z/OS 3.2 IBM Education Assistant

Solution Name: zERT Monitoring Enhancements

Solution Element(s): z/OS Communications Server

July 2025

Agenda

- Trademarks
- Objectives
- Overview
- Usage & Invocation / Function Externals
- Interactions & Dependencies
- Upgrade & Coexistence Considerations
- Installation & Configuration
- Summary
- Appendix

Trademarks

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

Objectives

- z/OS Communications Server 3.2 has enhanced z/OS Encryption Readiness Technology to easily distinguish between TLS/SSH connections (successful and failed) and unprotected connections and provide all the information necessary to ensure that the network traffic is protected per network policy.

- This is to address the following ideas
  - Idea ZOS-I-3371 – Enable zERT to Detect Partial/Failed Secure Handshakes
  - Idea ZOS-I-339 – Certificate details, particularly serial and expiry date, on SMF119-12
  - Idea ZOS-I-3963 – Add Client/Server role indicator into ZERT SMF 119 Subtype 11 Records

Overview: zERT background

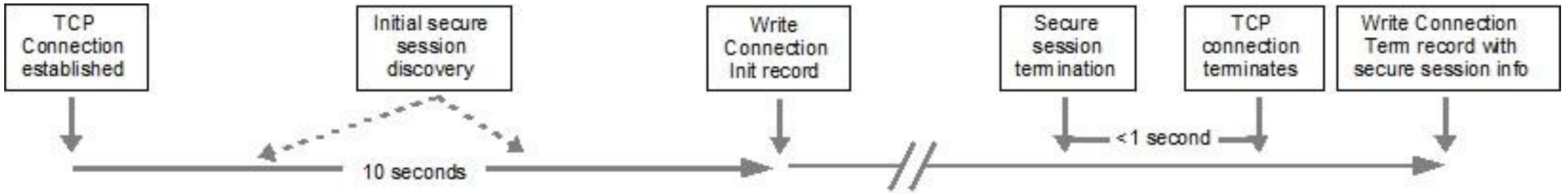
- zERT positions the TCP/IP stack as a central collection point and repository for cryptographic protection attributes for:
  - **TCP** connections that are protected by **TLS, SSL, SSH, IPsec or** have **no recognized cryptographic protection**
  - **Enterprise Extender** connections that are protected by **IPsec or** have **no recognized cryptographic protection**
    - Each peer-to-peer UDP port is considered a separate EE connection
    - In this presentation, we'll focus on TCP examples

- Two methods for discovering the security sessions and their attributes:
  - **Stream observation** (for TLS, SSL and SSH) – the TCP/IP stack observes the protocol handshakes as they flow over the TCP connection
  - **Advisory observation** by the cryptographic protocol provider (System SSL, ZERTJSSE, OpenSSH, TCP/IP's IPsec support)

- Reported through SMF 119 records via:
  - **SMF** and/or
  - **Real-time NMI** services

Overview: zERT monitoring background

- "zERT is not perfect"
  - Cannot predict which security protocols (if any) will be used on any given connection
  - Relies on stream observation for initial collection of TLS and SSH cryptographic attributes
  - When TLS or SSH is recognized, more detailed information *might be* provided by the z/OS protocol provider if it is aware of zERT (System SSL, ZERTJSSE, IPsec, and z/OS OpenSSH). However, zERT doesn't know which protocol provider is being used.
  - Timing of TLS and SSH handshakes is unpredictable – and can sometimes take an extended period of time (over 10 seconds in some cases)
- zERT uses timers in various situations when protection state is unclear. Example:



- Using timers **mitigates much of the unpredictability**, **but can also cause "unprotected"** sessions to be reported where they may not be expected
- The effect of these extraneous unprotected sessions may be **magnified when using zERT policy-based enforcement** rules for unprotected traffic

Overview: zERT monitoring and recording deficiencies

The current zERT monitoring and recording implementation suffers from the following deficiencies:

1) Writing unprotected record when a TLS/SSL or SSH **handshake fails** with no indication of the attempt
   - zERT only reports on *successful* TLS/SSL and SSH handshakes. If a handshake fails, zERT reports it as an Unprotected* session – there is no indication that a handshake was attempted and failed

