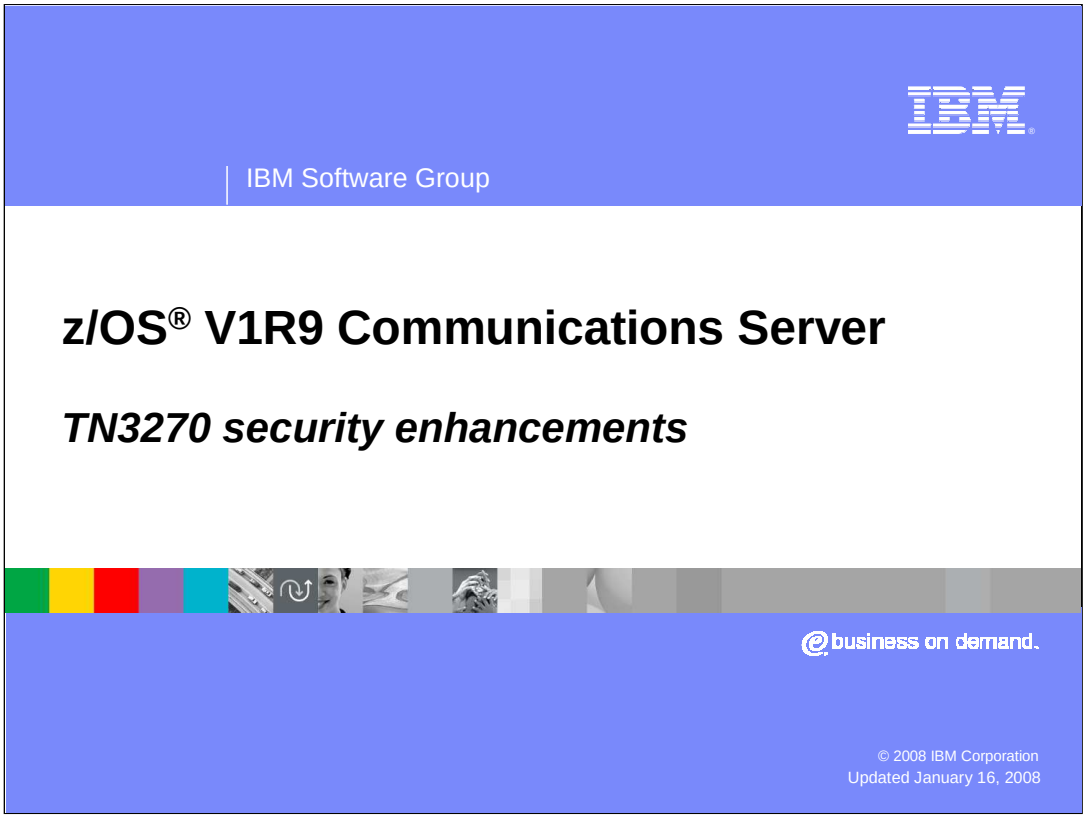
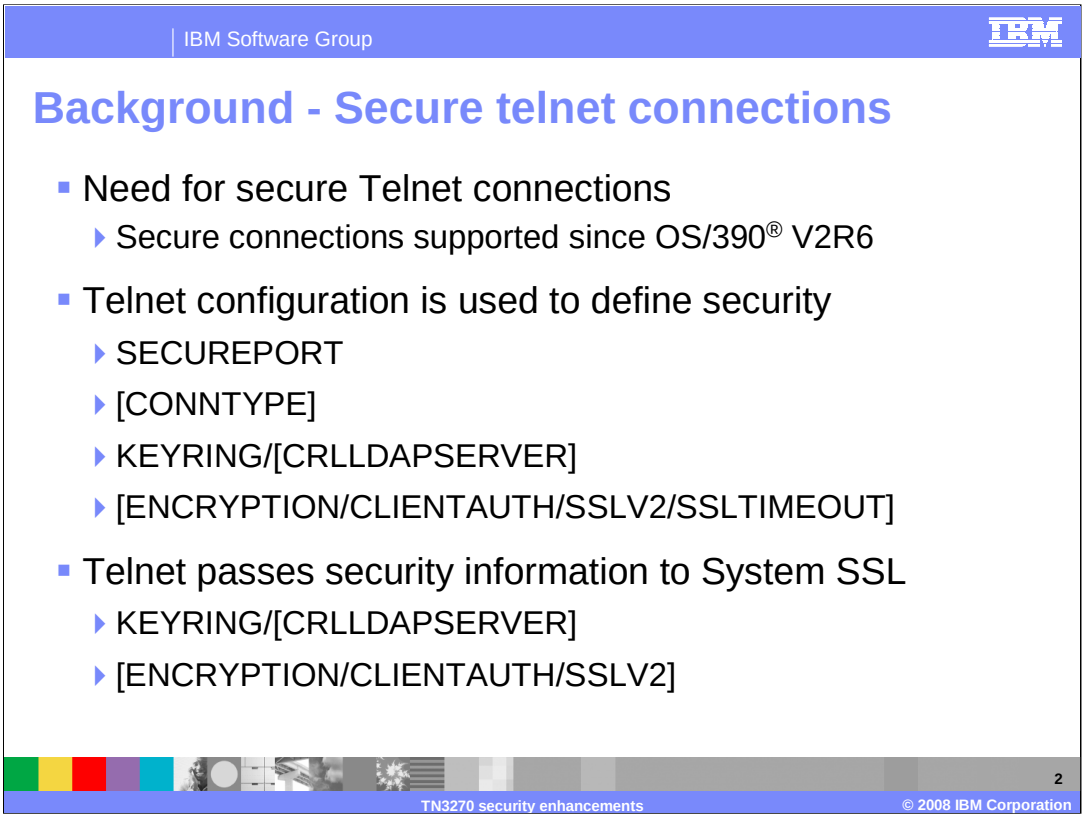


This presentation discusses the TN3270 security enhancements in the z/OS V1R9 Communications Server.

