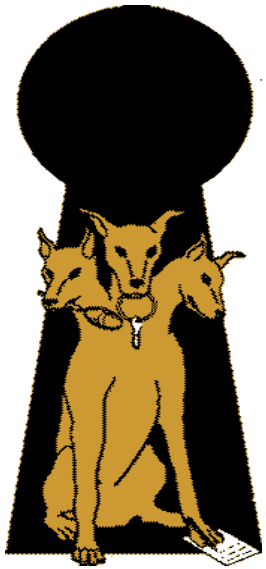
Kerberos on z/OS

Interaction with

Active Directory

On

Windows Server 2008



+

William Mosley
z/OS NAS Development
wmosley@us.ibm.com

December 2011

Agenda

- Updates to Windows Server 2008
- Setting up Cross-Realm Trust
- Using Active Directory as Primary KDC
- Miscellaneous Information
- Useful tools
- Session Summary

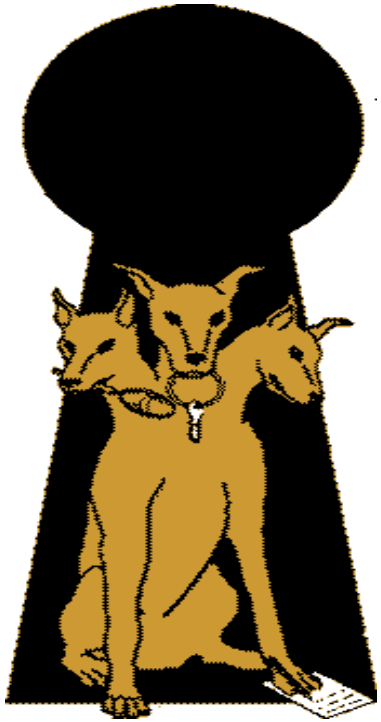
Trademarks

- The following are trademarks or registered trademarks of the International Business Machines Corporation:
  - IBM, DB2, OS/390, RACF, SecureWay, z/OS, AS/400, AIX
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.
- SOLARIS is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries
- Kerberos is a trademark of MIT
- Other company, product, and service names may be trademarks or service marks of others.

Changes in Windows Server 2008

AES

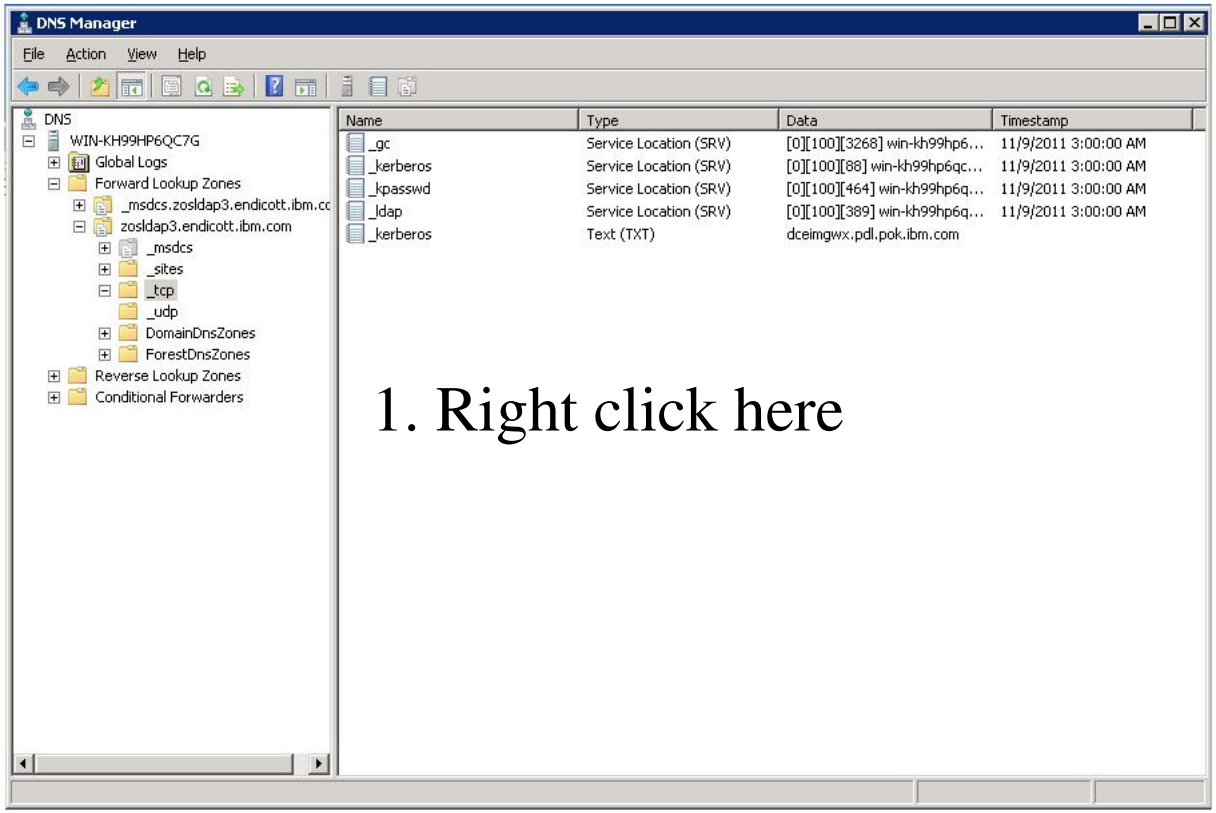
- Default for TGT, service key and session key
- GSSAPI support for AES



