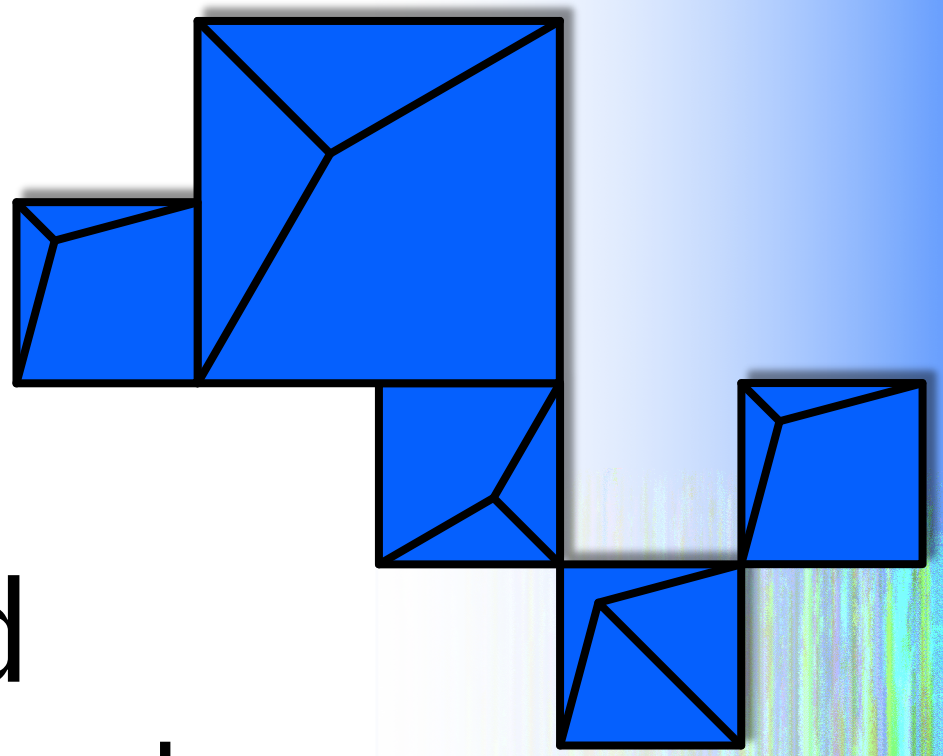


z/TPF Communication and Security Enhancements

Raymond Fan
z/TPF Development

Disclaimer

Any reference to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

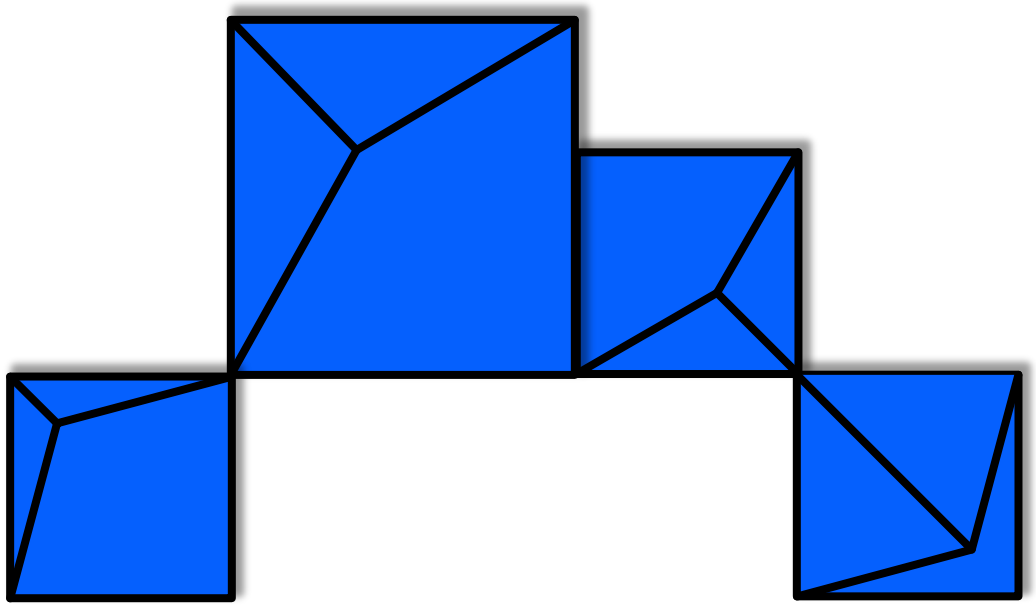
Agenda

Recent Deliverables

- OpenSSL Upgrade
- User Exits for logging HTTP request/responses, including REST messages
- SHA-256 Digital Signature support
- z/TPF HTTP Server Enhancements
- High-speed connector Enhancements

What's next?

- Hardware Compression Support for the HTTP server



OpenSSL Upgrade

Problem Statement

OpenSSL 1.0.2j is no longer supported by the OpenSSL community end of year 2019.

Value Statement

- Upgrading support from OpenSSL 1.0.2j to OpenSSL 1.1.1b improves the security of the z/TPF system.

- OpenSSL 1.1.1b is supported by the OpenSSL community. By upgrading, z/TPF can readily continue to pick up changes (i.e. security vulnerability patches) from the community moving forward.

