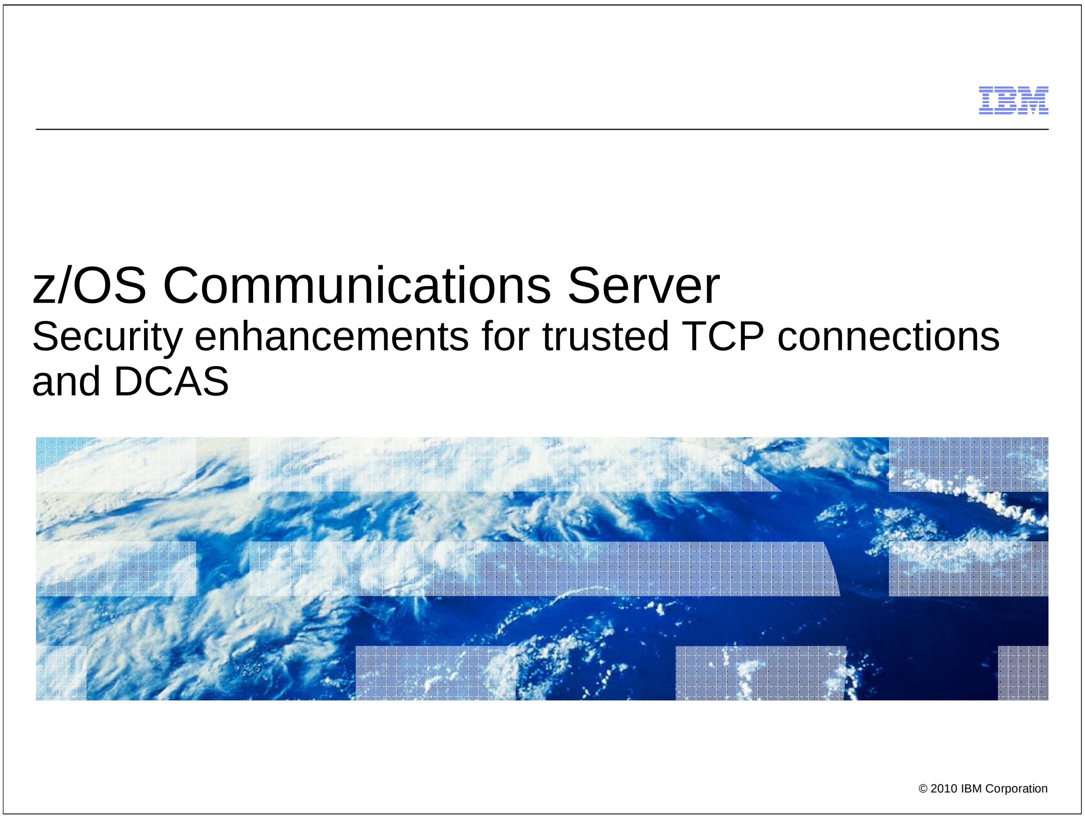
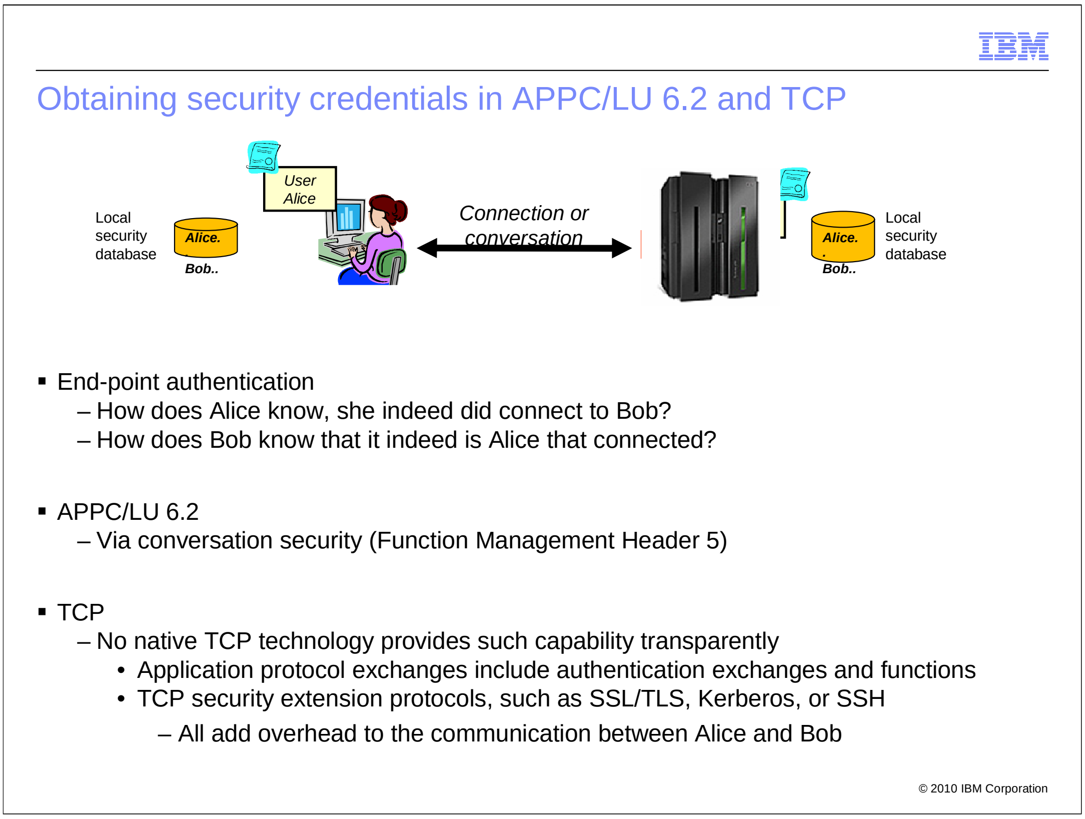


z/OS Communications Server

Security enhancements for trusted TCP connections and DCAS

There are several new security functions in z/OS V1R12 Communications Server. This presentation describes trusted TCP connections and an update to the digital certificate access server (DCAS).



