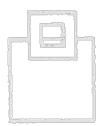


# Understand, manage and love certificates in z/OS and USS

#### Ulf Heinrich SOFTWARE ENGINEERING

u.heinrich@seg.de





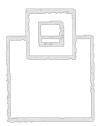
© 2024 SOFTWARE ENGINEERING GMBH and SEGUS Inc.

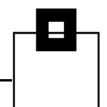
Agenda

#### General basics

- Where/what are certificates used for?
- How is it used/realized?
- Real examples from the ZOWE ecosystem,
  - as well as z/OSMF, UMS, SQLDI, Db2
- Managing certificates in USS and z/OS
- Analyzing certificate issues





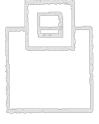


#### \_\_\_\_\_

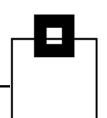
**General Basics** 

- Like an official identity card in the analog world, a certificate reliably proves an identity in the digital world to
  - Protect from fake identities
  - Refer to an authority that proves the identity
  - Acknowledge the data by the electronic signature of the authority
  - Relate a public key (owner) to an identity
  - Associate a public key to the identity data of
    - a person
    - an organization
    - a device



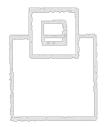


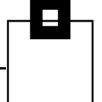


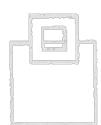


#### **General Basics**

- Digital certificates, or public key certificates, or identity certificates are used to identify and validate an unknown origin and to communicate securely with it
  - It includes information about the owner/<u>subject</u>, plus typically a certificate of the entity/<u>issuer</u> that has verified the owner/subject
  - 2. It includes a public key that allows asymmetric, one-way encryption
    - The public key is intended to be shared
      - A Public key enables anybody to encrypt content
      - Only the corresponding private key of a public/private key pair can decrypt the content







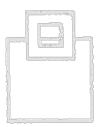
### **General Basics**

Conclusion:

- $\rightarrow$  A certificate is an electronic document used to
  - 1. prove an identity and
  - 2. to provide a key

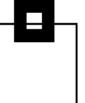
which is part of the document

- $\rightarrow$  Once a certificate is verified to be trustworthy the validity proves
  - sender/integrity of an e-mail (S/MIME)
  - authenticity of a payment card for transactions (EMV)
  - owner/integrity/genuine of apps/binaries (code signing)
  - Document, eID, role, ...
  - device (domain/host/IP) (TLS/SSL)
  - Further, the public key can be used for secure communication with a
    - Person, or organization (e.g. e-mail, messaging)
    - Device (https, ftps, sftp, ssh, VPN, RDP...)





		1



Digital Signature: Valid	×
	Runtime Stats Maintenance Packa
From: db2support Signed Bv: db2support@segus.com	
	message is Valid and Trusted.
I ne digital signature on this	message is Valid and Trusted.
For more information about the message. click Details.	the certificate used to digitally sign
	Digital Signature: Valid X
Warn me about errors in digitally sig	Subject:RE: IBM VerifyFrom:Heinrich, UlfSigned Bv:u.heinrich@segus.com
	The digital signature on this message is Valid and Trusted.
	For more information about the certificate used to digitally sign the message. click Details.
	Details
	<u>W</u> arn me about errors in digitally signed email before message opens.
	Close

Adobe Acrobat Reader Installer

Verified publisher: Adobe Inc. File origin: Downloaded from the Internet

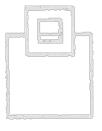
Show more details

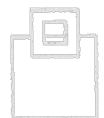
## Program signing and verification

This chapter provides information about enabling users to digitally sign programs and enabling RACF® to verify signed programs.

<sup>|<</sup>This chapter also provides instructions for enabling RACF support for Validated Boot for z/OS. Here, you must perform some set-up activities before using Validated Boot for z/OS to sign IPL data. The term *IPL data* includes IPL text and system load modules, such as the system residence volume (SYSRES) contents.With Validated Boot for z/OS, your installation can ensure that its IPL data is intact, untampered-with, and originates from a trusted build-time source. Information about RACF support for Validated Boot for z/OS is provided in IPL data signing for Validated Boot for z/OS.<sup>>|</sup>







/here/what are c	ertificates used for?	
Menu 🏠 🛠 flyer-bhc.pdf X + Create	⑦ .	
tools Edit Convert E-Sign	Find text or tools Q 🛛 🛱 🏠 🖗 🖉 🖾	
Certified by Software Engineering GmbH <db2announcement( CA 3-1 2013.</db2announcement( 	@seg.de>, Marketing, certificate issued by D-TRUST Application Certificates (Signature Panel)	
┌┲┱┐	Pocket Tools	
	Ø.	
Download our <u>licensed freeware</u> *	BufferPool HealthCheck for Db2 z/OS is a light fast Db2 for z/OS Local and Croup Bufferpool et thresholds that can cause performance degradat. The Db2 Buffer pools are the central area of Pt Lines are use and by everything and, arguably, they have been forgotten zover the past years. especially the GROUP BUFFERPOOLs have all fallen into the shadows. Afterall, just add some CPs, ZilPs or some memory and all is well?	
Licensed Freeware: Comes as a lightweight Batch job Validates more than 30 rules (system residency, prefetch, VPSEQT,)	So, what does <b>BufferPool HealthCheck for Db2 z/OS</b> then do? It issues -DISPLAY BUFFERPOOL and -DISPLAY GROUPBUFFERPOOL commands, reads the results, computes various values and checks if these values have broken any pre-defined thresholds or not.	
	Freeware Version 1	
The full WLX version: Supports BP simulation	The Freeware Version is a simple Batch job that outputs to SYSPRINT.	

#### nttps://www.seg.ae

Connection is secure

	~/
1-0	- X

websites

This site has a valid certificate, issued by a trusted authority.

This means information (such as passwords or credit cards) will be securely sent to this site and cannot be intercepted.

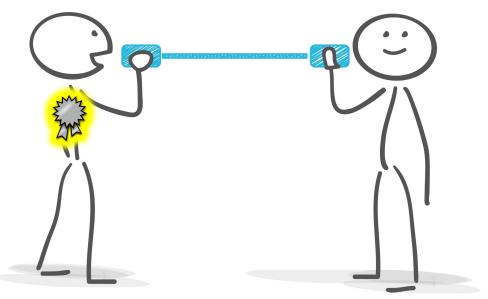
Always be sure you're on the intended site before entering any information.

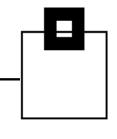
#### Learn more

The technology is always the same, but today we focus on secure client – server communication:

- 1. Assure that a subject is really the one it supposes to be.
- 2. Assure that the information exchanged isn't manipulated.
- 3. Assure that the communication is treated confidentially.







