z/OS 2.4 IBM Education Assistant (IEA)

Solution (Epic) Name: TLS 1.3 handshake and algorithm support

Element(s)/Component(s): System SSL and ITDS-LDAP

Agenda

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

Trademarks

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

Session Objectives

- At the end of this presentation, you will have an understanding of:
  - An overview of the TLS 1.3 protocol support.
  - An overview of how to configure a z/OS application to use the TLS 1.3 protocol support provided by System SSL.
  - How to configure ITDS-LDAP (server and client) to use the TLS 1.3 protocol support provided by System SSL.
  - Understand how these enhancements affect installation, migration and coexistence.

Overview

- Who (Audience)
  - Provide latest industry standard updates to the TLS protocol

- What (Solution)
  - Implement TLS 1.3 RFC 8446
  - Allow exploiters the ability to use TLS 1.3 in their client and server applications

- Wow (Benefit / Value, Need Addressed)
  - TLS 1.3 handshake is encrypted after initial client and server handshake messages
  - Less confusion with configuration of cipher specifications. Cipher specifications are no longer tied to the certificate that is in use
  - Removal of legacy/weak algorithms; focus on usage of Authenticated Encryption With Associated Data (AEAD)
  - Several key changes retrofitted back to TLS 1.2

Usage & Invocation – System SSL TLS 1.3 Support

- TLS 1.3 protocol is a major rewrite from prior protocol versions that has been in the works for many years and is now a formal RFC 8446.

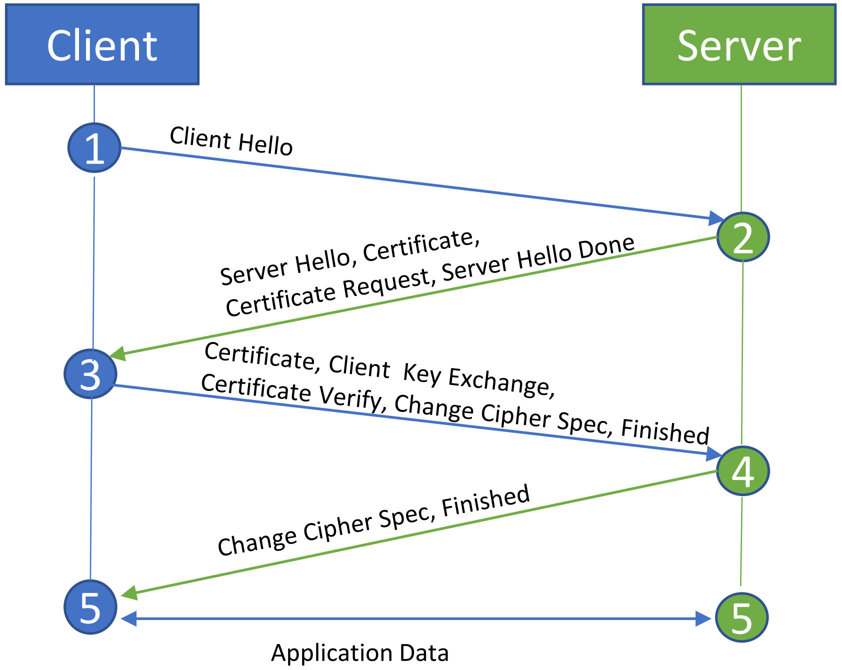
- TLS 1.3 promotes establishing secure connections faster (*elimination of unnecessary handshake steps*) as well as it addressed insecurities from prior protocol versions and forced use of newer encryption methods (*AEAD*).

- Under TLS 1.2 and earlier protocols, the initial connection opens up a dialog about which kind of encryption to use, which a server and client have to agree upon. Once agreed, they begin sharing encryption keys.

- TLS 1.3 eliminates the debate over what form of encryption to use. Instead, the initial connection is a statement from the client saying what it plans to access, the server provides an encryption key, the client provides a session key, and then the connection takes place.

Usage & Invocation – System SSL TLS 1.2 vs TLS 1.3

TLS 1.2 (Full handshake)



Client

Server

1 Client Hello

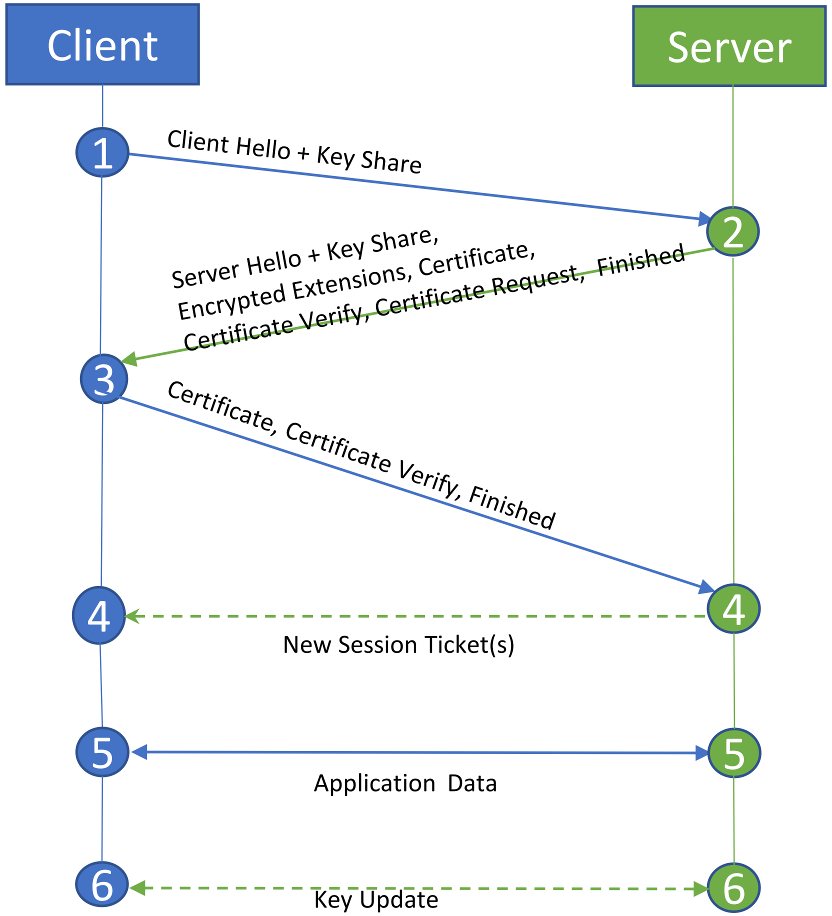
2 Server Hello, Certificate, Certificate Request, Server Hello Done

3 Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Finished

4 Change Cipher Spec, Finished

5 Application Data 5

TLS 1.3 (Full handshake)



Client

Server

1 Client Hello + Key Share

2 Server Hello + Key Share, Encrypted Extensions, Certificate, Certificate Verify, Certificate Request, Finished

3 Certificate, Certificate Verify, Finished

4 New Session Ticket(s) 4

5 Application Data 5

6 Key Update 6

Usage & Invocation – System SSL TLS 1.3 Support

- How is TLS 1.3 different then TLS 1.2 and earlier protocols?
  - TLS 1.3 encrypts the handshake messages once the initial hellos are exchanged. Earlier protocols did not encrypt until the FINISHED message.
  - Encrypted handshake messages are presented as payload messages (1703…). This means applications can no longer perform read ahead processing to determine if the message is a handshake, payload or alert message.
  - RSA key exchange has been removed and key exchanges being supported are ECDHE based which ensure perfect forward secrecy.
    - RSA based certificates can continue to be used to identify the endpoints of the connection. Certificates are now used for authentication purposes only.
    - When using RSA end entity certificates, RSASSA-PSS must be identified as an allowed signature algorithm
  - Key exchange driven through key share settings and not cipher definition
  - DSA, Brainpool, and DH certificates are not allowed with TLS 1.3

Usage & Invocation – System SSL TLS 1.3 Support

- TLS 1.3 support provided through existing APIs. No new APIs being added at this time.

- API external changes mainly isolated to the set/get attribute APIs.
  - gsk_attribute_[sg]et_enum()
  - gsk_attribute_[sg]et_buffer()

- Key update will continue to use the gsk_secure_socket_misc() API.
  - GSK_RESET_CIPHER – Updates the local read and write keys and instructs remote peer to do the same
  - GSK_RESET_WRITE_CIPHER – Updates the local write key and instructs remote peer to update its read key

- TLS 1.3 attributes not enabled by default.
  - Protocol enablement
  - Ciphers – not added to the default cipher list
  - Key share
  - Elliptical curves – Added x25519 group to end of default list

- Enablement can be done either through APIs or environment variables (if application supports 4 character ciphers)

Usage & Invocation – System SSL TLS 1.3 Support

- Unlike prior protocol versions, TLS 1.3 requires ICSF to be available.

- ICSF provides support for:
  - Elliptical curve processing – key pair generation, digital signature generation/verification and key derivation.
  - CHACHA20/Poly1305
  - AES-GCM

- TLS 1.3 is not currently supported in FIPS mode

- Minimum key sizes for local and remote peer's end entity certificates are enforced by System SSL:
  - RSA – 2048
  - ECC – 256 (secp256r1)
  - GSK_PEER_RSA_MIN_KEY_SIZE and GSK_PEER_ECC_MIN_KEY_SIZE settings will be used if more restrictive

- System SSL uses RFC 5280 certificate validation mode unless GSK_CERT_VALIDATION_MODE is explicitly set. (Overrides the current GSK_CERT_VALIDATION_MODE default of ANY)

Usage & Invocation – System SSL TLS 1.3 Extensions Supported by System SSL

Extension	Description
Certificate authorities	Provides information about the certificate authorities (CAs) supported by the end entity. If server is enabled for client authentication, the certificate authorities extension will be automatically encoded in the CERTIFICATE-REQUEST message using the CAs in the trusted certificate store.
Key share	Indicates the group(s) that contains the endpoint’s cryptographic parameters
Max fragment length (MFL)	Allow TLS clients and server to negotiate the maximum fragment length to be sent.
Pre-shared key	Used to negotiate the identity of the pre-shared key to be used with a given handshake in association with PSK key establishment.
PSK key exchange modes	Used along with the pre-shared key extension

Usage & Invocation – System SSL TLS 1.3 Extensions Supported by System SSL

Extension	Description
Signature algorithms	Specifies the list of hash and signature algorithm pair specifications that are supported by the client or server in order of preference for use in digital signatures of X.509 certificates and TLS handshake messages.
Signature algorithms cert	Specifies the list of hash and signature algorithm pair specifications that are supported by the client or server in order of preference for use in digital signatures of X.509 certificates.
Status request	Retrieve the OCSP revocation information for the remote peer’s certificate(s) Server will provide OCSP revocation support for the server’s certificate(s) when enabled for OCSP revocation checking via the GSK_OCSP_ENABLE and/or GSK_OCSP_URL setting and OCSP stapling has been enabled with the GSK_SERVER_OCSP_STAPLING setting.
Supported groups	Indicates the named groups that the client supports for key exchange
Supported versions	Used by the client to indicate which versions of TLS it supports and by the server to indicate which version it is using.

Usage & Invocation – System SSL TLS Extensions no longer supported for TLS 1.3

- Multi-Status Requests – Server side support only
- Renegotiation – replace by key update
- Truncated HMAC
- EC Points
- If System SSL application is configured for the above and TLS 1.3 is the only protocol enabled, the above extensions will not be included in the handshake message.
- If the partner application provides these extensions in the client hello, they will be silently ignored if TLS 1.3 is selected.

Usage & Invocation – System SSL TLS 1.3 Environment Variables and Attribute Types

Attribute	Description	Values	Comments
GSK_PROTOCOL_TLSV1_3 (New)	Specifies whether the TLS V1.3 protocol is supported.	Environment variable allowed settings: • ON, ENABLED, or 1 – Enables TLS 1.3 support • OFF, DISABLED, or 0 – Disables TLS 1.3 support gsk_attribute_[sg]et_enum() allowed settings (connection or environment): • GSK_PROTOCOL_TLSV1_3_ON • GSK_PROTOCOL_TLSV1_3_OFF	Default: OFF Enablement of TLS 1.3 disables TLS 1.0. If needed, must be explicitly enabled.
GSK_V3_CIPHER_SPECS_EXPANDED (Existing)	Specifies cipher specifications in order of preference as a string consisting of 1 or more 4-character values.	Updated to allow specification of 3 new TLS 1.3 4-character cipher specifications: • **1301** - TLS_AES_128_GCM_SHA256 • **1302 -** TLS_AES_256_GCM_SHA384 • **1303 -** TLS_CHACHA20_POLY1305_SHA256 gsk_attribute_[sg]et_buffer() (connection or environment)	Must include at least one TLS 1.3 cipher specification when TLS 1.3 is enabled 4-character cipher specifications must be enabled when TLS 1.3 is enabled. Application must call gsk_attribute_set_enum() with attribute GSK_V3_CIPHERS and enum value GSK_V3_CIPHERS_CHAR4

Usage & Invocation – System SSL TLS 1.3 Environment Variables and Attribute Types

Attribute	Description	Values	Comments
GSK_CLIENT_ECURVE_LIST (Existing)	Specifies the list of elliptic curves or supported groups that are supported by the client in order of preference for use. For TLS V1.2 and earlier protocols, this list is used by the client to guide the server as to which elliptic curves are preferred when using ECC-based cipher suites. For TLS V1.3, this list is used by the client to guide the server as to which elliptic curves are preferred and guide group selection for encryption and decryption of handshake messages.	Allowed values: • 0019 (secp192r1) • 0021 (secp224r1) • 0023 (secp256r1) • 0024 (secp384r1) • 0025 (secp521r1) • 0029 (x25519) • 0030 (x448) gsk_attribute_[sg]et_buffer() (connection or environment)	Default: 002100230024002500190029 If application is only enabled for TLS V1.3, the 0021 and 0019 elliptic curves or supported groups are silently ignored. If the application is not enabled for TLS V1.3, the 0029 and 0030 elliptic curves or supported groups are ignored
GSK_SERVER_TLS_KEY_SHARES (New) GSK_CLIENT_TLS_KEY_SHARES (New)	Specifies the list of the key share groups that are supported by either the server or client. The server uses the client's preferred key share group order and selects a group that is in common with the GSK_SERVER_TLS_KEY_SHARES list. The selected group is use to encrypt and decrypt TLS V1.3 handshake messages.	Allowed values: • 0023 (secp256r1) • 0024 (secp384r1) • 0025 (secp521r1) • 0029 (x25519) • 0030 (x448) Example: 002300240025 gsk_attribute_[sg]et_buffer() (connection or environment)	Default: No default value The client sends the key share groups that are in common and in the same order as the supported groups list (GSK_CLIENT_ECURVE_LIST).

Usage & Invocation – System SSL TLS 1.3 Environment Variables and Attribute Types

Attribute	Description	Values	Comments
GSK_TLS_SIG_ALG_PAIRS (Existing)	Specifies the list of hash and signature algorithm pair specifications that are supported by the client or server as a string consisting of 1 or more 4-character values in order of preference for use in digital signatures of X.509 certificates and TLS handshake messages. Signature algorithm pair specification only has relevance for sessions using TLS V1.2 and TLS V1.3.	Updated to allow specification of 3 new signature and hash algorithms (for RSASSA-PSS): • 0804: SHA-256 with RSASSA-PSS • 0805: SHA-384 with RSASSA-PSS • 0806: SHA-512 with RSASSA-PSS If only enabled for TLS 1.3, the following signature algorithms are not sent to the remote partner as they are not allowed in TLS 1.3: 0101, 0201, 0202, 0203, 0301, 0302, 0303 and 0402 • 0101 – RSA with MD5 • xx02 – DSA algorithms • 03yy – SHA-224 gsk_attribute_[sg]et_buffer() (connection or environment)	Default: 0601060305010503040104030402030103030302020102030202080608050804 The RSASSA-PSS signature and hash algorithms have been added to the end of the default list.

Usage & Invocation – System SSL TLS 1.3 Environment Variables and Attribute Types

Attribute	Description	Values	Comments
GSK_TLS_CERT_SIG_ALG_PAIRS (New)	Specifies the list of hash and signature algorithm pair specifications that are supported by the client or server in order of preference for use in digital signatures of X.509 certificates. Certificate signature algorithm pair specification only has relevance for TLS V1.2 client or TLS V1.3 client and server sessions.	Allowed values: Same values that are allowed with GSK_TLS_SIG_ALG_PAIRS in addition to the following new ones for RSASSA- PSS: • 0804: SHA-256 with RSASSA-PSS • 0805: SHA-384 with RSASSA-PSS • 0806: SHA-512 with RSASSA-PSS If only enabled for TLS 1.3, the following signature algorithms are not sent to the remote partner as they are not allowed in TLS 1.3: 0101, 0201, 0202, 0203, 0301, 0302, 0303 and 0402 • xx01 – MD5 algorithms • xx02 – DSA algorithms • 03yy – SHA-224 gsk_attribute_[sg]et_buffer() (connection or environment)	Default: No default value Optional setting The GSK_TLS_CERT_SIG_ALG_PAIRS setting overrides the GSK_TLS_SIG_ALG_PAIRS setting when checking the digital signatures of the remote peer's X.509 certificates. The certificate signature algorithm pair specification only has relevance for sessions using TLS V1.2 or higher protocols.

Usage & Invocation – System SSL TLS 1.3 Environment Variables and Attribute Types

Attribute	Description	Values	Comments
GSK_CERT_VALIDATION_MODE (Existing)	Specifies which Internet Standard is to be used for certificate validation.	Allowed values: • ANY – Certificate validation against RFC 2459 initially – if that fails, validate against RFC 3280 – if that fails validate against RFC 5280 • 2459 – Certificate validation against RFC 2459 only • 3280 - Certificate validation against RFC 3280 only • 5280 - Certificate validation against RFC 5280 only gsk_attribute_[sg]et_enum() (environment)	Default: ANY If TLS V1.3 is negotiated for a secure connection, certificate validation is done according to RFC 5280 unless specifically set.

Usage & Invocation – System SSL TLS 1.3 Middlebox Compatibility Mode

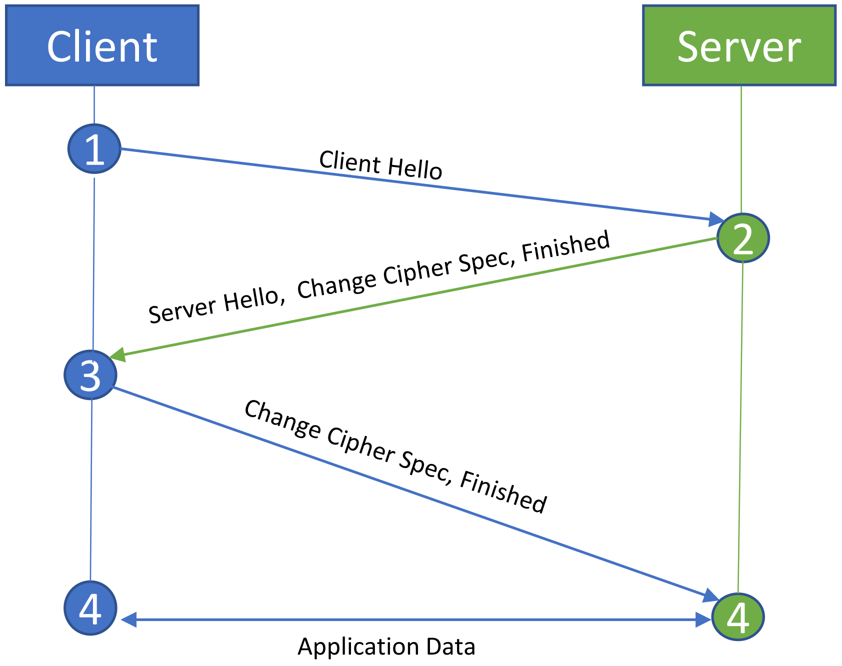
- Early TLS 1.3 drafts changed the CLIENT-HELLO and SERVER-HELLO message formats which caused middleboxes issues parsing these messages
- IETF TLS working group introduced a middlebox compatibility mode:
  - Restored the hello message formats to what they were in protocols prior to TLS 1.3
  - If client is enabled for compatibility mode:
    - A non-empty session ID is provided in the CLIENT-HELLO message
    - A dummy CHANGE-CIPHER-SPEC message is sent before client sends its encrypted handshake messages
  - Server can send a dummy CHANGE-CIPHER-SPEC message after its SERVER-HELLO or HELLO-RETRY-REQUEST messages
  - These updates make the TLS 1.3 handshake process resemble TLS 1.2 session resumption

Usage & Invocation – System SSL TLS 1.3 Environment Variables and Attribute Types

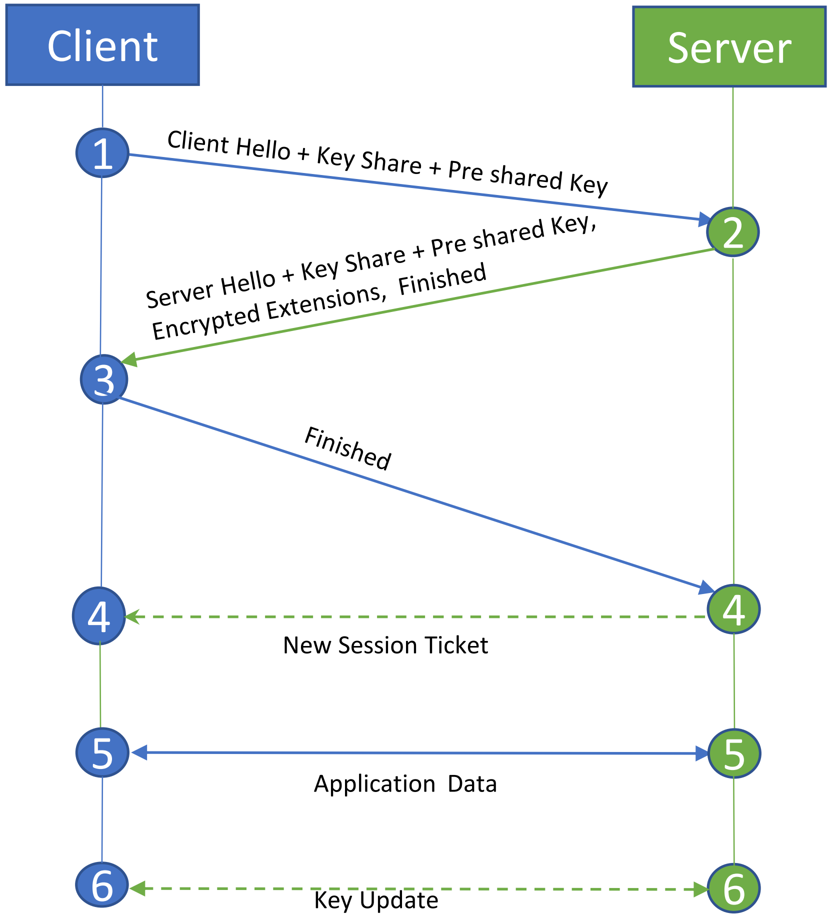
Attribute	Description	Values	Comments
GSK_MIDDLEBOX_COMPAT_MODE (New)	Specifies if the TLS V1.3 handshake process ought to use or tolerate handshake messages in a manner compliant with earlier TLS protocols to alleviate possible issues with middleboxes or proxies	Environment variable allowed settings: • ON, ENABLED, or 1 – Indicates that the TLS V1.3 handshake process ought to use or tolerate handshake messages in a manner compliant with earlier TLS protocols to alleviate possible issues with middleboxes or proxies • OFF, DISABLED, or 0 – Indicates that the TLS V1.3 handshake process ought to use the pure TLS V1.3 handshake message format gsk_attribute_[sg]et_enum() allowed settings (environment): • GSK_MIDDLEBOX_COMPAT_MODE_ON • GSK_MIDDLEBOX_COMPAT_MODE_OFF	Default: OFF

Usage & Invocation – System SSL TLS 1.2 vs TLS 1.3

TLS 1.2 (Cached/Resumed handshake)


Client
Server
1
Client Hello
2
Server Hello, Change Cipher Spec, Finished
3
Change Cipher Spec, Finished
4
4
Application Data

TLS 1.3 (Resumed handshake)


Client
Server
1
Client Hello + Key Share + Pre shared Key
2
Server Hello + Key Share + Pre shared Key,
Encrypted Extensions, Finished
3
Finished
4
4
New Session Ticket
5
5
Application Data
6
6
Key Update

Usage & Invocation – System SSL TLS 1.3 session resumption

- TLS 1.3 has changed the way cached handshakes or session resumption occurs
- In protocols prior to TLS 1.3, both client and server maintained a session ID cache
  - Prior to establishing a new connection, client would search its cache to see if a connection had previously been negotiated with the desired server.
    - If not, the session ID in the client's initial handshake (client hello) message is empty
    - If one is found and the new connection attributes are the same as a prior cached session, the session ID is retrieved and put into its initial handshake (client hello) message.
  - Server receives the client hello message
    - If it contains a session ID, server checks its cache to ensure that the connection attributes are the same. If it is, server retrieves the state of the prior client connection from its cache.
    - If there is not a session ID present, a new session ID is generated by the server and stored in its cache for the new connection.
    - Server sends either the new or accepted Session ID to the client in the server's initial handshake (server hello) message.

Usage & Invocation – System SSL TLS 1.3 session resumption

- TLS 1.3 now uses what are called session tickets to allow for session resumption to occur
    - Session tickets may now optionally be sent from the server to the client after a successful handshake
        - Encrypted by the server and session tickets contain full session state
        - Server does not store session tickets in its own cache
        - Have a defined lifetime
        - Server may send any number of session tickets any time after a TLS 1.3 handshake is completed
        - Client is fully responsible for maintaining the session tickets sent to it from the server.
    - Prior to establishing a new connection, client would search its cache to see if a connection had previously been negotiated with the desired server.
        - If a session ticket is not found in its cache, a full handshake is done with the server.
        - If a session ticket is found and the new connection attributes are the same as a prior cached session, the ticket is included in the client's initial handshake (client hello) message as a pre-shared key extension
        - If server accepts the ticket, it is indicated in the server's initial handshake (server hello) message.
        - Server may issue additional session tickets after a successful resumption handshake

Usage & Invocation – System SSL TLS 1.3 session resumption

- Session ticket server configuration settings
  - Session tickets created by server after completed handshake
  - Number of session tickets to send after completed handshake
  - Session ticket lifetime
  - Encryption algorithm to be used for encrypting tickets
  - Encryption key refresh interval
- Session ticket client configuration settings
  - Session tickets allowed to be stored in cache
  - Maximum session ticket size allowed to be stored in cache
- System SSL server applications automatically send session tickets after a successful TLS 1.3 handshake
  - Controlled by session ticket count after a full TLS 1.3 handshake
  - Successful TLS 1.3 resumption handshake results in a new session ticket being sent
- Server applications can also control the sending of session tickets by using the new GSK_SEND_SESSION_TICKET attribute ID on the gsk_secure_socket_misc() routine.
- Client applications may now need to call gsk_secure_socket_read() to read in any session tickets received after the TLS 1.3 handshake completes in gsk_secure_socket_init()
- Session reuse – Will continue to work the same way as it does with protocols prior to TLS 1.3

Usage & Invocation – System SSL TLS 1.3 Environment Variables and Attribute Types

Attribute	Description	Values	Comments
GSK_SESSION_TICKET_CLIENT_ENABLE (New)	Specifies if the client supports caching session tickets received from a server after a TLS V1.3 handshake has completed and if it supports TLS V1.3 resumption attempts to the server.	Environment variable allowed settings: • ON, ENABLED, or 1 – Enables client session ticket caching • OFF, DISABLED, or 0 – Disables client session ticket caching gsk_attribute_[sg]et_enum() allowed settings: (environment) • GSK_SESSION_TICKET_CLIENT_ENABLE_ON • GSK_SESSION_TICKET_CLIENT_ENABLE_OFF	Default: ON
GSK_SESSION_TICKET_CLIENT_MAXSIZE (New)	Specifies the maximum size in bytes of a session ticket that can be stored in the client session ticket cache. Setting the maximum session ticket size too small could implicitly disable session ticket caching on the client side.	The valid sizes are 0 through 2147483647. gsk_attribute_[sg]et_numeric_value() (environment)	Default: 8192 A value of 0 disables checking the session ticket size and allows a session ticket of any size.

Usage & Invocation – System SSL TLS 1.3 Environment Variables and Attribute Types

Attribute	Description	Values	Comments
GSK_SESSION_TICKET_SERVER_ALGORITHM (New)	Specifies the algorithm to be used by the server to encrypt and decrypt the session tickets used for TLS V1.3 and later session resumption.	Environment variable allowed settings: • AESCBC128 – Uses AES-CBC 128-bit encryption • AESCBC256 – Uses AES-CBC 256-bit encryption gsk_attribute_[sg]et_enum() allowed settings (environment): • GSK_SESSION_TICKET_SERVER_ALGORITHM_AESCBC128 • GSK_SESSION_TICKET_SERVER_ALGORITHM_AESCBC256	Default: AESCBC128
GSK_SESSION_TICKET_SERVER_COUNT (New)	Specifies the number of TLS V1.3 session tickets that will be sent by the server after the initial handshake has completed. Each subsequent resumed handshake will send a single session ticket to replace the one used for resumption.	The valid sizes are 0 through 16. gsk_attribute_[sg]et_numeric_value() (environment)	Default: 2

Usage & Invocation – System SSL TLS 1.3 Environment Variables and Attribute Types

Attribute	Description	Values	Comments
GSK_SESSION_TICKET_SERVER_ENABLE (New)	Specifies if the server supports sending session tickets after a TLS V1.3 handshake has completed, and if it will accept resumption attempts from the client.	Environment variable allowed settings: • ON, ENABLED, or 1 – Enables TLS V1.3 server session ticket resumption • OFF, DISABLED, or 0 – Disables TLS V1.3 server session ticket resumption gsk_attribute_[sg]et_enum() allowed settings (environment): • GSK_SESSION_TICKET_SERVER_ENABLE_ON • GSK_SESSION_TICKET_SERVER_ENABLE_OFF	Default: On

Usage & Invocation – System SSL TLS 1.3 Environment Variables and Attribute Types

Attribute	Description	Values	Comments
GSK_SESSION_TICKET_SERVER_KEY_REFRESH (New)	Specifies the key refresh interval of the encryption key used by the server to encrypt session tickets, in seconds. In order to encrypt and decrypt session tickets, GSK_SESSION_TICKET_SERVER_ENABLE must be ON and the server must be configured to send session tickets, either via GSK_SESSION_TICKET_SERVER_COUNT or via the GSK_SEND_SESSION_TICKET option in **gsk_secure_socket_misc().** When the encryption key is refreshed, and a new primary encryption key is generated, the former encryption key will be retained as a secondary key that can be used only for decryption until the subsequent refresh occurs. When the ticket is decrypted, it will only accept the ticket if the GSK_SESSION_TICKET_SERVER_TIMEOUT has not yet passed.	The valid values are 0 through 604800 gsk_attribute_[sg]et_numeric_value() (environment)	Default: 300 A value of 0 disables session ticket encryption key refresh

Usage & Invocation – System SSL TLS 1.3 Environment Variables and Attribute Types

Attribute	Description	Values	Comments
GSK_SESSION_TICKET_SERVER_TIMEOUT (New)	Specifies the maximum lifetime (in seconds) of a resumed session from the time of the initial handshake. The server will continue to generate new session tickets for each new resumed handshake until the timeout has been reached, provided GSK_SESSION_TICKET_SERVER_COUNT is greater than 0 and GSK_SESSION_TICKET_SERVER_ENABLE is set to ON. Each session ticket generated by the server will be valid until the timeout has passed. Because the key used for encryption must be available when the client attempts resumption, the GSK_SESSION_TICKET_SERVER_KEY_REFRESH value will impact the lifetime of a session ticket.	The valid values are 1 through 604800 gsk_attribute_[sg]et_numeric_value() (environment)	Default: 300

Usage & Invocation – System SSL TLS 1.3 Environment Variables and Attribute Types

Attribute	Description	Values	Comments
GSK_V3_SIDCACHE_SIZE (Modified)	Specifies the size in number of entries in the SSL V3 to TLS V1.2 session identifier and TLS V1.3 session ticket cache. The oldest entry will be removed when the cache is full to add a new entry. The range is 0-64000 and defaults to 512. Session identifiers and session tickets are not remembered if a value of 0 is specified. For the SSL V3, TLS V1.0, TLS V1.1, and TLS V1.2 protocols, the cache stores session identifiers for use on the server and client sides. For the TLS V1.3 protocol on the client side, the cache is used to store session tickets when GSK_SESSION_TICKET_CLIENT_ENABLE is set to ON. The session identifier and session ticket cache is allocated by using the requested size rounded up to a power of 2 with a minimum size of 16.	The valid values are 0 through 64000 gsk_attribute_[sg]et_numeric_value() (environment)	Default: 512 A value of 0 disables session identifier and session ticket caching

Usage & Invocation – System SSL TLS 1.3 Environment Variables and Attribute Types

Attribute	Description	Values	Comments
GSK_V3_SESSION_TIMEOUT (Modified)	Specifies the session timeout value in seconds for the SSL V3 to TLS V1.2 session identifiers and TLS V1.3 session tickets in the cache. This is the number of seconds until an SSL V3, TLS V1.0, TLS V1.1, and TLS V1.2 session identifier or TLS V1.3 session ticket expires. The range is 0-86400 and defaults to 86400. System SSL remembers SSL V3, TLS V1.0, TLS V1.1, and TLS V1.2 session identifiers or TLS V1.3 session tickets for this amount of time. This reduces the amount of data exchanged during the SSL handshake when a complete initial handshake has already been performed. Session identifiers and session tickets are not remembered if a value of 0 is specified.	The valid values are 0 through 86400 gsk_attribute_[sg]et_numeric_value() (environment)	Default: 86400 Session identifiers and session tickets are not remembered if a value of 0 is specified.

Usage & Invocation – System SSL TLS 1.2 changes

- TLS 1.3 RFC 8446 specifies several updates that also apply to TLS 1.2 implementations
  - A limited version downgrade protection mechanism is provided within the SERVER-HELLO message
  - Signature_algorithms_cert extension
    - This extension allows a client to indicate which signature algorithms that it can validate in X.509 certificates. GSK_TLS_CERT_SIG_ALG_PAIRS setting provides this support in System SSL
  - Allow RSA certificates with an RSASSA-PSS signature algorithm to be used on TLS 1.2 handshakes
    - RSASSA-PSS signature algorithms can now be specified on the GSK_TLS_SIG_ALG_PAIRS and GSK_TLS_CERT_SIG_ALG_PAIRS settings
  - Supported_versions extension for enhanced protocol negotiation
    - When enabled for TLS 1.2 and later, client will send the supported_versions extension with all enabled TLS protocols
    - When enabled for TLS 1.2 and later, server will use the TLS protocol versions indicated in the client's supported_versions extension to negotiate the TLS protocol that is to be used for the handshake. In prior protocols the client was only able to indicate support for its highest enabled protocol.

Usage & Invocation – System SSL Sample TLS 1.3 Configurations

- Server configuration:
  - Client authentication is not enabled
  - Support various clients that may connect to the server with any of the following characteristics:
    - Only allow TLS 1.3 connections with any allowed TLS 1.3 cipher specification in this preference order: 130313021301
    - Allow secp256r1, secp384r1, or secp521r1 for the key share groups that are used to encrypt/decrypt TLS 1.3 handshake messages
    - Allow TLS 1.3 handshake messages to be signed by any of the following signature algorithms
      - SHA-256 with RSASSA-PSS
      - SHA-384 with RSASSA-PSS
      - SHA-512 with RSASSA-PSS
  - Server uses a RSA 2048-bit certificate that has a signature algorithm of RSA with SHA-256
- Client configuration:
  - Enable only TLS 1.3
  - Use TLS 1.3 cipher specification 1302
  - Use secp256r1 for the negotiated key share group for encrypting the TLS 1.3 handshake messages
  - Allow secp256r1, secp384r1, and secp521r1 for the supported groups
  - Allow the server certificate chain to contain certificates that use a signature algorithm of RSA with SHA-256
  - Indicate support for the SHA-384 with RSASSA-PSS signature algorithm on the TLS 1.3 handshake messages

Usage & Invocation – System SSL Sample TLS 1.3 Configurations

Environment Variable Setting	Server Configuration	Client configuration
GSK_PROTOCOL_TLSV1_3	ON	ON
GSK_V3_CIPHER_SPECS_EXPANDED Note: 4-character cipher specifications must be enabled in application via a call to gsk_attribute_set_enum() with attribute ID GSK_V3_CIPHERS and enum value GSK_V3_CIPHERS_CHAR4	130313021301	1302
GSK_SERVER_TLS_KEY_SHARES	002100220023	<Not set>
GSK_CLIENT_TLS_KEY_SHARES	<Not set>	0021
GSK_CLIENT_ECURVE_LIST	<Not set>	002100220023
GSK_TLS_SIG_ALG_PAIRS	080408050806	0805
GSK_TLS_CERT_SIG_ALG_PAIRS	0401	0401

Usage & Invocation – System SSL Sample TLS 1.3 Configurations

- Server configuration:
    - Client authentication is not enabled
    - Support various clients that may connect to the server with any of the following characteristics:
        - Only allow TLS 1.3 connections with any allowed TLS 1.3 cipher specification in this preference order: 130313021301
        - Allow secp256r1, secp384r1, or secp521r1 for the key share groups that are used to encrypt/decrypt TLS 1.3 handshake messages
        - Allow TLS 1.3 handshake messages to be signed by any of the following signature algorithms
            - SHA-256 with RSASSA-PSS
            - SHA-384 with RSASSA-PSS
            - SHA-512 with RSASSA-PSS
    - Server uses a RSA 4096-bit certificate that has a signature algorithm of RSA with SHA-512.
    - Session ticket configuration
        - Use AESCBC256 encryption support
        - Allow session tickets to be generated and send 4 session tickets after a successful full TLS 1.3 handshake
        - Set the server key refresh to 30 minutes (1800 seconds)
        - Set the session ticket lifetime to 30 minutes (1800 seconds)

