```
openssl.exe x509 -req -in printer.csr -CA myCA.crt -CAkey myCA.key -CAcreateserial -out printer.crt
-days 1825 -sha256 -extfile printer.ext
```

Step 8) Pack the printer private key and signed public certificate into a PKCS#12 file required by the printer; **make note of the password you set here**:

Ubuntu

```
openssl pkcs12 -export -out printer.pfx -inkey printer.key -in printer.crt
```

Windows

```
openssl.exe pkcs12 -export -out printer.pfx -inkey printer.key -in printer.crt
```

Step 9) Import the printer certificate (printer.pfx) using the "Password Protected PKCS#12 Certificate" option on the printer webpage; use the **SSL/TLS -> Certificate Import** link on the left-hand tab. Supply the password from Step 8) when adding the new certificate. See Figure 2.2.a for what this looks like on the Epson TM-T88VI printer.

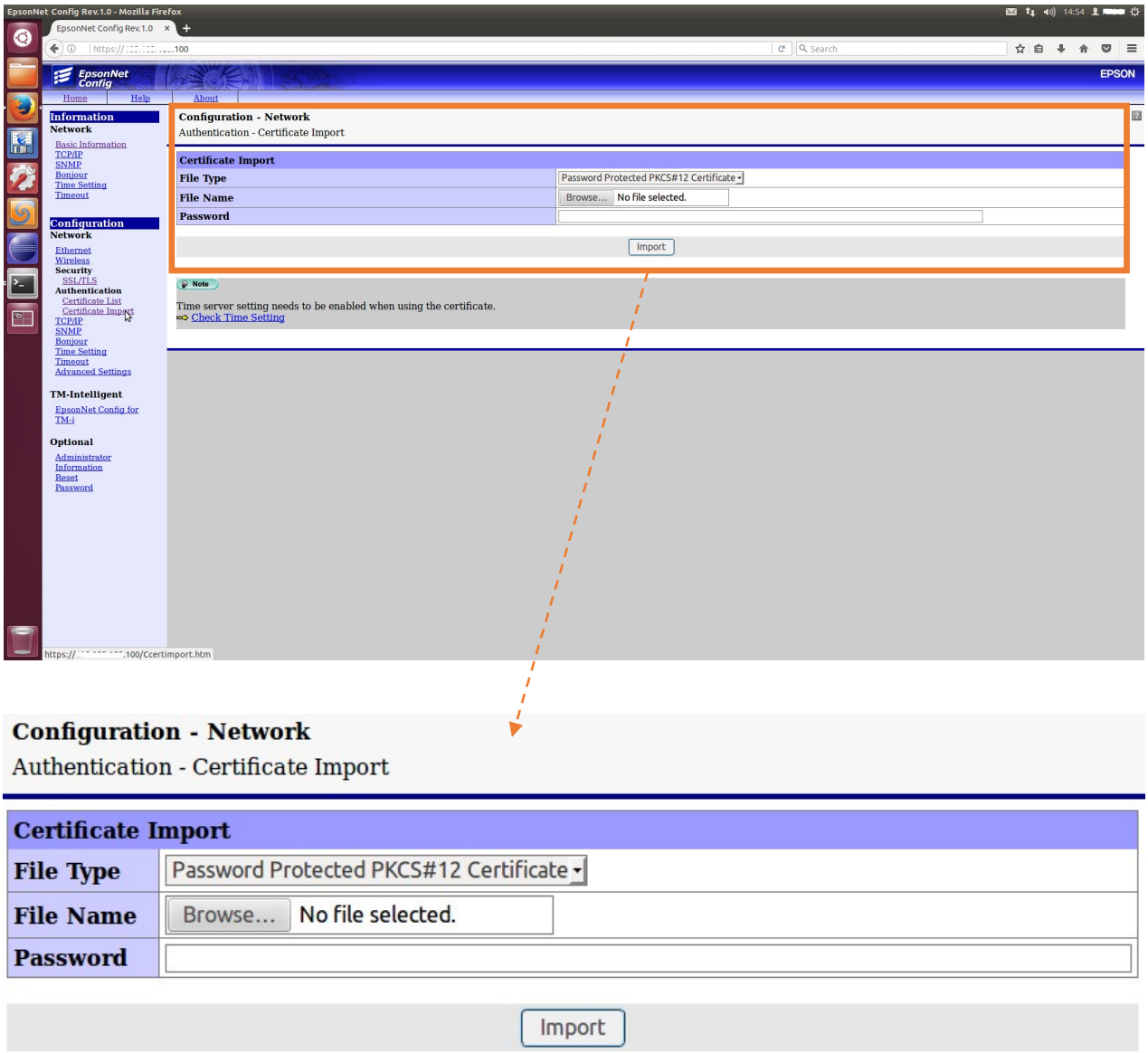

Figure 2.2.a: Certificate Import webpage on the Epson TM-T88VI

3. Limitations

3.1. Maximum Number of SAN Entries in Certificate

According to the RFC standard linked below, the number of SAN entries per certificate extension is defined by the set [1…MAX], where MAX is undefined and thus implementation specific.

https://tools.ietf.org/html/rfc5280

Consequently, different providers have established different limits on how many SAN entries their certificates can support. Among the limiting factors are the database size they must maintain and the maximum key size they support.

For example, consider the following three providers:

1. Microsoft: MAX is variable, limited by the internal (IIS) database entry size of 4 kB. This typically translates to about 200-250 entries of type: IP, or about 150 entries of type: dNS.
   See:
   https://social.technet.microsoft.com/wiki/contents/articles/3306.pki-faq-what-is-the-maximum-number-of-names-that-can-be-included-in-the-san-extension.aspx

2. SSL.com: MAX is 2000 entries per certificate.
   See: https://www.ssl.com/faqs/what-is-a-san-certificate/

3. LetsEncrypt.org: MAX is 100 entries per certificate.
   See: https://letsencrypt.org/docs/rate-limits/

Suggestion: *Epson Canada Ltd have successfully tried a single certificate using a 2048-bit private key and holding 256 distinct IP type entries, with the PKCS#12 file size amounting to under 4.5 kB, which is within the file size range of all "Intelligent" Epson TM products.*

4. Appendix

4.1. Check Supported Ciphers in OpenSSL

OpenSSL supports several ciphers, some of which may be deprecated by the time this document is being read. The reader is encouraged to explore all options as follows:

```
openssl ciphers -v
openssl enc -ciphers
```

To focus the list on currently relevant ciphers, try:

```
openssl ciphers -v
'ECDH+AESGCM:ECDH+CHACHA20:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS'
```

REVISION SHEET

Revision	Date	Author	Changes
1.00	2018-11-20	SL	- First draft
1.10	2019-05-23	SL	- Add Section 1.3: Ubuntu 18.04 Setup - Add certificate installation details to Section 2.1
1.10	2019-05-30	SL	- Add Section 1.4: Microsoft Windows ™ Setup - Add Windows instructions to Section 2: Procedure
1.20	2019-07-19	SL	- Recommend aes256 instead of des3 for private key generation