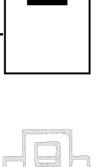


Let's have a closer look at secure client – server communication:

- A standardized process,...
  - 1987 Secure Data Network System (SDNS) project initiated
  - 1996 using SSL 3.0 under governance of the IETF to develop internet-standards
  - since 1999 continuously enhanced as transport layer security (TLS)
- ... that anybody understand/supports
  - Any current client (e.g. browser, desktop, smartphone) and server (e.g. mail, web, database) supports secure communication via the X.509 based mechanisms
    - TLS handshake
    - TLS record







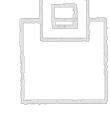
Secure client – server communication starts with a secure connection request, (e.g. https, ftps, ...) and often requires to specify a secure port:

https://s0w1.dus.seg.de:10443/zosmf

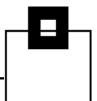
- 1. Connection request from a client to a server
- 2. Server replies with its certificate
- 3. Verification of the replying server and its trustworthiness by the client
- 4. Connection dependent handshake of the encryption between client and server
- Optionally: Certificate authentication of the client

Verification of the client by the server

5. Start encrypted communication



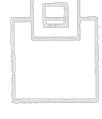


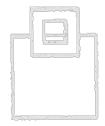


After we've received the certificate (including a key) from a server how is the information verified to guarantee its identity?

- A certificate alone does not guarantee the identity shown, nor its trustworthiness!
  - An identity can only be proved by a trusted entity
  - Trustworthiness can only be judged by the communication partner
- So, how can a client know if the communication partner is safe and trustworthy?
  - 1. Either the provided certificate is individually categorized trustworthy,
  - 2. or a superior certificate authority (CA) is trusted that confirms the identity shown (certificate chain)

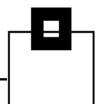
This is the major concept used throughout X.509-based TLS.





	· ····································		
1			

13



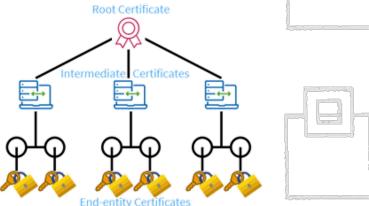
Who is a *superior certificate authority* (CA)?

- Higher instance in a certificate chain of trust (intermediate, or root)
  - Reputable, commonly trusted organizations\*
    - May assign limited duties to external identity authorities
  - Companies usually have an "internal" CA to simplify certificate management
- Validates the content of a certificate (signing request CSR) and can issue/revoke certificates inheriting trustworthiness

 $\rightarrow$  Certificates signed by a trusted CA are automatically trusted!

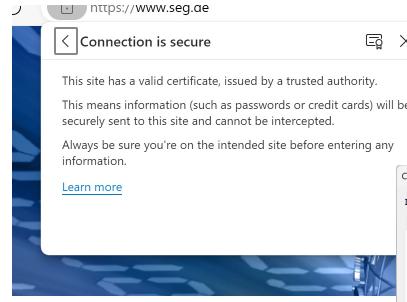
\*The Certification Authority Browser Forum (CA/Browser Forum) is a voluntary gathering of certificate Issuers and suppliers of internet browser software and other applications that use certificates.





			7
-			
I			

#### Who is a superior certificate authority (CA)?



Certificate Viewer: seg.de	$\sim$
General <b>Details</b>	
Certificate Hierarchy	
▼ ISRG Root X1	
▼ R3	
seg.de	

hor	rity (CA)?	
ିଲ୍ଲ ity. ls) will ng any	× be	
×	Certificates       ×         Intended purpose: <all>         Other People       Intermediate Certification Authorities       Trusted Root Certification Authorities         Issued To       Issued By       Expiration         Friendly Name       9/20/2028       D-TRUST Root CA 3 2013         GlobalSign       GlobalSign       3/18/2029         GlobalSign Root CA       GlobalSign Root CA       1/28/2028         Go Daddy Class 2 Certification A       Go Daddy Root Certificate Authon       Go Daddy Root Certificate Authon         Go Daddy Root Certificate Authon       Go Daddy Root Certificate Authon       Go Daddy Root Certificate Authon       Go Daddy Root Certificate Authon         IdenTrust Commercial Root CA 1       IdenTrust Commercial Root X1       6/4/2035       ISRG Root X1         Microsoft ECC Product Root Cert       Microsoft ECC Product Root Cert       1/1/2/2043       Microsoft Auth         Microsoft ECC TS Dest Certificate       Microsoft ECC TS Dest Certificate       3/27/2043       Microsoft ECC</all>	
	Import       Export       Remove       Advanced         Certificate intended purposes       Client Authentication, Server Authentication       View         Use       Use       Close	

© 2024 SOFTWARE ENGINEERING GMBH and SEGUS Inc.

Besides the verification of an identity we want to initiate the secure connection, but

- Client and server may not know each others yet
- Communicating securely requires that both parties are able to encrypt and to decrypt the information sent/received

#### BUT:

- Without a common (symmetric) encryption key, no encryption!
- If they'd negotiate a key to start encryption, it would need to be unencrypted and someone else on the network could use a network sniffer, steal the key and compromise the encryption

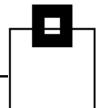
The solution:

 $\rightarrow$  Client and server negotiate the symmetric encryption key using asymmetric encryption





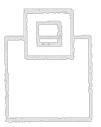
1	
6	{ {

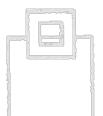


- TLS encryption is based on X.509 certificates that identify the owner and provide the public key from a public/private key pair
- The public key coming with this certificate can be used to initiate asymmetric encrypted communication
  - Therefore, the *public* key provided along with the certificate at connection request is used by the recipient to check integrity and create and return an encrypted pre-master-key
  - The encrypted pre-master-key can only be decrypted with the appropriate *private* key, which is then used for the further encryption
  - Public key can encrypt, but only private key can decrypt (asymmetric encryption)
- Due to the fact that the private key should <u>never ever</u> be accessible by someone else but the owner, certificates are typically generated manually by the owner, or as part of an installation by the owner (like ZOWE does):
  - E.g.:

