z/OS 2.4 IBM Education Assistant (IEA)

Solution (Epic) Name: PKI EST support, NAS FAST support, NAS Crypto Currency

Element(s)/Component(s): PKI Services, NAS

Top Validation Item: No

Agenda

- Trademarks
- Session Objectives
- Overview
- Usage & Invocation
- Session Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- Additional Trademarks:
  - None

Session Objectives

- Provide continuous enhancements to fulfill customer requirements and product currency
- At the end of this presentation, you should have an understanding of the support from:
  - NAS Flexible Authentication Secure Tunneling (FAST) support (Epic 83031)
  - NAS Crypto Currency (Epic 199113)
  - PKI Enrollment over Secure Transport (EST) support (Epic 199103)
  - PKI Synchronous GENCERT from web page (Epic 91556)
- The following epics will be covered in the TLS 1.3 presentation together with other components
  - RACF RACDCERT TLS1.3 support (Epic 216629)
  - PKI TLS 1.3 support (Epic 199109)

NAS Flexible Authentication Secure Tunneling (FAST) support (Epic 83031)

Overview

- Kerberos is a protocol for verifying the identity of principals on an open network
- The protocol provides pre-authentication mechanisms through pre-authentication data (padata)
- One of the common padata type is Flexible Authentication Secure Tunneling (FAST)
- RFC6113 - A Generalized Framework for Kerberos Pre-Authentication - defines the implementation of FAST
- z/OS NAS implements the mandatory FAST requirements specified by RFC6113

Overview

- In order to support FAST, NAS needs to support parts from other RFCs. The key RFCs used are:
  - Kerberos Principal Name Canonicalization and Cross-Realm Referrals – RFC6806
  - Anonymity Support for Kerberos – RFC8062
  - Public Key Cryptography for Initial Authentication in Kerberos (PKINIT) – RFC4556
- FAST negotiation was introduced as a mechanism for a client to determine if a KDC supports FAST pre-authentication
  - Defined by RFC6806

Overview

- FAST employs an armoring scheme
  - The armor can be a Ticket Granting Ticket (TGT) obtained by the client's machine using the host keys to pre-authenticate with the KDC, or an anonymous TGT obtained based on anonymous PKINIT
  - The z/OS implementation only provides for the use of anonymous PKINIT, not a TGT based on host keys.

Overview

- Who (Audience)
  - z/OS NAS customers who wants to use an encrypted pre-authentication protocol without the need for client certificates
- What (Solution)
  - NAS will support FAST through two updated Kerberos user commands and a new Kerberos API
    - Command **kinit**
    - Command **klist**
    - API **krb5_set_fast_armor_ticket**
- Wow (Benefit / Value, Need Addressed)
  - This function will benefit Kerberos customers who want to achieve interoperability with other platforms which support the FAST protocol
  - Lift the burden to manage strong passwords for Kerberos authentication

Usage & Invocation

- Use kinit new
- Use kinit –n to request a fully anonymous PKINIT ticket (armor ticket)
- Use klist –f to list the ticket flags to see if it is an anonymous ticket
- Use kinit –T to use the anonymous PKINIT ticket for FAST pre-authentication

Usage & Invocation

- Example:
  - Obtain an anonymous PKINIT TGT and store it in a credential cache file named armor.cache in the /home/gumby directory:
    - kinit –n –c /home/gumby/**armor.cache**
  - List the ticket flags to see if it is an armor ticket
    - klist –f /home/gumby/armor.cache

    Ticket cache: FILE:/home/gumby/armor.cache
    Default principal: WELLKNOWN/ANONYMOUS@WELLKNOWN:ANONYMOUS

    Server: krbtgt/DCEIMGHG.PDL.POK.IBM.COM@DCEIMGHG.PDL.POK.IBM.COM
    Valid 2019/04/08-18:25:28 to 2019/04/09-04:25:28
    Flags: FIA**a**
  - Obtain a TGT for the Gumby principal using FAST pre-authentication (using the anonymous PKINIT armor ticket in /home/gumby/armor.cache)
    - kinit –T /home/gumby/armor.cache Gumby

NAS Crypto Currency (Epic 199113)