Setting up Cross-Realm Trust

1. Map z/OS KDC host name to Windows domain
2. Setup peer-to-peer relationship between Windows and z/OS
3. Make sure that the encryption types of the cross-realm TGT are compatible
4. Define location of the z/OS KDC on Windows
5. Restart Windows server for changes to take affect

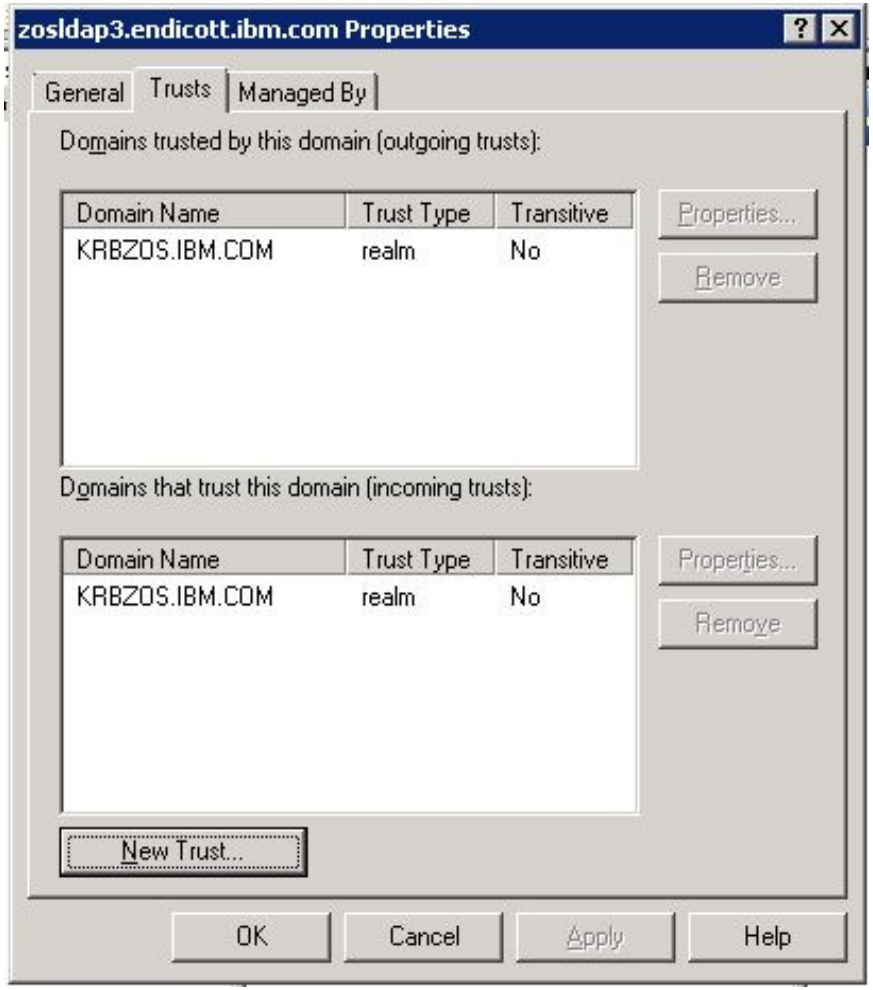
Mapping host name



2. Select “Other New Records”
3. Scroll down to “Text(TXT)”
4. Click Create Record
5. Record name is _kerberos
6. Text is domain name or IP address

Create a text record to map z/OS KDC to Windows domain controller for _udp and _tcp.