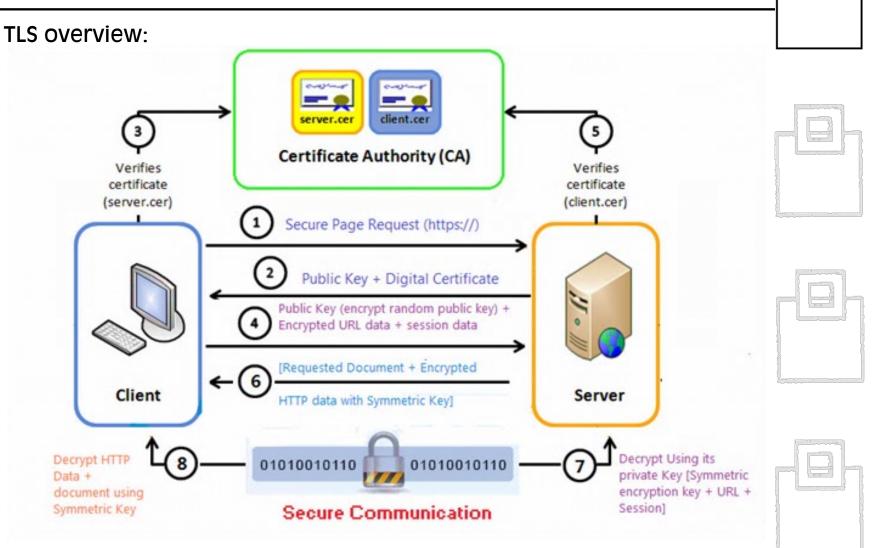
openssl req -x509 -newkey rsa:4096 -keyout key.pem out cert.pem **OR** certsigreq.csr -days 365







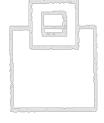




z/OSMF, ZOWE and Db2 work exactly this way:

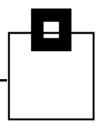
- 1. Connection request against z/OSMF, ZOWE, Db2 (secure port!)
- 2. Reply by z/OSMF, ZOWE, Db2 with its certificate (incl. certificate chain with a certificate authority if applicable)
- 3. Trustworthiness verification of the certificate, resp. of the root/intermediate certificate authority
- 4. Generation and return of the pre-master-key by the client using the servers public key
- 5. Generation of the encryption of an individual connection and start of the encrypted communication
  - Manipulation can be detected by an individual message authentication code

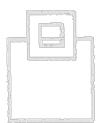


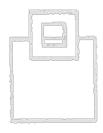


- The standardized certificate based on TLS is used
- Certificates are managed either in a KEYSTORE/TRUSTSTORE, or...
  - https://docs.zowe.org/stable/user-guide/configurecertificates-keystore
- by RACF KEYRINGs
  - https://docs.zowe.org/stable/user-guide/configurecertificates-keyring
- More detailed information about certificate generation/management for application development extending ZOWE is available at
  - https://docs.zowe.org/stable/extend/extend-apiml/onboardplain-java-enabler/#api-security

Reminder: It's all about trustworthiness!







 The certificate store is specified in the ZOWE configuration (zowe.yaml, formerly instance.env), as a java keystore/truststore, or...

```
certificate:
```

keystore:

type: PKCS12

file: /zowe/keystore/localhost/localhost.keystore.p12

password: password

alias: localhost

truststore:

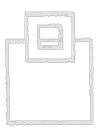
type: PKCS12

file: /zowe/keystore/localhost/localhost.truststore.p12
password: password

pem:

```
key: /zowe/keystore/localhost/localhost.key
certificate: /zowe/keystore/localhost/localhost.cer
certificateAuthorities: /zowe/keystore/local_ca/local_ca.cer
verifyCertificates: STRICT
```





#### ... as a RACF keyring

certificate:

keystore:

type: "JCERACFKS"

file: "safkeyring:///ZWESVUSR/ZOWEKEYS"

password: "password"

alias: "ZWESRV"

truststore:

type: "JCERACFKS"

file: "safkeyring:///ZWESVUSR/ZOWEKEYS"

password: "password"

pem:

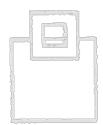
key: ""

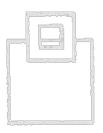
certificate: ""

certificateAuthorities:

"safkeyring:///ZWESVUSR/ZOWEKEYS&SEGROOTCA" verifyCertificates: "STRICT"







☐ localhost **KEYSTORE**: localhost.keystore.cer Stores its own certificate localhost.keystore.cer-ebcdic TRUSTSTORE localhost.keystore.csr Stores trusted certificates localhost.keystore.jwtsecret.cer localhost.keystore.jwtsecret.pem localhost.keystore.key localhost.keystore.p12 localhost.keystore\_signed.cer localhost.truststore.p12 local\_ca localca.cer RACE KEYRING localca.cer-ebcdic Stores both localca.keystore.p12 Ring: ZOWEKEYS Certificate Label Name Cert Owner USAGE DEFAULT **SEGROOTCA** CERTAUTH CERTAUTH NO

ID(ZWESVUSR) PERSONAL

YES

ZWESRV

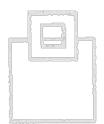
## Real examples from UMS and z/OSMF

 IBM Unified Management Server uses ZOWE's keystore/truststore/keyring by default, unless you specify something else in UMS's parmlib member

```
certificate:
    allowSelfSigned: true
    truststore:
    location: "safkeyring:///ZWESVUSR/IZPRING"
    type: "JCERACFKS"
    keystore:
    location: "safkeyring:///ZWESVUSR/IZPRING"
    type: "JCERACFKS"
    alias: "UMSSRV"
```

 For z/OSMF you can specify the RACF keyring in the IZU PARMLIB member

```
(...)
KEYRING_NAME('ZOSMFKEYS')
(...)
```

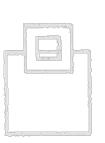


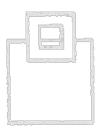
### Real examples from SQLDI and Db2

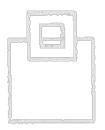
 For SQL Data Insights you are prompted to specify the RACF keyring when running the installation script sqldi.sh

Enter your keystore information > SQLDIID.SQLDIKEYRING

```
For Db2 you have to configure the TLS setup via PAGENT
  TTLSRule DD10SecureServer
  { LocalPortRange 15151
    JobName DD10DTST
    Direction Inbound
    TTLSGroupActionRef DD10SecureGrpAct
    TTLSEnvironmentActionRef DD10SecureEnvAct
    TTLSConnectionActionRef DD10SvrAuthConn
  TTLSGroupAction DD10SecureGrpAct
  { TTLSEnabled On
    Trace 15
  TTLSEnvironmentAction DD10SecureEnvAct
    TTLSKeyRingParms
         Keyring SEGDB2KEYRING
    (...)
```



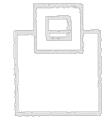


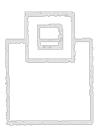


#### How to manage keystores, truststores, keyrings?

• A keystore/truststore can be managed using the keytool

>keytool	
Key and Certificate	Management Tool
Commands:	
-certreq	Generates a certificate request
-changealias	Changes an entry's alias
-delete	Deletes an entry
-exportcert	Exports certificate
-exportseckey	Export a batch of secret keys
-genkeypair	Generates a key pair
-genseckey	Generates a secret key
-gencert	Generates certificate from a certificate request
-importcert	Imports a certificate or a certificate chain
-importpass	Imports a password
-importkeystore	Imports one or all entries from another keystore
-importseckey	Import a batch of secret keys
-keypasswd	Changes the key password of an entry
-list	Lists entries in a keystore
-printcert	Prints the content of a certificate
-printcertreq	Prints the content of a certificate request
-printcrl	Prints the content of a CRL file
-storepasswd	Changes the store password of a keystore







How to manage keystores, truststores, keyrings?

