IBM Systems MEDIA

Monitor the Cryptographic Protection of Your z/OS Network Traffic

z/OS Communications Server V2.3 introduces a new feature: z/OS Encryption Readiness Technology (zERT).

By Chris Meyer, Dave Wierbowski, Michael Gierlach

10/09/2017

Imagine being directed by your manager to identify the overall quality of the cryptographic protection for your z/OS network. What exposures do you have? Who is using unapproved protection protocols, or worse, not using any protection at all? Where would you start?

There are numerous methods for cryptographically protecting TCP/IP traffic available on z/OS such as System SSL, Java Secure Sockets Extension (JSSE), Communications Server's Integrated IPSec, z/OS OpenSSH and AT-TLS. The typical configuration of these methods allows for the specification of multiple permutations of acceptable security attributes. Even if you acquired a deep understanding of each method's configuration, you would only know which security attributes were possible in your network. You would still not know which attributes had been negotiated by the security endpoints for any given connection. Existing SMF records and trace records might give you some hints, but you would have to work hard and long to turn those hints into a

comprehensive understanding of the security of your network's traffic patterns. In the end, you would still probably have some missing pieces.

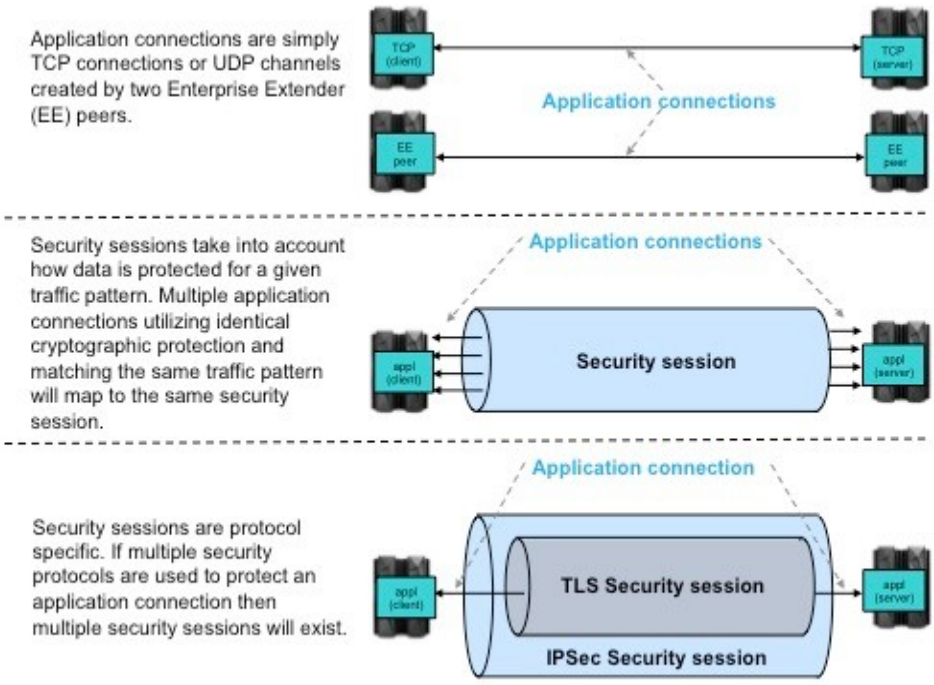
Protection Monitoring

z/OS Communications Server V2.3 introduces a new feature to provide you with what you need to answer your manager's request: z/OS Encryption Readiness Technology (zERT). zERT allows a z/OS network security administrator to determine which TCP and Enterprise Extender (EE) traffic patterns to and from their z/OS systems meet approved network encryption policies and which don't. It does this by collecting and recording, in SMF record format, the cryptographic protection attributes for TLS, SSL, SSH, and IPSec security sessions that terminate on the local stack. The following terms, as used within zERT, are:

- An application connection is a TCP connection or EE User Datagram Protocol (http://www-01.ibm.com/software/globalization/terminology/u.html#x2042771) channel over which two application programs communicate with each other. An application connection might or might not have cryptographic security, and the type and level of security coverage might change during the life of the connection.
- A security session is the combination of a unique set of security attributes for a specific security protocol as applied to one or more application connections that are associated with the same traffic pattern. For TCP, a traffic pattern means a distinct client IP address and server IP address/port combination. For EE, a traffic pattern means a distinct remote peer IP address/port and local IP peer address/port combination. For zERT, a security session might be used repeatedly by many different application connections over time.

Zero or more security sessions can protect a given application connection at any given time. For example, a connection might be flowing in the clear (e.g., no security sessions), it might be protected by a TLS session only (e.g., one security session) or it might be protected by both TLS and IPSec at the same time (e.g., multiple security sessions). Additionally, the type and level of security coverage might change during the life of the connection. For instance, an FTP control connection can transmit data as clear text, change to use TLS security protocols and return to transmitting data as clear text.

Security sessions take into account how data is protected for a given traffic pattern. Multiple application connections using identical cryptographic protection and matching the same traffic pattern map to the same security session. A single application connection using multiple security protocols map to multiple security sessions. Figure 1 (/mainframe/hot-topics/hot-topics/figure-1/) shows the relationship between application connections and security sessions.



How Protection Is Monitored

zERT provides two functions at no charge to assist a network administrator in determining how data in flight is being protected:

- zERT Discovery tracks and records significant events regarding the security sessions that are used to protect each individual application connection. zERT Discovery obtains this security information in two ways, through:

1. Limited observation of an application connection's data stream. Stream observation obtains basic cryptographic information that’s exchanged during an initial TLS, SSL or SSH handshake, and only applies to TCP connections.
2. Direct notification from z/OS zERT-enabled cryptographic protocol providers (CPPs). These are z/OS IPSec, System SSL and z/OS OpenSSH. Observation yields less comprehensive information than what’s provided by CPPs, but it does provide coverage for security sessions that use providers that aren’t zERT-enabled such as the JSSE or OpenSSL.

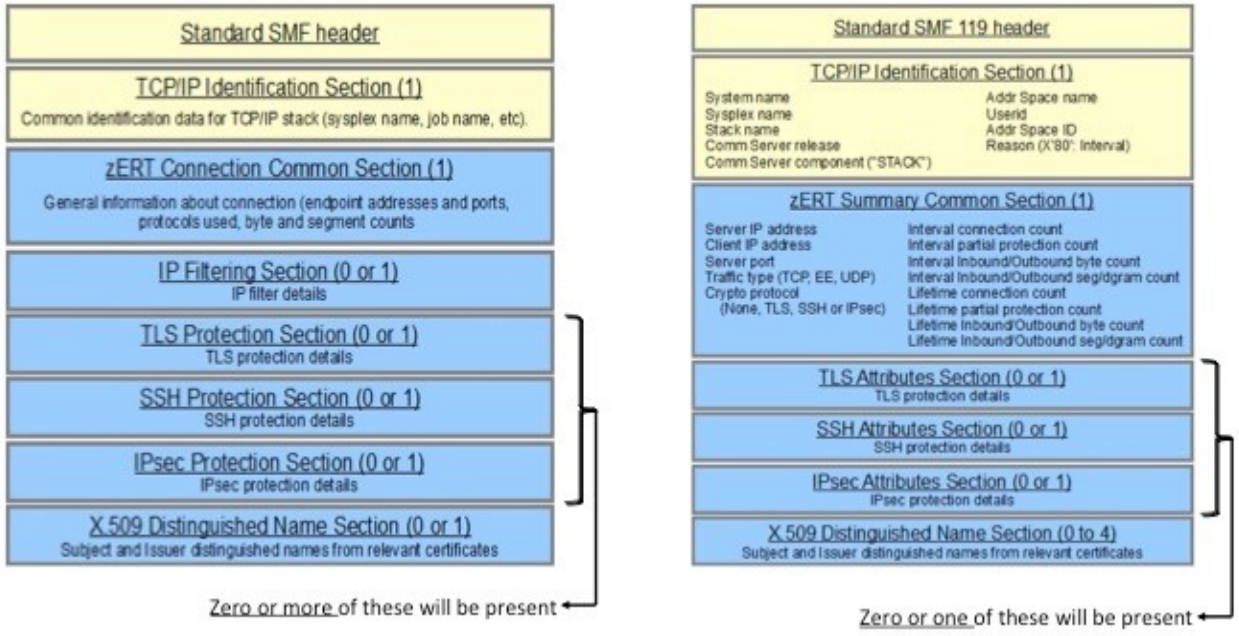
zERT Discovery is available with z/OS V2.3.

- zERT Aggregation provides summary information for the security sessions terminating on the local stack. zERT Aggregation uses the information that’s accumulated by zERT Discovery as a basis for the summary records. Since multiple application connections can use the same security session over time, zERT Aggregation typically results in the creation of significantly fewer SMF records than zERT Discovery, while still providing significant cryptographic detail. zERT Aggregation will be available on z/OS V2.3 with APAR PI83362 in 1Q 2018.

Both zERT Discovery and zERT Aggregation functions record the collected information as SMF type 119 records:

- zERT Discovery information is recorded as zERT Connection Detail (subtype 11) records. These records are written when application connections start, end or have a change in cryptographic coverage. A zERT Connection Detail record contains information about the zero or more security sessions currently protecting an application connection.
- zERT Aggregation information is recorded as zERT Summary (subtype 12) records. These records are written at user-defined intervals. Each zERT Summary record represents a single security session, and includes both significant security characteristics and usage statistics for the security session.

Figure 2 (/mainframe/hot-topics/hot-topics/figure-2/) shows the basic layout of the zERT subtype 11 and subtype 12 records.



Each SMF record provides information about the connection endpoints as well as appropriate information about the local z/OS socket owner, such as user ID and jobname. For each protocol included in the SMF record, a protocol-specific section is included that contains important cryptographic attributes of the security session being recorded. For instance, the protocol-specific section can include values identifying the security session's authentication method, the cryptographic algorithms used for encryption, message authentication, key exchange and the key lengths used for each algorithm (where appropriate). However, it's important to understand that zERT doesn't store or record the values of secret keys, initialization vectors or any other secret values that are negotiated or derived during cryptographic protocol handshakes. For complete details on the SMF 119 zERT Connection Detail record formats, see the z/OS Communications Server: IP Programmer's Guide and Reference, SC27-3659 (https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.halx001/zertcon.htm).

Enabling the zERT functions is simple. The zERT Discovery function is enabled by specifying the GLOBALCONFIG ZERT parameter in the TCP/IP profile data set, while the zERT Aggregation function is enabled by coding GLOBALCONFIG ZERT AGGREGATION. Independent SMFCONFIG and NETMONITOR controls in the TCP/IP profile data set are available to direct zERT to write the SMF records to the z/OS SMF, to

a network management application by using the real-time Network Management Interface for zERT information, or both. Table 1 summarizes the configuration tasks required to use zERT.

Task	TCP/IP Profile statement	Comments
Enable zERT Discovery function	GLOBALCONFIG ZERT	
Enable zERT Aggregation function	GLOBALCONFIG ZERT AGGREGATION	To enable the aggregation function the discovery function must also be enabled.
Enable generation of SMF 119 subtype 11 records	SMFCONFIG TYPE119 ZERTDETAIL	The discovery function must be enabled before SMF 119 subtype 11 records will be generated.
Enable generation of SMF 119 subtype 12 records	SMFCONFIG TYPE119 ZERTSUMMARY	The aggregation function must be enabled before SMF 119 subtype 12 records will be generated.
Enable collection of SMF 119 subtype 11 records by using the real time NMI	NETMONITOR ON ZERTSERVICE	The discovery function must be enabled before SMF 119 subtype 11 records will be generated.
Enable collection of SMF 119 subtype 12 records by using the real-time NMI	NETMONITOR ON ZERTSERVICE	The aggregation function must be enabled before SMF 119 subtype 12 records will be generated.

Balance of Information

For more detailed information on configuring zERT, see the z/OS Communications Server: IP Configuration Reference, SC27-3657 (https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.halz001/pro.htm). For more information on the real-time zERT NMI service, see the z/OS Communications Server: IP Programmer's Guide and Reference, SC27-3659 (https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.halx001/inttcp.htm).

As you plan to use the zERT functions, it's important to realize that while zERT Discovery provides the most thorough level of cryptographic coverage information, the number of zERT Connection Detail records can be large. How large depends on the number of connections that are supported by your z/OS system and the frequency with which those connections are created, deleted or changed from a cryptographic coverage perspective. Using the zERT Aggregation function typically reduces the number of SMF records generated, but provides less information at an individual application connection level. The combination of functions might be what you need to provide your manager with that in-depth analysis of your network cryptographic coverage.

Christine Smith contributed to the technical and editorial review of this article

About the author
Chris Meyer, CISSP, is a senior software engineer in Research Triangle Park, N.C.

Dave Wierbowski is a senior software engineer in Endicott, N.Y.

Michael Gierlach is an advisory software engineer in z/OS Communications Server in Research Triangle Park, N.C.