Technical Details

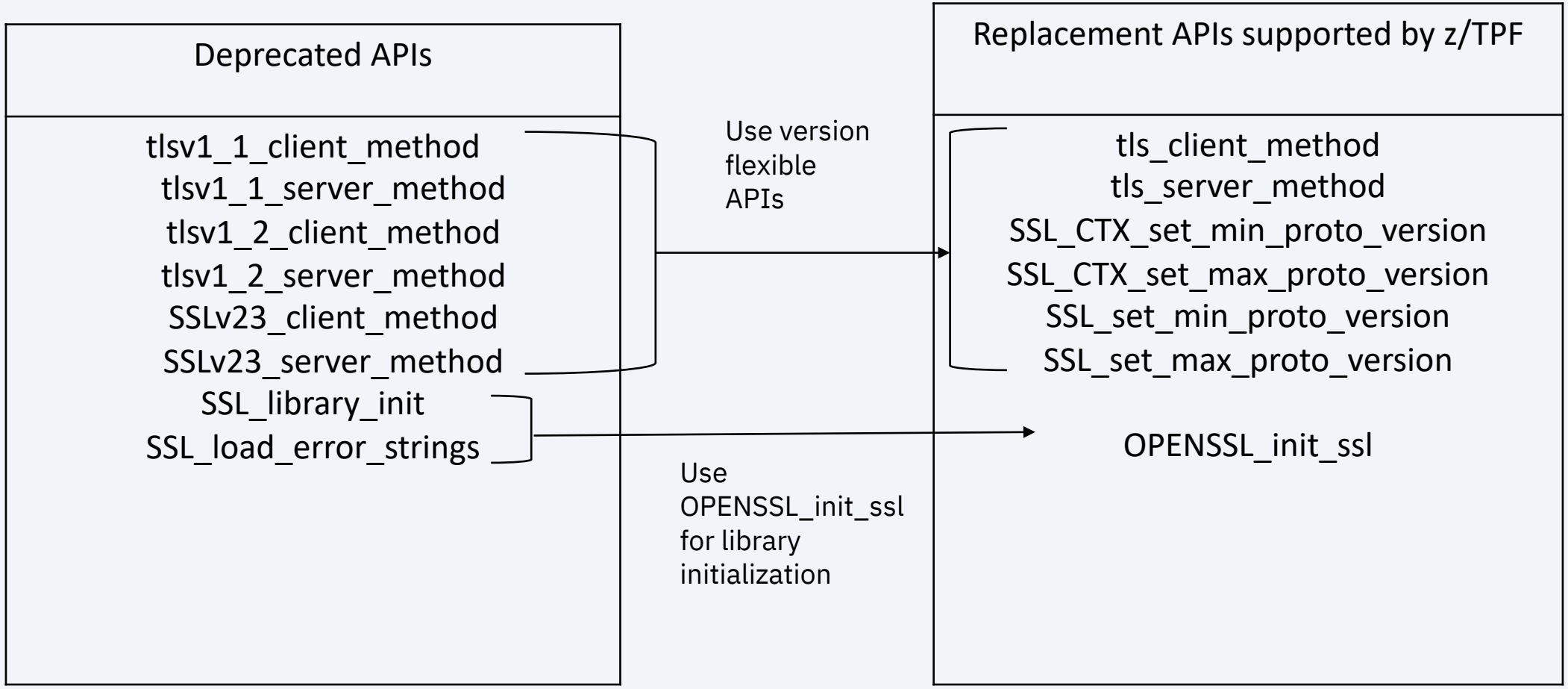
- Cipher algorithms
  - DES no longer supported by OpenSSL. DES-CBC-SHA cipher no longer supported.
    - **DES is insecure.**
    - Software implementation of DES was removed from OpenSSL code.
    - Applications using DES-CBC-SHA cipher must be updated to use a more current cipher prior to installing OpenSSL APARs.
  - z/TPF APIs dependent on DES in software have been updated
    - tpf_cryptc and CRYPC returns error if using DES or DESCBC and the hardware support for DES is not available on the processor.
    - CIFRC assembler macro and cifrc() C function obsoleted. Use tpf_cryptc or CRYPC instead.

Technical Details

- Migration Considerations
  - z/TPF applications using OpenSSL do not need to be re-compiled. Documented OpenSSL APIs for z/TPF still work.
  - SSL portions of Apache will no longer function or compile if SSL modules of z/TPF Apache server are being used.
    - z/TPF dropped support for Apache in November 2018.
  - APIs deprecated by OpenSSL community will generate compiler warnings upon rebuilding applications that use them.
    - Deprecated APIs still exist but are not guaranteed to exist in the next OpenSSL version.

Technical Details

- APIs
  - OpenSSL APIs that replace deprecated APIs are supported by z/TPF.



Technical Details

- Software and Configuration Changes
  - VERSION parameter of application configuration files for SSL now indicates the minimum SSL version allowed for the application.
  - Applies to z/TPF middleware packages (MQ, MongoDB, INETD, HTTP Client, Enhanced HTTP Client, FTP Client).
  - Application configuration files are saved on the z/TPF file system in /etc/ssl.

Example:
Prior to SSL upgrade, if a secure HTTP server application configuration file for SSL specified a VERSION of TLSv1, it only needed TLSv1 ciphers.

With the SSL upgrade clients can now connect to the secure HTTP server using TLSv1.1 or TLSv1.2 as well (TLSv1=minimum version allowed). May need to update cipher list.

See documentation for SSL_set_cipher_list for a list of z/TPF supported ciphers for each version of TLS.

https://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.2020/gtpc2/cpp_ssl_set_cipher_list.html

Technical Details

- If using z/TPF OpenLDAP, OpenSSL 1.1.1b requires OpenLDAP 2.4.48.
  - If using OpenSSL and CONFIG SIP macro: OPENLDAP=yes, this update is required as the current OpenLDAP 2.4.37 will no longer compile.
  - APEDIT recommends that OpenLDAP 2.4.48 is installed before the OpenSSL APARs to minimize change. New OpenLDAP version builds and functions against either OpenSSL 1.0.2j or OpenSSL 1.1.1b.
  - If installed in advance, the OpenLDAP code must be again rebuilt and reloaded as part of the OpenSSL APAR installation.
  - Alternatively, can apply the OpenLDAP and OpenSSL upgrade at the same time to avoid double rebuilding of OpenLDAP.
  - For more information, see “Installing OpenLDAP” section of z/TPF Knowledge Center.”

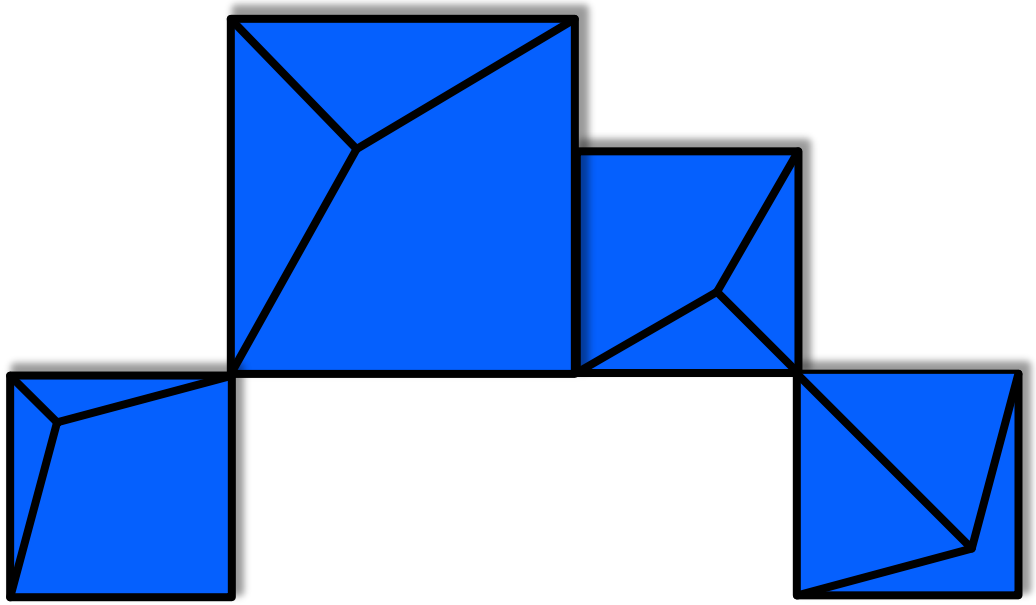
    https://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.2019/gtpl3/l3installingopenldaponztpf.html