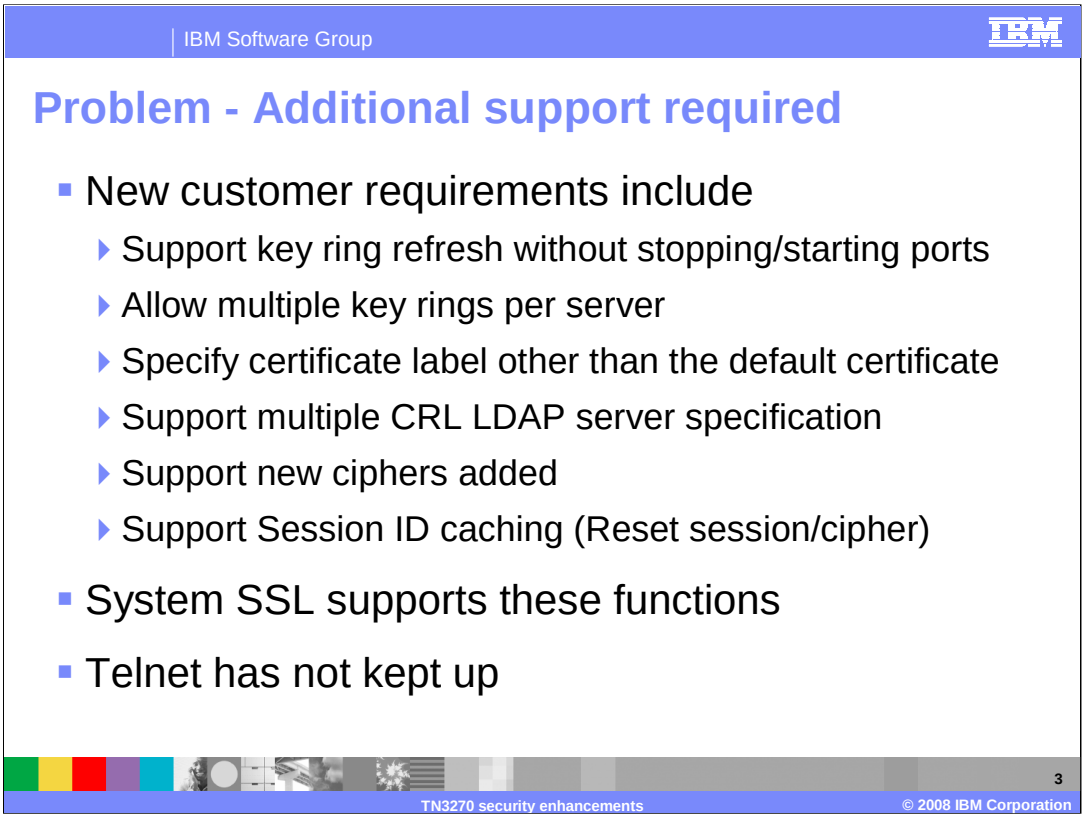


For several years, Telnet connections have needed to be secure. Telnet first implemented secure connections in OS/390 V2R6 which was generally available in September 1998.

Telnet was developed with a direct interface to the System Secure Sockets Layer, SSL, component. Telnet configuration is used to define the parameters needed for System SSL to set up its environment to support secure connections. SECUREPORT designates that the connections to the Telnet port will be secure and KEYRING specifies the key ring name System SSL will use. Additional, optional parameters are shown in brackets.

In a later release, OS/390 V2R10, the Conntype statement was added to allow both secure and non-secure connections on the same port.

System SSL needs a key ring to properly set up the System SSL environment used by Telnet. Optionally, CRLLDAPServer, Encryption, ClientAuth, and SSLv2 values can be specified to further customize the System SSL environment.



When Telnet first implemented secure connections on OS/390 V2R6, System SSL was not as robust as it is today. System SSL allowed only one active environment to support telnet connections. Telnet security setup was developed around that assumption and others based on System SSL capability at the time. For example, because only one System SSL environment could be activated, Telnet allows only one key ring name for all ports.

Customer have asked for Telnet to support different key rings on different ports and even different key rings on the same port. Customers have a need to be able to refresh security parameters without having to stop/restart the secure ports. This is particularly useful when the default certificate expires and must be replaced. Some customers have backup Certificate Revocation List Lightweight Directory Access Protocol, CRL LDAP, servers and would like to specify these backups. Customers would like to quickly use new ciphers that are periodically added. Customers have client emulators that support session ID caching and renegotiation of a cipher key during an active secure session. Customers want to specify a certificate label to be used instead of the default key ring certificate.