Obtaining security credentials in APPC/LU 6.2 and TCP

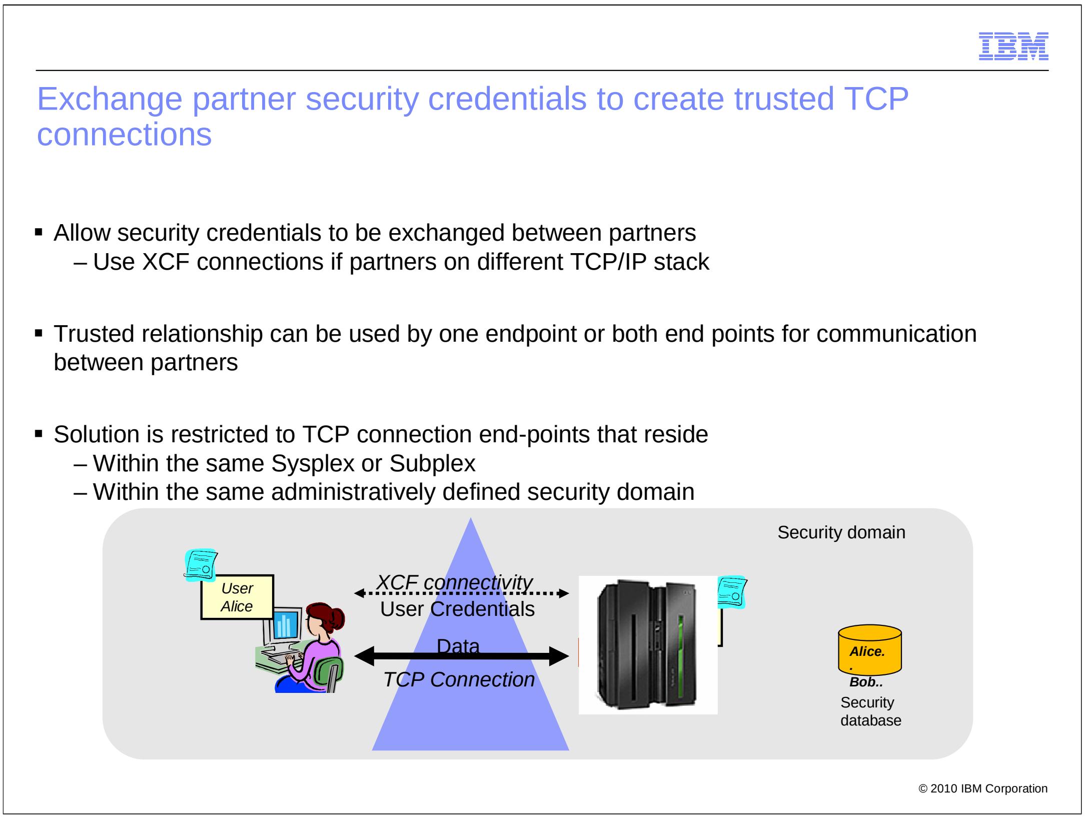
- End-point authentication
  - How does Alice know, she indeed did connect to Bob?
  - How does Bob know that it indeed is Alice that connected?
- APPC/LU 6.2
  - Via conversation security (Function Management Header 5)
- TCP
  - No native TCP technology provides such capability transparently
    - Application protocol exchanges include authentication exchanges and functions
    - TCP security extension protocols, such as SSL/TLS, Kerberos, or SSH
      - All add overhead to the communication between Alice and Bob

When client/server communication requires end-point authentication, the TCP protocol in itself does not provide such support transparently.

SNA/APPC provides such capabilities with the conversation attach request (the Function Management Header 5). For SNA applications, SNA LU 6.2 communication can return user-specific security credentials in an LU 6.2 transaction initiation request. The user-specific security credentials can contain the user ID, password and group. This information can be used by the LU 6.2 attach manager function on the server side to authenticate the transaction user. It can also be used to establish a user-specific security environment in which the LU 6.2 transaction program starts executing on the server node.

For TCP, there are in general two methods available for end-point authentication. The first is to add the exchange of end-user credentials to the TCP payload protocol (application protocol), where each application protocol implements exchange of end-user credentials. The various security extensions can be SSL/TLS (Secure Sockets Layer/Transport Layer Security), Kerberos, or SSH (Secure Shell) encrypted. A good example is the FTP USER and PASS commands.

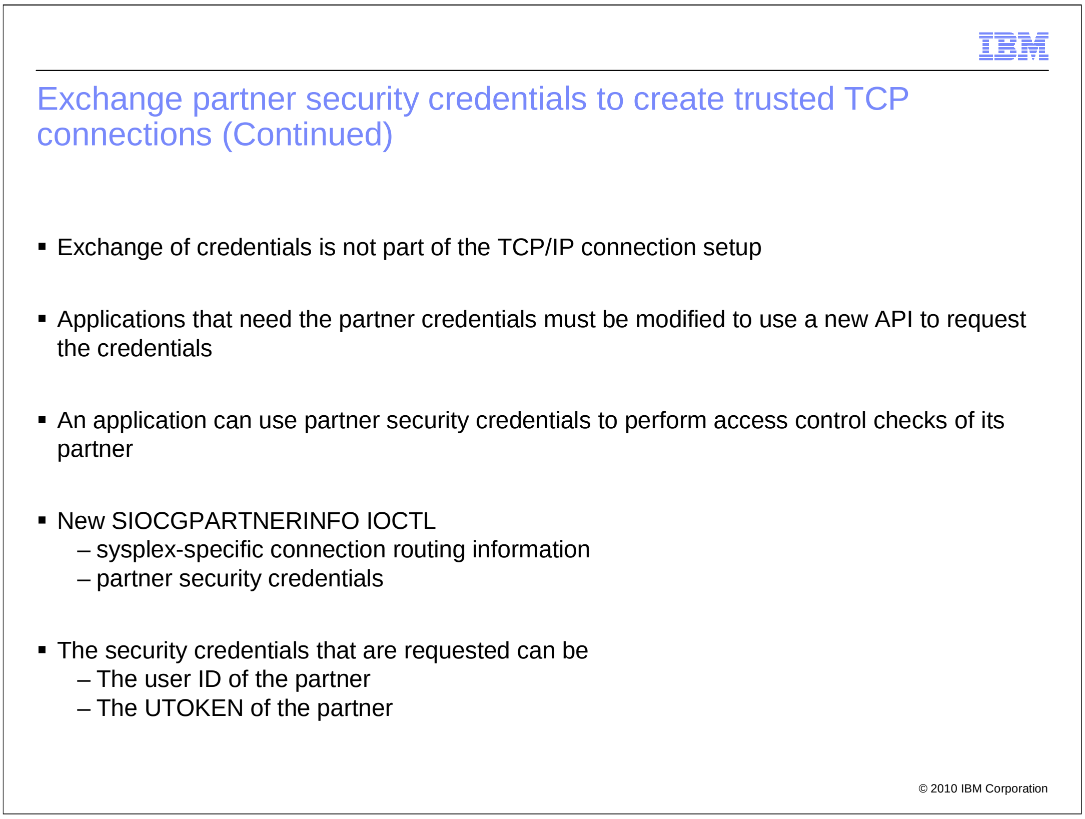
The other method is to extend the TCP protocol with one or more security protocols, such as SSL/TLS, Kerberos, or SSH. SSL/TLS protocols combined with operating-system specific support for mapping X.509 certificates to user definitions are widely used. SSL/TLS can provide both single-sided authentication for server authentication only or mutual authentication for client and server authentication.



Exchange partner security credentials to create trusted TCP connections

- Allow security credentials to be exchanged between partners
  - Use XCF connections if partners on different TCP/IP stack
- Trusted relationship can be used by one endpoint or both end points for communication between partners
- Solution is restricted to TCP connection end-points that reside
  - Within the same Sysplex or Subplex
  - Within the same administratively defined security domain

z/OS V1R12 Communications Server provides a way to exchange security credentials between partners. If the partners are on different TCP/IP stacks, the exchange is across an XCF (cross-system coupling facility) connection. This trusted relationship can be used for one-way or two-way communication. One-way communication is when one endpoint extracts partner security credentials. Two-way communication is when both end points extract partner security credentials.