Overview

- RFC 8009 AES Encryption with HMAC-SHA2 for Kerberos 5 defines two encryption types and two checksum types using 128 bits and 256 bits AES keys
- The algorithms are used to produce more secure encryption, integrity protection, and checksum operations

Overview

- Who (Audience)
  - Kerberos Administrators who want to use the most current encryption algorithms in all the Kerberos operations
- What (Solution)
  - Support the two new encryption types in all Kerberos operations in KDC, client and server
    - aes128-cts-hmac-sha256-128
    - aes256-cts-hmac-sha384-192
- Wow (Benefit / Value, Need Addressed)
  - z/OS customers can run Kerberos with the most current algorithms

Usage & Invocation

- For Kerberos client, add the two new types in the following entries in krb5.conf

default_tkt_enctypes = aes256-cts-hmac-sha384-192,aes128-cts-hmac-sha256-128, aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1, des-hmac-sha1,des-cbc-md5,des-cbc-md4,des-cbc-crc

default_tgs_enctypes = aes256-cts-hmac-sha384-192,aes128-cts-hmac-sha256-128,aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1, des-hmac-sha1,des-cbc-md5,des-cbc-md4,des-cbc-crc

- For KDC, add the two new types in the following entry in the KDC envar file

SKDC_TKT_ENCTYPES= aes256-cts-hmac-sha384-192, aes128-cts-hmac-sha256-128,aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96,des3-cbc-sha1, des-hmac-sha1,des-cbc-md5,des-cbc-md4,des-cbc-crc

Usage & Invocation

- If the KDC's registry is SAF, whenever a user is added with the KERB segment, the two new encryption types AES128SHA2 and AES256SHA2 are included in the default list of key encryption type
  - **ADDUSER** JOE KERB(KERBNAME(JOE))
  - LISTUSER JOE NORACF KERB

    USER=JOE

    KERB INFORMATION

    KERBNAME= JOE

    KEY ENCRYPTION TYPE= DES DES3 DESD AES128 AES256 AES128SHA2 AES256SHA2
  - If you want to exclude them when the user is added, you need to specify explicitly with the 'NO' version string, eg
    - **ADDUSER** JOE KERB(KERBNAME(JOE) ENCRYPT(**NOAES256SHA2**))

    KERBNAME= JOE

    KEY ENCRYPTION TYPE= DES DES3 DESD AES128 AES256 AES128SHA2 NOAES256SHA2

Usage & Invocation

- If you want to exclude them after the user is added using the default, you can specify explicitly with the ‘NO’ version string in the ALTUSER command, eg
  - **ALTUSER** JOE PASSWORD(JOE) NOEXPIRE KERB(ENCRYPT(NOAES128SHA2)

KERBNAME= JOE

KEY ENCRYPTION TYPE= DES DES3 DESD AES128 AES256 NOAES128SHA2 AES256SHA2

- The new AES128SHA2 and AES256SHA2 ENCTYPEs are enable by default in new and existing KERB segments of USER profiles and REALM class profiles.
  - Although enabled by default, there will not be any AES128SHA2 or AES256SHA2 keys generated until the next password change for the USER or REALM profile.

Usage & Invocation

- Similarly when the profile in the REALM class is defined with the KERB segment, the two new encryption types AES128SHA2 and AES256SHA2 are included in the default list of key encryption type
  RDEFINE REALM /.../KERB10.ENDICOTT.IBM.COM/KRBTGT/KER2000.EDICOTT.IBM.COM KERB(PASSWORD(12345678)))
  - RLIST REALM /.../KERB10.ENDICOTT.IBM.COM/KRBTGT/KER2000.EDICOTT.IBM.COM KERB

```
CLASS     NAME
-----     ----
REALM     /.../KERB10.ENDICOTT.IBM.COM/KRBTGT/KER2000.ENDICOTT.IBM.COM
KERB INFORMATION
----------------
KEY VERSION= 001
KEY ENCRYPTION TYPE= NODES NODES3 NODESD NOAES128 NOAES256 AES128SHA2 AES256SHA2
CHECK ADDRESSES= NO
```