2) Writing unprotected record when a TLS/SSL or SSH **handshake is in progress** (10 second timer pops)
   - When the init timer pops, we cut an unprotected record. An additional protection change record is written if the handshake completes and the cryptographic protocol provider tells us

3) Writing unprotected protection change record when a **TLS session is terminated, and the TCP connection takes >1 second to terminate but no further application data flows**
   - The TLS connection is explicitly terminated, causing System SSL to notify zERT of the change AND no additional data passes over the TCP connection after the TLS session is terminated by an inbound RST packet, BUT more than one second passes before the TCP connection is terminated

Missing information in SMF119 subtype 11/12

1) The client and server **certificate information** are currently only recorded in the subtype 11 records, not in subtype 12

2) Flag to indicate whether the **local socket is acting as the TCP client or server** is only in subtype 12 records, not in subtype 11

Overview

- Who (Audience)
  - As a z/OS Network Security Administrator

- What (Solution)
  - I wish to use zERT SMF records to easily distinguish between TLS/SSH connections (successful and failed) and unprotected connections

- Wow (Benefit / Value, Need Addressed)
  - to ensure and prove that my network traffic is protected in compliance with my network security policy

Overview - zERT monitoring enhancements (1 of 2)

1) When a **TLS/SSH handshake fails**
   - zERT will not generate an unprotected record anymore. Instead, it will write -
     - Subtype 11 detail record - with a TLS/SSH section and a new indicator of ‘Failed handshake’
     - Subtype 12 summary record – new counts for the failed handshakes during that interval

2) When a **TLS/SSH handshake is in progress**
   - If zERT has detected the beginnings of a handshake within the first 10 seconds, it will write a record for the connection only when the TLS/SSH handshake completes successful or fails. It will not write a record when the handshake is in progress.

3) When a **TLS session is terminated**
   - zERT will not write a record until the TCP connection is terminated (term or short-term record) or data flows on the TCP connection in the clear (protection state change record)

Overview - zERT monitoring enhancements (2 of 2)

Missing fields in subtype 11/12

1) zERT aggregation will record the **certificate expiration dates and certificate serial** in subtype 12 records (similar to what we do in subtype 11).
   - SMF119SS_ proto_xCert_Serial_Len, SMF119SS_proto_xCert_Serial, SMF119SS_ proto_xCert_Time_Type and SMF119SS_ proto_xCert_Time (where x is S for server certificate information and C for client certificate information, Lcl for IKE Local certificate information and Rmt for IKE Peer certificate information; proto is TLS or IPsec)

2) zERT discovery will record the indicator that tells whether the **local socket is acting as the TCP client or server** in subtype 11 records (similar to what we have in subtype 12)

Usage & Invocation / Function Externals – SMF 119 Subtype 11 updates (1 of 3)

Updates (in red) to SMF 119, subtype 11 (zERT connection detail record)

Table 2. zERT connection detail common section

2(X'2')	SMF119SC_SAFlags	1	Binary	Flags: • X'80': IPv6 connection • X'40': AT-TLS cryptographic data protection operations are bypassed for this connection as part of a stack optimization for intra-host connections. Only AT-TLS peer authentication operations are executed in this case. • X'20': Connection reset by zERT policy-based enforcement - Can only be set when event type (SMF119SC_SAEvent_Type) is connection termination or short connection termination. Otherwise, 0 • x'10': The local socket of this connection is acting as the server (only meaningful when SMF119SC_SAIPProto indicates TCP) • x'08': The local socket of this connection is acting as the client (only meaningful when SMF119SC_SAIPProto indicates TCP)

Usage & Invocation / Function Externals – SMF 119 Subtype 11 updates (2 of 3)

Table 4. zERT TLS protocol attributes section

3(X'3')	SMF119SC_TLS_Handshake_Type	1	Binary	Handshake type: • X'01': Full handshake • X'02': Abbreviated handshake • X'03': Failed handshake. When set, all the fields in this section will contain binary zeroes or blanks except the SMF119SC_TLS_Source, SMF119SC_TLS_Protocol_Provider and SMF119SC_TLS_Neg_Cipher fields.