Technical Details

- PJ45904 (Nov 2019) – Updates to libcurl package
  - Must be applied along with the two other OpenSSL APARs PJ45794 (Nov 2019) and PJ45842 (Nov 2019).
  - Required for libcurl to compile and function with two aforementioned APARs.

Conclusion

- z/TPF's version of OpenSSL is current with the OpenSSL community, enabling us to respond quickly to future security concerns.
- Upgraded OpenSSL improves the overall security of z/TPF system.
- **See APEDIT which identifies the Migration Considerations related to OpenSSL upgrade.**
- Delivered with PJ45794, PJ45842, PJ45904 (Nov 2019)



User Exits for Logging HTTP Request/Responses, including REST messages

Problem Statement

z/TPF HTTP Server and z/TPF Enhanced HTTP client lacks mechanism that allows for customers to log individual requests and responses.

Pain Points

- Current support lacks ability to log inbound or outbound z/TPF HTTP messages.

- Currently, if logging were to be called from an application, some information has been parsed out (i.e. original message that flows on the wire). That information is not available for logging.

Technical Details

- Two new z/TPF user exits for raw HTTP messages as well as REST messages.
  - UHCR (client) is invoked when a z/TPF HTTP client request is sent to the network and again when the response is received.
  - UHSR (server) is invoked when an inbound HTTP request is received by the HTTP server and again when the application sends the HTTP response.
  - If user exit was called for logging a request, it will always be called for logging the response, regardless of whether the response was successful or not (i.e. timeout).

Technical Details

- Data passed into the user exits.
  - Pointer to the HTTP message sent or received from the network.

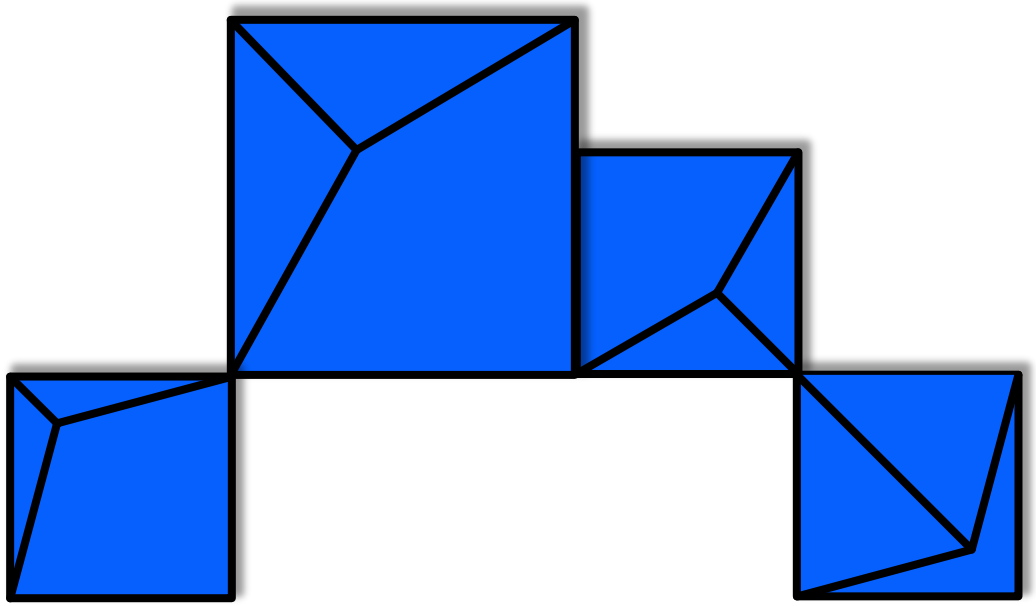
  - Inbound or outbound request type.
  - Socket descriptor the message came on.
  - Indication if connection was SSL connection or not.
  - Structure containing remote and local IP/Port.
  - HTTP server request structure. NULL if response.
  - HTTP server response structure. NULL if request.

Setup routines take care of parsing this data out and passing it right into the user exit.

Conclusion

- Users can now log inbound or outbound z/TPF HTTP messages, including REST consumer and provider messages using their own enterprise logging infrastructure.