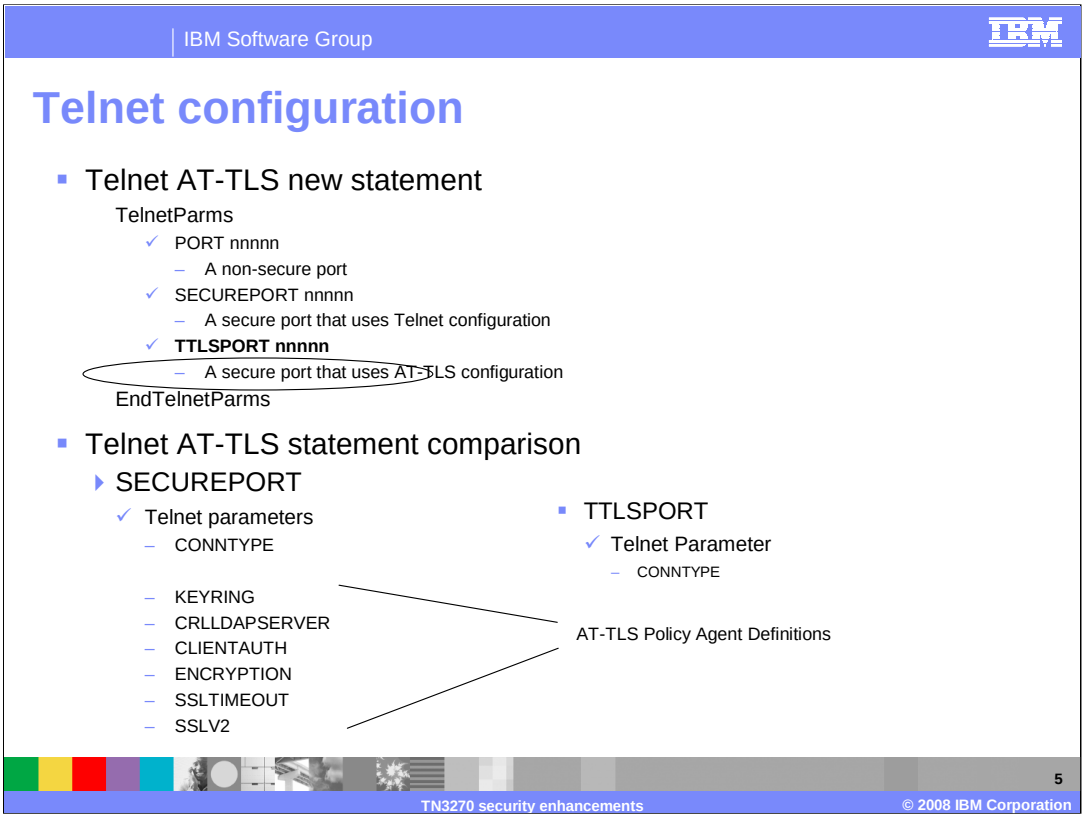
System SSL has continued to improve and now supports these functions.

Telnet configuration has not been enhanced to take advantage of the new System SSL function. Two AES ciphers have recently been added to Telnet, but Telnet still does not support all the ciphers available for use by System SSL.

Application Transparent Transport Layer Security (AT-TLS) was introduced in z/OS V1R7 and supports all of the new functions in System SSL. AT-TLS is the z/OS Communications server strategic application security option and will continue to be updated as new System SSL functions become available.

To satisfy existing Telnet security requirements, the choices were to make additional updates to Telnet configuration to make use of the new System SSL function or to enable Telnet to fully use AT-TLS. Because AT-TLS is strategic and provides System SSL functions beyond the current requirements, the choice was made to enable Telnet for AT-TLS. With AT-TLS, you will be able to specify multiple key rings for different ports or the same port, change key rings without stopping ports, specify up to five CRL LDAP servers, specify new ciphers immediately, cache session IDs, manage session IDs and cipher renegotiation, and use a certificate other than the default certificate during the SSL negotiations.

Telnet provides you with much flexibility through its current configuration options. That flexibility had to be retained while moving to AT-TLS. Being able to specify Conntype and client authentication at very granular levels is a popular Telnet feature.



Ports are defined as either non-secure basic ports or secure ports by specifying either PORT nnnnn or SECUREPORT nnnnn. A basic port can support only non-secure connections. A SECUREPORT interacts with System SSL to create a single System SSL environment used by the port. A SECUREPORT can support secure or non-secure connections depending on the Conntype value. Security-related parameters can be specified only for a SECUREPORT.

A new port definition, TTLSPORT, is created to define a Telnet port that uses AT-TLS security configuration instead of using Telnet configuration. To ease migration to TTLSPORT, the SECUREPORT option is retained at the current level of functionality.

Several security configuration statements used by SECUREPORT to create a System SSL environment are moved to AT-TLS when a TTLSPORT is defined. AT-TLS configuration is managed using the Policy Agent. For either port type, a System SSL environment must be created. If SECUREPORT is used, Telnet configuration defines the values used to create the environment. If TTLSPORT is used, AT-TLS configuration defines the values used to create the environment.

Notice the Conntype statement remains for both port types. Conntype is not a System SSL environment variable. Conntype is used by Telnet to decide if System SSL should be used to set up security for a particular connection.

The notes page that follows, titled “Telnet/AT-TLS Conversion”, gives the conversion for all Telnet to AT-TLS statements and a detailed discussion of how to implement client authentication in AT-TLS. Following the notes page is an example showing both the Telnet configuration and the AT-TLS configuration.

Telnet/AT-TLS conversion

Telnet statement	AT-TLS equivalent statement	Location of AT-TLS statement
CLIENTAUTH NONE	HANDSHAKEROLE SERVER	TTLSENVIRONMENTACTION
CLIENTAUTH SSLCERT	HANDSHAKEROLE SERVERWITHCLIENTAUTH CLIENTAUTHTYPE REQUIRED	TTLSENVIRONMENTACTION TTLSENVIRONMENTADVANCEDPARMS
CLIENTAUTH SAFCERT	HANDSHAKEROLE SERVERWITHCLIENTAUTH CLIENTAUTHTYPE SAFCHECK	TTLSENVIRONMENTACTION TTLSENVIRONMENTADVANCEDPARMS
CRLLDAPSERVER	GSK_LDAP_SERVER GSK_LDAP_SERVER_PORT	TTLSGSKLDAPPARMS
ENCRYPTION	TTLSCIPHERPARMS	TTLSENVIRONMENTACTION
KEYRING	KEYRING	TTLSKEYRINGPARMS
SSLV2	SSLV2	TTLSENVIRONMENTADVANCEDPARMS
SSLTIMEOUT	HANDSHAKETIMEOUT	TTLSENVIRONMENTADVANCEDPARMS