Usage & Invocation

- If you want to exclude them when the profile is defined, you need to specify explicitly with the ‘NO’ version string, eg
- **RDEFINE** REALM /.../KERB10.ENDICOTT.IBM.COM/KRBTGT/KER2000.E DICOTT.IBM.COM **KERB**(PASSWORD(12345678) ENCRYPT(**NOAES128SHA2**))
- RLIST REALM /.../KERB10.ENDICOTT.IBM.COM/KRBTGT/KER2000.EDICOTT.IBM.COM **KERB**

```
CLASS      NAME
-----      ----
REALM      /.../KERB10.ENDICOTT.IBM.COM/KRBTGT/KER2000.ENDICOTT.IBM.COM
KERB INFORMATION
----------------
KEY VERSION= 001
KEY ENCRYPTION TYPE= NODES NODES3 NODESD NOAES128 NOAES256 NOAES128SHA2 AES256SHA2
CHECK ADDRESSES= NO
```

Usage & Invocation

- You may also exclude them by specifying the ‘NO’ value string in the KERB segment of the RALTER command after the profile is defined, eg
- **RALTER** REALM /.../KERB10.ENDICOTT.IBM.COM/KRBTGT/KER2000.EDICOTT.IBM.COM KERB(PASSWORD(12345678) ENCRYPT(**NOAES256SHA2**))
- RLIST REALM /.../KERB10.ENDICOTT.IBM.COM/KRBTGT/KER2000.EDICOTT.IBM.COM **KERB**

```
CLASS     NAME
-----     ----
REALM     /.../KERB10.ENDICOTT.IBM.COM/KRBTGT/KER2000.ENDICOTT.IBM.COM
KERB INFORMATION
----------------
KEY VERSION= 001
KEY ENCRYPTION TYPE= NODES NODES3 NODESD NOAES128 NOAES256 AES128SHA2 NOAES256SHA2
CHECK ADDRESSES= NO
```

Usage & Invocation

- If you use the RACF Callable Service **R_kerbinfo** to retrieve and update Kerberos information stored in RACF, you need to increase the minimum length of the data in order to handle the new encryption types, otherwise the new types will not be returned
  - For USER profile: 838 bytes → 942 bytes
  - For REALM profile: 849 bytes → 953 bytes

PKI Enrollment over Secure Transport (EST) support (Epic 199103)

Overview

- Enrollment over Secure Transport (EST) is the successor to Simple Certificate Enrollment Protocol (SCEP).
- SCEP has not been standardized. EST is standardized by RFC7030.
- This protocol aims to provision certificates in a more robust manner than the traditional SCEP.
- It also supports ECC certificates. Cisco IOS Software and Cisco IOS XE already support EST.

Overview

- Who (Audience)
  - PKI administrator who wants to set up a PKI CA for the EST clients, like the Cisco routers, to request certificates
- What (Solution)
  - PKI Services can be set up as a EST CA to issue a certificate for an EST client if it
    - was preregistered by the PKI administrator, or
    - could authenticate itself with a client certificate, which is trusted by the PKI EST CA
- Wow (Benefit / Value, Need Addressed)
  - Provide more options for our PKI customers to request certificates
  - Enable clients with a large number of Cisco routers to obtain RSA and ECC certificates automatically

Usage & Invocation

- Make sure the instance of the CA is capable of accepting EST request. The CA certificate must meet one or the following requirements:
  - contain the id-kp-cmcRA extended key usage extension, or
  - contain Domain or IP in the Subject Alternate Name extension with a value that matches the host name or the IP address in the URI that was used for the request or
  - the Common Name in the Subject Distinguished Name with a value that matches the host name in the URI that was used for the request, if there is no Domain or IP is present in the Subject Alternate Name extension
- If your current CA fulfills the requirement, it can be used too

Usage & Invocation

- Update the pkiserv.conf file to enable the EST function

```
[CertPolicy]
# Enable the Enrollment over Secure Transport Protocol (EST)
# T = True, EST is enabled
# F = False, EST is disabled (default if not specified)
#
EnableEST=T

# Specify the full pathname containing the EST CA file in DER format, required if EnableEST=T
#
ESTCAFile=/var/pkiserv/estcafile.der

# Specify the template nickname corresponds to the EST template
# in pkiserv.tmpl or pkitmpl.xml for EST preregistration and
# certificate fulfillment. Maximum length is 8.
# Ignored if the value is greater than 8 characters or if
# EST is not enabled.
#
ESTTemplate=2YESTP
```

Usage & Invocation

- Update the HTTP configuration files used for server authentication and client authentication
  - Vhost file for server authentication, eg vhost443.conf

ScriptAliasMatch ^/.well-known/est/(.*) “/usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/pkiest”

Or, for a domain called employees

ScriptAliasMatch ^/.well-known/est/employees/(.*) “/usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/pkiest”

  - Vhost file for client authentication, eg vhost1443.conf

ScriptAliasMatch ^/.well-known/est/(.*) “/usr/lpp/pkiserv/PKIServ/clientauth-cgi-bin/pkiest”

Or, for a domain called employees

ScriptAliasMatch ^/.well-known/est/employees/(.*) “/usr/lpp/pkiserv/PKIServ/clientauth-cgi-bin/pkiest”

Usage & Invocation

- Preregister the EST client
  - use the EST preregistration template in pkiserv.tmpl or pkitmpl.xml – the shipped sample is called ‘2-Year EST Certificate – Preregistration’, after your customization
  - Use the pkiprereg utility (the same utility used by SCEP)
- If the client already has a certificate to identify itself, the preregistration process can be skipped
- Let the client know the URL that can reach to the EST CA to perform the three supported functions

Usage & Invocation

- Function **cacerts**: Get the EST CA certificates (Note: This is usually the first step the EST client needs if it has not already obtained the CA certificate(s) from other means.)

https://www.YourCompany.com:<server authentication port>/.well-known/est/cacerts, or

https://www.YourCompany.com:<server authentication port>/.well-known/est/*<ca_domain>*/cacerts

- Function **simpleenroll**: Request a certificate from the EST CA

https://www.YourCompany.com:<server authentication port>/.well-known/est/simpleenroll, or

https://www.YourCompany.com:<server authentication port>/.well-known/est/*<ca_domain>*/simpleenroll

- Function **simplereenroll**: Renew a certificate from the EST CA

https://www.YourCompany.com:<client authentication port>/.well-known/est/simplereenroll, or

https://www.YourCompany.com:<client authentication port>/.well-known/est/*<ca_domain>*/simplereenroll

PKI Synchronous GENCERT from web page (Epic 91556)

Overview

- When a requestor submits a request to PKI Services through the web page interface, they need to wait for the PKI daemon’s certificate creation timer event to kick in to create the certificate
- Although there is an attribute to control this mode of operation through the R_PKIServ callable service, this control is never exposed on the web page interface

Overview

- Who (Audience)
  - PKI administrator who wants to set up template for users to get the certificate right away after submitting the request through the web page
- What (Solution)
  - Provide a switch to request certificate synchronously through the PKI Services web interface
- Wow (Benefit / Value, Need Addressed)
  - To enable the automation of certificate generation and retrieval using the PKI web interface

Usage & Invocation

- If the web page is hosted through the HTTP server, update the pkiserv.tmpl file
  - Add the synchronous switch to the template that does not require administrator's approval on the request -- Absence of the <ADMINAPPROVE> tag

    <SYNCHRONOUS=Y>

- If the web page is hosted through the Websphere or Liberty server, update the pkitmpl.xml file
  - Add the synchronous switch to the template that does not require administrator's approval on the request – presence of the <tns:AutoApprove> tag

    <tns:Synchronous>Y</tns:Synchronous>

Session Summary

- Now you should have an understanding of the support from:
  - NAS Flexible Authentication Secure Tunneling (FAST) support (Epic 83031)
  - NAS Crypto Currency (Epic 199113)
  - PKI Enrollment over Secure Transport (EST) support (Epic 199103)
  - PKI Synchronous GENCERT from web page (Epic 91556)
- The following epics will be covered in the TLS 1.3 presentation together with other components
  - RACF RACDCERT TLS1.3 support (Epic 216629)
  - PKI TLS 1.3 support (Epic 199109)

Appendix

- Publication references
  - Security Server Command Language Reference
  - Security Server Callable Services
  - Cryptographic Services PKI Services Guide and Reference
  - Integrated Security Services Network Authentication Service Administration
  - Integrated Security Services Network Authentication Service Programming