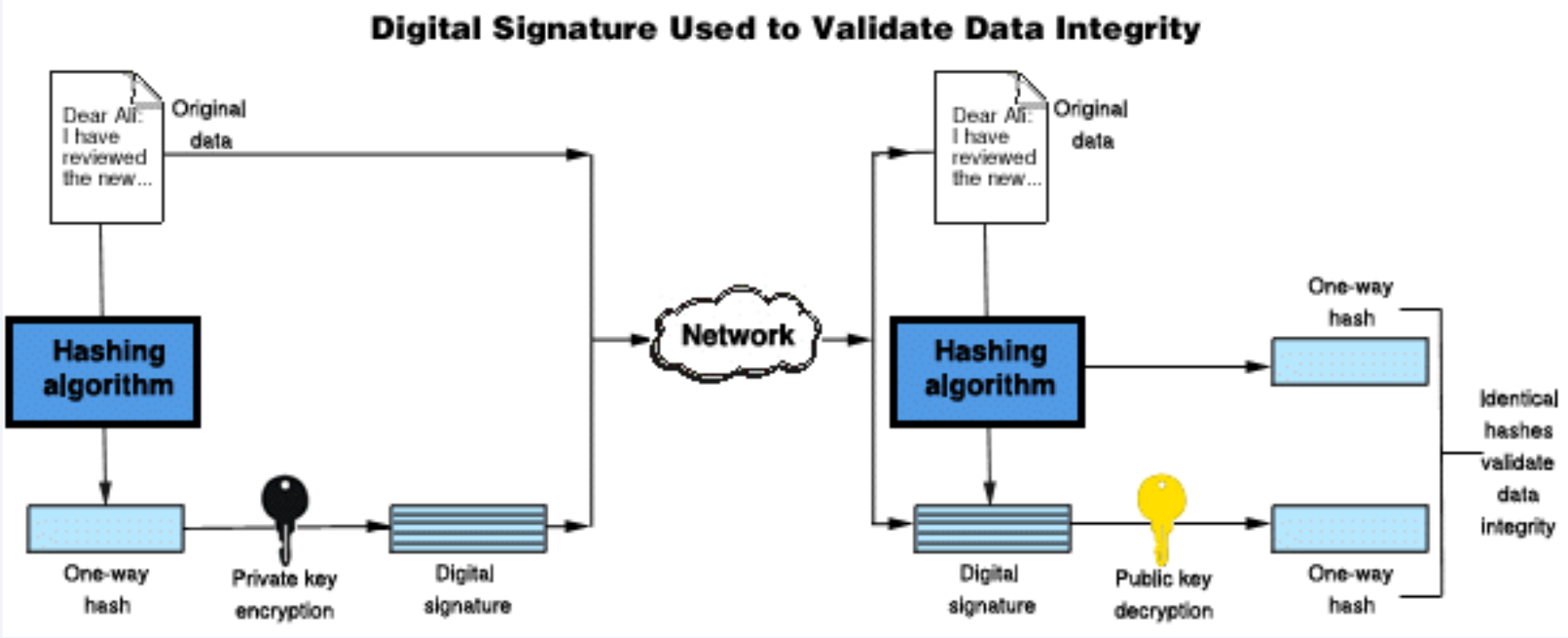
- Delivered with PJ45786 (Nov 2019).



SHA-256 Digital Signature Support

Background

- Digital signatures address data integrity issue (tampering and impersonation).



Background

- Digital signature support was available on z/TPF since July 2009.
  - tpf_RSA_sign_data
  - tpf_RSA_sign_digest
  - tpf_RSA_verify_data
  - tpf_RSA_verify_digest

- Used SHA1 or MD5 as message digest (hashing) algorithm.

Problem Statement

The message digest algorithms used by the four APIs z/TPF provides for creating and verifying a digital signature do not comply with the latest security standards.

Technical Details

- 2 new APIs for signing/verifying contiguous data
  - tpf_RSA_sign
  - tpf_RSA_verify
- 6 new APIs for signing/verifying discontiguous data
  - tpf_RSA_sign_init
  - tpf_RSA_sign_update
  - tpf_RSA_sign_final
  - tpf_RSA_verify_init
  - tpf_RSA_verify_update
  - tpf_RSA_verify_final

- New APIS use SHA-256 as the message digest algorithm, which is compliant to the latest security standards.

Technical Details

- Contiguous APIs
  - Makes it easy to sign or verify in a single step.

- Discontiguous APIs
  - APIs for signing/verifying discontiguous data requires initialization of a context which you would update with data, before finalizing the creation or verification of a digital signature.
  - Can be used when data is not all available at once, for example if it is being obtained in chunks through the wire.
  - Discontiguous APIs are similar to OpenSSL’s method of creating a hash of discontiguous data

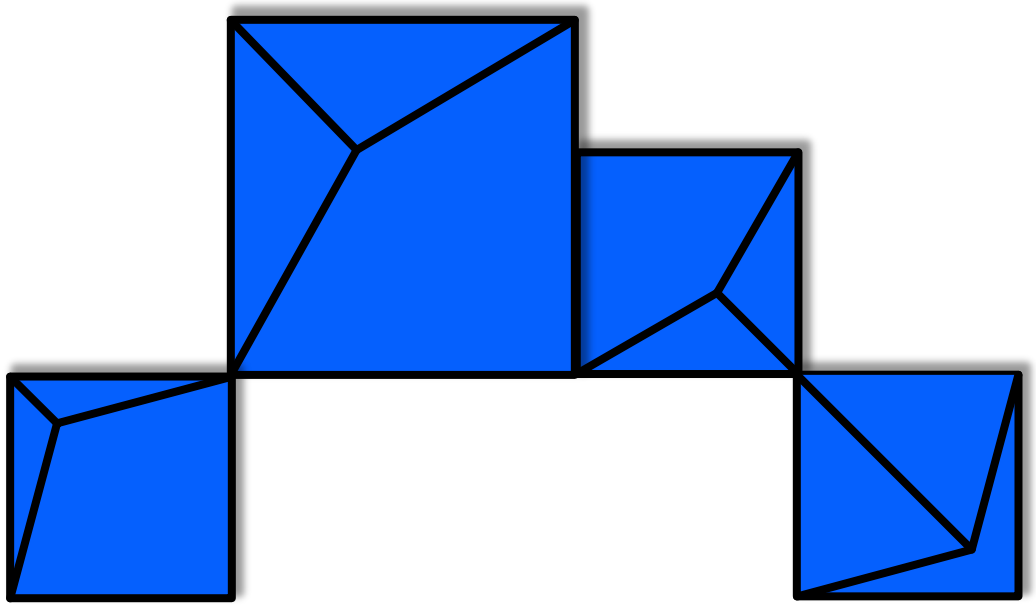
Technical Details

- Obsoleted APIs for signing and verifying data:
  - tpf_RSA_sign_data
  - tpf_RSA_sign_digest
  - tpf_RSA_verify_data
  - tpf_RSA_verify_digest

Conclusion

- Can create and verify digital signatures that conform to the latest security standards on z/TPF.