HANDSHAKEROLE and TTLSCIPHERPARMS can also be in TTLSCONNECTIONACTION

SSLV2 and SSLTIMEOUT can also be in TTLSCONNECTIONADVANCEDPARMS

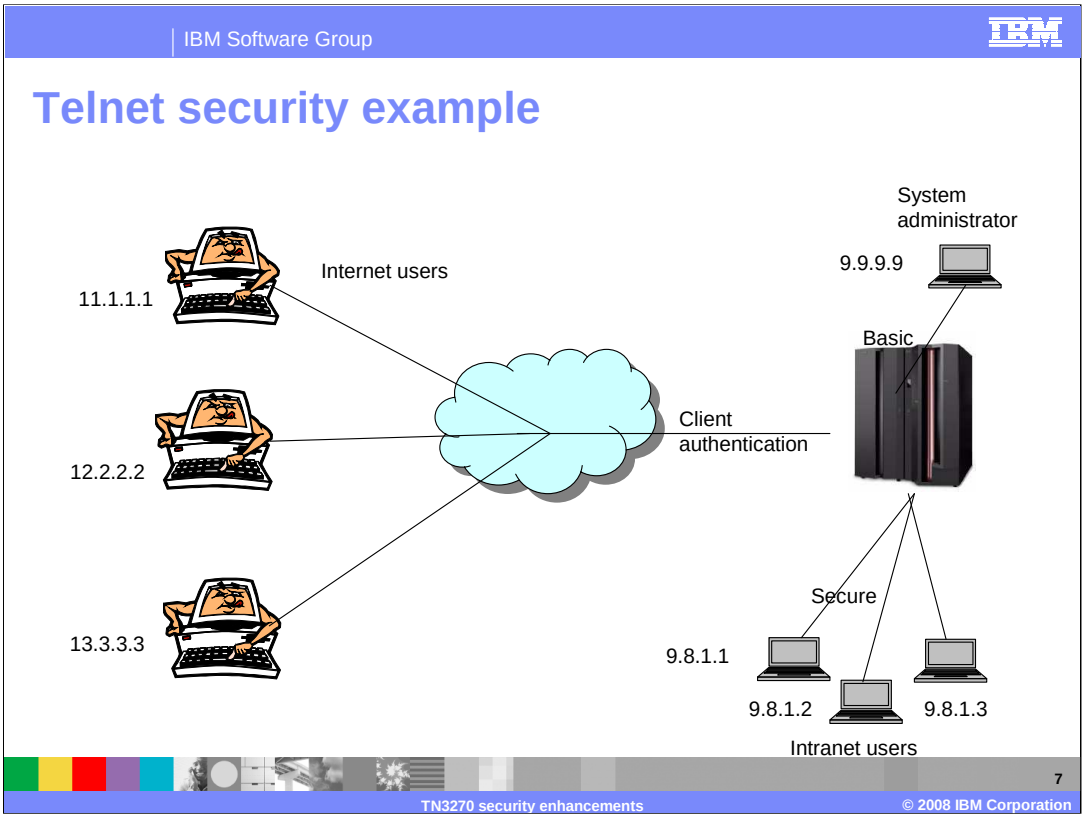
This table shows the AT-TLS equivalent statements for the Telnet security statements.

There are many variations possible with the Telnet profile statement CLIENTAUTH. In AT-TLS, whether client authentication is done or not done is controlled by the HandshakeRole parameter on either the TTLSEnvironmentAction or TTLSConnectionAction statements. If the connection needs client authentication, the level of authentication is controlled with the ClientAuthType parameter on the TTLSEnvironmentAdvancedParms statement.

If you have both CLIENTAUTH SSLCERT and CLIENTAUTH SAFCERT in different ParmsGroup statements in your Telnet configuration, you need two TTLSEnvironmentAction statements; one TTLSEnvironmentAction statement for ClientAuthType Required and one TTLSEnvironmentAction statement for ClientAuthType SAFCheck. Two TTLSRule statements, each referencing a different TTLSEnvironmentAction statement in AT-TLS, replace the two PARMSMAP statements in the Telnet profile.

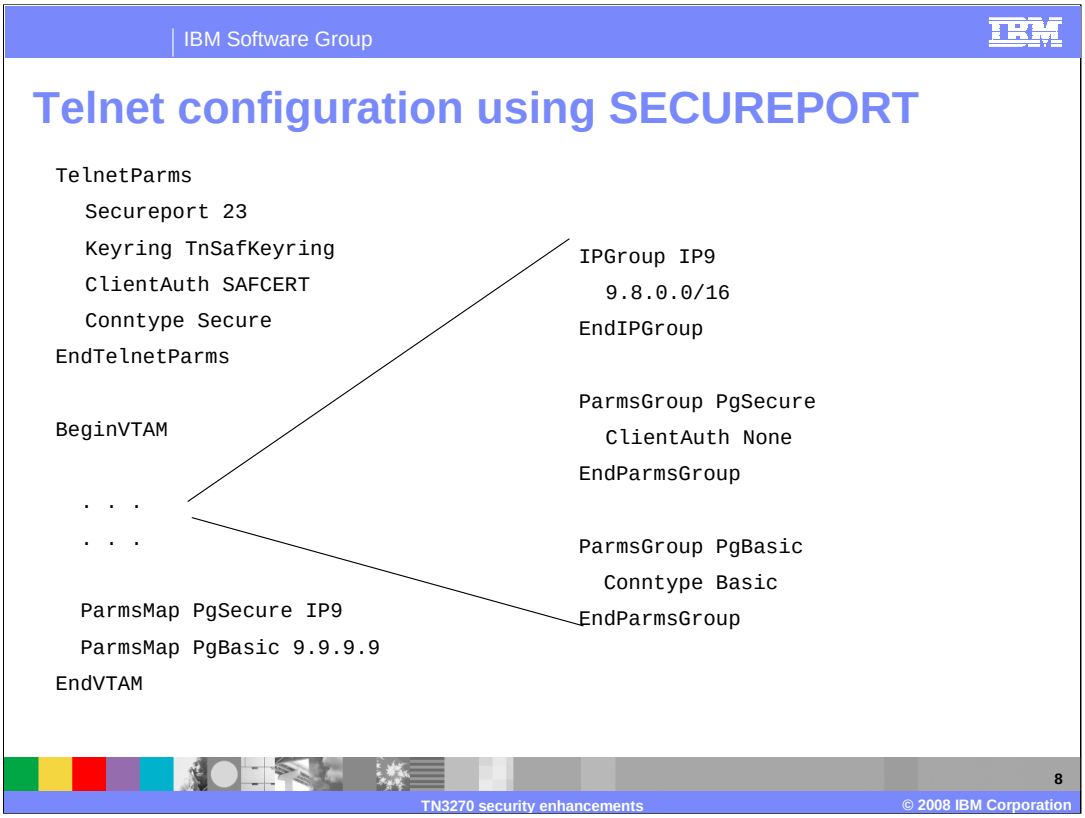
If you have a mixture of CLIENTAUTH NONE and CLIENTAUTH SAFCERT, you need a TTLSEnvironmentAction statement with HandshakeRole ServerWithClientAuth, and a TTLSConnectionAction statement with HandshakeRole Server. Two TTLSRule statements in AT-TLS (one with the TTLSConnectionAction statement and one without) replace the two PARMSMAP statements in the Telnet profile. You could instead create a second TTLSEnvironmentAction statement with HandshakeRole Server, but many more resources are associated with a TTLSEnvironmentAction statement compared to a TTLSConnectionAction