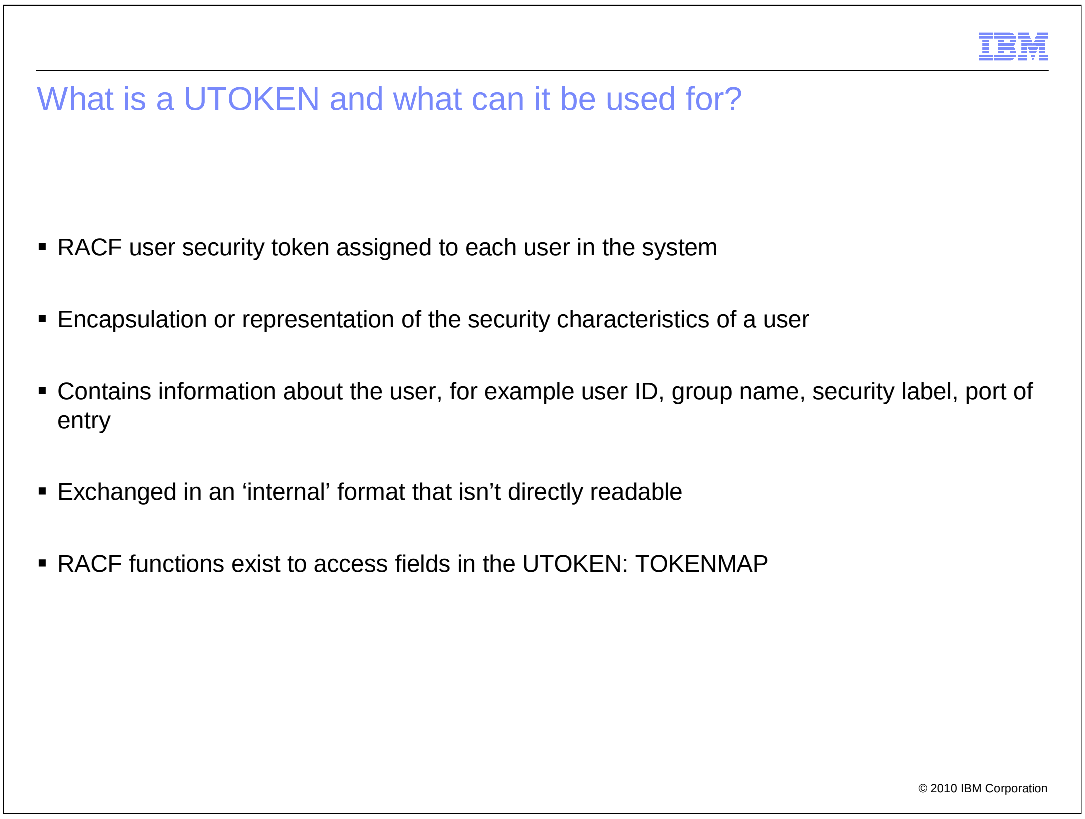
IBM

Exchange partner security credentials to create trusted TCP connections (Continued)

- Exchange of credentials is not part of the TCP/IP connection setup
- Applications that need the partner credentials must be modified to use a new API to request the credentials
- An application can use partner security credentials to perform access control checks of its partner
- New SIOCGPARTNERINFO IOCTL
  - sysplex-specific connection routing information
  - partner security credentials
- The security credentials that are requested can be
  - The user ID of the partner
  - The UTOKEN of the partner

The exchange of security credentials is not part of the TCP/IP connection setup. Only the applications requesting credentials need modification. The exploiting socket partners can use the partner security credentials to perform access control checks. For example they can check the partner has sufficient privileges and has a “Trust relationship” with the partner to create a trusted TCP connection.

A new API SIOCGPARTNERINFO IOCTL command was created to provide multiple functions. This IOCTL returns the sysplex-specific connection routing information and the partner security credentials.

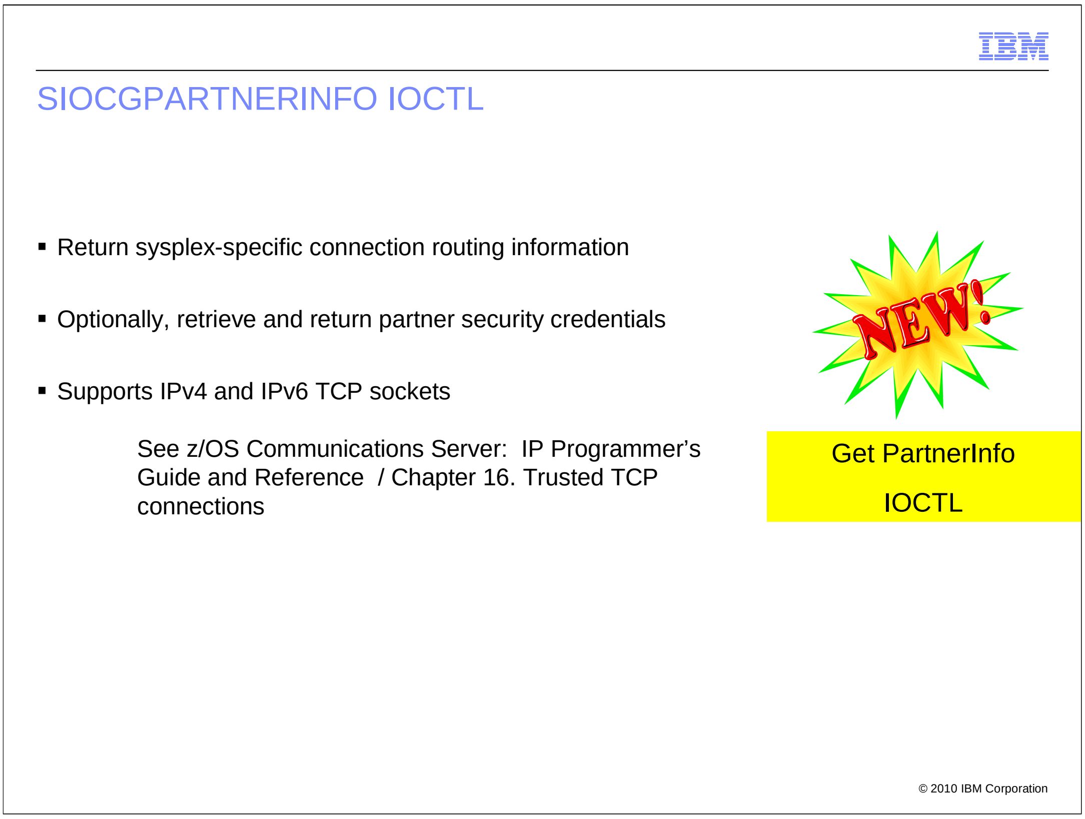


What is a UTOKEN and what can it be used for?

- RACF user security token assigned to each user in the system
- Encapsulation or representation of the security characteristics of a user
- Contains information about the user, for example user ID, group name, security label, port of entry
- Exchanged in an ‘internal’ format that isn’t directly readable
- RACF functions exist to access fields in the UTOKEN: TOKENMAP

Without requesting a password, an authorized program can use a UTOKEN to establish a security environment for a user from which the UTOKEN came. Other security products also provide the concept of a UTOKEN, but the content can differ from the UTOKEN provided by RACF. In general, a UTOKEN cannot be used between different security products.

For information about what is provided in the UTOKEN by the ICHRUTKN macro, see *z/OS Security Server RACF Data Areas.*



SIOCGPARTNERINFO IOCTL

- Return sysplex-specific connection routing information
- Optionally, retrieve and return partner security credentials
- Supports IPv4 and IPv6 TCP sockets