Domains and Trust



RALTER REALM
/.../KRBZOS.IBM.COM/krbtgt/KRB2008.IBM.COM
KERB(PASSWORD(Pa55w0rd))

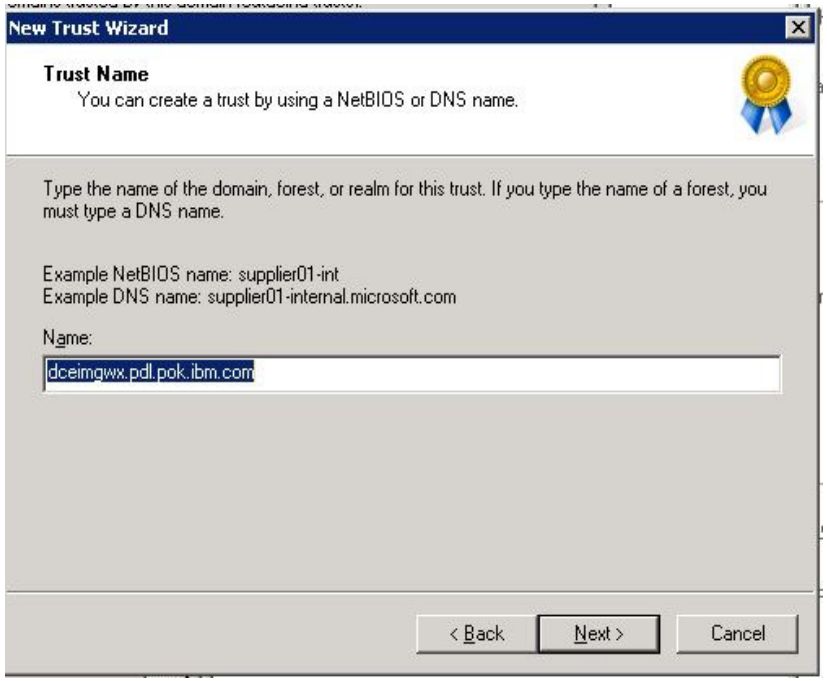
RALTER REALM
/.../KRBZOS.IBM.COM/krbtgt/KRB2008.IBM.COM
KERB(ENCRYPT(NODES NODESD NODES3 AES128
AES256))

RALTER REALM
/.../KRB2008.IBM.COM/krbtgt/KRBZOS.IBM.COM
KERB(PASSWORD(Pa55w0rd))

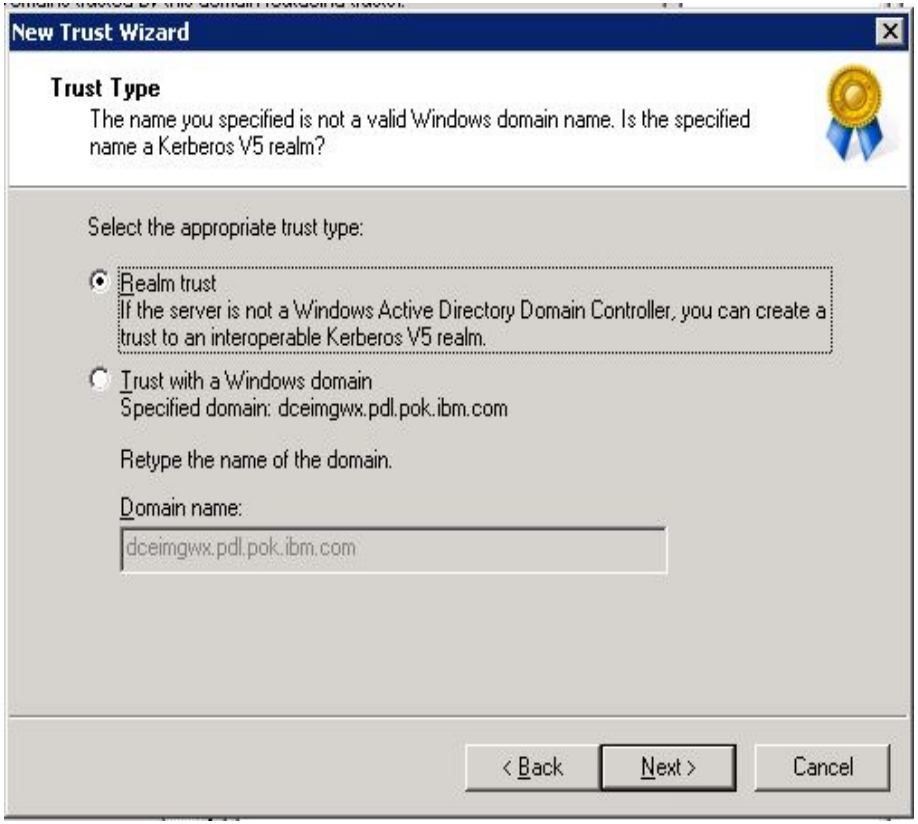
RALTER REALM
/.../KRB2008.IBM.COM/krbtgt/KRBZOS.IBM.COM
KERB(ENCRYPT(NODES NODESD NODES3 AES128
AES256))