System SSL defines client authentication type only at the environment level and controls whether client authentication is performed by the handshake role which can be specified at the environment or connection level.



The easiest way to show the difference between SECUREPORT and TTLSPORT configuration is through an example.

Assume the following environment. There are several users on the Internet who require secure connections with server authentication and a client authentication level that requires a mapping of the client certificate to a Security server user ID. Several more users are on the company intranet where secure connections with only server authentication are required. Finally, the system administrator has a fixed IP address and does not require any security.

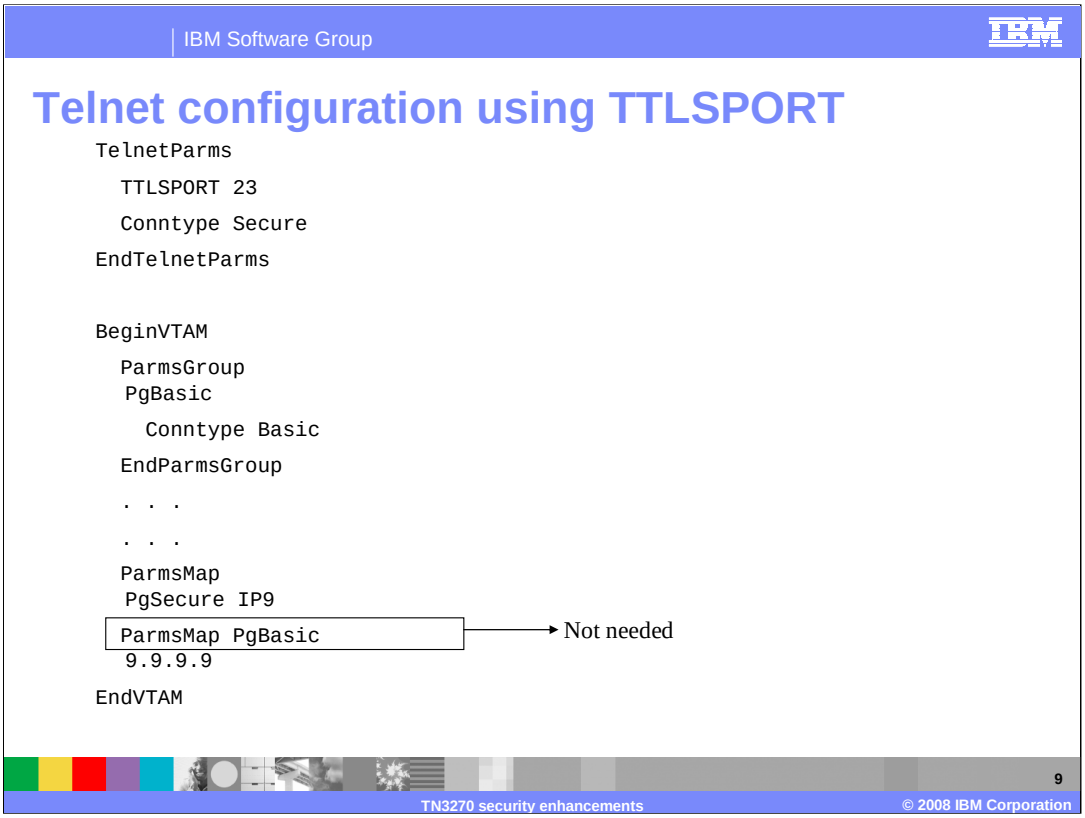


First here is a look at the security statements needed when SECUREPORT is used to define port security for the example from the previous slide.

Within TelnetParms the key ring name (TNSafKeyring), client authentication level (SAFCert), and connection type (Secure) are specified. Conntype secure is the default but is shown for completeness. The key ring and client authentication levels are passed to System SSL to create the System SSL environment used for securing connections at the port level. If nothing else were coded, all connections would be secure and require client authentication, satisfying the internet requirement on the previous slide.

An IPGroup, IP9, defines a subset of connections that represent all intranet company users. A Parmsgroup, PgSecure, defines internal security with client authentication off. The ParmsMap statement is used to map PgSecure to IP9. The result is any connection IP address starting with 9.8 will not be required to provide client authentication, satisfying the intranet requirement on the previous slide.

Another Parmsgroup, PgBasic, defines a connection type as basic (non-secure). This Parmsgroup is mapped to the system administrator's IP address directly giving the system administrator a basic connection, satisfying the administrator requirement on the previous slide.

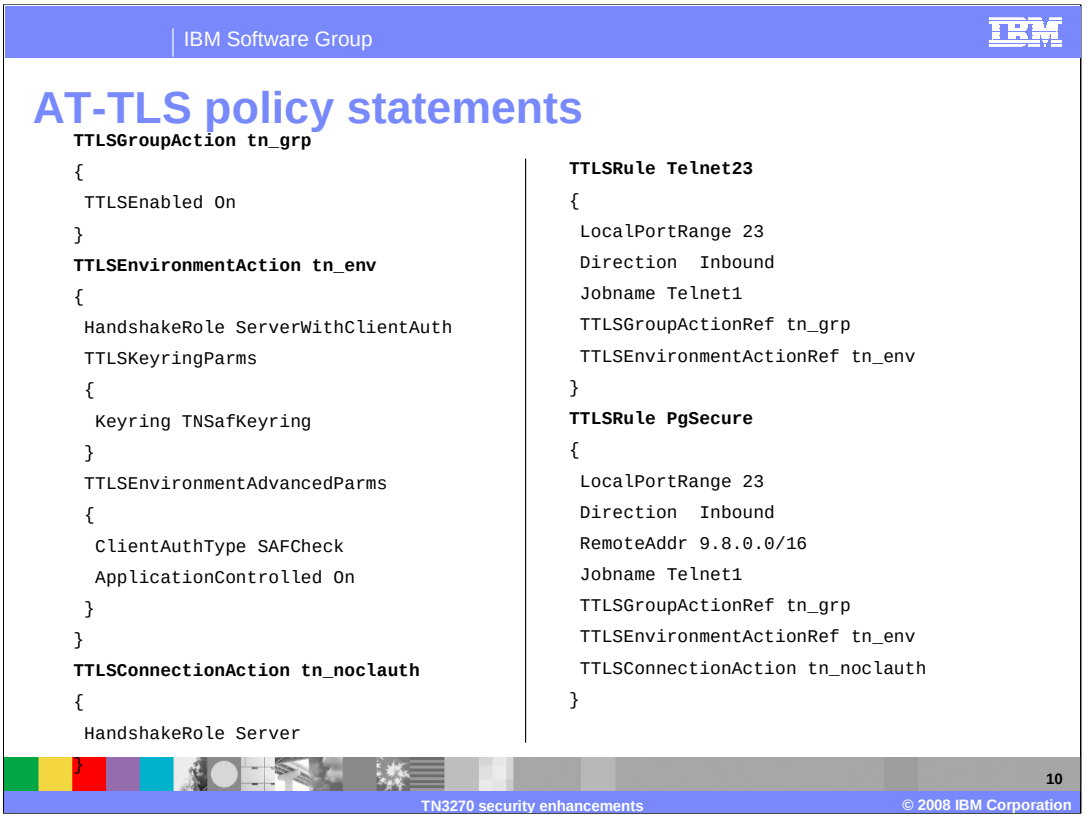


Telnet security statements are not needed when TTLSPORT is used to define the secure port.

TelnetParms uses the TTLSPORT statement and does not need the Key ring and client authentication statements that are now configured in AT-TLS policy in the Policy Agent. Connection type is controlled by Telnet and must still be specified. Conntype secure is the default but is shown for completeness. Combined with the AT-TLS statements on the next slide the internet requirements from the example are satisfied.

Because client authentication control is done by AT-TLS the IP9 IPGroup and PgSecure Parmsgroup are not needed in Telnet. Combined with the AT-TLS PgSecure rule on the next slide the intranet requirements from the example are satisfied.

Because Telnet controls whether security is initiated, the PgBasic Parmsgroup remains in Telnet and continues to be mapped to IP address 9.9.9.9 to give the system administrator a basic, non-secure, connection which satisfies the administrator requirement from the example.

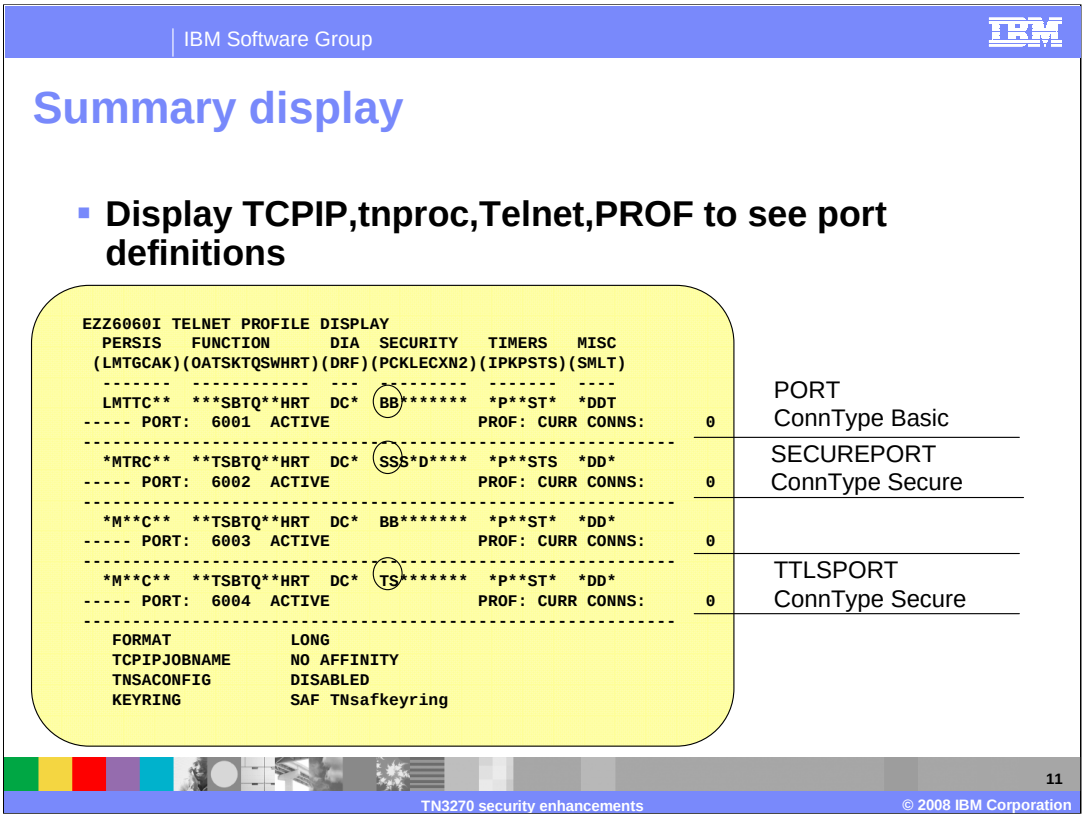


TTLSRULEs require group and environment actions. Connection actions are optional. The group action, TTLSGroupAction, defines high level settings such as if AT-TLS is enabled. The Environment action, TTLSEnvironmentAction, defines the System SSL environment and is where key ring and client authentication are defined. The connection action, TTLSConnectionAction, is used to override certain environment settings at the connection level.