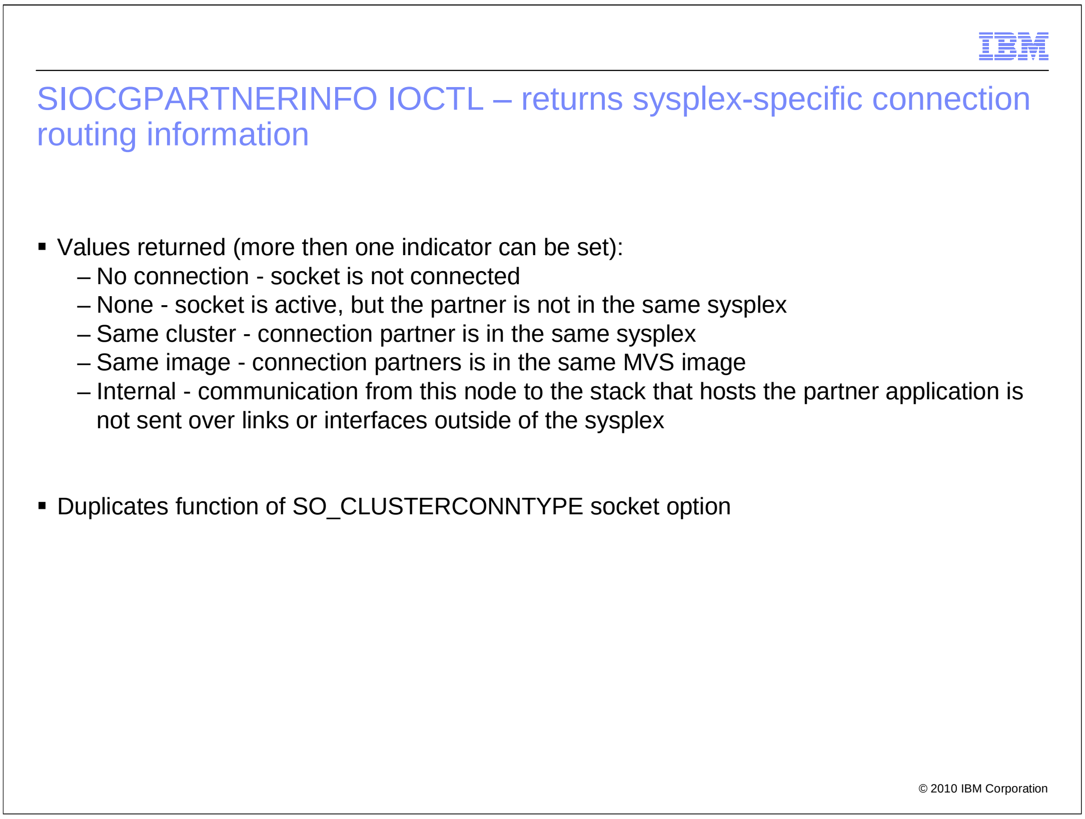
See z/OS Communications Server: IP Programmer’s Guide and Reference / Chapter 16. Trusted TCP connections

NEW!

Get PartnerInfo

IOCTL

The new SIOCGPARTNERINFO IOCTL command always returns the sysplex-specific connection routing information. It will optionally retrieve and return the partner security credentials. The sysplex-specific connection routing information was previously only supported in the SO_CLUSTERCONNTYPE socket option command. The partner security credentials are new information returned to an application. This IOCTL command is supported on both IPv4 and IPv6 TCP sockets.



SIOCGPARTNERINFO IOCTL – returns sysplex-specific connection routing information

- Values returned (more then one indicator can be set):
  - No connection - socket is not connected
  - None - socket is active, but the partner is not in the same sysplex
  - Same cluster - connection partner is in the same sysplex
  - Same image - connection partners is in the same MVS image
  - Internal - communication from this node to the stack that hosts the partner application is not sent over links or interfaces outside of the sysplex
- Duplicates function of SO_CLUSTERCONNTYPE socket option

The SIOCGPARTNERINFO IOCTL always returns the connection routing information. The connection routing information can be one or more of these values.

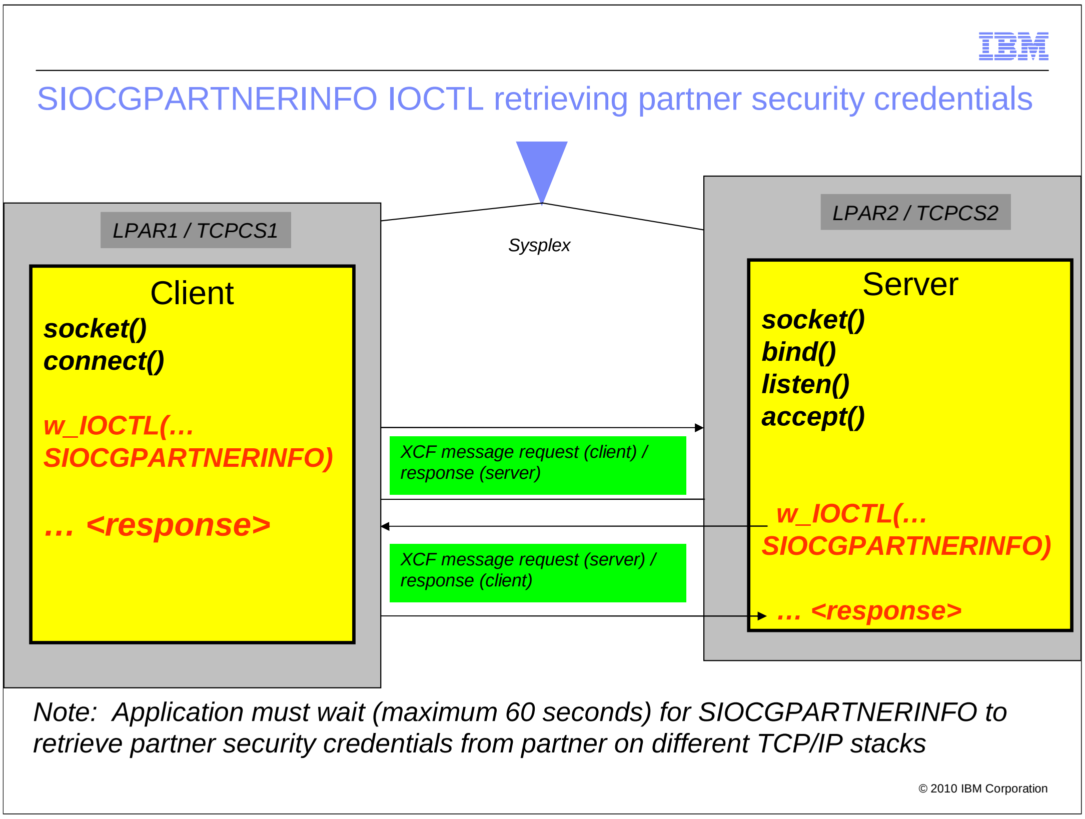
A “No connection” value is set if the socket is not connected.

A “None” value is set if the socket is active, but the partner is not in the same sysplex. A “same cluster” value is set if the connection partner is in the same sysplex.

A “same image” value is set if the connection partner is in the same MVS image.

A “internal” value is set if the communication from this node to the stack that hosts the partner application is not sent over links or interfaces outside of the sysplex.

The connection routing information is a duplicate function of SO_CLUSTERCONNTYPE socket option (getsockopt). The existing SO_CLUSTERCONNTYPE socket option is only supported in the TCP/IP C API and z/OS UNIX System Services API.



SIOCGPARTNERINFO IOCTL retrieving partner security credentials

Sysplex

LPAR1 / TCPCS1

Client
socket()
connect()

w_IOCTL(…
SIOCGPARTNERINFO)

… <response>

XCF message request (client) / response (server)

XCF message request (server) / response (client)

LPAR2 / TCPCS2

Server
socket()
bind()
listen()
accept()

w_IOCTL(…
SIOCGPARTNERINFO)

… <response>

Note: Application must wait (maximum 60 seconds) for SIOCGPARTNERINFO to retrieve partner security credentials from partner on different TCP/IP stacks

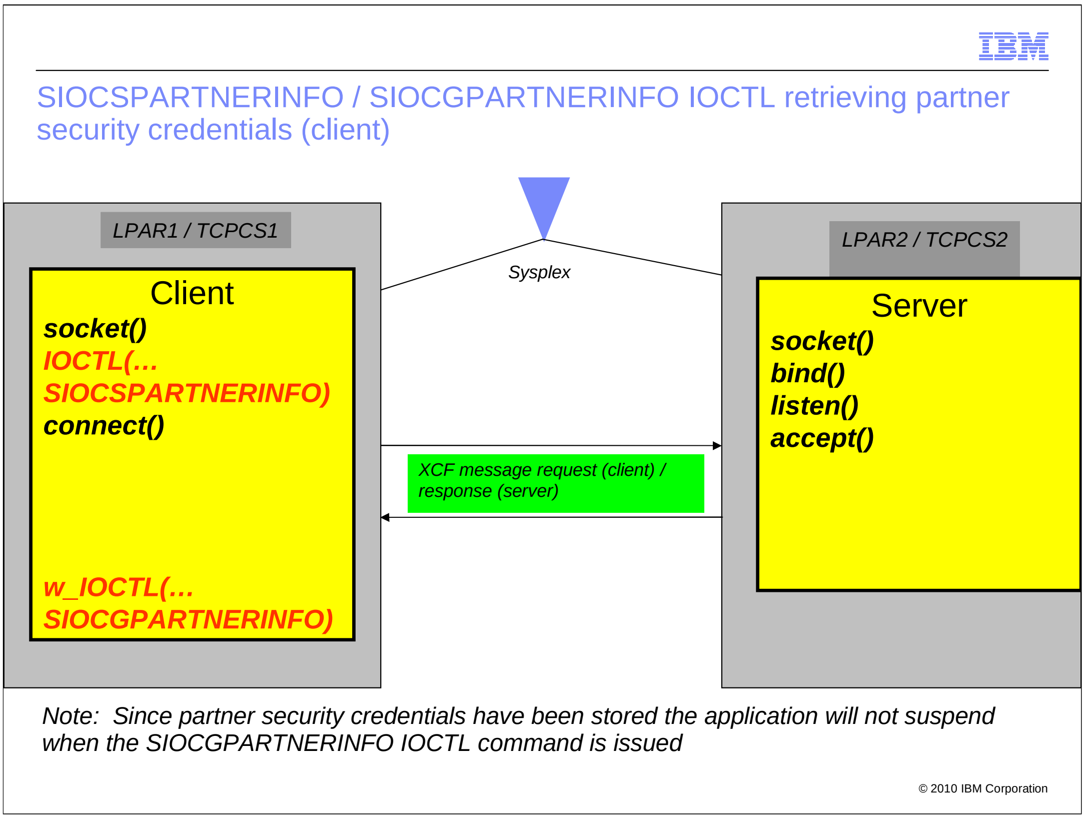
The SIOCGPARTNERINFO IOCTL command, if requested, can return your partner security credentials. Based on the input request type, a user ID, user security token (UTOKEN) or both can be returned. The address-space user ID and UTOKEN, and if available the task-level user ID and UTOKEN, are returned. Retrieving this information has minimal performance impact during TCP connection time.

The partner security credentials can only be retrieved if the application runs APF-authorized, in supervisor state, or at least read access assigned to the user in the SERVAUTH class for profile EZB.IOCTL.sysname.tcpprocname.PARTNERINFO.

In this example, when the SIOCGPARTNERINFO IOCTL is issued, the application is suspended. The application must wait for the SIOCGPARTNERINFO command to retrieve and return the partner security credentials from applications on different stacks.

The client application must issue the SIOCGPARTNERINFO IOCTL after the connect command is completed. The input PartnerInfo PI_TIMEOUT is set to a non-zero time value of 1-60 seconds indicating that it can be suspended. The XCF message request is sent from the client stack (TCPCS1) to the server stack (TCPCS2). The partner security credentials are returned to the client stack and then given to the client application.

The server application must issue the SIOCGPARTNERINFO IOCTL after the accept command is complete. The input PartnerInfo PI_TIMEOUT is set to a non-zero time value of 1-60 seconds indicating that it can be suspended. The XCF message request is sent from the server stack (TCPCS2) to the client stack (TCPCS1). The partner security credentials are returned to the server stack and then given to the server application.



SIOCSPARTNERINFO / SIOCGPARTNERINFO IOCTL retrieving partner security credentials (client)

LPAR1 / TCPCS1

Client

socket()

IOCTL(…
SIOCSPARTNERINFO)

connect()

w_IOCTL(…
SIOCGPARTNERINFO)

Sysplex

XCF message request (client) / response (server)

LPAR2 / TCPCS2

Server

socket()

bind()

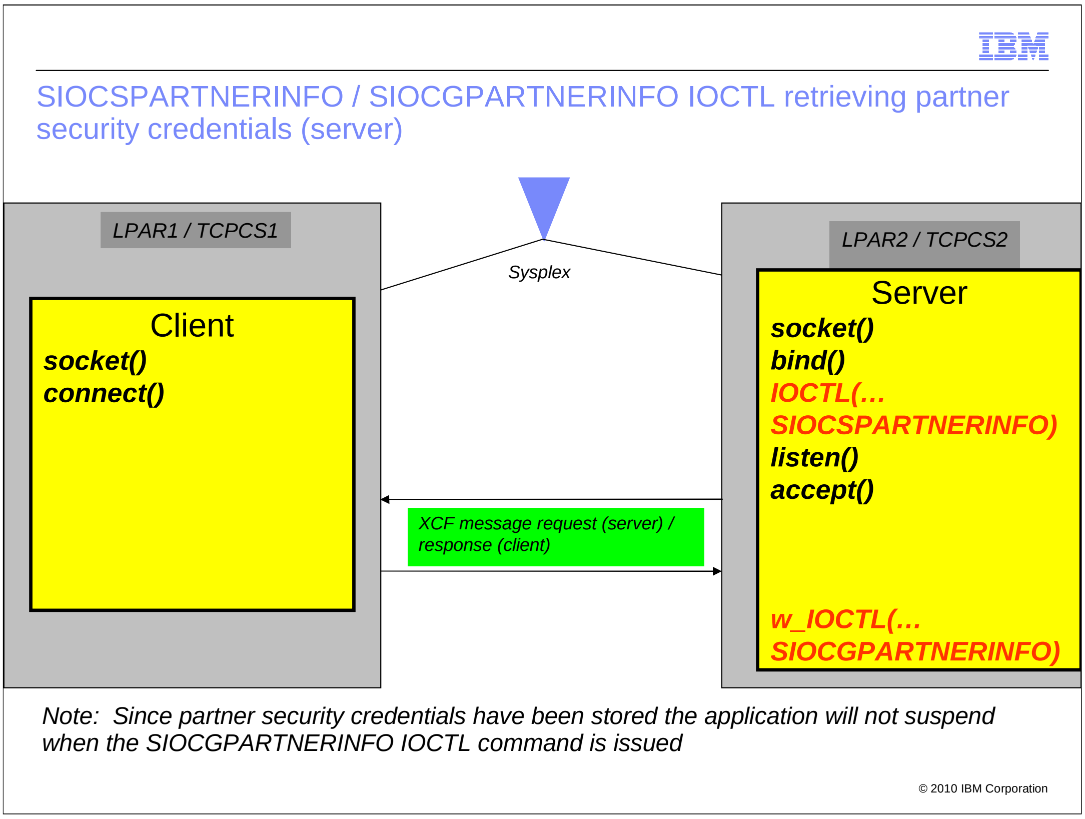
listen()

accept()

Note: Since partner security credentials have been stored the application will not suspend when the SIOCGPARTNERINFO IOCTL command is issued

You can use the SIOCSPARTNERINFO IOCTL command to avoid suspending when requesting your partner security credentials on a different TCP/IP stack using the SIOCGPARTNERINFO IOCTL command. The SIOCSPARTNERINFO command is supported on both IPv4 and IPv6 TCP sockets.

In this example, the client application issues the SIOCSPARTNERINFO IOCTL before the connect command. After the connect is complete, an XCF message request is sent by the client stack (TCPCS1) to the server stack (TCPCS2) for the partner security credentials. The client stack will store the received information before the SIOCGPARTNERINFO IOCTL is issued. The client application then issues the SIOCGPARTNERINFO IOCTL for the partner security credentials and it is returned immediately without suspending.



SIOCSPARTNERINFO / SIOCGPARTNERINFO IOCTL retrieving partner security credentials (server)

LPAR1 / TCPCS1

Client

socket()
connect()

Sysplex

XCF message request (server) / response (client)

LPAR2 / TCPCS2

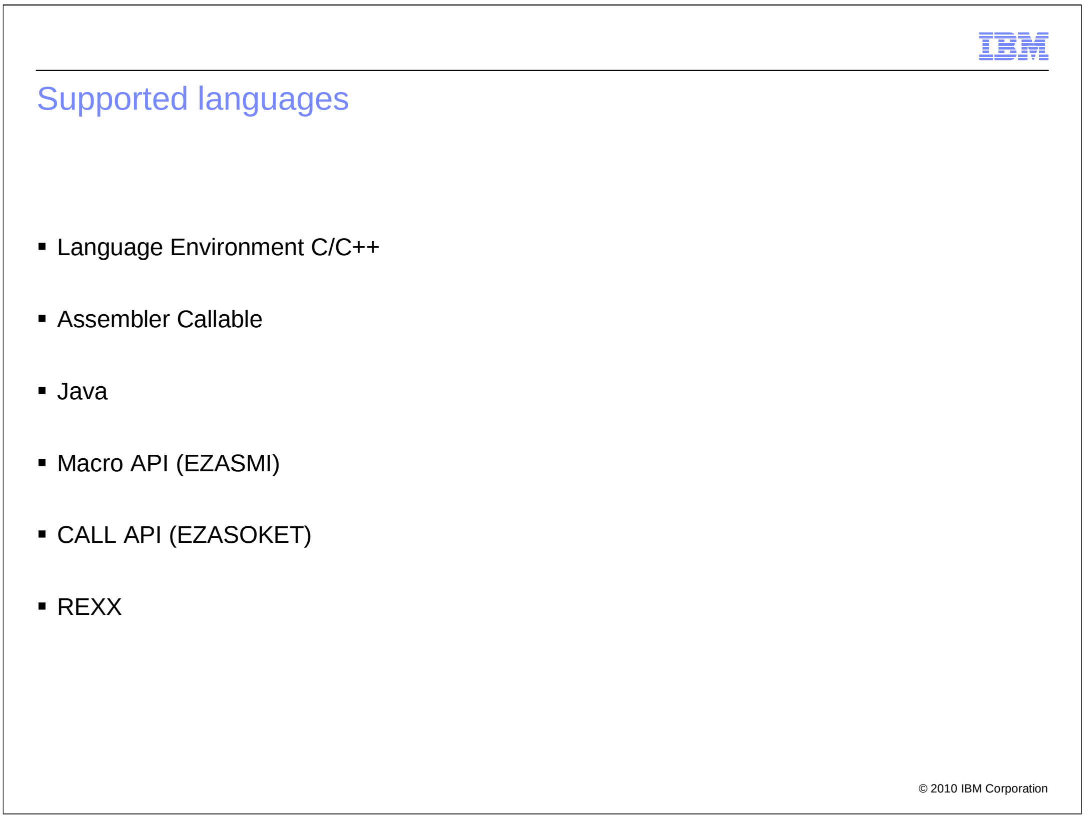
Server

socket()
bind()
IOCTL(… SIOCSPARTNERINFO)
listen()
accept()

w_IOCTL(… SIOCGPARTNERINFO)

Note: Since partner security credentials have been stored the application will not suspend when the SIOCGPARTNERINFO IOCTL command is issued

In this example, the server application issues the SIOCSPARTNERINFO IOCTL before the listen command. After the accept command is complete, an XCF message request is sent by the server stack (TCPCS2) to the client stack (TCPCS1) for the partner security credentials. The server stack will store the received information before the SIOCGPARTNERINFO IOCTL is issued. The server application then issues the SIOCGPARTNERINFO IOCTL for the partner security credentials and it is returned immediately without suspending.



Supported languages

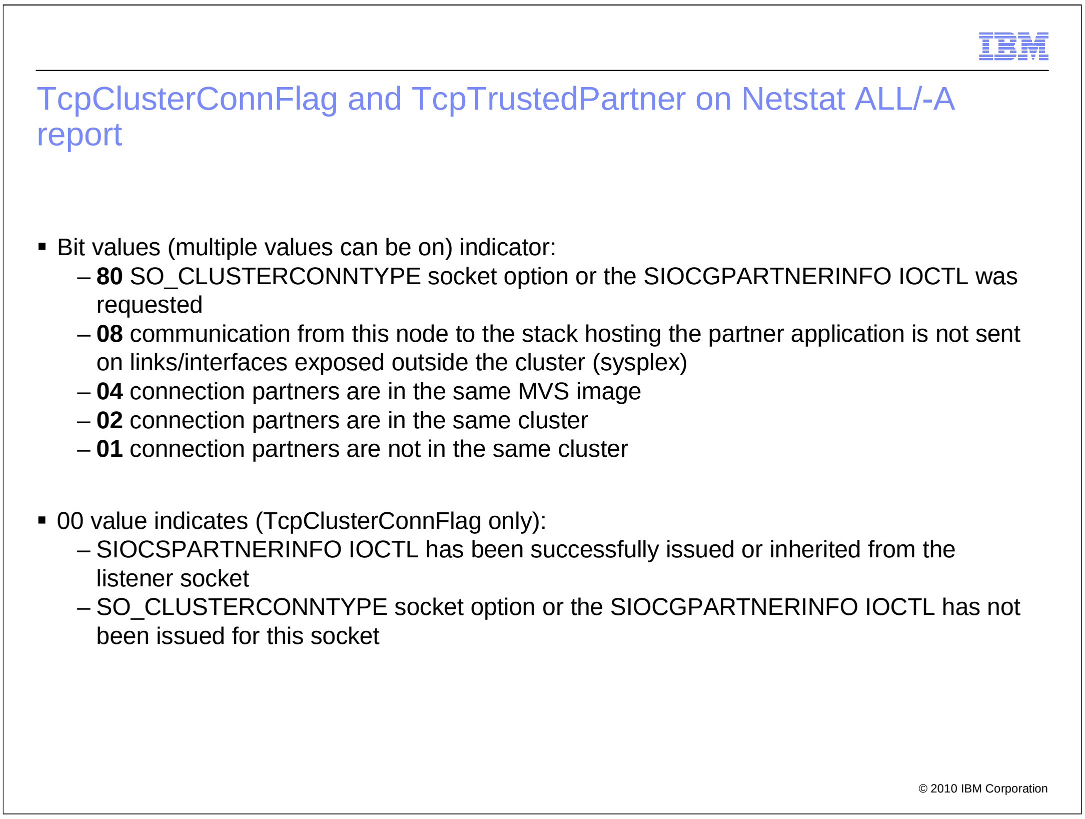
- Language Environment C/C++
- Assembler Callable
- Java
- Macro API (EZASMI)
- CALL API (EZASOKET)
- REXX

For the LE C/C++ API, the SIOC**S**PARTNERINFO uses the IOCTL command and the SIOCGPARTNERINFO uses the w_IOCTL command.

For the Assembler Callable API, the BPX1IOC function is used for 31 bit and BPX4IOC function is used for 64 bit.

The Java API uses the EZBTrustedPartner.jar for 31 bit support and the EZBTrustedPartner64.jar for 64 bit support. It uses the EZBTrustedPartnerdoc.jar for JavaDoc (classes and use). Install these jar files at /usr/include/java_classes

Other supported languages are the macro API (EZASMI), the CALL API (EZASOKET), and the REXX API.



TcpClusterConnFlag and TcpTrustedPartner on Netstat ALL/-A report

- Bit values (multiple values can be on) indicator:
  - **80** SO_CLUSTERCONNTYPE socket option or the SIOCGPARTNERINFO IOCTL was requested
  - **08** communication from this node to the stack hosting the partner application is not sent on links/interfaces exposed outside the cluster (sysplex)
  - **04** connection partners are in the same MVS image
  - **02** connection partners are in the same cluster
  - **01** connection partners are not in the same cluster

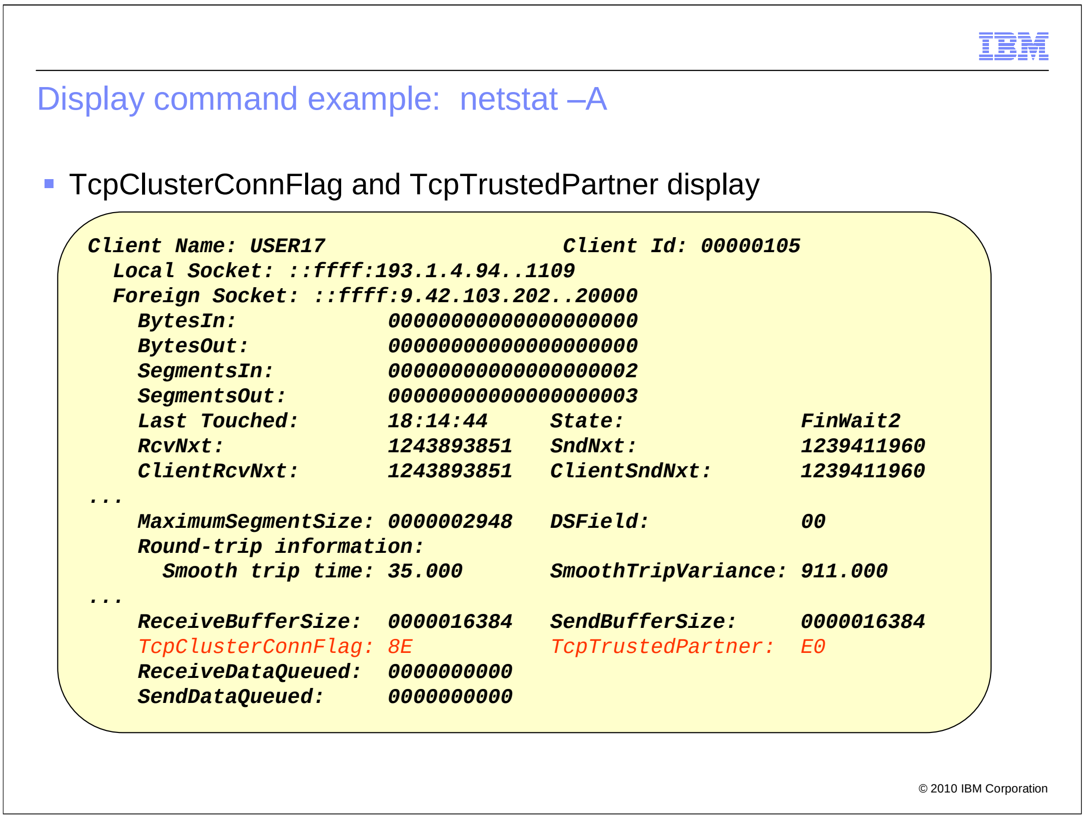
- 00 value indicates (TcpClusterConnFlag only):
  - SIOCSPARTNERINFO IOCTL has been successfully issued or inherited from the listener socket
  - SO_CLUSTERCONNTYPE socket option or the SIOCGPARTNERINFO IOCTL has not been issued for this socket

The Netstat ALL/-A report displays the TcpClusterConnFlag and the TcpTrustedPartner values.

The TCP cluster connection type field indicates that sysplex-specific connection routing information is set if an SO_CLUSTERCONNTYPE socket option or an SIOCGPARTNERINFO IOCTL has been issued for this connection. The TcpClusterConnFlag field returns the same information regardless of whether the application issued the SIOCGPARTNERINFO IOCTL or SO_CLUSTERCONNTYPE socket option.

For information about trusted TCP/IP connections and the SIOCGPARTNERINFO and SIOCSPARTNERINFO IOCTL calls, see *z/OS Communications Server: IP Programmer's Guide and Reference.*

For information on the Netstat command see *z/OS Communications Server: IP System Administrator Commands.*



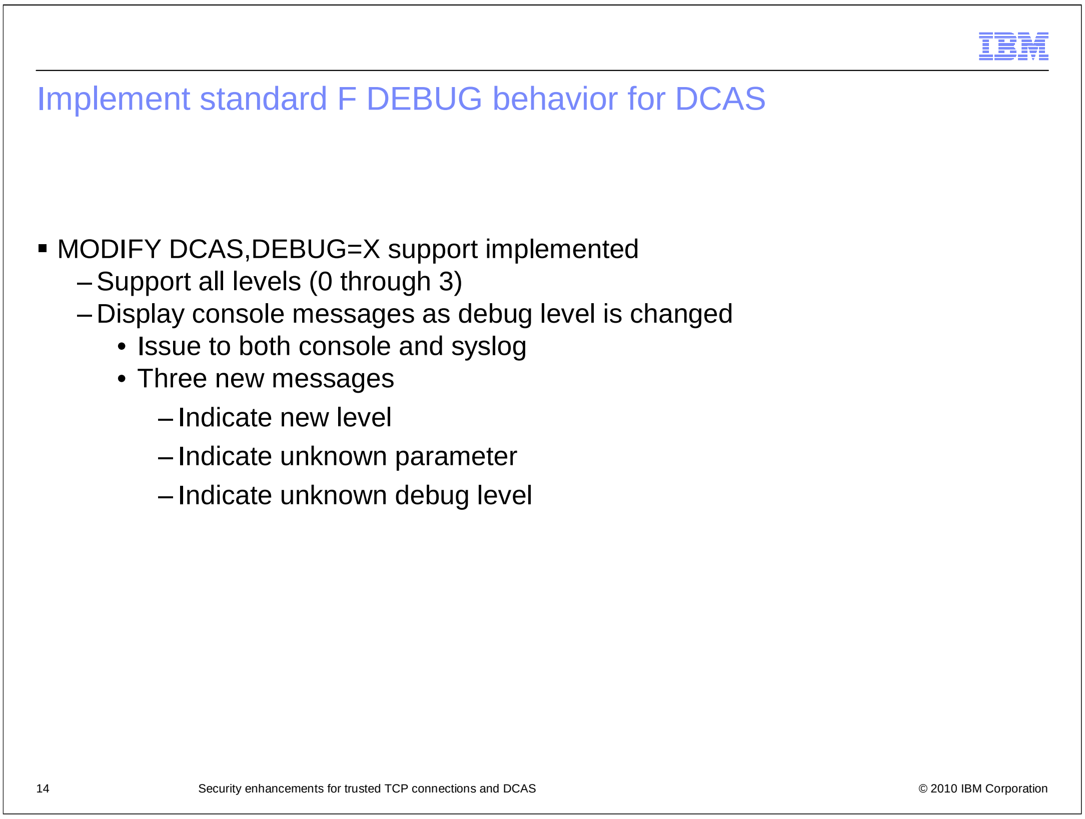
Display command example: netstat –A

- TcpClusterConnFlag and TcpTrustedPartner display

```
Client Name: USER17                   Client Id: 00000105
  Local Socket: ::ffff:193.1.4.94..1109
  Foreign Socket: ::ffff:9.42.103.202..20000
    BytesIn:            00000000000000000000
    BytesOut:           00000000000000000000
    SegmentsIn:         00000000000000000002
    SegmentsOut:        00000000000000000003
    Last Touched:       18:14:44     State:              FinWait2
    RcvNxt:             1243893851   SndNxt:             1239411960
    ClientRcvNxt:       1243893851   ClientSndNxt:       1239411960
...
    MaximumSegmentSize: 0000002948   DSField:            00
    Round-trip information:
      Smooth trip time: 35.000       SmoothTripVariance: 911.000
...
    ReceiveBufferSize:  0000016384   SendBufferSize:     0000016384
    TcpClusterConnFlag: 8E           TcpTrustedPartner:  E0
    ReceiveDataQueued:  0000000000
    SendDataQueued:     0000000000
```

For the TcpClusterConnFlag value, x’8E’, the ‘8’ indicates that the SO_CLUSTERCONNTYPE socket option or SIOCGPARTNERINFO has been issued. The ‘E’ is the routing information and can be broken down to the returned values of 08, 04 and 02. The 08 indicates that the communication from this node to the stack hosting the partner application is not sent on links or interfaces exposed outside the cluster (sysplex). The 04 indicates that the connection partners are in the same MVS image. The 02 indicates that they are in the same cluster.