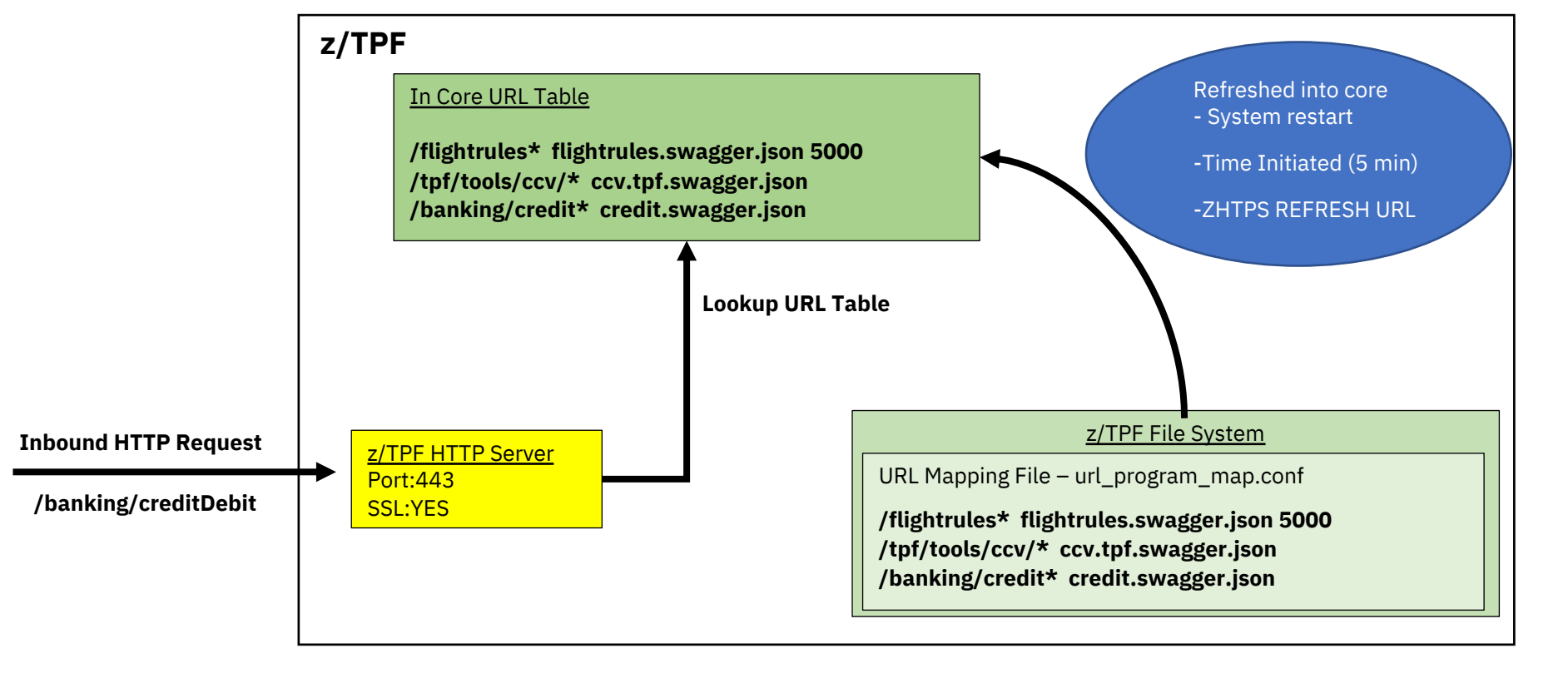
- Delivered with PJ45762 (Aug 2019).



z/TPF HTTP Server Enhancements

URL Mapping As-is: Deploying a new REST service

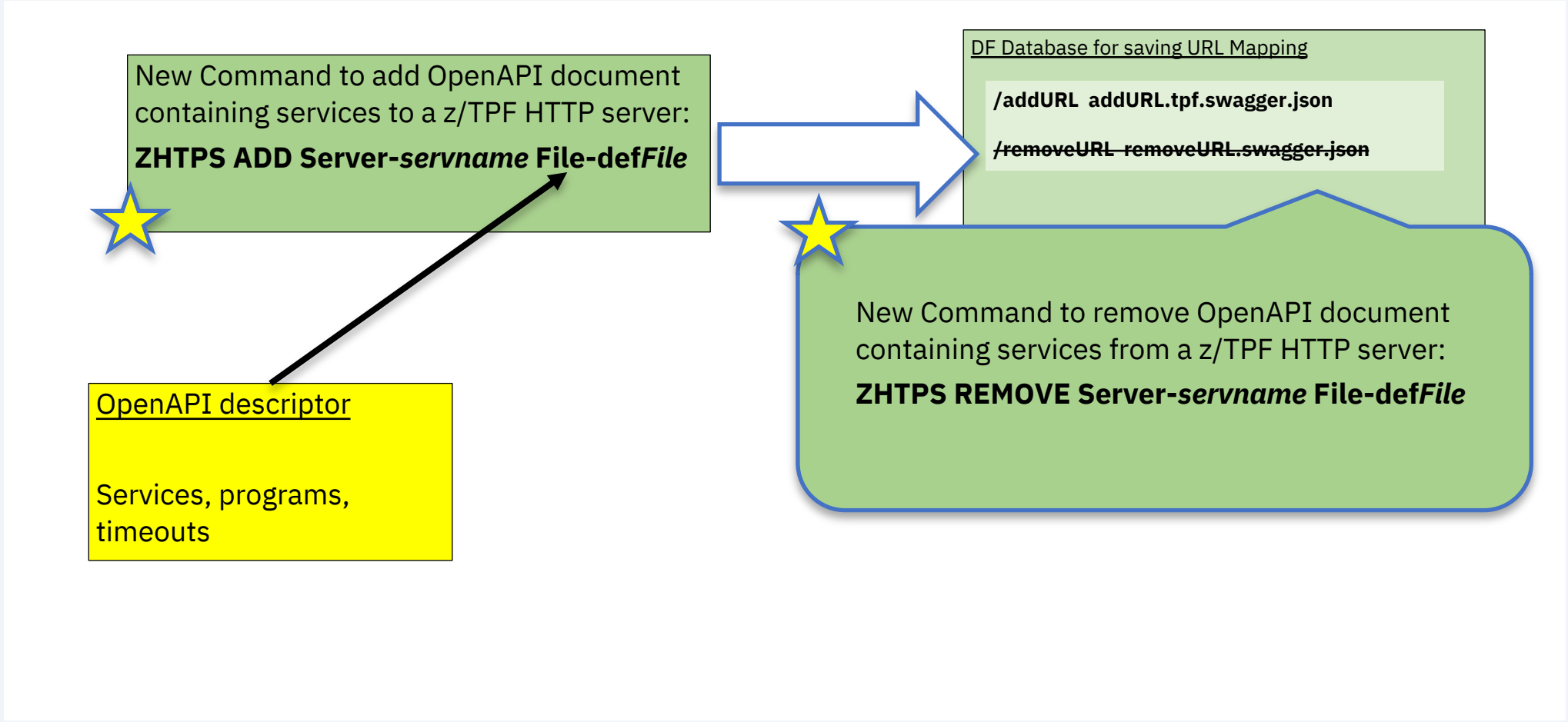
1. Update to the URL mapping file.
2. FTP of that file into the z/TPF system.
3. Refresh the newly FTPed file into memory (ZHTPS REFRESH URL).



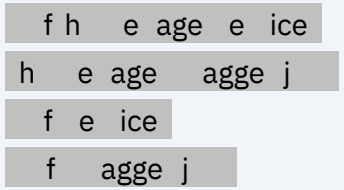
Problem Statement

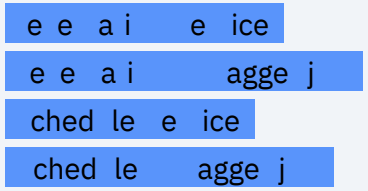
- Deploying a new REST service to a HTTP Server instance is not very user friendly and currently cannot be automated.
- Services are available system wide. SSL services could be serviced on non-SSL ports.
  - Example: If you had two HTTP server ports, one that is SSL enabled and one that does not use SSL, there is no way to deploy a service and have it serviced ONLY on SSL-enabled HTTP servers on TPF.

New Commands to add/remove REST services



New Command to display URL mapping entries





URL mapping entries added through new ZHTPS ADD command.

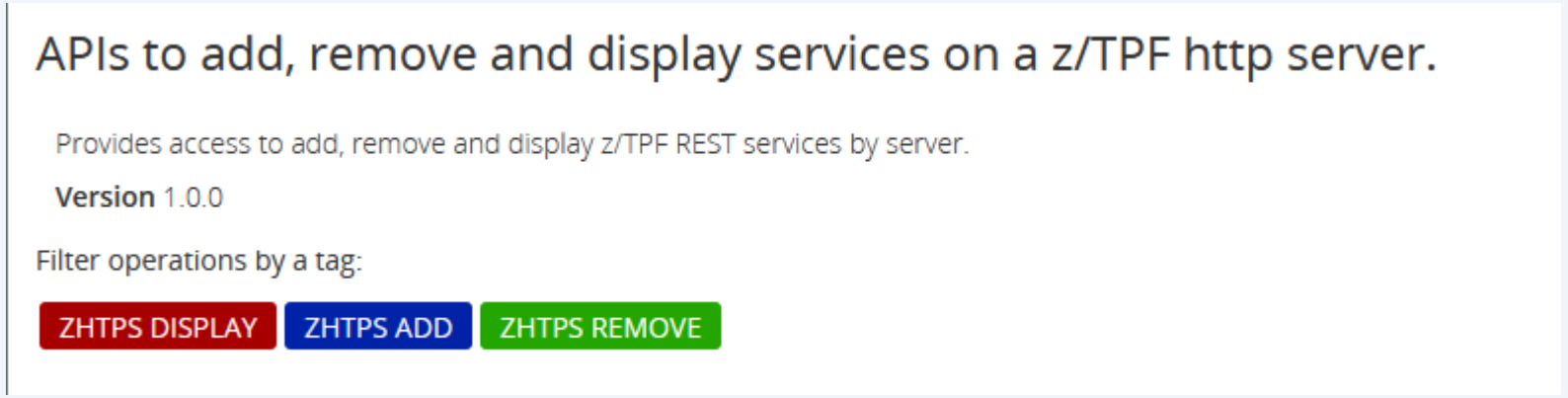
Entries are tied to an individual server entry (server port number)

URL mapping entries in the URL program mapping file.

Entries are accessible system wide.

New REST APIs delivered by z/TPF

- Add REST service : Deploys a service for an HTTP server and adds it to URL mapping table.
- Remove REST service: Removes a service from the URL mapping table.
- Display REST service: Displays services on an HTTP server.

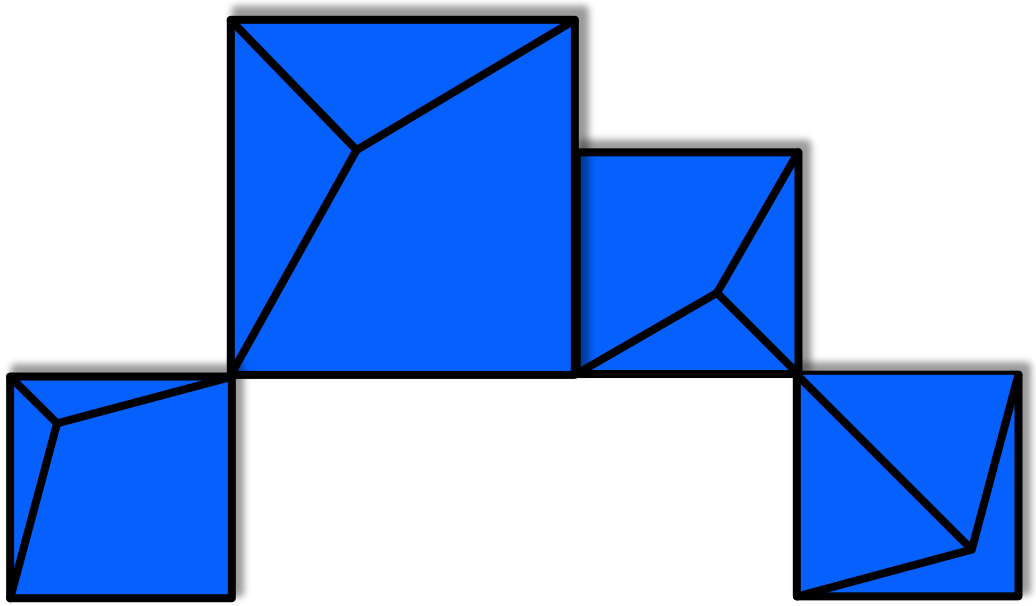


Technical Details

- Old URL Mapping will still exist. Mappings defined by the new commands have search priority.
- New mapping will no longer allow mapping base paths to four character program names. Documentation will be provided to convert these to REST services.
- Optimized search used on new mapping entries.
- URL mapping information automatically refreshed to all processors on ZHTPS ADD/REMOVE.

Conclusion

- REST service deployment is simplified by enabling users to add or remove a URL mapping with a single command.
- Services can be assigned to a specific server on a port instead of being available system wide.
- Automate the deployment of REST services through new system services.
- To be delivered with PJ46010 (Aug 2020).