TTLSRULE Telnet23 is similar to the TelnetParms statement defining security for the entire port and will define the security for the internet users from the example. The group action enables AT-TLS. The environment action specifies that key ring TNSafKeyring should be used and the client authentication level is SAFCheck. The AT-TLS parameter SAFCheck is equivalent to the Telnet parameter SAFCert. Combined with the Telnet statements on the previous slide the internet requirements from the example are satisfied.

TTLSRULE PgSecure uses the RemoteAddr statement to specify the internal subset of port 23 connections. The connection action is also specified. The connection action turns off client authentication by setting handshake role to Server instead of ServerWithClientAuth. The result is all internal clients will have secure connections without client authentication. This TTLSRULE essentially replaces the PgSecure ParmsGroup, IP9 IPGroup, and the ParmsMap statement in Telnet. Combined with the Telnet statements on the previous slide the intranet requirements from the example are satisfied.

AT-TLS is not referenced by the system administrator connection based on the Telnet Conntype Basic statement.



The Telnet profile summary display shows how ports are defined. Under the security column the first character position indicates the port definition and the second character indicates the connection type.

Port 6001, BB, indicates a Basic port defined by the PORT statement and a Connection type of Basic.

Port 6002, SS, indicates a Secure port defined by the SECUREPORT statement with Telnet configuration and a Connection type of Secure.

Port 6003 is the same as port 6001.

Port 6004, TS, indicates a Secure port defined by the TTLSPORT statement with AT-TLS configuration and a Connection type of Secure.

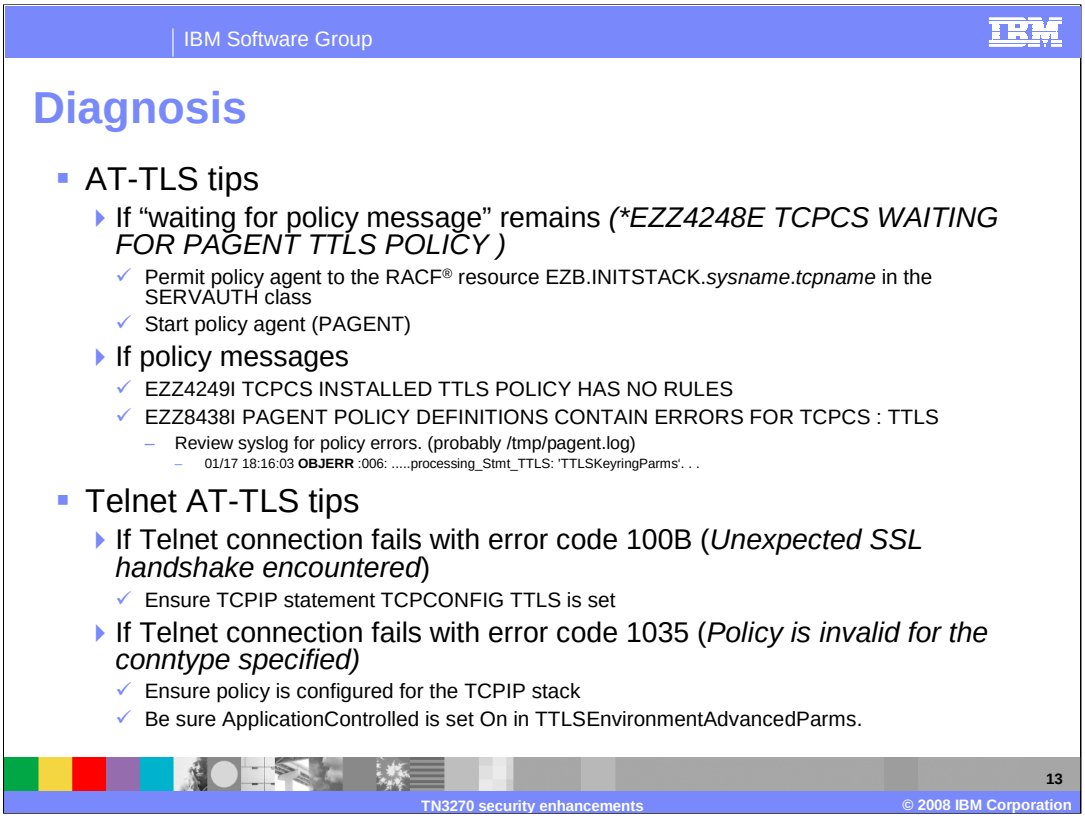
Detail display

- **Display TCPIP,tnproc,Telnet,PROF,DETail for detail**

```
EZZ6080I TELNET PROFILE DISPLAY
  PERSIS   FUNCTION      DIA  SECURITY   TIMERS   MISC
 (LMTGCAK)(OATSKTQSWHRT)(DRF)(PCKLECXN2)(IPKPSTS)(SMLT)
  -------  ------------  ---  ---------  -------  ----
  *******  **TSBTQ***RT  EC*  BB**D****  *P**STS  *DD* *DEFAULT
  -M--C--  ------------  DC-  ---------  -------  ---- *TGLOBAL
  -------  ---------H--  ---  TS--*----  ------*  ---- *TPARMS
  *M**C**  **TSBTQ**HRT  DC*  TS*******  *P**ST*  *DD* CURR
 PERSISTENCE
   NOLUSESSIONPEND
   . . .
 SECURITY
  TTLSPORT                    6004
   CONNTYPE                  SECURE
   KEYRING                   TTLS
   CRLLDAPSERVER             TTLS
   ENCRYPTION                TTLS
   CLIENTAUTH                TTLS
   NOEXPRESSLOGON
   NONACUSERID
   SSLV2                     TTLS
TIMERS
```


The Telnet profile detail display shows the values of all parameters in the profile. With TTLSPORT, most configuration variables have moved from Telnet configuration to AT-TLS configuration which is represented by the TTLS values.

