

Things that you need to know about Digital Certificates on z/OS

RACF User Group

March 15th 2017

Wai Choi, CISSP
IBM Corporation
RACF/PKI Development & Design
Poughkeepsie, NY

e-mail: wchoi@us.ibm.com



Agenda

- **Discussion on digital certificate set up for secure communication**
- **Overview of certificate / key ring utilities, Certificate Authority available on z/OS**

First encounter with digital certificate

- **Do you know you come across it every day?**
- **Do you ever take a look at it?**

1)The cert issued by the Certificate Authority vouches for amazon's identity
2)The cert is used in the process of encrypting the communication between your browser and the amazon site

The screenshot shows the Amazon.com homepage in a browser. The address bar displays 'https://www.amazon.com/' with a red circle around the 'https' and a lock icon. The page features a search bar, navigation links like 'Departments', 'Your Amazon.com', 'Today's Deals', 'Gift Cards & Registry', 'Sell', 'Help', 'Account', 'Try Prime', 'Lists', and 'Cart'. Promotional banners for 'AMAZON DEVICES' (featuring a Fire tablet for \$49.99) and 'echodot \$49.99' are visible. Below these are sections for 'Welcome' (with a 'Sign in securely' button), 'Holiday Gift Guides' (with sub-sections for Electronics and Toys), 'Amazon Echo and Alexa', and 'Amazon.com Gift Cards' (with a 'Shop now' link).

amazon's certificate and its issuer

The image displays two side-by-side screenshots of a Windows Certificate dialog box, illustrating the details of Amazon's certificate and its issuer.

Left Screenshot (Amazon's Certificate):

- General Tab:** Shows the certificate's details.
- Field/Value Table:**

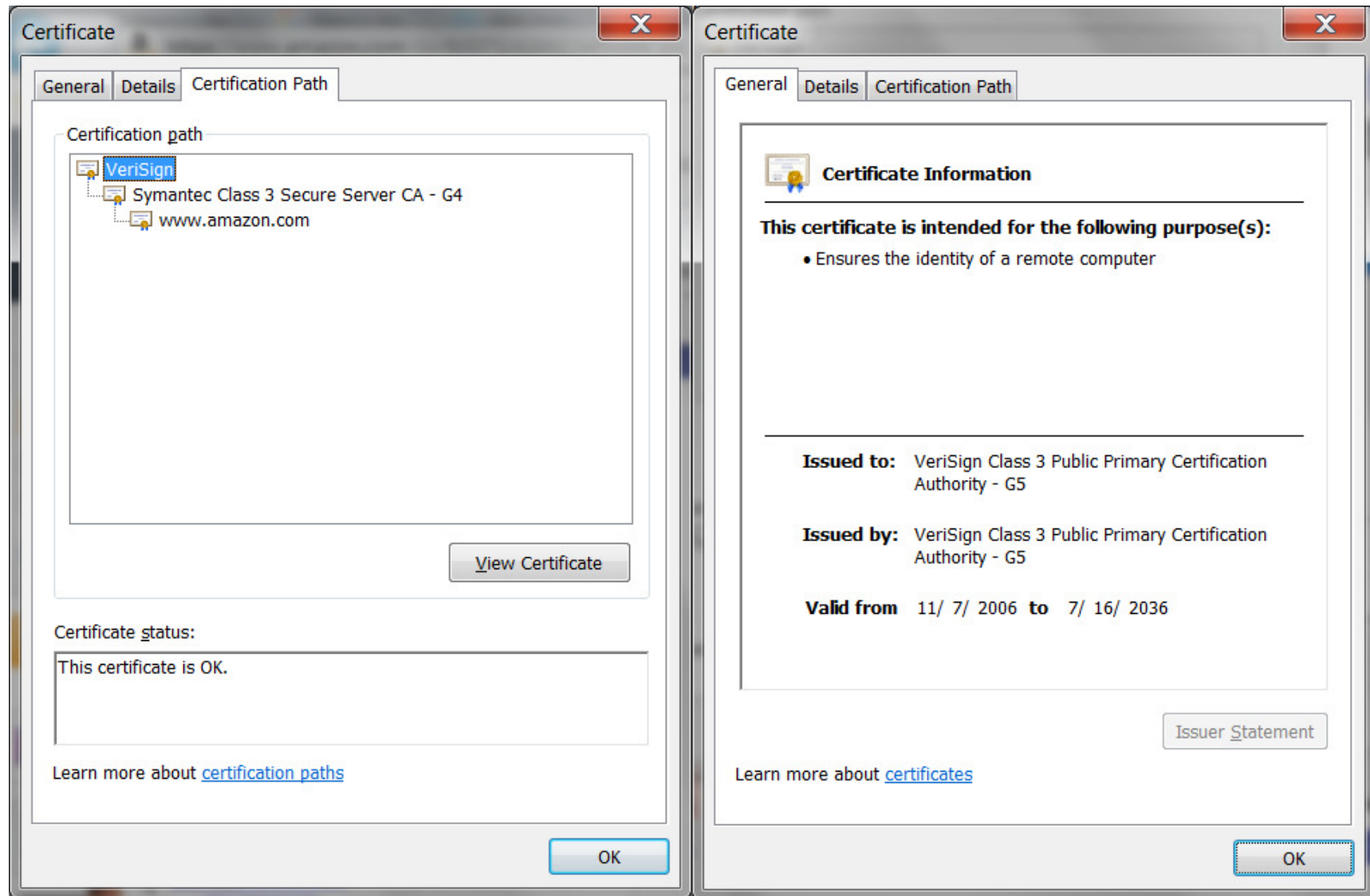
Field	Value
Version	V3
Serial number	7e 49 96 45 90 9b 4c 62 ...
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	Symantec Class 3 Secure...
Valid from	Tuesday, May 17, 2016 7...
Valid to	Friday, December 30, 20...
Subject	www.amazon.com, Amaz...
- Details Section:**
 - CN = www.amazon.com
 - O = Amazon.com, Inc.
 - L = Seattle
 - S = Washington
 - C = US
- Callout:** A blue box with the text "The name matches that in the URL" points to the CN field.

Right Screenshot (Certificate Issuer):

- Field/Value Table:**

Field	Value
Version	V3
Serial number	7e 49 96 45 90 9b 4c 62 ...
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	Symantec Class 3 Secure...
Valid from	Tuesday, May 17, 2016 7...
Valid to	Friday, December 30, 20...
Subject	www.amazon.com, Amaz...
- Details Section:**
 - CN = Symantec Class 3 Secure Server CA - G4
 - OU = Symantec Trust Network
 - O = Symantec Corporation
 - C = US

Certificate chain and the root CA certificate



What's inside a Certificate?