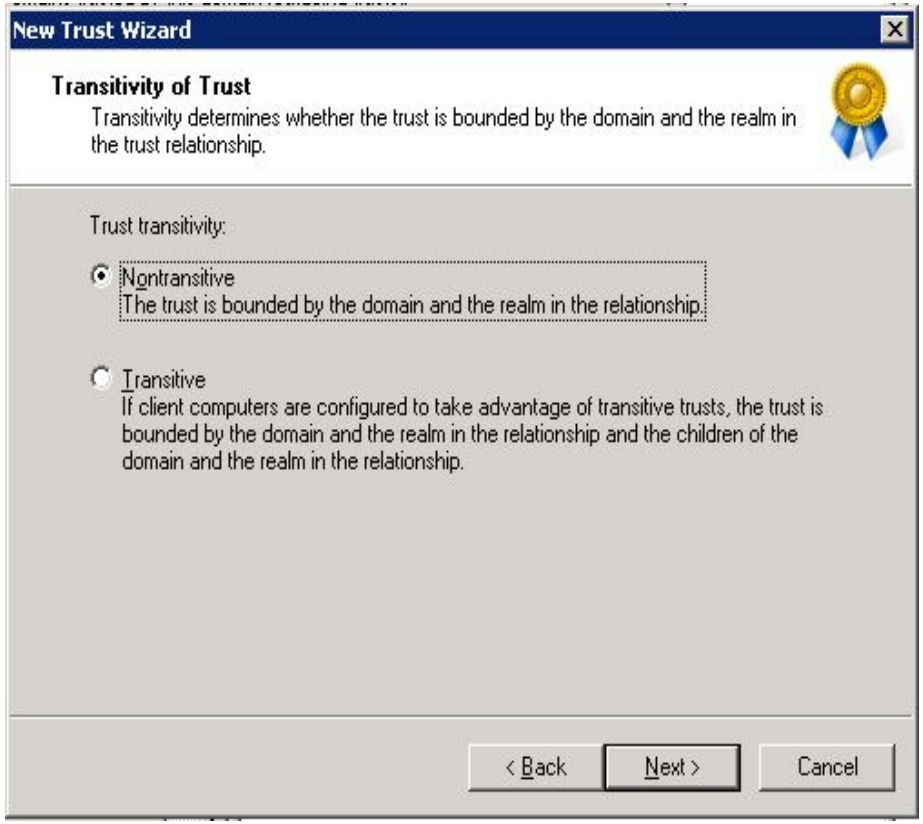
Password should match password in RACF REALM class