High-speed connector Enhancements

Background – z/TPF High-speed connector

- Provides high availability
- Administrator can define groups of servers
  - Defined through configuration

- Supports TLS sessions as well as non-TLS
- Supports synchronous or asynchronous communication

- The z/TPF system handles
  - Load balancing
  - Error handling and automatic session establishment
  - Statistics
  - Topology can be changed without affecting existing applications

- Easier for applications to communicate with remote servers
  - Single API to send requests

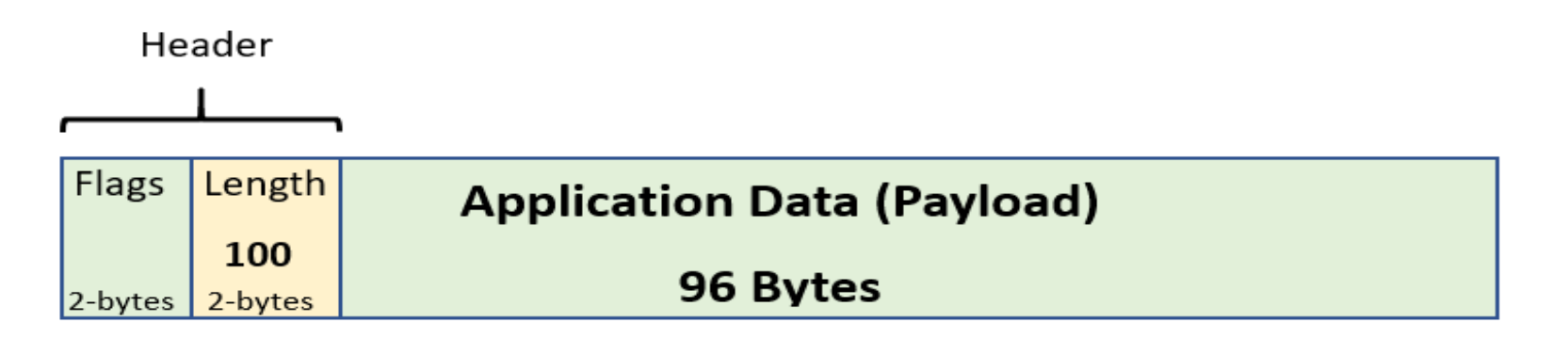
Problem Statement

- High-speed connector requires a specific message header format. Existing remote servers may be expecting a different header format.
- High-speed connector cannot be used to communicate with existing servers that expect a different header layout.

Technical Details

A system administrator can define a high-speed connector group and supply the format of the header the server expects to see.

- New parameters in the endpoint group descriptor to describe the header layout the server expects.

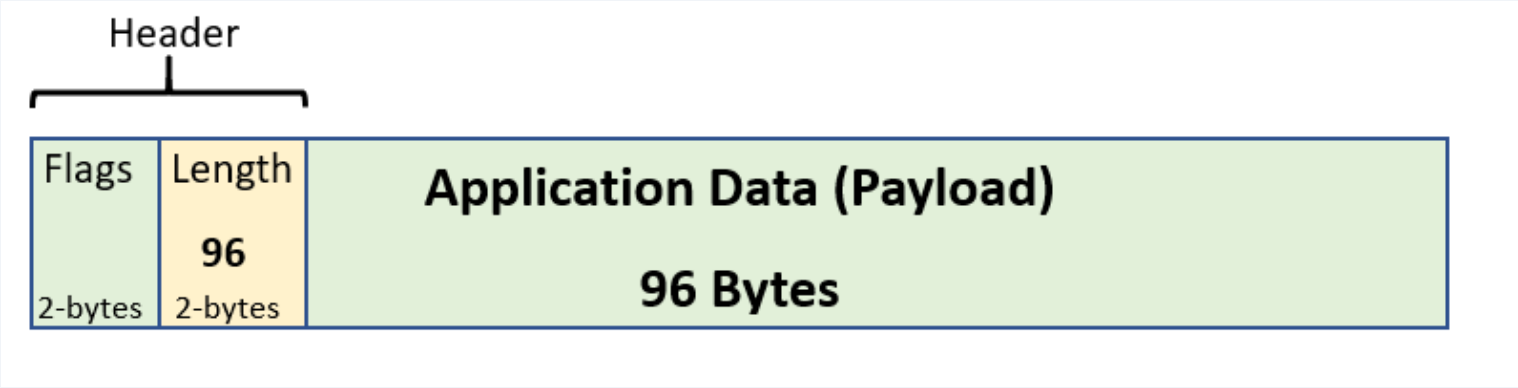


- Endpoint group descriptor would indicate
  - lenOffset=2
  - lenFieldLen=2
- Applications send data using the tpf_send_message or tpf_send_async_message APIs using the format described above.

Allows z/TPF client applications to communicate with any TCP/IP server in the enterprise.

Some implementations do not include the length of the header (or the length field) in the length specified in the message.

Additional optional parameter can be added to account for this :



- Endpoint group descriptor would indicate
  - lenOffset=2
  - lenFieldLen=2
  - headerLen=4

z/TPF will adjust the length to include the length of the header as well.

Additional high-speed connector enhancements delivered

- Ability to define HSC groups as “high priority”.
  - New endpoint group descriptor option.
  - Application responses and acknowledgements to outbound data are processed even while in input list shutdown.
- New options to control the frequency of messages being displayed to the console.
- Improved HSC command diagnostics and displays.
  - ZCONN CONFIG provides the loaded configuration
  - ZCONN TOPOLOGY provides the topology of the High-speed connector group including some additional diagnostic information

Value Statement

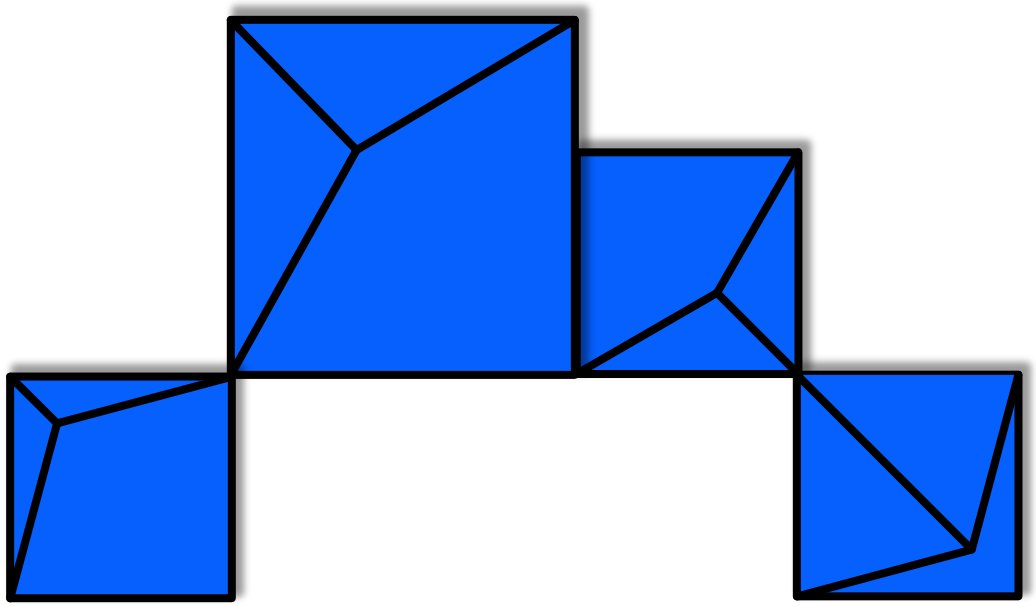
With the high-speed connector enhancements, z/TPF applications can communicate with existing servers in the enterprise as well as improve the usability of high-speed connector on z/TPF.

Conclusion

The high-speed connector enhancements were delivered in stages:

- Delivered with PJ46068 (June 2020)
  - Configurable high-speed connector headers
  - High priority sessions and sockets

- Delivered with PJ46148 (August 2020)
  - Configurable high-speed connector console warnings
  - Improved high-speed connector displays and diagnostics



Future Deliverables

Hardware Compression Support for the HTTP server

- z15 provides hardware compression support.
- Compression reduces
  - CPU consumed
  - Network bandwidth requirements
- Plan to leverage this support for the z/TPF HTTP server as a future item.
- Reducing the time, CPU consumption, and network bandwidth required to send large z/TPF HTTP server responses.

Call for Sponsor Users

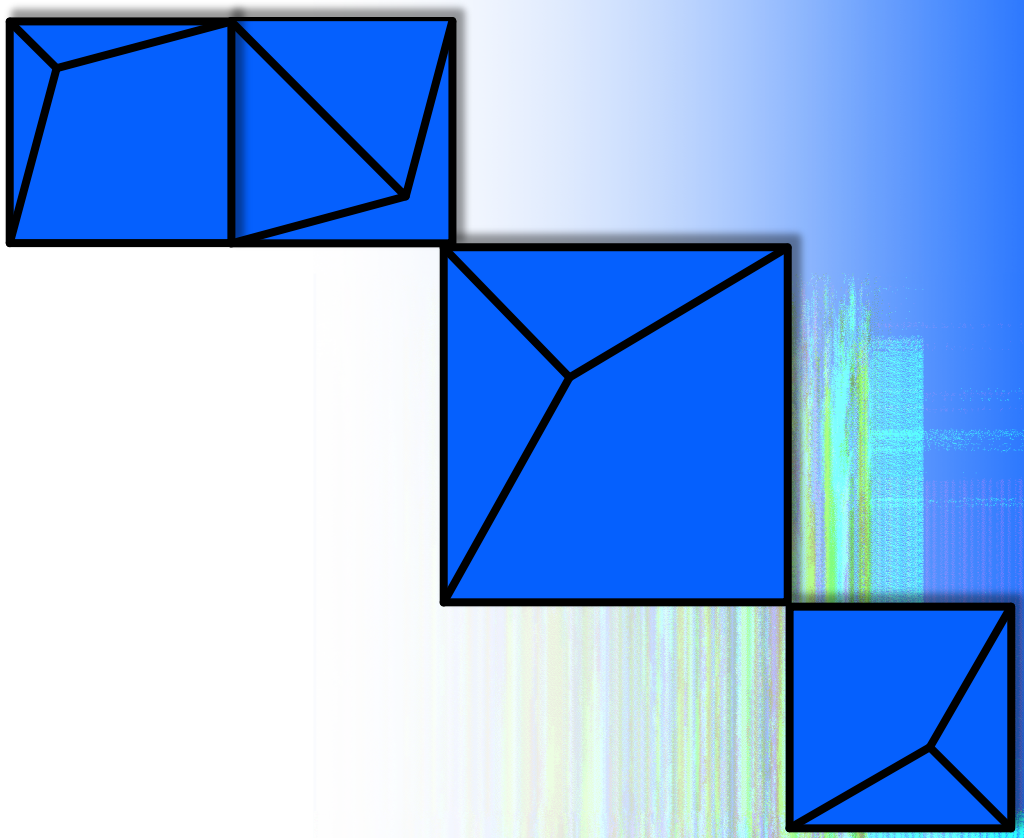
- Will be looking for Sponsor Users to assist in design and implementation, targeting the following personas:
  - z/TPF system administrators
  - z/TPF operators and coverage
  - z/TPF solution architects

- If you would like to be involved, contact:

  Jamie Farmer (jvfarmer@us.ibm.com) or Danielle Tavella (Danielle.Tavella@ibm.com)

Thank You

Questions? Comments?



Virtual TPFUG Q&A

Summary of Q&A from the virtual TPFUG event:

Question	Answer
Q: Do you receive an indication that the user exit was activated as a result of time out? If so, what is the indication?	A: Yes, the return code of HTTP 408 would be returned on a timeout to the request and is specified in the response structure that is passed to the user exit.
Q: HS connector is for client side. Will there be similar server in the future to support SSL-enabled sockets?	A: The Internet Daemon already has SSL enablement with SSL models for server side on TPF. But you are correct, there is no seamless transition to make a non-SSL server an SSL server in INETD without changing application code. Does sound like it may be useful to have moving forward. In higher level protocols like HTTP this is abstracted from the application, but for a purely TCP application that seamless transition does not exist today.
Q: Is the length field for the high-speed connector in integer and if so, is there any concern with big endian and little endian?	A: The length field is an integer. Endianness could potentially be a problem. If the lengths are expected to be in little endian, we can easily add an option to the configuration of configurable headers to address this. We will likely be working on additional High-Speed Connector enhancements in the semi-near future where we can incorporate such changes.
Q: In addition to configuration changes for z/TPF middleware packages such as INETD, do we still have to change existing socket APIs to SSL socket APIs such as SSL_read and SSL_write?	A: Correct. If you are converting an existing application to use SSL, you need to change things like socket read() and write() to SSL_read and SSL_write APIs.

Trademarks

Notes

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.