For TcpTrustedPartner value, x’E0’, the ‘E’ can be broken down to the returned values of 08, 04 and 02. The 08 indicates the partner user ID has been retrieved. The 04 indicates the partner UTOKEN has been retrieved. The 02 indicates the SIOCSPARTNERINFO IOCTL has been successfully issued.

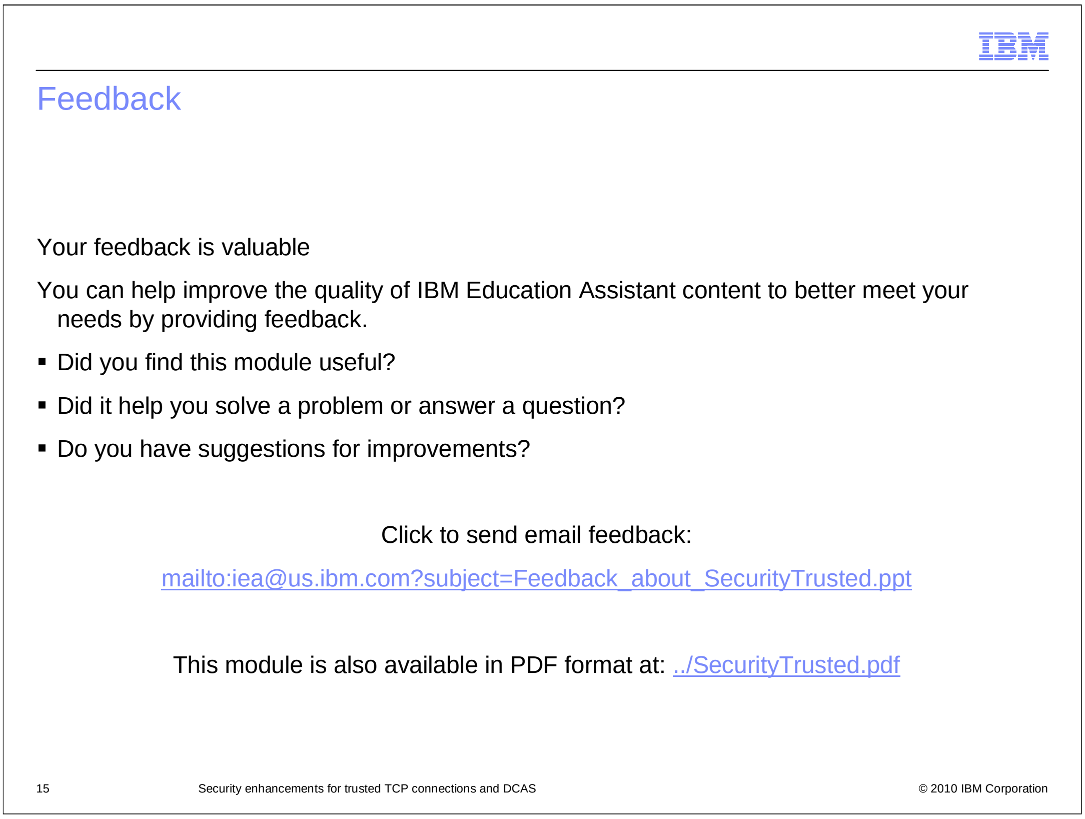


Implement standard F DEBUG behavior for DCAS

- MODIFY DCAS,DEBUG=X support implemented
  - Support all levels (0 through 3)
  - Display console messages as debug level is changed
    - Issue to both console and syslog
    - Three new messages
      - Indicate new level
      - Indicate unknown parameter
      - Indicate unknown debug level

14 Security enhancements for trusted TCP connections and DCAS © 2010 IBM Corporation

Before z/OS V1R12, DCAS needed to be restarted to change the debug level. In z/OS V1R12, DCAS supports using a modify command to change the debug level. The new debug level is displayed in console and syslog messages.



Feedback

Your feedback is valuable

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

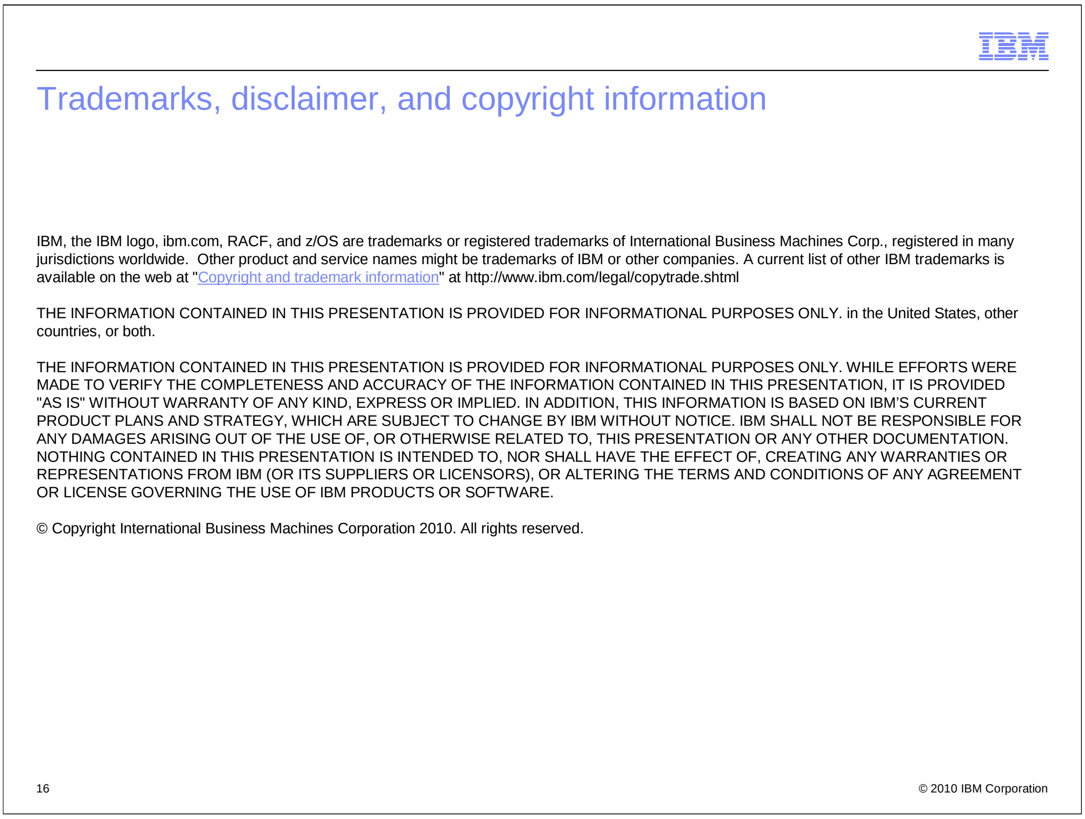
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

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

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

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



Trademarks, disclaimer, and copyright information

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. in the United States, other countries, or both.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.