Usage & Invocation – System SSL Sample TLS 1.3 Configurations

- Client configuration:
  - Enable only TLS 1.3
  - Use TLS 1.3 cipher specification 1303
  - Use secp521r1 for the negotiated key share group for encrypting the TLS 1.3 handshake messages
  - Allow secp256r1, secp384r1, and secp521r1 for the supported groups
  - Allow the server certificate chain to contain certificates that use a signature algorithms of RSA with SHA-512
  - Indicate support for the SHA-512 with RSASSA-PSS signature algorithm on the TLS 1.3 handshake messages
  - Client session ticket support
    - Allow session tickets to be stored in the cache with a maximum size of 4096 bytes
    - Specify a cache entry and session ticket lifetime of 15 minutes (900 seconds)
    - Allow 256 cache entries

Usage & Invocation – System SSL Sample TLS 1.3 Configurations

Environment Variable Setting	Server Configuration	Client configuration
GSK_PROTOCOL_TLSV1_3	ON	ON
GSK_V3_CIPHER_SPECS_EXPANDED Note: 4-character cipher specifications must be enabled in application via a call to gsk_attribute_set_enum() with attribute ID GSK_V3_CIPHERS and enum value GSK_V3_CIPHERS_CHAR4	130313021301	1303
GSK_SERVER_TLS_KEY_SHARES	002100220023	<Not set>
GSK_CLIENT_TLS_KEY_SHARES	<Not set>	0023
GSK_CLIENT_ECURVE_LIST	<Not set>	002100220023
GSK_TLS_SIG_ALG_PAIRS	080408050806	0806
GSK_TLS_CERT_SIG_ALG_PAIRS	0601	0601

Usage & Invocation – System SSL Sample TLS 1.3 Configurations

Environment Variable Setting	Server Configuration	Client configuration
GSK_SESSION_TICKET_SERVER_ALGORITHM	AESCBC256	<Not set>
GSK_SESSION_TICKET_SERVER_ENABLE	ON	<Not set>
GSK_SESSION_TICKET_SERVER_COUNT	4	<Not set>
GSK_SESSION_TICKET_SERVER_KEY_REFRESH	1800	<Not set>
GSK_SESSION_TICKET_SERVER_TIMEOUT	1800	<Not set>
GSK_SESSION_TICKET_CLIENT_ENABLE	<Not set>	ON
GSK_SESSION_TICKET_CLIENT_MAXSIZE	<Not set>	4096
GSK_V3_SESSION_TIMEOUT	<Not set>	900
GSK_V3_SIDCACHE_SIZE	<Not set>	256

Overview – ITDS-LDAP TLS 1.3 Support

- ITDS LDAP server allows LDAP clients and applications to secure connections using SSL (V2/V3) and TLS (V1.0, V1.1 and V1.2) protocols

- In z/OS V2R4, ITDS will exploit the System SSL TLS 1.3 support through environment variables

Usage & Invocation – ITDS-LDAP TLS 1.3 Server Support

- The **sslCipherSpecs** server configuration option should be set to GSK_V3_CIPHER_SPECS_EXPANDED to allow the TLS 1.3 cipher specifications (1301, 1302, or 1303) to be used.
- Server configuration file, in the general section:

```
sslCipherSpecs GSK_V3_CIPHER_SPECS_EXPANDED
```

- Server environment variables file:

```
GSK_PROTOCOL_TLSV1_3=ON
LDAP_SSL_CIPHER_FORMAT=CHAR4
GSK_SERVER_TLS_KEY_SHARES=002300240025
GSK_V3_CIPHER_SPECS_EXPANDED=130113021303
```

Usage & Invocation – ITDS-LDAP client C-API

- The **ldap_set_option**() routine can be used to explicitly set the cipher specifications with the **LDAP_OPT_SSL_CIPHER_EXPANDED** option

- The cipher specifications can be specified using the list of z/OS System SSL 4 character values in string form.
  - ldap_set_option(ld, LDAP_OPT_SSL_CIPHER_EXPANDED, "1301");
  - ldap_set_option(ld, LDAP_OPT_SSL_CIPHER_FORMAT,
    LDAP_SSL_CIPHER_FORMAT_CHAR4);

Usage & Invocation – ITDS-LDAP command line utilities

- The following LDAP command line utilities which use the C-API will use the environment variables:
    - ldapchangepwd
    - ldapcompare
    - ldapdelete
    - ldapmodify/ldapadd
    - ldapmodrdn
    - ldapsearch

    - db2pwden
    - ds2ldif -r (remote option, using extended operation)
    - ldapexop

Usage & Invocation – ITDS-LDAP command line utilities

- Example
  - export LDAP_SSL_CIPHER_FORMAT=CHAR4
  - export GSK_PROTOCOL_TLSV1_3=ON
  - export GSK_V3_CIPHER_SPECS_EXPANDED=1301
  - export GSK_CLIENT_TLS_KEY_SHARES=0023

  - ldapsearch -p 636 -Z -K *my.kdb* -P my*kdbpw* -N *mykeylabel* -D *bindDN* -w *mybindpw* -b *basedn* “objectclass=*”

Interactions & Dependencies

- To exploit this item, all systems in the Plex must be at the new z/OS level: No

- Software Dependencies
  - ICSF is required to use TLS 1.3

- Hardware Dependencies
  - None

- Exploiters
  - AT-TLS (Application Transparent – TLS)
  - IBM Tivoli Directory Server (ITDS) for z/OS

Session Summary

- You should now be able to:
  - Understand the TLS 1.3 support provided by System SSL and exploited by the IBM Tivoli Directory Server (ITDS) for z/OS (client and server)
  - Be able to identify relevant documentation
- Any Questions?

Appendix

- z/OS Cryptographic Services System SSL Programming
- IBM Tivoli Directory Server Plug-in Reference for z/OS
- IBM Tivoli Directory Server Administration and Use for z/OS
- IBM Tivoli Directory Server Messages and Codes for z/OS
- IBM Tivoli Directory Server Client Programming for z/OS