TTLS indicates the variable is defined in the AT-TLS policy and not in Telnet.

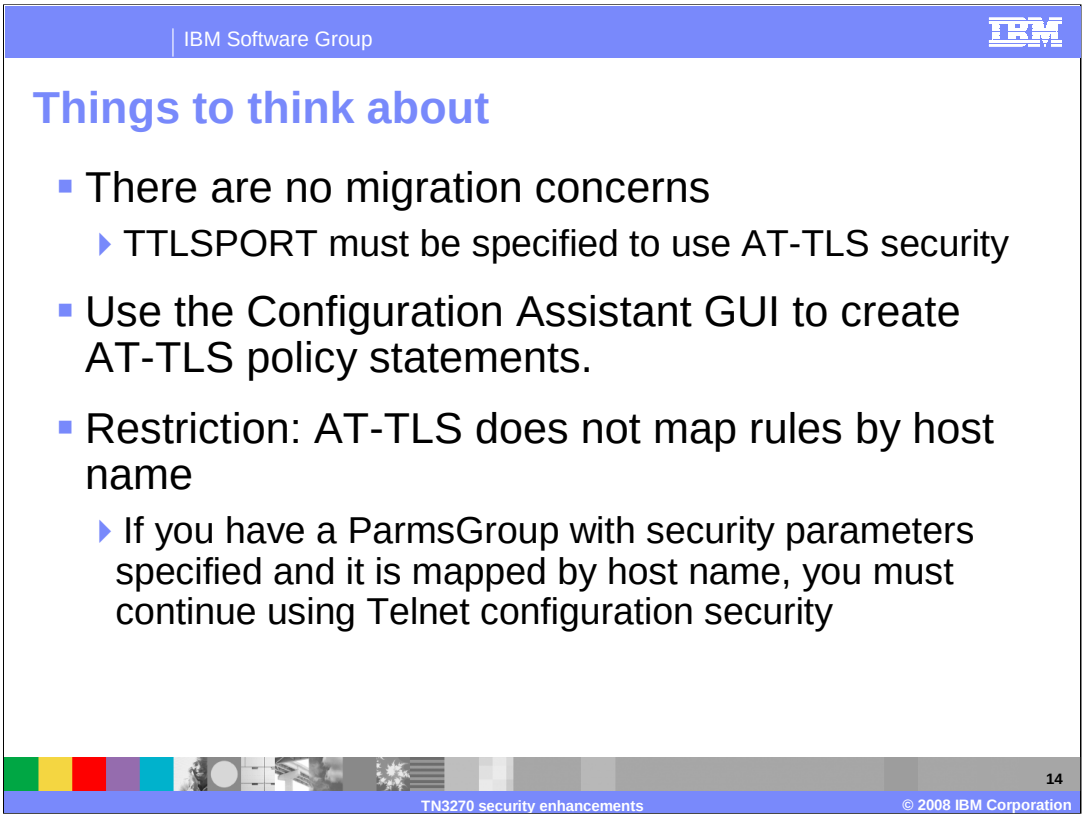


If message EZZ4248E remains, you need to permit the Policy Agent to the INITSTACK Resource Access Control Facility, RACF, resource in the SERVAUTH class or start the Policy Agent.

If EZZ4249I indicates there are no rules for TTLS, verify the policy data sets are set up properly. There are many ways to define policy. One common way is to define TCP Image statements in the pagent configuration file. Ensure there is a TCPImage statement for the TCPIP stack used by Telnet and ensure the TCPImage policy file has a TTLSCONFIG statement pointing to the configuration file that defines the Telnet rules. If EZZ8438I indicates an error with a policy statement, review the pagent syslog output. It is probably in /tmp/pagent.log. Look for the keyword OBJERR immediately following the timestamp.

If Telnet connections fail with error code 100B, “Unexpected SSL handshake encountered.”, AT-TLS is probably not enabled in the stack. All connections are considered non-secure but the client is sending a secure handshake.

If Telnet connections fail with error code 1035, “Policy is invalid for the conntype specified.”, either there is no policy configured for the connection or the policy did not specify that for Telnet, AT-TLS must be application controlled.



There are no migration issues with AT-TLS for Telnet. You can continue to define secure ports with the SECUREPORT statement, but you will have access to many more System SSL functions if you use TTLSPORT.

In some cases the creation of AT-TLS policy files for Policy Agent can be difficult. The Configuration Assistant GUI will, through a series of wizards and online help panels, generate AT-TLS configuration files for any number of z/OS images with any number of TCPIP stacks per image.

From the example earlier, you saw that TTLSRULE was, in some cases, a replacement for the ParmsGroup and ParmsMap statements. In Telnet, you can define a client identifier as a host name or a host name group and map a ParmsGroup to that client identifier. That ParmsGroup might contain security parameters. TTLSRULE does not support the concept of a connection matching a specified host name. If you use host names as a client identifier to differentiate security variables you will have to continue to use the Telnet configuration for security.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_TN3270_Security.ppt

This module is also available in PDF format at: ../TN3270_Security.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

OS/390 RACF z/OS

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance 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 or performance improvements equivalent to the ratios stated here.