- A keyring can be managed using RACF
  - Services option menu

RACF - SERVICES OPTION MENU

OPTION ===>

SELECT ONE OF THE FOLLOWING:

- 1 DATA SET PROFILES
- 2 GENERAL RESOURCE PROFILES
- 3 GROUP PROFILES AND USER-TO-GROUP CONNECTIONS
- 4 USER PROFILES AND YOUR OWN PASSWORD
- 5 SYSTEM OPTIONS
- 6 REMOTE SHARING FACILITY

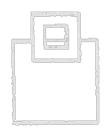
7 DIGITAL CERTIFICATES, KEY RINGS, AND TOKENS

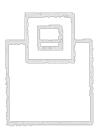
99 EXIT

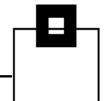
RACDCERT (Manage RACF digital certificates)

"Use the RACDCERT command to install and maintain digital certificates, key rings, and digital certificate mappings in RACF."

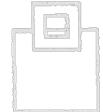




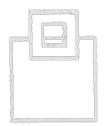


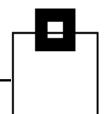


- Using KEYSTORE/TRUSTSTORE with self-signed certificates might be ok for testing,
  - 📥 Easy setup without additional RACF
  - Unix/USS OPENSSL and KEYTOOL usage as usual
  - $\P$  Has to be trusted by the ZOWE user
  - No centralized certificate management
- but at the end, a RACF KEYRING with company CA-signed certificates is a better choice
  - le Centralized z/OS/USS certificate management
  - lmplicitly trusted for all employers
  - Requires RACDCERT knowledge and authorization
  - Some (Db2) require additional PAGENT definition



1	~~~~~	





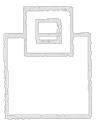
\$\$

#### Managing certificates in USS and z/OS

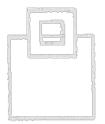
#### RACDCERT example of a certificate + company CA

#### 1. Create a company CA to make any of your certificates trustworthy

```
//GENCACRT EXEC PGM=IKJEFT01, REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSTSIN
           DD DDNAME=RACF
//RACF
           DD DATA, DLM=$$, SYMBOLS=JCLONLY
   RACDCERT GENCERT CERTAUTH +
            SUBJECTSDN( +
              CN('SOFTWARE ENGINEERING ROOT CA') +
              OU('DEVELOPMENT') +
              O('SOFTWARE ENGINEERING GMBH') +
              L('DUESSELDORF') +
              SP('NORTH RHINE WESTPHALIA') +
              C('DE')) +
            SIZE(2048) +
            NOTAFTER (DATE (2033-01-07)) +
            WITHLABEL ('SEGROOTCA') +
            KEYUSAGE (CERTSIGN)
```







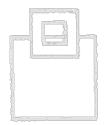
\$\$

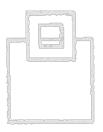
#### Managing certificates in USS and z/OS

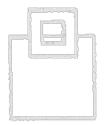
#### RACDCERT example of a certificate + company CA

#### 2. Create a certificate signed with the CA created before

```
//GENSVCRT EXEC PGM=IKJEFT01, REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD DDNAME=RACF
           DD DATA, DLM=$$, SYMBOLS=JCLONLY
//RACF
   RACDCERT GENCERT ID(IZUSVR1) +
            SUBJECTSDN( +
              CN('ZOSMF MANAGEMENT SERVICE') +
              OU('DEVELOPMENT') +
              O('SOFTWARE ENGINEERING GMBH') +
              L('DUESSELDORF') +
              SP('NORTH RHINE WESTPHALIA') +
              C('DE')) +
            SIZE(2048) +
            NOTAFTER (DATE (2025-04-02)) +
            WITHLABEL ('IZUSRV') +
            KEYUSAGE (HANDSHAKE) +
            ALTNAME(IP(192.168.9.98) +
                DOMAIN('SOW1.DUS.SEG.DE')) +
            SIGNWITH(CERTAUTH LABEL('SEGROOTCA'))
```





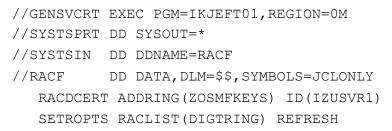


31

### Managing certificates in USS and z/OS

#### RACDCERT example of a certificate + company CA

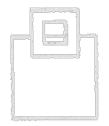
#### 3. Create a keyring for the certificates created

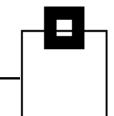












\$\$

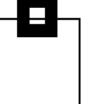
#### Managing certificates in USS and z/OS

#### RACDCERT example of a certificate + company CA

#### 4. Add the certificates created to the keyring created

//GENSVCRT EXEC PGM=IKJEFT01,REGION=0M //SYSTSPRT DD SYSOUT=\* //SYSTSIN DD DDNAME=RACF //RACF DD DATA,DLM=\$\$,SYMBOLS=JCLONLY RACDCERT CONNECT(CERTAUTH LABEL('SEGROOTCA') + RING(ZOSMFKEYS)) + ID(IZUSVR1) RACDCERT CONNECT(ID(IZUSVR1) + LABEL('IZUSRV') + RING(ZOSMFKEYS) + USAGE(PERSONAL) DEFAULT) + ID(IZUSVR1)





#### RACDCERT example of a certificate + company CA

#### 5. Permit access to the keyring created

- //GENSVCRT EXEC PGM=IKJEFT01,REGION=0M
- //SYSTSPRT DD SYSOUT=\*
- //SYSTSIN DD DDNAME=RACF
- //RACF DD DATA,DLM=\$\$,SYMBOLS=JCLONLY
  RDEFINE RDATALIB IZUSVR1.ZOSMFKEYS.LST UACC(NONE)
  PERMIT IZUSVR1.ZOSMFKEYS.LST CLASS(RDATALIB) ID(IZUSVR1) +
  ACCESS(CONTROL)
- /\* Uncomment this command to allow other user to access key ring ... \*/
- /\* PERMIT IZUSVR1.ZOSMFKEYS.LST CLASS(RDATALIB) ID(<USER>) +
- /\* ACCESS (READ)

SETROPTS RACLIST (RDATALIB) REFRESH

PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(IZUSVR1) +

ACCESS (READ)

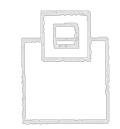
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(IZUSVR1) +

ACCESS (READ)

SETROPTS RACLIST (FACILITY) REFRESH

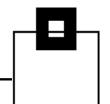
\$\$





\*/

\*/



### Analyzing certificate issues

Trustworthy or not, that's the question!