New Trust Wizard

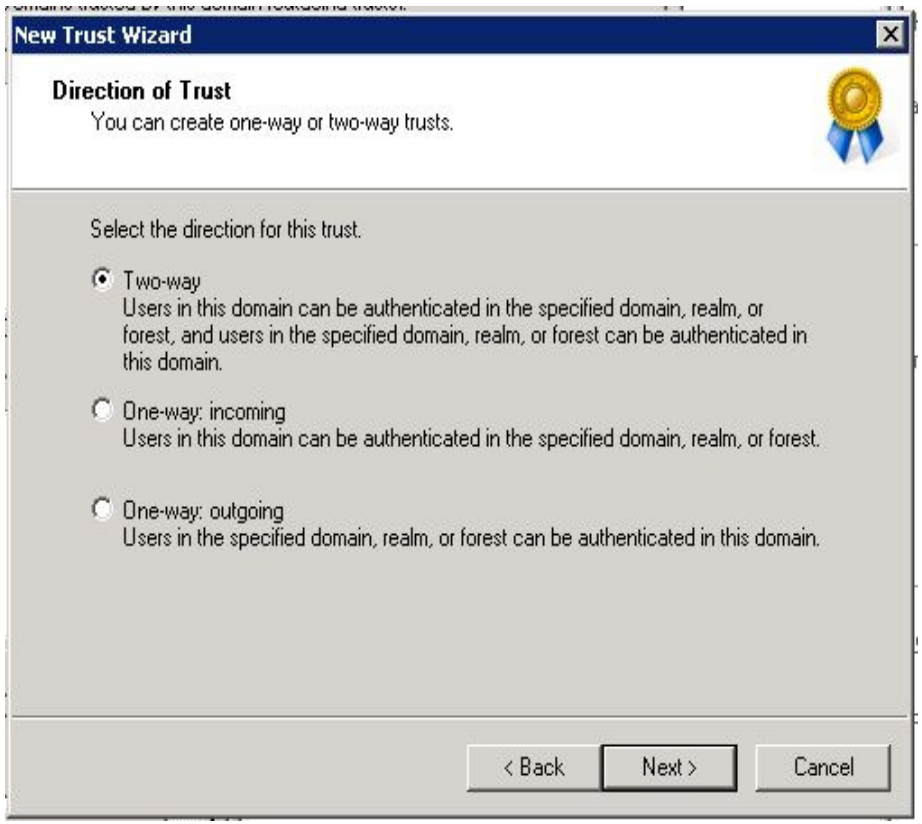


New Trust Wizard...

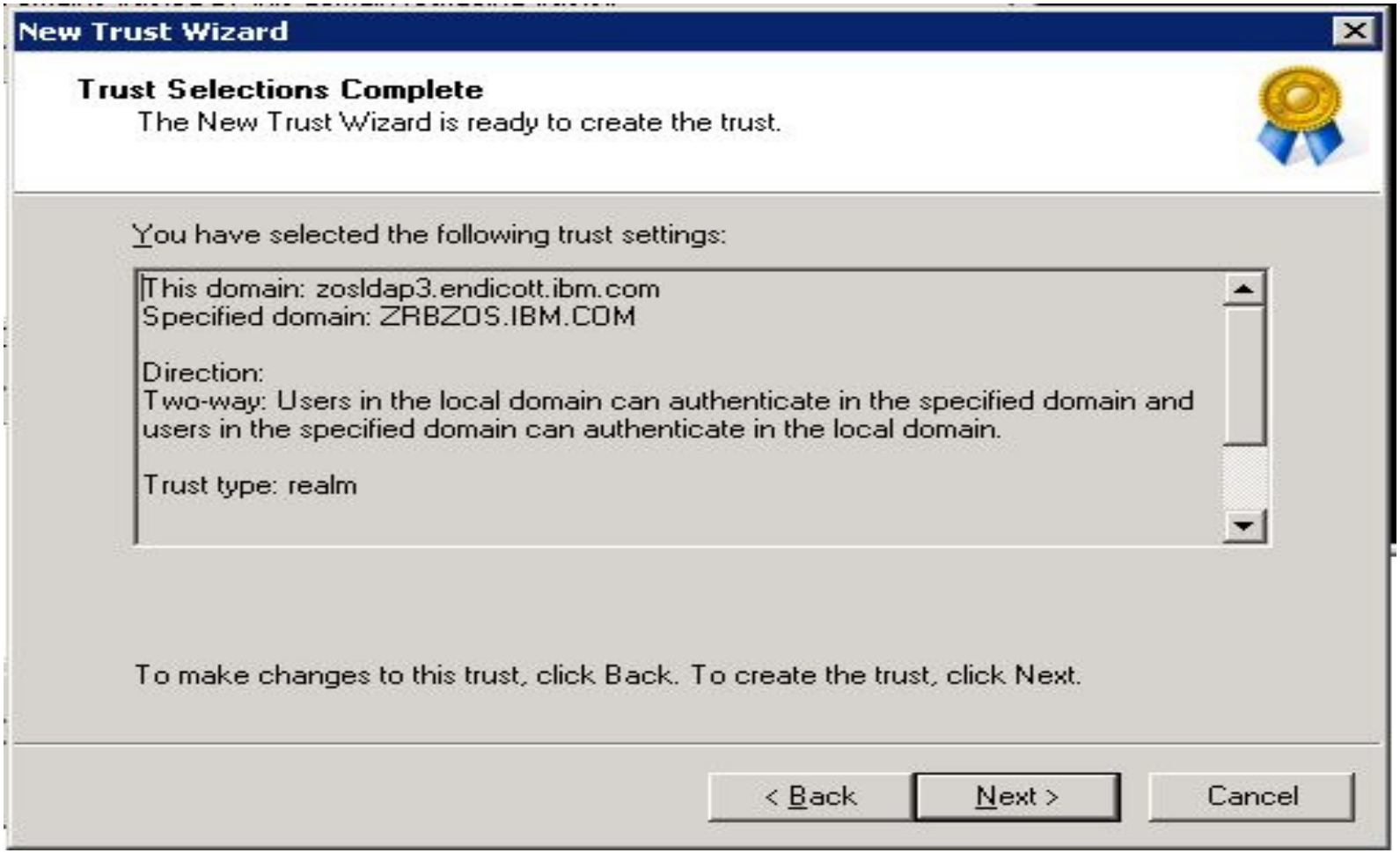




New Trust Wizard...



New Trust Wizard...



Define z/OS KDC on Windows

- ksetup /addkdc <realmName> [kdcName]
- ksetup /addkdc KRBZOS.IBM.COM dceimgwx.krbzos.ibm.com

Don't forget to restart the Windows server.

Using AD as Primary KDC

A service account associated with the remote application server must be created on the Windows Server.

1. Service Principal Name must be unique
2. Create SPN for application server
3. Export service key to keytab file
4. Transmit keytab file to remote machine
5. Merge keytab file

Checking for existing SPN's

- SPN consists of <service type>/<host name>
- Windows will allow you to create multiple SPN's without complaining
- If there is a duplicate SPN, the Kerberos api call will return the error code **0x96c73a07**
  - (Server principal is not found in security registry)

```
WINDOWS:To find duplicate SPN's ….
ldifde -f spn.out -l serviceprincipalname -r
"(serviceprincipalname=*)"
```

```
WINDOWS:To remove duplicate SPN's ….
setspn -d <service type>/<host name>  <account name of  SPN>
```

* Logoff and logon accout for changes to take affect

Creating a SPN

```
WINDOWS:To create a SPN...
ktpass princ<service-name>/<domain>@<REALM>
/crypto AES128-SHA1 /mapuser <account-name>

OR

setspn -a <service type>/<host name> <account name of  SPN>
```

* Logoff and logon for the change to take affect

Exporting service key to keytab file

- The key version number defined in AD must match the key version number of the key in the keytab file
- If the key version numbers do not match there will be an error

```
WINDOWS:To find key version number...
ldifde -f <file name> -t 3268 -l *,msDS-KeyVersionNumber
    -r "(servicePrincipalName=<service name>/<host name>*)"
    -p subtree
```

Look in the output file for msDS-KeyVersionNumber:<value>
Use this value in the following command.

```
WINDOWS:To export service key...
ktpass princ <service-name>/<domain>@<REALM>
 /crypto AES128-SHA1 /kvno <key-version number>
 /out <keytab.filename> /pass <account-password>
```

Transmitting and importing keytab file

- Ftp the keytab file to the remote machine running the application server
- Depending on the environment the keytab may be used as is or it may be merged with an existing keytab file.

```
ZOS:To merge keytab file...
keytab merge <file name>
```

Miscellaneous Information

- DES is disabled by default
- Z/OS does not support RC4
- Service names are not case sensitive.
- A kinit to the Windows KDC may be unsuccessful if preauthentication is required and the UDP network protocol is used.
  - Specify kdc_use_tcp = 1 in krb5.conf

Useful tools

- Kerbtray – GUI tool that displays ticket information
- Ldifde – useful for searching for service principal names and key version numbers
- Ktpass – export keytab file from windows to another machine
- Klist – views and deletes tickets granted to current logon session
- Ksetup – useful for configuring Windows for Kerberos interoperability
- Wireshark – useful for viewing Kerberos packets

References…

- **IBM Books**
- **SA22-7687 z/OS Security Server RACF Command Language Reference**
- **SC24-5926 z/OS Integrated Security Services Network Authentication and Privacy Service Administration**
- **SC24-5927 z/OS Integrated Security Services Network Authentication and Privacy Service Programming**

➢Internet

- http://web.mit.edu/kerberos/www/
- http://msdn.microsoft.com/en-us/library/ff649429.aspx
- http://technet.microsoft.com/en-us/library/cc749438%28WS.10%29.aspx
- **http://social.technet.microsoft.com/wiki/contents/articles/kerberos-interoperability-step-by-step-guide-for-windows-server-2003.aspx**

References

RFCs

- RFC 1510 - The Kerberos Network Authentication Service (V5)
- RFC 4120 - The Kerberos Network Authentication Service (V5)
- RFC 1964 - The Kerberos Version 5 GSS-API Mechanism
- RFC 2078 - Generic Security Service Application Program Interface (V2)
- RFC 2744 - Generic Security Service Application Program Interface (V2): C Bindings
- RFC 3962 - Advanced Encryption Standard (AES) Encryption for Kerberos
- RFC 4121 - The Kerberos V5 GSSAPI Mechanism: Version 2
- RFC 4537 – Kerberos Cryptosystem Negotiation Extension

- RFC 2025 - The Simple Public-Key GSS-API Mechanism (SPKM)
- RFC 2847 - LIPKEY - A low infrastructure mechanism Using SPKM
- RFC 3962 - Advanced Encryption Standard (AES) Encryption for Kerberos
- RFC 4121 - The Kerberos V5 GSSAPI Mechanism: Version 2
- RFC2253 UTF-8 String Representation of Distinguished names
- RFC2459 X.509 Public Key Infrastructure

Session Summary

- What we have covered:
  - Windows Server 2008 AD Kerberos changes
  - Overview of Cross-Realm setup
  - Setup z/OS Application server with AD
  - Miscellaneous info
  - Useful tools

Questions ?

Reference

SPKM-3

- **The Simple Public-Key GSS-API Mechanism (SPKM) is based on a public key infrastructure, not the Kerberos symmetric-key infrastructure**
  - **SPKM-3 does not use secure timestamps, enabling secure authentication in environments without access to secure time**
  - **Designed to be flexible, for example providing Algorithm Identifiers for specifying various algorithms to be used by communicating peers**
  - **Provides support for asymmetric algorithm-based digital signatures**
  - **Data formats and procedures are designed to be as similar to the Kerberos mechanism as possible for ease of implementation by applications which are already Kerberos enabled**
- **SPKM-3 uses the same certificate infrastructure as SSL**

LIPKEY

- **LIPKEY (a Low Infrastructure Public Key Mechanism using SPKM) is a GSS-API security mechanism which can be used when the initiator (client) does not have a certificate and instead uses user ID and password for authentication**
- **It consists of a client with no public key certificate, accessing a server with a public key certificate (in contrast, in SPKM-3, both client and server require access to certificates)**
- **The server must have access to a user ID/password repository (we use the __passwd system routine, with setup/restrictions documented in the z/OS Network Authentication Service Programming Guide)**

How LIPKEY works

A client using the LIPKEY mechanism

- Obtains the server's certificate
- Verifies that it was signed by a trusted CA
- Generates a random session symmetric key
- Encrypts the session key with the server's public key
- Sends the encrypted session key to the server
- At this point, the client and server have a secure channel, so the client can provide a user name and password for authentication

R_ticketserv (IRRSPK00)

- Parse or extract Kerberos principal
  - Function code
    - TKTS_RETURN_NAME (1) - Parse specified ticket and return Kerberos principal name
      - ☒ GSS-API context token is input
      - ☒ Principal name is output

R_usermap (IRRSIM00)

- Map application user
  - Function codes:
    - UMAP_R_TO_K (5) -- return the Kerberos application user identity for the supplied RACF user ID
    - UMAP_K_TO_R (6) -- return the RACF user ID associated with the supplied Kerberos application user identity

R_admin (IRRSEQ00)

- Functions supported
  - ADMN_ADD_USER, ADMN_ALT_USER, ADMN_LST_USER ADMN_ADD_GENRES, ADMN_ALT_GENRES, ADMN_LST_GENRES to support KERB segment fields
- Fields
  - KERBNAME - realm or principal name
  - MAXTKTLF - realm or principal maximum ticket life
  - MINTKTLF - realm wide minimum ticket life
  - DEFTKTLF - realm wide default ticket life
  - PASSWORD - realm password