56(X'38')	SMF119SC_TLS_Neg_Cipher	6	EBCDIC	Negotiated cipher suite identifier
				• If the TLS version is Unknown, this field is set to "FFFFFF" (Unknown). The TLS version is Unknown for a failed TLS handshake.
				• If the TLS version is SSLv3 or higher, this is a four character value in the first 4 bytes of this field, padded with trailing blanks. Refer to the TLS Cipher Suite registry at http://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml for a complete list of the 4-hexadecimal-character values.
				• If the TLS version is SSLv2, then all 6 bytes are used:
				  • "010080": 128-bit RC4 with MD5
				  • "020080": 40-bit RC4 with MD5
				  • "030080": 128-bit RC2 with MD5
				  • "040080": 40-bit RC2 with MD5
				  • "050080": 128-bit IDEA with MD5
				  • "060040": DES with MD5
				  • "0700C0": 3DES with MD5

Usage & Invocation / Function Externals – SMF 119 Subtype 11 updates (3 of 3)

Table 5. zERT SSH protocol attributes section

0(X'0')	SMF119SC_SSH_Prot_Ver	1	Binary	Protocol version: • X'00': Unknown • X'01': Protocol version 1 • X'02': Protocol version 2
4(X'4')	SMF119SC_SSH_Flags	1	Binary	Flags: •X’80’: Failed handshake. When set, all the fields in this section will contain binary zeroes or blanks except the SMF119SC_SSH_Source and SMF119SC_SSH_Protocol_Provider.

Usage & Invocation / Function Externals – SMF 119 Subtype 12 updates (1 of 6)

Updates to SMF 119, subtype 12 (zERT summary record)

Table 2. zERT summary record common section

204 (x'CC')	SMF119SS_SAInitFailedHsConnCnt	4	Binary	Count of failed TLS or SSH handshakes for the life of this security session at the beginning of the summary interval. When this field is > 0, all the fields in the corresponding security session (TLS or SSH) section will contain binary zeroes or blanks except SMF119SS_TLS_Source and SMF119SS_TLS_Neg_Cipher (when the SMF119SS_SecProto is TLS) or SMF119SS_SSH_Source (when the SMF119SS_SecProto is SSH).
208 (x'D0')	SMF119SS_ SAEndFailedHsConnCnt	4	Binary	Count of failed TLS or SSH handshakes for the life of this security session at the end of the summary interval. When this field is > 0, all the fields in the corresponding security session (TLS or SSH) section will contain binary zeroes or blanks except SMF119SS_TLS_Source and SMF119SS_TLS_Neg_Cipher (when the SMF119SS_SecProto is TLS) or SMF119SS_SSH_Source (when the SMF119SS_SecProto is SSH).

Note: Count of failed handshakes is a subset of the connections reported in SMF119SS_SAInitLifeConnCnt that attempted a TLS or SSH (indicated by the SMF119SS_SecProto field of the zERT summary common section) handshake that failed. Valid when SMF119S1Len >= SMF119SS_S1Len_V2.

Usage & Invocation / Function Externals – SMF 119 Subtype 12 updates (2 of 6)

Table 3. zERT summary record TLS protocol attributes section

Server certificate information

42 (x'2A')	SMF119SS_TLS_SCert_Serial_Len	1	Binary	Server certificate serial number length in bytes.
43 (x'2B')	SMF119SS_TLS_SCert_Serial	20	Binary	Server certificate serial number, left justified.
63 (x'3F')	SMF119SS_TLS_SCert_Time_Type	1	Binary	Format of server certificate "not after" time: • X'01': Coordinated Universal Time (UTC) • X'02': Generalized Time (GT)
64 (x'40')	SMF119SS_TLS_SCert_Time	15	EBCDIC	Server certificate "not after" time: • If the time type is UTC (SMF119SS_TLS_SCert_Time_Type = X'01'), the first 13 bytes of this field contain the time in UTC format (YYMMDDhhmmssZ). • If the time type is GT (SMF119SS_TLS_SCert_Time_Type = X'02'), all 15 bytes of this field contain the time in GT format (YYYYMMDDhhmmssZ).