#### This Connection Is Not Private

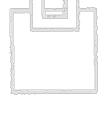
This website may be impersonating "s0w1.dus.seg.de" to steal your personal or financial information. You should go back to the previous page.

Go Back

Safari warns you when a website has a certificate that is not valid. This may happen if the website is misconfigured or an attacker has compromised your connection.

To learn more, you can <u>view the certificate</u>. If you understand the risks involved, you can <u>visit this website</u>.

#### How to fix this???







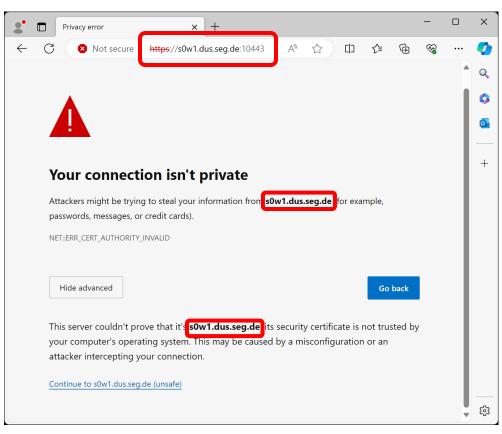
34

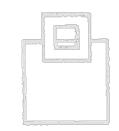
#### © 2024 SOFTWARE ENGINEERING GMBH and SEGUS Inc.

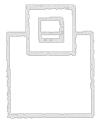
### Analyzing certificate issues

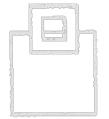
Trustworthy or not, that's the question!

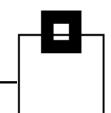
#### 1. Make sure the host, or IP is correct!







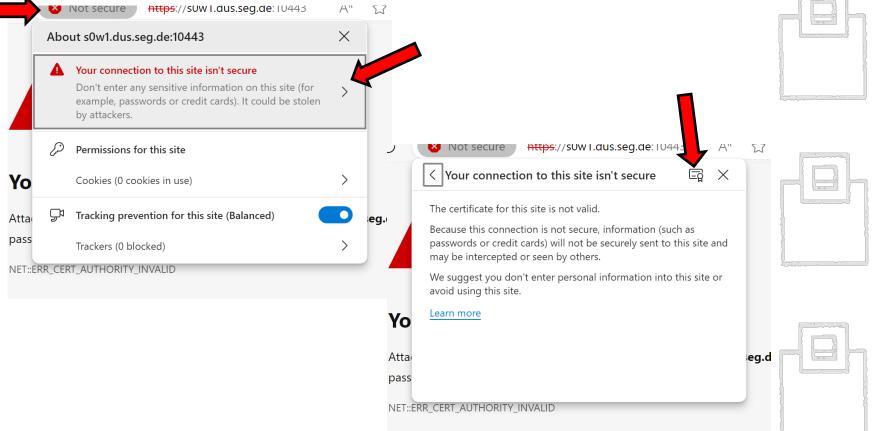




### Analyzing certificate issues

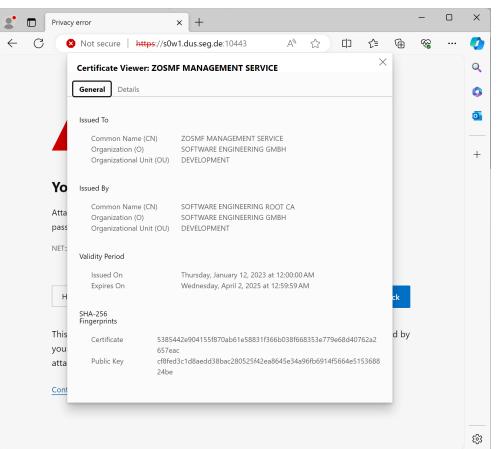
#### Trustworthy or not, that's the question!

#### 2. Verify the certificate



Trustworthy or not, that's the question!

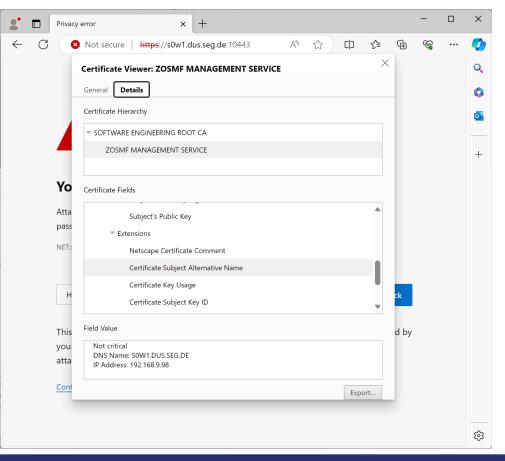
#### 2. Verify the certificate's content

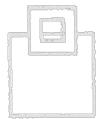




*Trustworthy or not, that's the question!* 

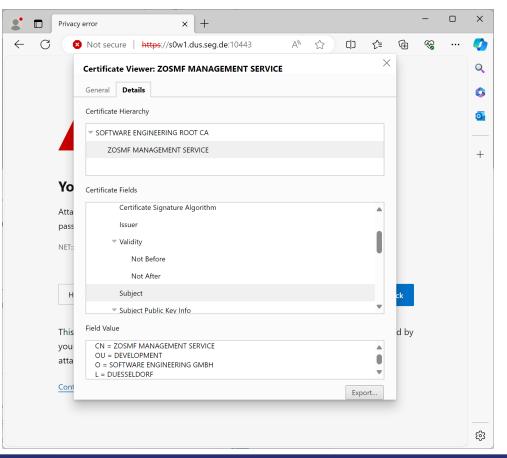
2. Verify the certificate's content



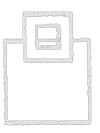


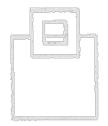
*Trustworthy or not, that's the question!* 

2. Verify the certificate's content



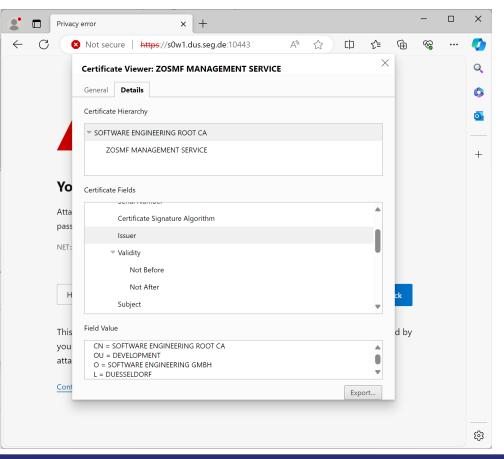


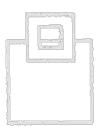


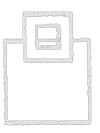


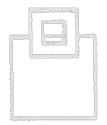
*Trustworthy or not, that's the question!* 