Certificate Info

- version
- serial number
- signature algorithm ID
- issuer's name
- validity period
- subject's name
- subject's public key
- extensions

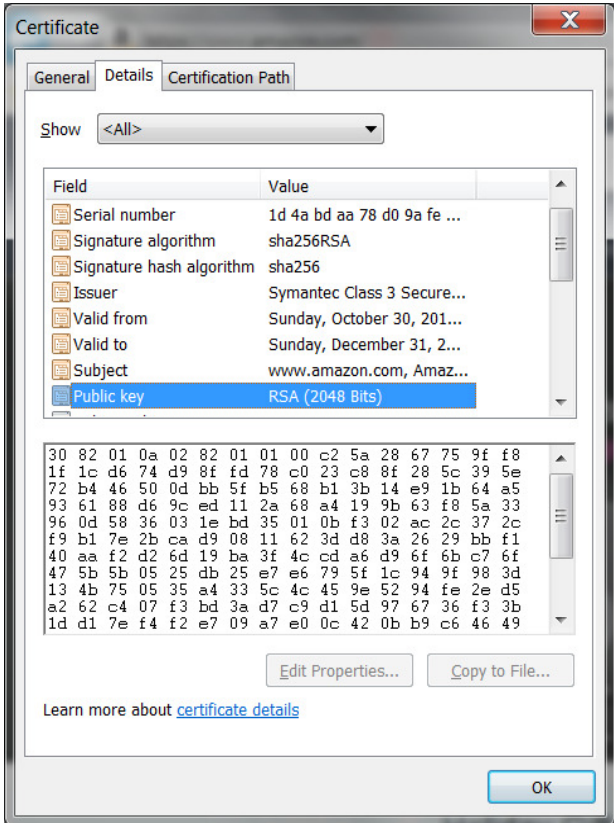
Certificate Signature

This is the hash/encrypt algorithm used in the signature, eg. sha256RSA

The certificate binds a public key to a subject

CA signs the above cert info by encrypting the hash with its **private** key

The private key is NOT in the certificate. It is kept in a key store



You can NOT change ANY of the certificate information!

Certificate Formats

- **X.509 certificates can be packaged differently**
 - Single certificate
 - PKCS#7 certificate package
 - Contains end entity certificate and its issuer(s)
 - PKCS#12 certificate package
 - Similar to PKCS#7, but also contains the private key associated with the end-entity certificate.
 - Packaged protected by a password
- **Package can be in binary or Base64 encoded format (containing Aa-Zz,0-9,/,+ (= is for padding) for easy cut and paste)**
- **Transfer the file in binary or text accordingly**
- **B64 format certificate helps the distribution of internal CA certificates**

-----BEGIN CERTIFICATE-----

```
MIICPTCCAaagAwIBAgIIR49S4QANLvEwDQYJKoZIhvcNAQEFBQAwNzELMAkGA1UE
BhMCVVMxDTALBgNVBAoTBFRlc3QxGTAXBgNVBAMMEFRlc3Rfc2VsZ19zaWduZWQw
HhcNMDgwMTE3MTMwNjQxWhcNMDkwMTE2MTMwNjQxWjA3MQswCQYDVOQGEWJVUzEN
MAsGA1UEChMEVGVzZdEzZmBcGA1UEAwQVGVzZdP9zZWxmX3NpZ251ZDCBnzANBgkq
hkiG9w0BAQEFAAOBjQAwgYkCgYEA9tK0v5gLaceozMfMeVd891fCjBVoR+dpzhwK
R2B/QcQYBGLfgS4YM/wGSh6YrmVyg00VxocriySbcxRuBayw3pE4/3JI2myINmLp
bFIdPCnqk/qvFK+1N+nrEnBK9y1s7NmxDIuQQF'sX/o/DpoxwzXf+JbWDwirQR
NyLiTGMCaWAAaNSMFAwHQYDVR0OBBYEFwDFLjOUcRa62BV53jVYHewuOWEMB8G
A1UdIwQYMBaAFAwDFLjOUcRa62BV53jVYHewuOWEMA4GA1UdDwEB/wQEAwIE8DAN
BgkqhkiG9w0BAQUFAAOBgQAC5sW1f3EdE0k9zc8wKNT1sczWkQBrVy4Rdr17ERqN
D2OfkBjQuXiNwN18pF6WPWFYg80MNwhP4oJSVePnzElh4Wzi2w1/zI8rINSW7px3
w16lz+8jE184q/N0q0tOPTAtEb6fIzwjkLttctt3oF+IjunvE5QoRsXRJbbTMD/BG
jw==
```

-----END CERTIFICATE-----

Behind the scene - handshake process

- You visit the amazon site to buy something
- **https** in the URL indicates you are communicating under a secure protocol - your browser sends a set of proposed algorithms that needs for **encrypting** the subsequent communication
- Two parties are involved:
 - Amazon server (server)
 - send a certificate to identify itself to your browser – the certificate's subject name matches that in the URL you entered (www.amazon.com)
 - send a set of algorithms that are matching with the proposed list
 - Your browser (client)
 - validate amazon's certificate and decides whether to trust it – *more details in the next slide*
 - generate a **session key** using the chosen algorithm
 - this key is wrapped by making use of the amazon's certificate and send to the server – *more details in the following slides*

Behind the scene - handshake process

- These steps are referred as the **handshake** process in the **SSL/TLS** (Secure Sockets Layer / Transport Layer Security) protocol
- Once the secure session is established, all the information you entered, like your credit card number, will be encrypted using the session key before sending to amazon
- This is an example of SSL/TLS **server authentication** (one way) – only the server needs to identify itself for the client to verify

Behind the scene - certificate verification

- **Which side performs checking?**

- Client (your browser)

- **Validation checks**

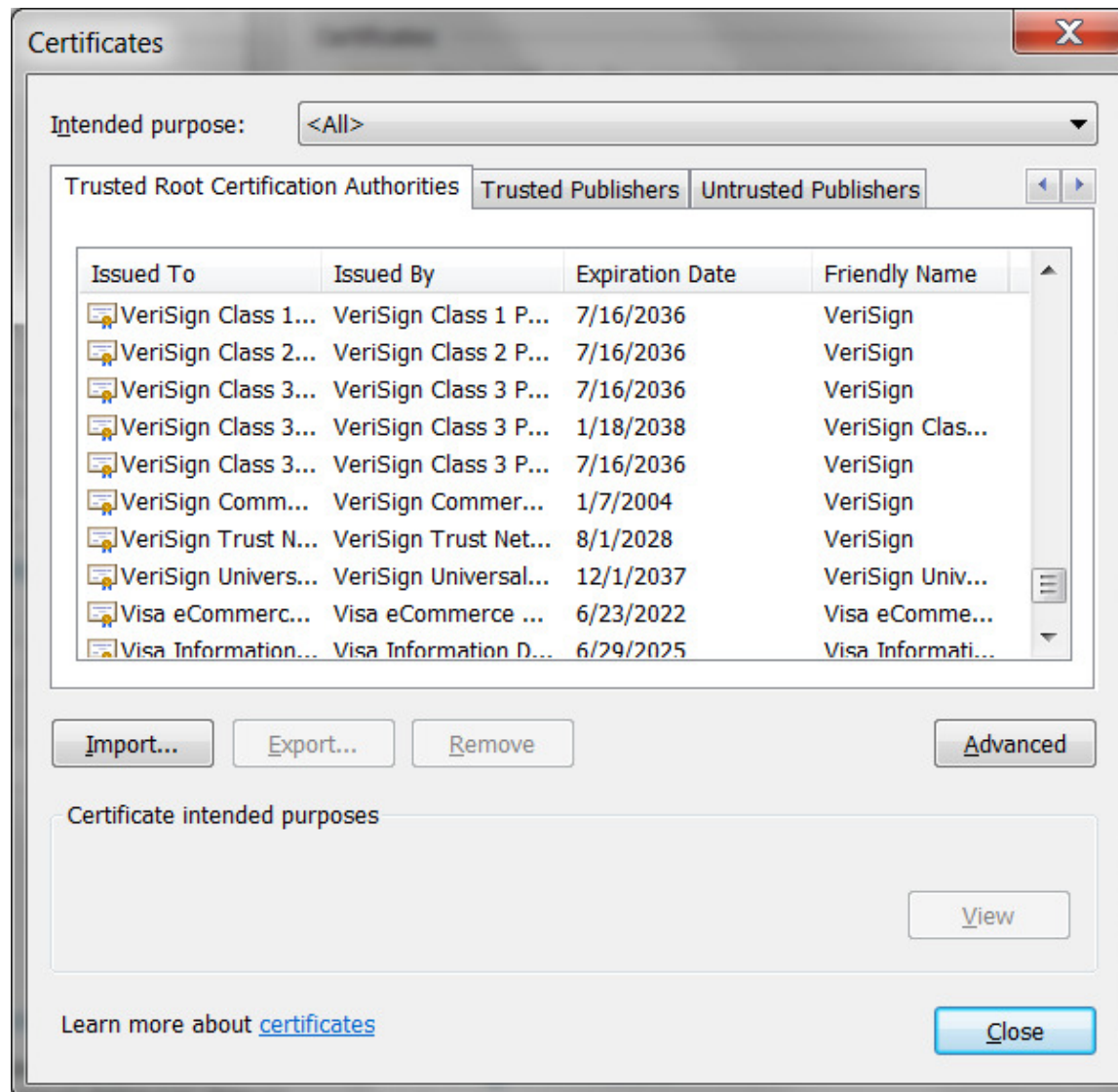
- Check the certificate's integrity by verifying the signature on the certificate – is it really issued by the CA it claims?
- Check if the certificate is expired by verifying the expiration date on the certificate
- Check if the certificate has been revoked – the issuer provides the revocation status through Certificate Revocation List(CRL) or Online Certificate Status Protocol(OCSP)

Note: The validation checks apply to the issuer certificate(s) too. All the certificates have to pass these checks

- **Trust check** - check if the root CA certificate is trusted

- Is the root CA certificate of the Amazon certificate in the Trust Root Certification Authorities in your browser?

Browser's certificate store - Trusted Root Certificate Authorities



Trust or not?

■ Who makes the decision to put a CA certificate in the browser's trusted root store

1) The application owner of your browser – Microsoft, Firefox, Google...

- The browser preloads a set of 'well known' CA certs when you first install it
- You may check to see what are the processes involved before the company decided to accept a CA in its trust store
- Each browser company may have different sets of rules to accept the CAs
- Some CAs charge a lot to issue a certificate, some are free.
- Usually the CA that charges more performs more thorough background check and validation on the requestor and provides warranty coverage on damage caused by the CA's negligence
 - **DV certificate** – Domain validation, just need to prove you are the owner of a domain. Usually free.
 - **OV certificate** – Organization validation, simple vetting through customer contact using reliable third party data. Less expensive.
 - **EV certificate** – Extended validation, extensive vetting using government registries. More expensive
- You trust the company to make the decision for you

How about an internal CA?

- **Who makes the decision to put a CA certificate in the browser's trusted root store**

2) Yourself

- You may put a CA that you know in the trust store if you know you will be contacting the server whose certificate was issued by that CA
- It is the server's responsibility to tell you what the root CA it used in the issuer(s)' chain for its server certificate (The server can skip this step if it chose a well known CA)
- It is your responsibility to decide if you want to trust that root CA (The client can skip this step if the server's root CA is a well known CA since the browser decided for you)

Certificate issued by an internal root CA

As long as you put this CA cert in your trusted root CA store, all the certificates issued by it are validated the same way as those issued by those 'well known' CAs.

Field	Value
Version	V3
Serial number	01
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	Sharb01 CA, Test, The S...
Valid from	Tuesday, October 04, 2011
Valid to	Wednesday, October 12, 2011
Subject	firstreq, Class 1 Inter...

CN = firstreq
OU = Class 1 Internet Certificate CA
O = The Sharb01 Firm
C = US

Certification path

- Demo Customer Design Centre Certificate Authority
 - Sharb01 CA
 - firstreq

Certificate status:
This certificate is OK.

How to distribute the internal root
certificate?
Let's see how the government do it...

The screenshot shows a web browser window with the URL <https://www.cnss.gov/cnss/>. A red error message in the address bar reads "Certificate error". The browser's address bar also contains the text "Red color on the untrusted URL". The website header includes the CNSS logo and navigation links: ABOUT, LIBRARY, HELP, LOGIN YOUR ACCOUNT, and SEARCH. A help menu is open over the LIBRARY link, listing: Login Help, Certificates, [Certificate Errors](#), Terms of Use, LOOKING FOR SOMETHING ELSE?, Search the Library, Site Map, and Contact Us. The main content area features a large graphic with the text "System Protecte" and the CNSS logo. A blue banner at the bottom of the page reads "Meeting Current and Future Threats". The address bar at the bottom of the browser shows <https://www.cnss.gov/CNSS/help/access.cfm>.

Committee on National Security Systems

Meeting Current and Future Threats

https://www.cnss.gov/CNSS/help/access.cfm

BU Guidance - Security - C... successfactors IBM Connections IBM CIO Business Transfor... IBM Standard Software In... IT Help Central IBM VIRUS Computer Eme... IBM Software RFE Commu...

ARE YOU GETTING SITE CERTIFICATE ACCESS ERRORS?

Note: The IAD.Gov website uses TLS 1.2, supported by a Department of Defense (DoD) PKI certificate, to ensure confidentiality and integrity for all users. IAD.Gov website users will need to have the current DoD Root and Intermediate Certificate Authorities (CA) loaded into their browsers to avoid receiving untrusted web site notifications. Instructions for these processes are provided on this page.

Portions of this web site use SSL protection to help secure our content. Access to these areas require that a site security certificate is loaded into your browser. Other areas can only be accessed if you have a Public Key Infrastructure (PKI), Personal Identity Verification (PIV) or Common Access Cards (CAC) correctly installed in your browser. Access to this site requires both your personal certificate and site security certificate. There are two ways to avoid site certificate error messages:

1. **Import a DoD Root CA Certificate (preferred).**
2. Add an exception for the web site (Mozilla Firefox only) or create a Trusted Site (IE only).

While adding an exception is the faster, easier process, you might have to repeat the process for multiple protected DoD web sites. Importing the DoD Root CA Certificate will take a few minutes, but it is the more thorough solution. You should only have to import it once per browser.

You may see some other messages, usually alerts, rather than error messages, even when everything is installed correctly.

3. Other common error messages.

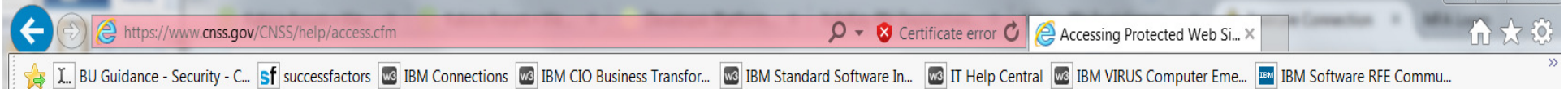
1. Import a DoD Root CA Certificate

Web site users will need to have the current DoD Root and Intermediate Certificate Authorities (CA) loaded into their browsers to avoid receiving untrusted web site notifications.

Please visit the [Information Assurance Support Environment \(IASE\) Tools](#) page to download the DoD Root and Intermediate CA Certificates. Select the "Trust Store" tab; you will find the InstallRoot 4.1: NIPR Windows Installer in this section. A user guide for InstallRoot 4.1 is available on the same page.

If you have any questions, please contact the IAD Client Contact Center at 410-854-4200 or IAD_CCC@nsa.gov.

Link to another site to download the DoD root and Intermediate CA certificates



2. Add Exception/Create Trusted Site

- **Add an Exception (Mozilla Firefox only)**

If you receive a Secure Connection Failed message in Mozilla Firefox, you have the option of simply adding an Exception, thereby making it a Trusted Site. To do so, complete the following steps:

1. On the error window, click **Or you can add an exception**; the page reloads.
2. Click **Add Exception**; the **Add Security Exception** window opens.
3. Click **Get Certificate**; the window reloads.
4. Check the **Permanently store this exception** box; then click **Confirm Security Exception**.

- **Create a Trusted Site (IE only)**

In IE, you may receive an error message stating that there is a problem with this website's security certificate. You have the option of making it this site a Trusted Site. To do so, complete the following steps:

1. Go to **Tools > Internet Options**.
2. Select the **Security** tab.
3. Click **Trusted Sites**.
4. To create a Trusted Site, click **Sites**; the **Trusted Sites** window opens.
5. Enter the URL of the desired site.
6. Click **Add**. The site is listed in the **Trusted Sites** box.
7. Check **Require server verification (https:) for all sites in this zone**.

3. Other Common Error Messages

- **Switching from HTTP to HTTPS Pages with IE**

If you enter the site URL starting with http, instead of https, or if the page you're coming from had a URL starting with http and the link to the secure site was coded with a relative link, **you may see a security warning. Select Yes** to proceed.

http://iase.disa.mil/pki-pke/Pages/tools.aspx

Not HTTPS – means no identity verification performed

i
IASE Information Assurance Support Environment

All Sites

Home Cybersecurity Training Topic Map STIGs Tools News Help RSS Feeds

Home > PKI-PKE > Tools

PKI and PKE Tools

*PKI = DoD PKI Certificate Required

Domain Management

Description

Password Hash Refresh Script *PKI

The DoD PKE Password Hash Refresh script can be used to periodically change passwords (and by extension, their associated hashes) for smart card-enforced accounts within specific OU containers and Groups in Microsoft Active Directory (AD). (ZIP Download) Size: 2 KB

Password Hash Refresh Script: User Guide *PKI

This guide provides step-by-step instructions for using the DoD PKE Password Hash Refresh script to periodically change passwords (and by extension, their associated hashes) for smart card enforced accounts. (PDF Download) Date: 02/11/2014 | Size: 686 KB

Smart Card Logon (SCL) Troubleshooting Tool 1.0 *PKI

The SCL troubleshooting Tool is designed to identify and diagnose SCL problems that are present on an Active Directory domain controller. The following operating systems are supported: Windows Server 2008, 2008 R2, 2012, and 2012 R2 . (MSI Download) Date: 02/26/2016 | Size: 14,161 KB

Smart Card Logon (SCL) Troubleshooting Tool 1.0 User Guide *PKI

This guide provides usage instructions for the Smart Card Logon (SCL) Troubleshooting Tool. (PDF Download) Date: 02/26/2016 | Size: 605 KB

Certificate Tools

PKI-PKE

- PKI-PKE Home
- + Getting Started
- + End Users
- PKE A-Z
- For Administrators, Integrators & Developers
- Tools
- SIPRNet PKI
- + Mobile Devices
- Interoperability
- For RAs, LRAs, KRAs & TAs
- Newsletters
- External Certification Authorities (ECA) Customer Support
- Policies
- SHA-256 Coordination
- Conferences
- Initiatives
- About
- Contact Us

This guide provides installation and usage instructions for the NSSdb CertLoader script for Linux environments. (PDF Download) Date: 07/09/2015 | Size: 333 KB

NSSdb CertLoader for Windows *PKI

This script facilitates population of trusted Certification Authority (CA) certificates in an NSS database on Windows operating systems. The script extracts all certificates from a specified PKCS#7 file, converts them to PEM format as necessary, then loads them into a specified NSS database. (ZIP Download) Size: 2 KB

NSSdb CertLoader for Windows User Manual *PKI

This guide provides installation and usage instructions for the NSSdb CertLoader script for Windows environments. (PDF Download) Date: 07/09/2015 | Size: 331 KB

PKI CA Certificate Bundles: PEM Self-Extracting ZIP

These signed self-extracting zip files contain all the Certification Authority (CA) certificates for the specified PKI in PEM format. Instructions for verifying the digital signatures on the files can be found in the Verifying Digital Signatures on DoD PKE Tools guide. Designed to be run on Microsoft Windows

- For DoD PKI Only - Version 5.0 - (EXE Download) Size: 190 KB
- For ECA PKI Only - Version 5.0.1 - (EXE Download) Size: 175 KB
- For JITC PKI Only - Version 5.0.1 - (EXE Download) Size: 204 KB
- For SIPR PKI Only - Version 5.0.1 *Download available on SIPRNet Only

PKI CA Certificate Bundles: PKCS#7

These zip files contain three PKCS#7 files that contain all the Certification Authority (CA) certificates for the specified PKI in different formats. One PKCS#7 file contains the certificates in DER format, another in PEM, and the last also in PEM but with a signature applied to the PKCS#7 file. Instructions for verifying the integrity of all three files using OpenSSL are included in the README

- For DoD PKI Only - Version 5.0 - (ZIP Download) Size: 214 KB
- For ECA PKI Only - Version 5.0.1 - (ZIP Download) Size: 83 KB
- For JITC PKI Only - Version 5.0.1 - (ZIP Download) Size: 346 KB
- For SIPR PKI Only - Version 5.0.1 *Download available on SIPRNet Only

Download DoD package as instructed



Content of the download package:

README.txt

DoD_Root_CA_4_0x01_DoD_Root_CA_4.cer

DoD_Root_CA_3_0x01_DoD_Root_CA_3.cer

DoD_Root_CA_2_0x05_DoD_Root_CA_2.cer

DoD_PKE_CA_chain.pem

Certificates_PKCS7_v5.0u1_DoD_OSX_CAsOnly.der.p7b

Certificates_PKCS7_v5.0u1_DoD_DoDRootCA4_withCAs_FirefoxChromeOS.der.p7b

Certificates_PKCS7_v5.0u1_DoD_DoDRootCA3_withCAs_FirefoxChromeOS.der.p7b

Certificates_PKCS7_v5.0u1_DoD_DoDRootCA2_withCAs_FirefoxChromeOS.der.p7b

Certificates_PKCS7_v5.0u1_DoD.sha256

Certificates_PKCS7_v5.0u1_DoD.pem.p7b

Certificates_PKCS7_v5.0u1_DoD.der.p7b

README.TXT

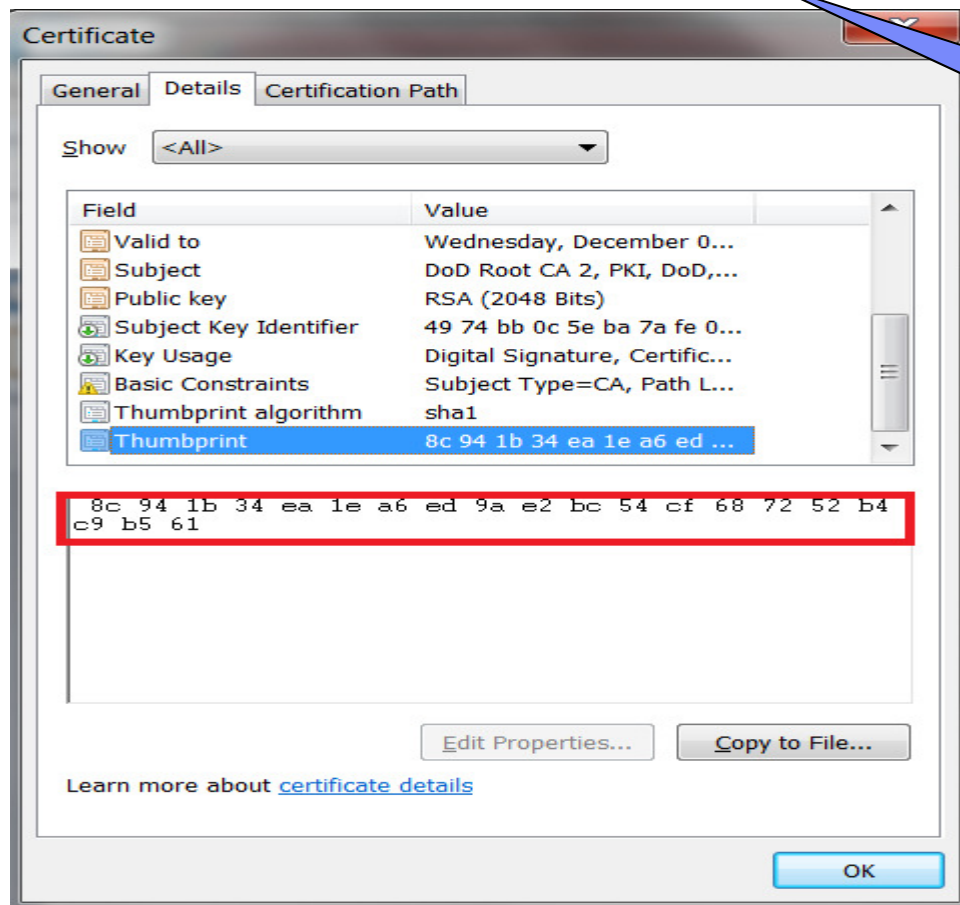
...

Verify the following output:

```
subject=/C=US/O=U.S. Government/OU=DoD/OU=PKI/CN=DoD Root CA 2  
issuer= /C=US/O=U.S. Government/OU=DoD/OU=PKI/CN=DoD Root CA 2  
SHA1 Fingerprint=
```

```
8C:94:1B:34:EA:1E:A6:ED:9A:E2:BC:54:CF:68:72:52:B4:C9:B5:61
```

Confirm output and verify the DoD Root CA 2 SHA1 Fingerprint by calling the DoD PKI at (844) 347-2457 or DSN 850-0032.



Call to confirm? How do you know this number is trust worthy?

A server wants to establish a secure session with a client using server authentication.

What are the steps?

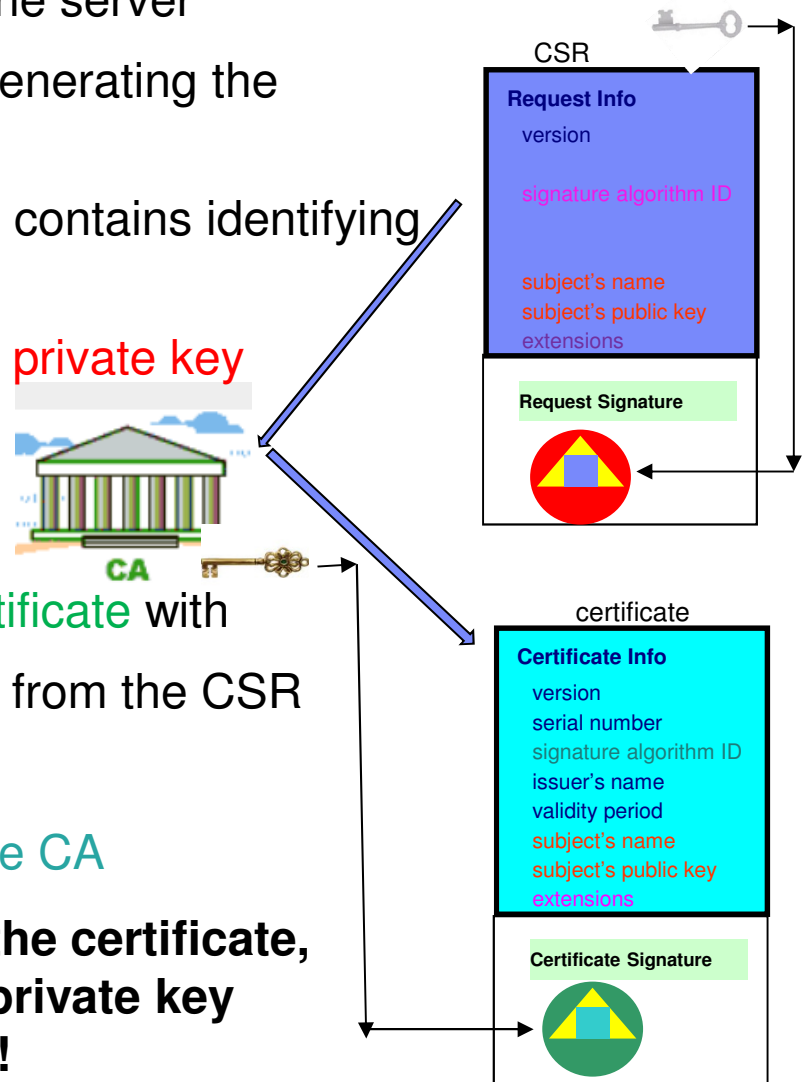
- 1) Get a certificate
- 2) Set up a certificate store / key ring and put the certificate there

Step 1: Get a certificate

- The server needs to obtain a certificate to identify itself. There are different options:
 - a) Use utilities from z/OS or other platforms – RACF RACDCERT or System SSL gskkyman, openssl
 - simple, but they do not provide any revocation status on the certificate
 - b) Buy one from some commercial CAs
 - costs money
 - c) Get a free one from free CAs
 - May not be accepted by some clients
 - > Especially client on z/OS – don't want to open up the communication with numerous servers that were issued without going through some identity checking
 - d) Request one from some internal CA, eg. z/OS PKI Services
 - needs set up. But if a large number of certificates are needed, it pays back.
 - control what certificates to be issued

Key pair ->CSR->certificate

- Need to have a public private **key pair** first for the server
 - The key pair is generated in the process of generating the certificate signing request (**CSR**)
 - The public key is put on the CSR, which also contains identifying information for the server
 - CSR is signed by the **server's corresponding private key**
 - The private key put in a safe place!!!
- The CSR is sent to the CA
- After the CA validates the CSR, it returns a **certificate** with
 - the public key and the identifying information from the CSR
 - other content that the CA decides
 - the signature created by the **private key of the CA**



Note: Secure the private key associated with the certificate, especially the CA's. Compromise of the CA's private key invalidates ALL the certificates it has issued!!!

Step 2: Set up a certificate store

- Certificate must be placed in a certificate store / key ring/ key database before it can be used by an application to perform identification and validation
- The server set up a certificate store /key ring / key database with these certificates (assuming the CA is a root cert):
 - the server certificate
 - the CA certificate
- The server sends the CA certificate to the client
- The client sets up a key ring / key database / certificate store with this certificate:
 - the CA certificate

Server certificate store

- FTP Server cert 
- CA cert that signed FTP server cert

- CA cert that signed FTP server cert

Client certificate store


Note: If the certificate store is managed by RACF, need not to create a real key ring to store the CA certificate, use a virtual key ring under certificate authority's predefined ID,ie CERTAUTH

If there are intermediate CAs in the chain...

- Root CA
 - Intermediate CA1
 - Intermediate CA2
 - Server
- Putting the intermediate CA in the client's store may cause validation failure if the intermediate CA has changed

Contains the server cert and the whole CA chain

Server certificate store

- FTP Server cert 
- Intermediate CA2
- Intermediate CA1
- Root CA of the server

- Root CA of the server


Client certificate store

Contains only the root CA

Why intermediate CA reissued

- SHA1 -> SHA2
 - 1024 ->2048
- (steps shown later)

Server certificate store before
intermediate CAs change

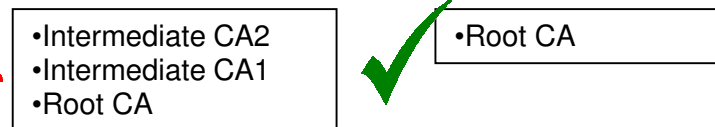
•FTP Server cert 
•Intermediate CA2
•Intermediate CA1
•Root CA

Server certificate store after
intermediate CAs change

•FTP Server cert' 
•Intermediate CA2'
•Intermediate CA1'
•Root CA



Client certificate store



Client certificate store

Details on these 2 steps

Generate certificates and create key rings using z/OS utilities

- **RACF RACDCERT**
 - create request, create certificate including self-signed one
 - manage certificates stored in a RACF key ring
 - specified with <ring owner>/<ring name>, eg. FTPID/ftpRing
 - key ring can be created before or after the certificates have been obtained
 - key rings are protected by RACF resource profiles in FACILITY or RDATA LIB class
 - RDATA LIB class provides granular control on individual key ring access

Generate certificates and create key rings using z/OS utilities

- System SSL gskkyman
 - a Unix based utility shipped as part of the System SSL product in the z/OS Cryptographic Services Element
 - create request, create only self-signed certificate
 - manage certificates stored in a key database file
 - specified with /<full directory path>/<db file name>, eg /etc/certStore/ftp.kdb
 - key database must be created before the operation on certificates
 - protected by the file system's permission bits and password

Eg - Renewing a CA and server Certificate with a new key Pair with SHA2 signing algorithm

Steps:

1) Create a new certificate based on the original certificate:

```
RACDCERT CERTAUTH REKEY(LABEL('original CA  
cert'))WITHLABEL('new cert')SIZE... NOTAFTER...
```

2) Generate a request based on the new certificate and put in a dataset 'req_dsn'

```
RACDCERT CERTAUTH GENREQ(LABEL('new CA cert'))  
DSN(req_dsn)
```

3) Send the request to the original certificate CA

4) The CA signed the new certificate with SHA256RSA signing algorithm

5) After receive the new certificate and save it in a dataset 'cert_dsn', add it back under the same ID:

```
RACDCERT CERTAUTH ADD(cert_dsn)
```

Eg - Renewing a CA Certificate with a new key Pair with SHA2 signing algorithm

Steps:

6) Perform step 1- 5 on the server certificate, replace the CERTAUTH in the command with the server ID

7) When ready, start to use this new CA certificate and new server certificate

```
RACDCERT CERTAUTH ROLLOVER(LABEL('original CA cert'))NEWLABEL('new CA cert')
```

```
RACDCERT ID(<server ID>) ROLLOVER(LABEL('original server cert'))NEWLABEL('new server cert')
```

8) Restart the server!!!

Missing an important CA function

- RACDCERT and gskkyman do not have all the functions of a real Certificate Authority
 - No revocation status not provided

Certificate Authority on z/OS

- **z/OS PKI Services** provides full certificate life cycle management
 - Included in z/OS base - 'free'
 - Request, create, renew, revoke certificate
 - Provide certificate status through Certificate Revocation List(CRL) and Online Certificate Status Protocol (OCSP)
 - Generation and administration of certificates via customizable web pages
 - Support Simple Certificate Enrollment Protocol (SCEP) for routers to request certificates automatically
 - Automatic notification or renewal of expiring certificates
- Cannot generate its own certificate to start
 - Use RACDCERT to generate or buy externally
- Cannot generate key ring / certificate store

Using z/OS PKI Services web pages

Cryptographic Services PKI Services Guide and Reference

<http://publibz.boulder.ibm.com/epubs/pdf/iky2a110.pdf>

User requests server certificate

PKI Services Certificate Generation Application

[Install the CA certificate to enable SSL sessions for PKI Services](#)



Choose one of the following:

- Request a new certificate using a model

Select the certificate template to use as a model

Select the certificate template to use as a model

5-Year PKI SSL Server Certificate
1-Year PKI SSL Browser Certificate
1-Year PKI S/MIME Browser Certificate
2-Year PKI Windows Logon Certificate
2-Year PKI Browser Certificate For Authenticating To z/OS
5-Year PKI SSL Server Certificate
5-Year PKI IPSEC Server (Firewall) Certificate
5-Year PKI Intermediate CA Certificate
2-Year PKI Authenticode - Code Signing Certificate
5-Year SCEP Certificate - Preregistration
1-Year PKI Generated Key Certificate
n-Year PKI Certificate for Extensions Demonstration

- Pick up a previously requested certificate

Enter the assigned transaction ID

Select the certificate return type

- Renew or revoke a previously issued browser certificate

- Recover a previously issued certificate whose key was generated by PKI Services

Enter the email address when the original certificate was requested

Enter the same pass phrase as on the request form

- Administrators click here

Here's your Certificate. Cut and paste it to a file

```

-----BEGIN CERTIFICATE-----
MIIGHwYJKoZIhvcNAQcCoIIGeDCCBnQCAQEXADALBgkqhkiG9w0BBwGgggZcMIID
9TCCA16gAwIBAgIBBDANBgkqhkiG9w0BAQUFADAYMQswCQYDVQQGEwJVUzEMMAoG
A1UEChMDSUJNMRUwEwYDVQQLExIUIBDZXJOIEF1dGgWYHcNMDQxMDA2MDAwMDAw
WhcNMDkxMDA0MjM1OTU5WjBQMQRwCQYDVQQGEwJVUzERMA8GA1UECBMTM3IF1v
cmsxFTATBgNVBAAoTDES1dyBz3JrIFJVZEXMBUGA1UEAxMOU1VHIFd1YiBTZXJ2
ZXIwZ28wDQYJKoZIhvcNAQEBBQADgYOAMIGJAoGBAJQLBDRIAAdlhNFYyQE/MOZ9S
eF+8zLv4AD6MyN1IP/Tr+Ij3T6c9mNYUB7fWqSpAIfmPt8W6KWLROMb3lHVuYYtB
oGaQ/FprcnHEkvP5QbOrvbxqfQZnrA1N4kGisGibGv6evZ1fLAHpOJNLAAJfC2/h
EbB0sdQ4RL8VCFzrSo2BAgMBAAGjggH7MIIB9zApBgNVHREEIjAghhhodHRwOi8v
d3d3LnJ1Z3N1cnZlc15jb22HBA17LUMwDgYDVROPAQH/BAQDAgWgMBMGA1UdJQQM
MAoGCCsGAQUFBwMBMIIBYwYDVROFBIIBWjCCAVYwSaBHoEWkQzBBMQswCQYDVQQG
DAJVUzEMMAoGA1UECgwDSUJNMRUwEwYDVQQLDAxIUIBDZXJOIEF1dGgWYHcNMDQx
BAMMBENSTDEwXaBboFmGV2xkYXA6Ly85LjU2LjU0LjEzMDozODkxMm5kZW50LjEzMDoz
VT11UyYUyMEN1cnZlc15jb22HBA17LUMwDgYDVROFBIIBWjCCAVYwSaBHoEWkQzBBMQsw
aW9uTG1zdDBxOG+gbyZrbGRhcDovL215b3RoZXJ1dGgWYHcNMDQxMm5kZW50LjEzMDoz
eS5jb206MzG5LONOPUNSTDES1U9SF11MjBDZXJOJTIwQXV0aCxpPpU1CTSxDpVVT
P2N1cnRpZm1jYXR1UmV2b2NhdGl1bGkxc3QwN6A1oDOGmWhOdHA6Ly93d3cubX1j
b21wYU55LmNvbS9QS01TZXJ2L2NhY2VydHMvQ1JMM55jcmwwHQYDVROOBYYEFFp6
TKC8zJOGnu/1vjWmjxq/S2+NMB8GA1UdIwQYMBaAFldu6pMUI9gIBAPXMeK3zu1Z
M+arMAOGCSqGSIb3DQEBAQA4GBADpjj6bl0eBL+z2GQmd9EQGXyP5zrPyoALIJ8
LP3ugJ5sI1R55mtNsUm358JzYwtT/46uP6zmDnn3hxAt6cwM1UWHNPkzIQHfx+O2
1SL/fX/5u8QCFhr8E7a18Z+Aeppcoi6/YxHfH1+5qIcMv5/oeKbH28foxSNw1Rb
n/tKWewmMIICXzCCAciGAWIBAgIBADANBgkqhkiG9w0BAQUFADAYMQswCQYDVQQGE
wJVUzEMMAoGA1UEChMDSUJNMRUwEwYDVQQLExIUIBDZXJOIEF1dGgWYHcNMDQx
MDA0MDQwMDAwWhcNMDQxMDA0MjM1OTU5WjBQMQRwCQYDVQQGEwJVUzEMMAoGA1UE
ChMDSUJNMRUwEwYDVQQLExIUIBDZXJOIEF1dGgWYHcNMDQxMDA0MDQwMDAwWhcN
MDQxMDA0MjM1OTU5WjBQMQRwCQYDVQQGEwJVUzEMMAoGA1UECBMTM3IF1v
gYOAMIGJAoGBALAbZJJN/FEu/VDi+mRnuJzpwK16V4ATqNHztjuEMbdz13rtIpaR
OqIh61atRRsddACuH4vkxaNxg/WHodzf/kkndHmRh1Ew1IwRLCEfU3LaiBgSURO
QiPhwV61cQUHSTW+uxnXJq56OKQAo4weiFr+GRm6ISA3i1/Yt4oIeIDAgMBAAGj
gYQwgYEWpWYJYIZIAyB4QgENBDITMEdlbmVyyXR1ZCBieSB0aGUgU2VjdXJpdHkg
U2VydWVYIGZvc1B6LO9TICHSQUNGKTAObgNVHQ8BAf8EBAMCAQYwDwYDVROTAQH/
BAUwAwEB/zAdBgNVHQ4EFgQUt27qkxQj2AgEA9cx4rf07Vkz5qswDQYJKoZIhvcN
AQEFBQADgYEAqWTnhDcf7GUAww7hBk5XWbODsT5N/A/P2mVFs7mSpJpT3Ildbe+I
Ipf4kRFRuoN6bIFDwOyFnCp71BbWH8dF/OnMwBGMsFEHLrF6Fjw12ovObWVqCiAE
-----END CERTIFICATE-----

```

email: webmaster@your-company.com



For z/OS server, put this content in a dataset. Then specify the dataset name in RACDCERT ADD

User requests browser certificate

PKI Services Certificate Generation Application

[Install the CA certificate to enable SSL sessions for PKI Services](#)



Choose one of the following:

- Request a new certificate using a model

Select the certificate template to use as a model

- Pick up a previously requested certificate

Enter the assigned transaction ID

Select the certificate return type

- Renew or revoke a previously issued browser certificate

- Recover a previously issued certificate whose key was generated by PKI Services

Enter the email address when the original certificate was requested

Enter the same pass phrase as on the request form

- Administrators click here

5-Year PKI SSL Server Certificate
1-Year PKI SSL Browser Certificate
1-Year PKI S/MIME Browser Certificate
2-Year PKI Windows Logon Certificate
2-Year PKI Browser Certificate For Authenticating To z/OS
5-Year PKI SSL Server Certificate
5-Year PKI IPSEC Server (Firewall) Certificate
5-Year PKI Intermediate CA Certificate
2-Year PKI Authenticode - Code Signing Certificate
5-Year SCEP Certificate - Preregistration
1-Year PKI Generated Key Certificate
n-Year PKI Certificate for Extensions Demonstration

Retrieve Your 1-Year PKI SSL Browser Certificate

Please bookmark this page

Since your certificate may not have been issued yet, we recommend that you create a bookmark to this location so that when you return to this bookmark, the browser will display your transaction ID. This is the easiest way to check your status.

Enter the assigned transaction ID

1dZ0dFXy4lw2Tc++++++

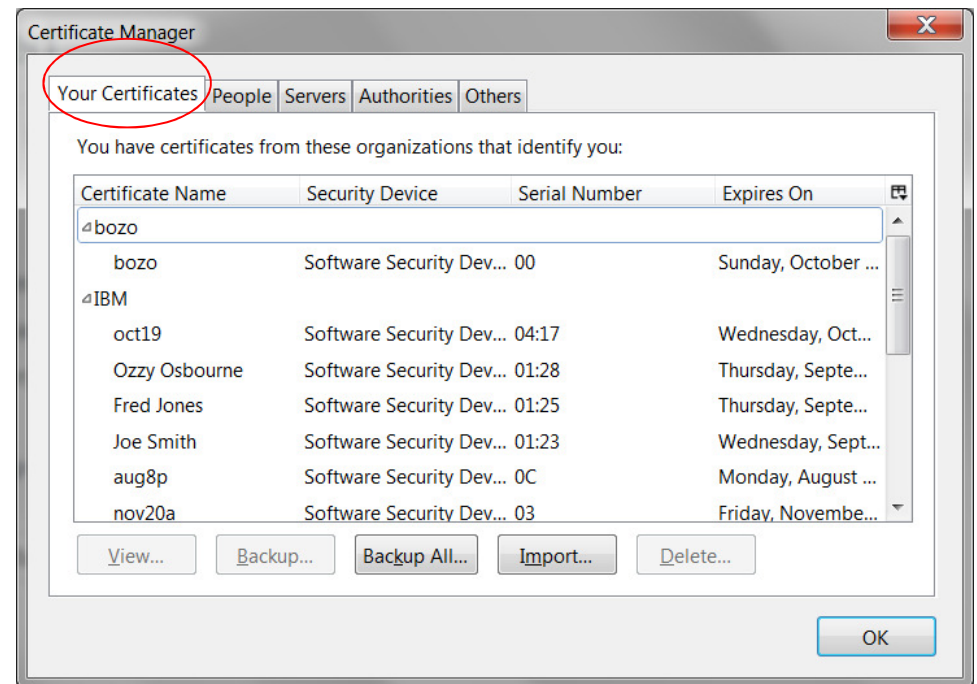
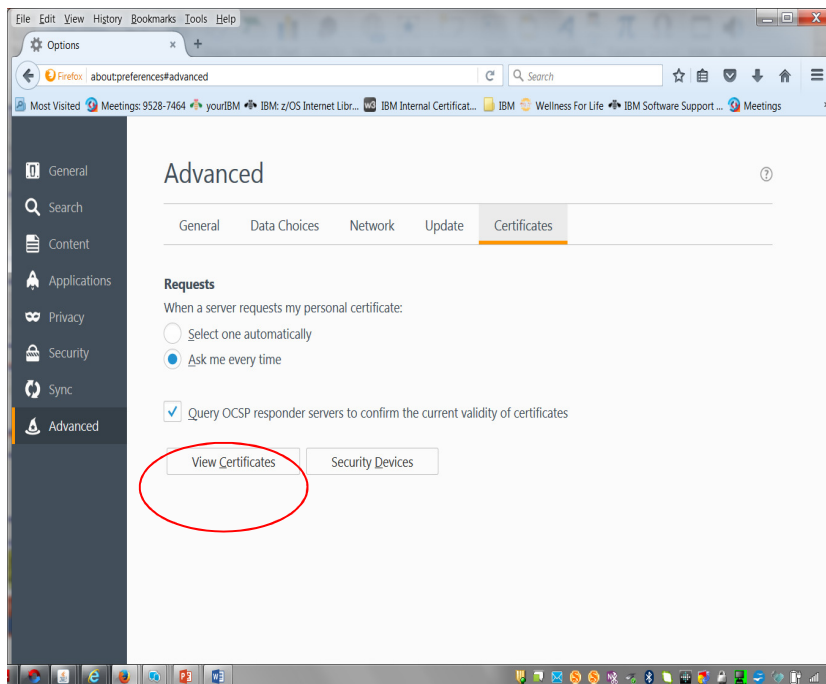
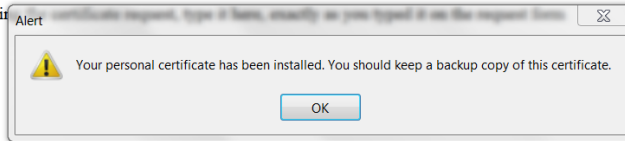
If you specified a pass phrase when submitting

•••••

Retrieve and Install Certificate

Home Page

email: webmaster@your-company.com



- **IBM PKI Redbooks**

- Managing Digital Certificates across the Enterprise**

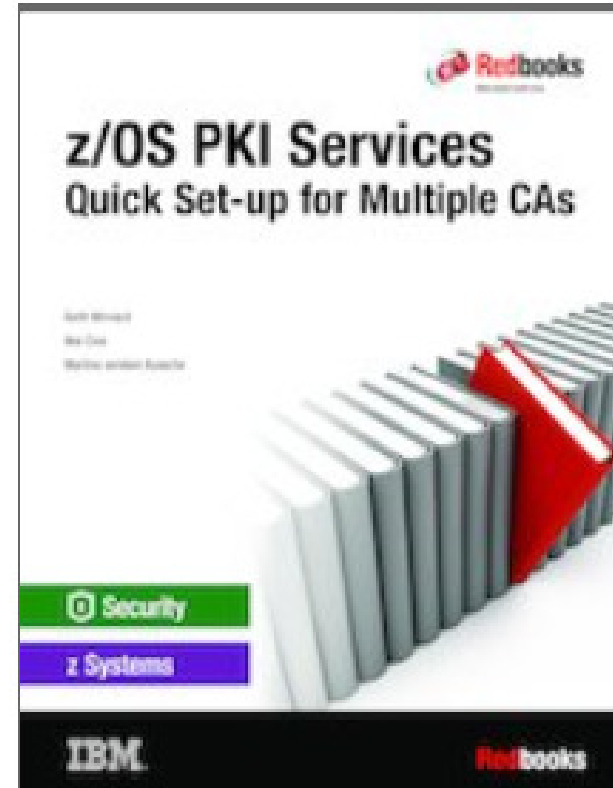
- <http://www.redbooks.ibm.com/abstracts/sg248336.html?Open>

- z/OS PKI Services: Quick Set-up for Multiple CAs**

- <http://www.redbooks.ibm.com/abstracts/sg248337.html?Open>

- Youtube on PKI web page usages**

- <https://www.youtube.com/watch?v=U0oqk6siKkA&feature=youtu.be>



Some suggestion on debugging certificate problem

- Very first question - Which side are you looking at – server or client?
- Find out where the key ring used by the application is specified, eg:
 - FTP server, IPSEC: AT-TLS policy
 - TN3270 Server: Telnet profile
 - HTTP server: httpd.conf / vhost.conf
- List the certificates in the key ring and record their labels (RACDCERT LISTRING)
 - Server ring – server certificate (usually marked as default), issuer CA, root CA
 - Client ring – root CA, same as that in server ring
- List the chain of the server certificate using its label to make sure the ring contains the right ones (RACDCERT LISTCHAIN)

Some suggestion on debugging certificate problem

- All the certificates in the key ring serve a purpose, for identification or for validation. If you can't tell why it is there, probably it is not needed.
- The more certificates in the ring, the longer the processing time, the harder to debug which one is causing the problem

References

- **PKI Services web site:**
<http://www.ibm.com/servers/eserver/zseries/zos/pki>
- **Cryptographic Server Manual**
Cryptographic Services PKI Services Guide and Reference
<http://publibz.boulder.ibm.com/epubs/pdf/iky2a110.pdf>
Cryptographic Services System Secure Sockets Layer Programming
<http://publibz.boulder.ibm.com/epubs/pdf/gsk2aa10.pdf>
- **Security Server Manuals:**
RACF Command Language Reference
<http://publibz.boulder.ibm.com/epubs/pdf/ich2a411.pdf>
RACF Security Administrator's Guide
<http://publibz.boulder.ibm.com/epubs/pdf/ich2a711.pdf>
- **RFCs**
RFC5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
- **IBM Education Assistant web site:**
<http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp>
- **RACF web site:**
<http://www.ibm.com/servers/eserver/zseries/zos/racf>

Questions ?