Usage & Invocation / Function Externals – SMF 119 Subtype 12 updates (3 of 6)

Table 3. zERT summary record TLS protocol attributes section

Client certificate information

79 (x'4F')	SMF119SS_TLS_CCert_Serial_Len	1	Binary	Client certificate serial number length in bytes.
80 (x'50')	SMF119SS_TLS_CCert_Serial	20	Binary	Client certificate serial number, left justified.
100 (x'64')	SMF119SS_TLS_CCert_Time_Type	1	Binary	Format of client certificate "not after" time: • X'01': Coordinated Universal Time (UTC) • X'02': Generalized Time (GT)
101 (x'65')	SMF119SS_TLS_CCert_Time	15	EBCDIC	Client certificate "not after" time: • If the time type is UTC (SMF119SS_TLS_CCert_Time_Type = X'01'), the first 13 bytes of this field contain the time in UTC format (YYMMDDhhmmssZ). • If the time type is GT (SMF119SS_TLS_CCert_Time_Type = X'02'), all 15 bytes of this field contain the time in GT format (YYYYMMDDhhmmssZ).

Usage & Invocation / Function Externals – SMF 119 Subtype 12 updates (4 of 6)

Table 5. zERT summary record IPSec protocol attributes section

IKE Local certificate information (will be populated if SMF119SS_IPSec_IKETunLocalAuthMeth indicates RSA, ECDSA, or Digital signature and local certificate information is available Otherwise, all fields set to zero.)

74 (x'4A')	SMF119SS_IPSec_LclCert_Serial_Len	1	Binary	Local IKE certificate serial number length in bytes.
75 (x'4B')	SMF119SS_IPSec_LclCert_Serial	20	Binary	Local IKE certificate serial number, left justified.
95 (x'5F')	SMF119SS_IPSec_LclCert_Time_Type	1	Binary	Format of local IKE certificate "not after" time: • X'00': Manual tunnel - unused • X'01': Coordinated Universal Time (UTC) • X'02': Generalized Time (GT)
96 (x'60')	SMF119SS_IPSec_LclCert_Time	15	EBCDIC	Local IKE certificate "not after" time: • If the time type is UTC (SMF119SS_IPSec_LclCert_Time_Type = X'01'), the first 13 bytes of this field contain the time in UTC format (YYMMDDhhmmssZ). • If the time type is GT (SMF119SS_IPSec_LclCert_Time_Type = X'02'), all 15 bytes of this field contain the time in GT format (YYYYMMDDhhmmssZ).

Usage & Invocation / Function Externals – SMF 119 Subtype 12 updates (5 of 6)

Table 5. zERT summary record IPSec protocol attributes section

IKE Peer certificate information (will be populated if SMF119SS_IPSec_IKETunRmtAuthMeth indicates RSA, ECDSA, or Digital signature and remote certificate information is available . Otherwise, all fields set to zero.)

111 (x'6F')	SMF119SS_IPSec_RmtCert_Serial_Len	1	Binary	Remote IKE certificate serial number length in bytes.
112 (x'70')	SMF119SS_IPSec_RmtCert_Serial	20	Binary	Remote IKE certificate serial number, left justified.
132 (x'84')	SMF119SS_IPSec_RmtCert_Time_Type	1	Binary	Format of remote IKE certificate "not after" time: • X'00': Manual tunnel - unused • X'01': Coordinated Universal Time (UTC) • X'02': Generalized Time (GT)
133 (x'85')	SMF119SS_IPSec_RmtCert_Time	15	EBCDIC	Remote IKE certificate "not after" time: • If the time type is UTC (SMF119SS_IPSec_RmtCert_Time_Type = X'01'), the first 13 bytes of this field contain the time in UTC format (YYMMDDhhmmssZ). • If the time type is GT (SMF119SS_IPSec_RmtCert_Time_Type = X'02'), all 15 bytes of this field contain the time in GT format (YYYYMMDDhhmmssZ).