2. Verify the certificate's content





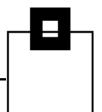




#### Trustworthy or not, that's the question!

#### 3. Verify that the CA (or the certificate) is trusted

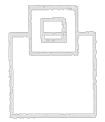
Privacy error × 🐯 Se	x +			- 0 ~~	×	
← C	Your browser is managed by your organization	☆ CD	Σ≡ \⊕	~~ ···		
	Tour <u>prowser is managed</u> by your organization				Q	
Settings	Security				۵.	
Q Search settings	Manage security settings for Microsoft Edge	M	7		_	
<ul> <li>Profiles</li> <li>Privacy, search, and services</li> </ul>	Manage certificates Manage HTTPS/SSL certificates and settings	2			+	
<ul><li>Appearance</li><li>Sidebar</li></ul>	Microsoft Defender SmartScreen Help protect me from malicious sites and downloads with Microsoft Defender SmartScreen	ð •				
<ul> <li>Start, home, and new tabs</li> <li>Share, copy and paste</li> <li>Cookies and site permissions</li> </ul>	Block potentially unwanted apps Blocks downloads of low-reputation apps that might cause unexpected behaviors					
Default browser	Website typo protection ⑦ Warn me if I have mistyped a site address and may be directed to a potentially malicious site.	۵ 💿				
쏭 Family safety A <sup>라</sup> Languages	Clear all previously allowed sites	Clear				
Consultation     Printers     System and performance	Use secure DNS to specify how to lookup the network address for websites By default, Microsoft Edge uses your current service provider. Alternate DNS providers may cause some sites to not be reachable.	8				
<ul><li>Reset settings</li><li>Phone and other devices</li></ul>	Use current service provider Your current service provider may not provide secure DNS				•	
C About Microsoft Luge						
	📕 Q 🎽 😨 📽	^	DEU 🖵 😋	11:06 AI	M 24	

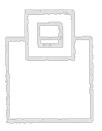


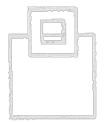
#### Trustworthy or not, that's the question!

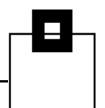
#### 3. Verify that the CA (or the certificate) is trusted

Certificates			×
Intended purpose: <a>All&gt;</a> Other People Intermediate Cert	tification Authorities	Root Certification Au	thorities
Issued To VeriSign Universal Root Certif VeriSign Class 3 Public Prima USERTrust RSA Certification A T-TeleSec GlobalRoot Class 2 Thawte Timestamping CA Symantec Enterprise Mobile R SwissSign Gold CA - G2 Starfield Class 2 Certification	ry Certifi VeriSign Class 3 Pul ry Certifi VeriSign Class 1 Pul Authority USERTrust RSA Cer T-TeleSec GlobalRod Thawte Timestampi Root for Symantec Enterpris SwissSign Gold CA Authority Starfield Class 2 Cer	bli 7/17/2036 bli 7/17/2036 tif 1/19/2038 ot 10/2/2033 n 1/1/2021 e 3/15/2032 - G2 10/25/2036 rti 6/29/2034	Friendly I VeriSign VeriSign Sectigo T-TeleSe Thawte 1 <none> SwissSigi Starfield</none>
Import Export	<u>R</u> emove		<u>A</u> dvanced
Certificate intended purposes			⊻iew
			<u>C</u> lose





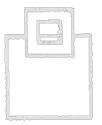




#### Trustworthy or not, that's the question!

3. Verify that the CA (or the certificate) is trusted – add it, if missing

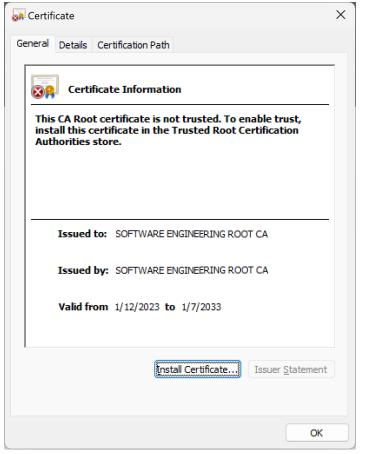
Certificate X
General Details Certification Path
Certification <u>p</u> ath
<u>Vi</u> ew Certificate
Certificate <u>s</u> tatus: This CA Root certificate is not trusted because it is not in the Trusted Root Certification Authorities store.
ОК



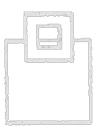
FBh
-----

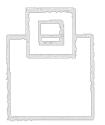
#### Trustworthy or not, that's the question!

3. Verify that the CA (or the certificate) is trusted – add it, if missing





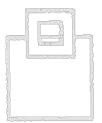




Trustworthy or not, that's the question!

3. Verify that the CA (or the certificate) is trusted – add it, if missing

🔶 🛿 🛃 Certificate Import Wizard	×
Welcome to the Certificate Import Wizard	
This wizard helps you copy certificates, certificate trust lists, and certificate revocation lists from your disk to a certificate store.	
A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.	
Store Location  Current User  Local Machine	
To continue, dick Next.	
<u>N</u> ext Can	el



in man m	

Trustworthy or not, that's the question!

← 🖉 Cer	tificate Import Viz	rd		×
	ficate Store Certificate stores are e	system areas where	certificates are kept.	
	he certificate.		ate store, or you can spe tore based on the type o	
7	Place all certificate store	ates in the following s e:	store	Browse



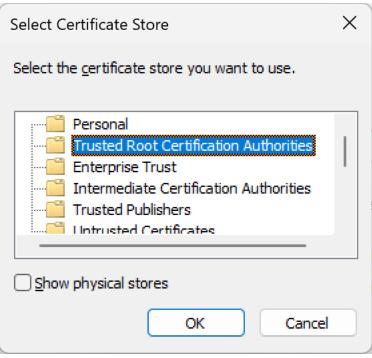
		~~~~~s2	1	
{			L	
	0.9			ł
				1
 		~~~~		

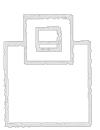
{		

#### © 2024 SOFTWARE ENGINEERING GMBH and SEGUS Inc.

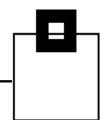
# Analyzing certificate issues

Trustworthy or not, that's the question!









Trustworthy or not, that's the question!

3. Verify that the CA (or the certificate) is trusted – add it, if missing

🗧 🛃 Certificate Import Wizard	×
Completing the Certificate Import Wizard	
The certificate will be imported after you click Finish.	
You have specified the following settings:	
Certificate Store Selected by User         Trusted Root Certification Authorities           Content         Certificate	5
<u> </u>	h Cancel



······································

32	 

#### © 2024 SOFTWARE ENGINEERING GMBH and SEGUS Inc.

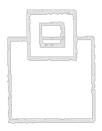
49	

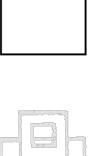
# Analyzing certificate issues

#### Trustworthy or not, that's the question!

Security	Warning	$\times$	
	You are about to install a certificate from a certification authority (CA) claiming to represent:		
	SOFTWARE ENGINEERING ROOT CA		
	Windows cannot validate that the certificate is actually from "SOFTWARE ENGINEERING ROOT CA". You should confirm its origin by contacting "SOFTWARE ENGINEERING ROOT CA". The following number will assist you in this process:	2	
	Thumbprint (sha1): 2EBB4B97 7684E537 008ED611 FC7296B8 3D36D433		
	Warning: If you install this root certificate, Windows will automatically trust any certificate issued by this CA. Installing a certificate with an unconfirmed thumbprint is a security risk. If you click "Yes" you acknowledge this risk.		
	Do you want to install this certificate?		
	Yes No		









#### Trustworthy or not, that's the question!

Certificate Import Wizard X	Certificates	×
The import was successful.	Intended purpose: <all>         Other People       Intermediate Certification Authorities       Trusted Root Certification Authorities</all>	ities 1 >
	QuoVadis Root CA 2         QuoVadis Root CA 2         11/2	iration 24/2031 31/2025 0/2023
ОК	SOFTWARE ENGINEERING ROOT CA       SOFTWARE ENGINEERING ROO 1/7/         Starfield Class 2 Certification Authority       Starfield Class 2 Certification Au 6/29         SwissSign Gold CA - G2       SwissSign Gold CA - G2	5
	Image: Second State Sta	2021 2/2033 0/2022
		Advanced
	Certificate intended purposes <all></all>	View
		Close

#### © 2024 SOFTWARE ENGINEERING GMBH and SEGUS Inc.

# Analyzing certificate issues

#### Trustworthy or not, that's the question!

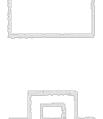
	Inteps.//sow i.dus.seg.ue. 10445/203111/20g011 anel.	Jsh	
	< Connection is secure 드립 .	×	
$\vec{\lambda}$	This site has a valid certificate, issued by a trusted authority.		
	This means information (such as passwords or credit cards) will b securely sent to this site and cannot be intercepted.	be	
	Always be sure you're on the intended site before entering any information.		r P B
	Learn more	$\mathbf{N}$	
		<sup>-</sup> he l	
		pera	
		7/05	

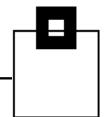
*Trustworthy, or not, that's the question!* 

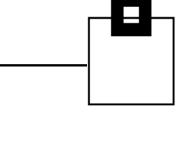
But what can you do if it's not a browser client, but an API, like a RESTful service?

 $\rightarrow$  OPENSSLs tls debugging is your friend!









```
CONNECTED(0000005)
TLS client extension "renegotiation info" (id=65281), len=1
0001 - <SPACES/NULS>
depth=1 C = DE, ST = NORTH RHINE WESTPHALIA, L = DUESSELDORF, O =
SOFTWARE ENGINEERING GMBH, OU = DEVELOPMENT, CN = SOFTWARE
ENGINEERING ROOT CA
verify error:num=19:self signed certificate in certificate chain
verify return:0
write W BLOCK
```

openssl s client -connect s0w1.dus.seg.de:15151 -tlsextdebug

```
___
```

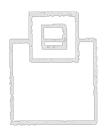
#### Certificate chain

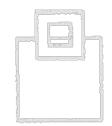
0 s:/C=DE/ST=NORTH RHINE WESTPHALIA/L=DUESSELDORF/O=SOFTWARE ENGINEERING GMBH/OU=DEVELOPMENT/CN=DB2 SECURE DISTRIBUTION SERVICE

i:/C=DE/ST=NORTH RHINE WESTPHALIA/L=DUESSELDORF/O=SOFTWARE ENGINEERING GMBH/OU=DEVELOPMENT/CN=SOFTWARE ENGINEERING ROOT CA

1 s:/C=DE/ST=NORTH RHINE WESTPHALIA/L=DUESSELDORF/O=SOFTWARE ENGINEERING GMBH/OU=DEVELOPMENT/CN=SOFTWARE ENGINEERING ROOT CA

i:/C=DE/ST=NORTH RHINE WESTPHALIA/L=DUESSELDORF/O=SOFTWARE ENGINEERING GMBH/OU=DEVELOPMENT/CN=SOFTWARE ENGINEERING ROOT CA



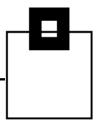


#### Server certificate

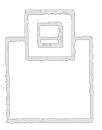
#### ----BEGIN CERTIFICATE----

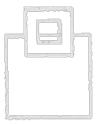
MIIEqDCCA9iqAwIBAqIBBDANBqkqhkiG9w0BAQsFADCBpDELMAkGA1UEBhMCREUx HzAdBgNVBAgTFk5PUlRIIFJISU5FIFdFUlRQSEFMSlHjj085BgNVBAcTC0RVRVNT RUXET1JGMSIwIAYDVQQKEx1TT0ZUV0FSRSBFTkdJTkVFUk1ORyBHTUJIMRQwEqYD VOOLEwtERVZFTE9OTUVOVDEkMCIGA1UEAxMbU09GVFdBUkUgRU5HSU5FUk1ORvBS T09UIENBMB4XDTIzMDExNTIzMDAwMFoXDTI1MDQwMTIyNTk1OVowqaqxCzAJBqNV BAYTAkRFMR8wHQYDVQQIExZOT1JUSCBSSElORSBXRVNUUEhBTE1BMRQwEqYDVQQH EwtEVUVTU0VMRE9SRjEiMCAGA1UEChDGC09GVFdBUkUgRU5HSU5FRVJJTkcgR01C SDEUMBIGA1UECxMLREVWRUxPUE1FTlQxKDAmBqNVBAMTH0RCMiBTRUNVUkUqRElT VFJJQlVUSU90IFNFUlZJQ0UwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB AQD70x0TZ5WsqsK6ZTy3b+Ry+xIcMTaw01+OeVG04dOPvrZEtVsvicS74vdllilB I10YncHNZ9/3E8RwxTv5qSxG4KW6PKsgd2Qpk7iBP4rMXKkrvp8rEp000W0LgPur 4sCtQpEytfYps/AFhNwPoT1hK1hZkXjywILn7/sJ3t9zYCesDDUJ1EJkywaO8U/V vgLh0SsEq2aUlaxSYhyc4KAPsdencU0QuzSZhbwMyA+4i0eSK4fqOsGUmSoACVc4 Tg0qvFLF6iTcPEXW9XNJqlVGqg1RaWuNwKG00Z01ETZUbAVZsam4exiYnRUiT6J9 oyPfzQnB8+w59ir2Jx3p8wfbAqMBAAGjqbYwqbMwPwYJYIZIAYb4QqENBDIWMEdl bmVyYXR1ZCBieSB0aGUqU2VjdXJpdHkqU2VydmVyIGZvciB6L09TIChSQUNGKTAq BgNVHREEGTAXgg9TMFcxLkRVUy5TRUcuREWHBMCoCWIwDgYDVR0PAQH/BAQDAgWg MB0GA1UdDgQWBBQlyjuoy6SipU3H23fH7cpw+ALB0zAfBgNVHSMEGDAWgBT/MgiN 4im65Gpt4iPBBGhEz1XpXzAhffEdq2iG9w0BAQsFAAOCAQEAkDFU531SDp31G1jH IPdA6w9MeJx344sqd/K4LPzfIGuzmmuHZrAHCHZNaA64BBMogeGOV2zoxenwf07A CIeTQpqE19TuNH2vyrulMd8p4c6VwUjto/N+GXobE3WmNt5nrdGLOIqrxutwmiMD 2HE101Ih7unsVqq24qfDczxHNVLapJ1Yy4qXiqC/UG8055GhjIwEaMvfEQ82GhcI v1pekhL7hK0p8xGOAYQVBUM0MrpVBCSiFYdVs2hPaTA86QcynqT9CGNrXf2JeTqk FIzH7h3nLdCRZd9KXQATQ5b24a9OXGzC6bKqiSD9unxWI8DYxBXOx3G3kufaXn2X kOE/EQ==

----END CERTIFICATE----









#### Base 64 encoded certificates can be decoded using OPENSSL:

Certificate: Data: Version: 3 (0x2) Serial Number: 4 (0x4) Signature Algorithm: sha256WithRSAEncryption ISSUER: C=DE, ST=NORTH RHINE WESTPHALIA, L=DUESSELDORF, O=SOFTWARE ENGINEERING GMBH, OU=DEVELOPMENT, CN=SOFTWARE ENGINEERING ROOT CA Validity Not Before: Jan 15 23:00:00 2023 GMT Not After : Apr 1 22:59:59 2025 GMT Subject: C=DE, ST=NORTH RHINE WESTPHALIA, L=DUESSELDORF, O=SOFTWARE ENGINEERING GMBH, OU=DEVELOPMENT, CN=DB2 SECURE DISTRIBUTION SERVICE Subject Public Key Info: Public Key Algorithm: rsaEncryption Public-Key: (2048 bit) Modulus: 00:fb:d3:1d:13:67:95:ac:aa:c2:ba:65:3c:b7:6f: e4:72:fb:12:1c:31:36:b0:3b:5f:8e:79:51:b4:e1: d3:8f:be:b6:44:b5:5b:2f:89:c4:bb:e2:f7:65:96: 29:41:23:53:98:9d:c1:cd:67:df:f7:13:c4:70:c5: 3b:f9:a9:2c:46:e0:a5:ba:3c:ab:20:77:64:29:93: b8:81:3f:8a:cc:5c:a9:2b:be:9f:2b:12:9d:34:39: 6d:0b:80:fb:ab:e2:c0:ad:42:91:32:b5:f6:29:b3: f0:05:84:dc:0f:a1:3d:61:2b:58:59:91:78:f2:c0: 82:e7:ef:fb:09:de:df:73:60:27:ac:0c:35:09:94: 42:64:cb:06:8e:f1:4f:d5:be:02:e1:d1:2b:04:ab: 66:94:95:ac:52:62:1c:9c:e0:a0:0f:b1:d7:a7:71: 4d:10:bb:34:99:85:bc:0c:c8:0f:b8:8b:47:92:2b: 87:e0:3a:c1:94:99:2a:00:09:57:38:4e:0d:2a:bc: 52:c5:ea:24:dc:3c:45:d6:f5:73:49:aa:55:46:aa: 0d:51:69:6b:8d:c0:a1:b4:d1:9d:25:11:36:54:6c: 05:59:b1:a9:b8:7b:18:98:9d:15:22:4f:a2:7d:a3: 23:df:cd:09:c1:f3:ec:39:f6:2a:f6:27:1d:e9:f3: 07:db Exponent: 65537 (0x10001) X509v3 extensions: Netscape Comment: Generated by the Security Server for z/OS (RACF) X509v3 Subject Alternative Name: DNS:SOW1.DUS.SEG.DE, IP Address:192.168.9.98 X509v3 Key Usage: critical Digital Signature, Key Encipherment X509v3 Subject Key Identifier: 25:CA:3B:A8:CB:A4:A2:A5:4D:C7:DB:77:C7:ED:CA:70:F8:02:C1:D3 X509v3 Authority Key Identifier: FF:32:08:8D:E2:29:BA:E4:6A:6D:E2:23:C1:04:68:44:CF:55:E9:5F Signature Algorithm: sha256WithRSAEncryption Signature Value: 90:31:54:e7:7d:52:0e:9d:e5:1b:58:c7:20:f7:40:eb:0f:4c: 78:9c:77:e3:8b:20:77:f2:b8:2c:fc:df:20:6b:b3:9a:6b:87: 66:b0:07:08:76:4d:68:0e:b8:04:13:28:81:e1:8e:57:6c:e8: c5:e9:f0:7f:4e:c0:08:87:93:42:9a:84:d7:d4:ee:34:7d:af: ca:bb:a5:31:df:29:e1:ce:95:c1:48:ed:a3:f3:7e:19:7a:1b: 13:75:a6:36:de:67:ad:d1:8b:38:8a:ab:c6:eb:70:9a:23:03: d8:71:25:3a:52:21:ee:e9:ec:56:aa:b6:e2:a7:c3:73:3c:47: 35:52:da:a4:99:58:cb:88:17:8a:a0:bf:50:6f:34:c3:b8:d0: 33:1c:04:68:cb:df:11:0f:36:1a:17:08:bf:5a:5e:92:12:fb: 84:ad:29:f3:11:8e:01:84:15:05:43:34:32:ba:55:04:24:a2: 15:87:55:b3:68:4f:69:30:3c:e9:07:32:9e:04:fd:08:63:6b: 5d:fd:89:79:38:24:14:8c:c7:ee:1d:e7:2d:d0:91:65:df:4a: 5d:00:13:43:96:f6:e1:af:4e:5c:6c:c2:e9:b2:a0:89:20:fd: ba:7c:56:23:c0:d8:c4:15:ce:c7:71:b7:92:e7:da:5e:7d:97: 90:e1:3f:11