Usage & Invocation / Function Externals – SMF 119 Subtype 12 updates (6 of 6)

Table 4. zERT summary record SSH protocol attributes section

2(X’2')	SMF119SC_SSH_Prot_Ver	1	Binary	Protocol version: • X'00': Unknown • X'01': Protocol version 1 • X'02': Protocol version 2

Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - None

Upgrade & Coexistence Considerations

- To exploit this solution, all systems in the Plex must be at the new z/OS level: No
- No Upgrade and co-existence considerations
- Reserved SMF fields have new values
- Applications that consume SMF type 119 subtype 12 records can detect the presence of the new fields in the existing sections by checking the length of the section in the associated triplet.

Installation & Configuration

- In the TCP/IP profile
  - Enable GLOBALCONFIG ZERT
  - Enable SMFCONFIG ZERTDETAIL and/or NETMONITOR ZERTSERVICE for SMF 119 subtype 11 records
  - Enable GLOBALCONFIG ZERT AGGREGATION, SMFCONFIG ZERTSUMMARY and/or NETMONITOR ZERTSUMMARY for SMF 119 subtype 12 records

Summary

- z/OS Communications Server 3.2 has enhanced z/OS Encryption Readiness Technology to easily distinguish between TLS/SSH connections (successful and failed) and unprotected connections and provide all the information necessary to ensure that the network traffic is protected per network policy.

Appendix

- IBM zERT “all-in-one” page
  - http://ibm.biz/thingsaboutzert

- z/OS Communications Server: IP Configuration Guide
  - Monitoring cryptographic network protection: z/OS encryption readiness technology (zERT)

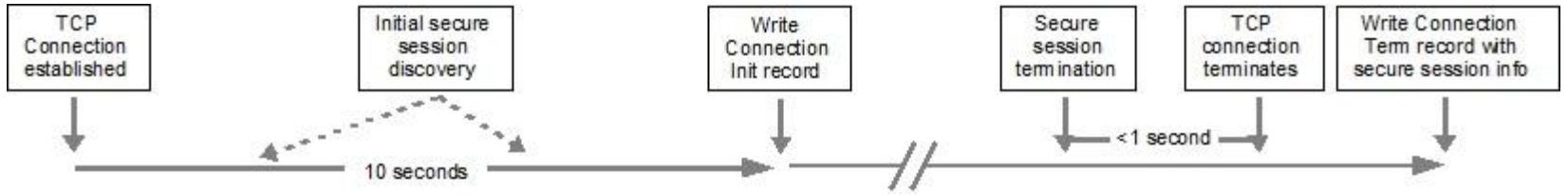
- z/OS Communications Server: IP Configuration Reference
  - GLOBALCONFIG statement
  - SMFCONFIG statement
  - NETMONITOR statement

- z/OS Communications Server: IP Programmer’s Guide and Reference
  - zERT connection detail record (subtype 11)
  - zERT Summary record (subtype 12)
  - Real-time NMI: SYSTCPER service
  - Real-time NMI: SYSTCPES service

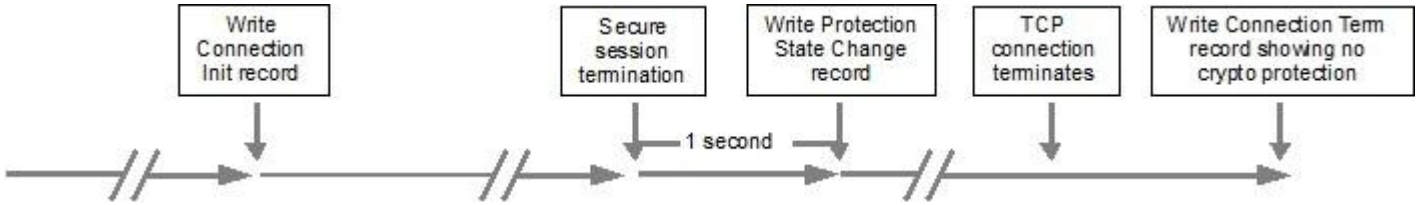
Backup

Background: Sample event sequences and use of timers to minimize records:

1. Typical TCP connection (not short-lived) with no state changes:



2. State change: Early termination of security session:



3. Typical short-lived TCP connection: