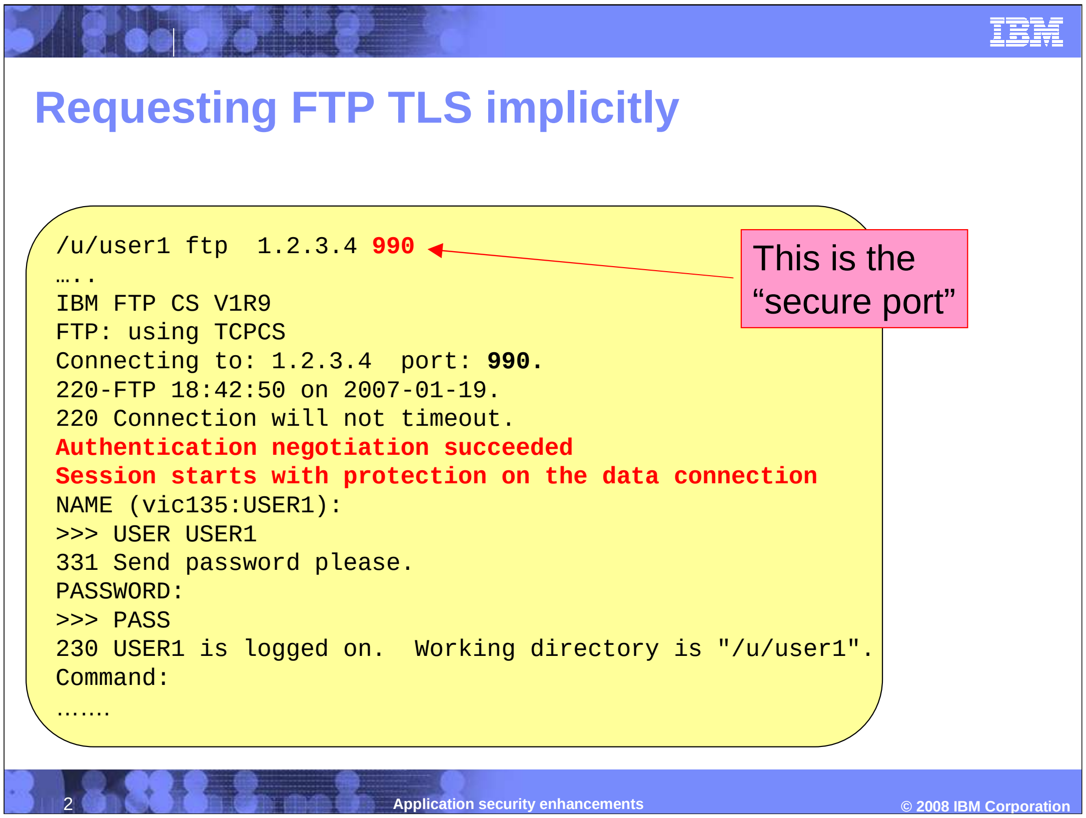
z/OS® V1R10 Communications Server

Application security enhancements

@business on demand software

This presentation covers application security enhancements.



Requesting FTP TLS implicitly

```
/u/user1 ftp  1.2.3.4 990
…..
IBM FTP CS V1R9
FTP: using TCPCS
Connecting to: 1.2.3.4  port: 990.
220-FTP 18:42:50 on 2007-01-19.
220 Connection will not timeout.
Authentication negotiation succeeded
Session starts with protection on the data connection
NAME (vic135:USER1):
>>> USER USER1
331 Send password please.
PASSWORD:
>>> PASS
230 USER1 is logged on.  Working directory is "/u/user1".
Command:
…….
```

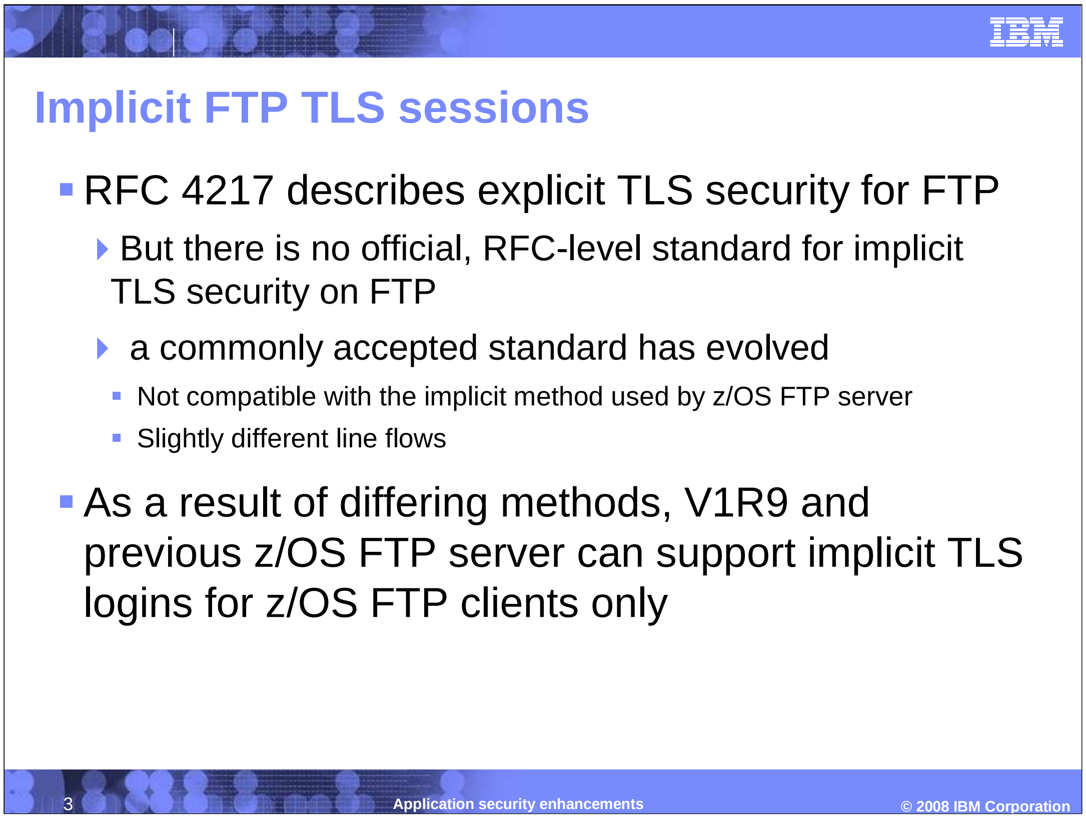
This is the “secure port”

z/OS FTP supports TLS security for FTP sessions, based on Revision 5 of the Internet Draft, *On Securing FTP with TLS.* This draft describes an explicit and an implicit method for securing an FTP session with TLS. This function is concerned only with **implicitly** secure TLS FTP sessions.

This example demonstrates how FTP establishes an **implicitly secure** connection. The Internet Draft, *On Securing FTP with TLS*, specified that connections to port 990 are assumed to be secure – no AUTH command is needed to secure the connection. In this example, the ftp client connects to port 990, the secure port.

The first line shows the ftp client being started with the host name parameter of 1.2.3.4, and the optional port parameter of 990.

Notice the client messages indicating the session is secure.



Implicit FTP TLS sessions

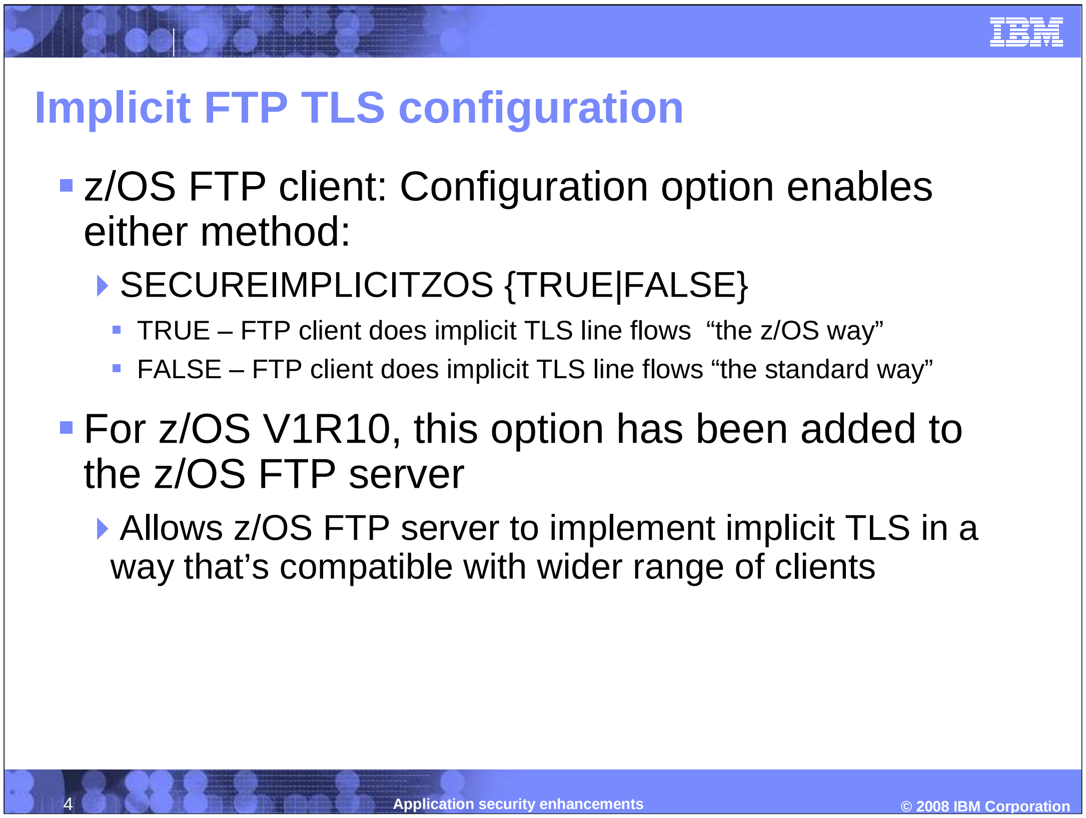
- RFC 4217 describes explicit TLS security for FTP
  - But there is no official, RFC-level standard for implicit TLS security on FTP
  - a commonly accepted standard has evolved
    - Not compatible with the implicit method used by z/OS FTP server
    - Slightly different line flows
- As a result of differing methods, V1R9 and previous z/OS FTP server can support implicit TLS logins for z/OS FTP clients only

3 Application security enhancements © 2008 IBM Corporation

When FTP clients log into a server’s TLSPORT, by convention they are TLS secured.

The z/OS FTP server implicit TLS support is compatible with z/OS FTP clients only.

The internet draft *On Securing FTP with TLS* became RFC 4217 as of October, 2005. RFC 4217 still describes explicitly securing the session with TLS, but has dropped the requirement to implicitly secure connections to FTP port 990. Nonetheless, FTP platforms started securing FTP sessions with TLS long before the Internet draft became an RFC, so some FTP platforms implement implicit TLS security. However since the method was never standardized, a commonly accepted standard has evolved that is not compatible with the z/OS implementation.



Implicit FTP TLS configuration

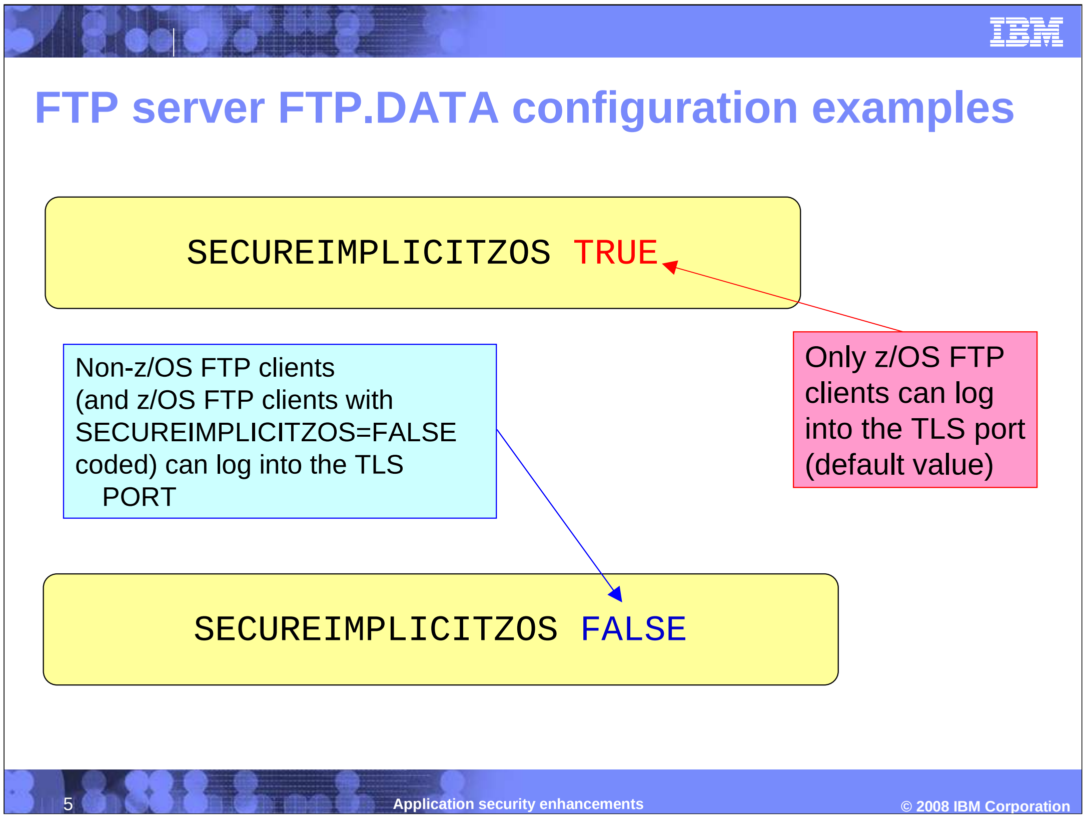
- z/OS FTP client: Configuration option enables either method:
  - SECUREIMPLICITZOS {TRUE|FALSE}
    - TRUE – FTP client does implicit TLS line flows “the z/OS way”
    - FALSE – FTP client does implicit TLS line flows “the standard way”
- For z/OS V1R10, this option has been added to the z/OS FTP server
  - Allows z/OS FTP server to implement implicit TLS in a way that’s compatible with wider range of clients

The V1R9 and earlier z/OS FTP client can log into an FTP server’s secure port using either method.   The SECUREIMPLICITZOS configuration option allows you to specify which method the client will use. You configure SECUREIMPLICITZOS TRUE when connecting to a z/OS FTP server’s secure port, and SECUREIMPLICITZOS FALSE when connecting to a non z/OS FTP server’s secure port.

You can code the SECUREIMPLICITZOS statement in FTP.DATA to set the value for the FTP client, and you can change the setting after the client is started with the locsite subcommand.

Although the client can be configured to use either method, no corresponding option existed for the FTP server. The z/OS FTP server always drove the security handshake after it sends the initial reply to the client (“the z/OS way”).

That is why only z/OS FTP clients can successfully log into the V1R9 z/OS FTP server TLSPORT. This has been fixed in V1R10 by a providing the same option for z/OS FTP server.



IBM

FTP server FTP.DATA configuration examples

```
SECUREIMPLICITZOS TRUE
```

Only z/OS FTP clients can log into the TLS port (default value)

Non-z/OS FTP clients (and z/OS FTP clients with SECUREIMPLICITZOS=FALSE coded) can log into the TLS PORT

```
SECUREIMPLICITZOS FALSE
```

5 Application security enhancements

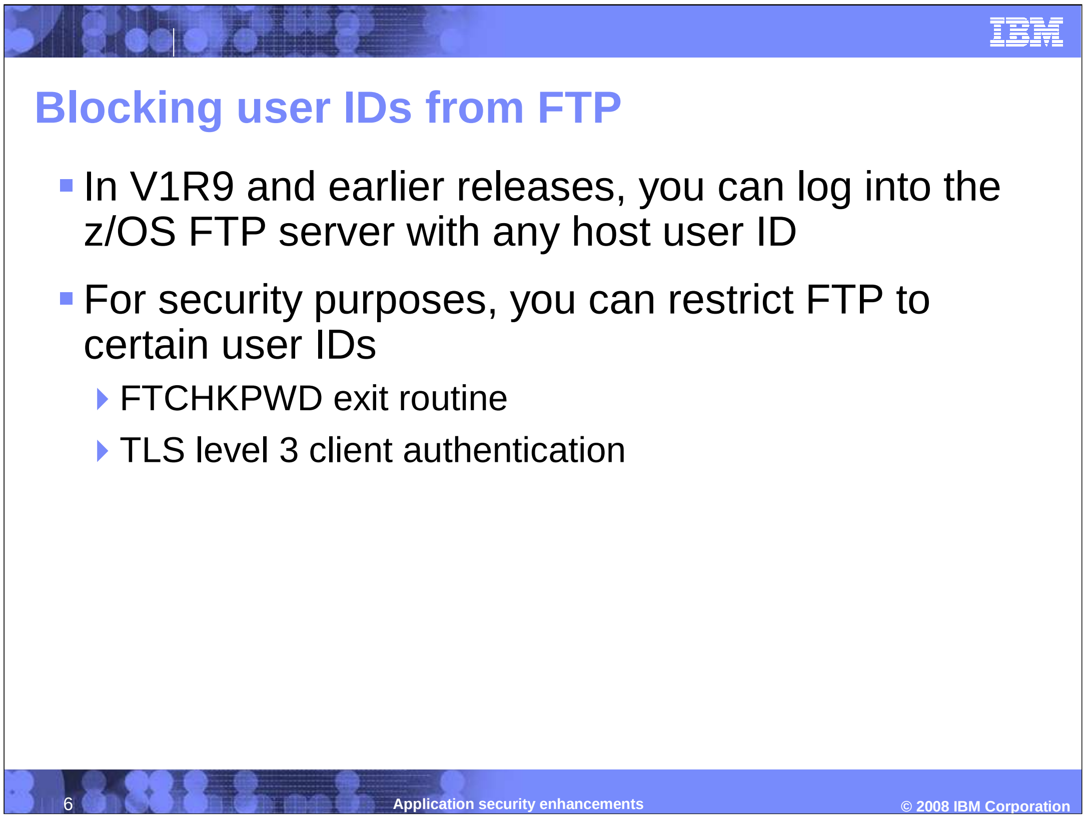
Here are some things to think about when configuring the server SECUREIMPLICITZOS value.

If you configure SECUREIMPLICITZOS TRUE for the z/OS FTP server, the server can support implicit TLS logins only from z/OS FTP clients. And even then, only if those clients have configured SECUREIMPLICITZOS TRUE (or are using SECUREIMPLICITZOS TRUE by default)

If you configure SECUREIMPLICITZOS FALSE for the FTP server, the server allows implicitly secured logins for non z/OS clients in addition to z/OS FTP clients that have configured SECUREIMPLICITZOS FALSE.

You will obtain the most flexibility at the server by configuring SECUREIMPLICITZOS FALSE.

The SECUREIMPLICITZOS setting applies whether you code TLSMECHANISM FTP or TLSMECHANISM ATTLS.



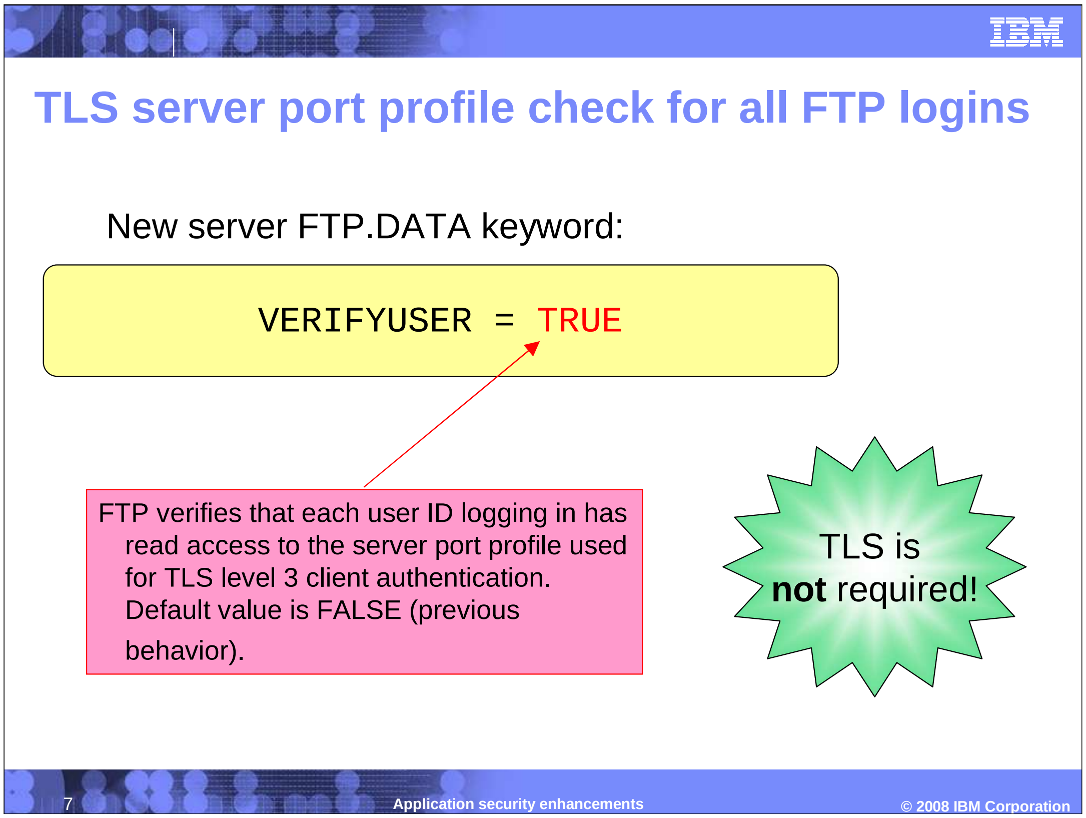
Blocking user IDs from FTP

- In V1R9 and earlier releases, you can log into the z/OS FTP server with any host user ID
- For security purposes, you can restrict FTP to certain user IDs
  - FTCHKPWD exit routine
  - TLS level 3 client authentication

By default, any user ID that is valid on the z/OS host can log into FTP. For security purposes, a customer might want to allow only certain user IDs to log into FTP on a certain host. z/OS FTP currently provides two ways to do this: you can code and install the FTCHKPWD exit routine, or you can configure TLS level 3 client authentication.

The FTCHKPWD exit routine is code written by you which is invoked by the FTP server as part of validating the user ID used to log into FTP. The sample FTCHKPWD in SEZAINST shows one method of using an exit routine to control which user IDs are allowed to log into the FTP server.

TLS level 3 client authentication adds a Security Access Facility (SAF) profile check to TLS level 2 client authentication. After configuring TLS level 2 client authentication, you can define a **server port profile** in the **SERVAUTH** class. You then grant READ access to those user IDs you want to allow to log into the FTP server. FTP will verify each user ID logging in with TLS has at least READ access to the profile.



TLS server port profile check for all FTP logins

New server FTP.DATA keyword:

```
VERIFYUSER = TRUE
```

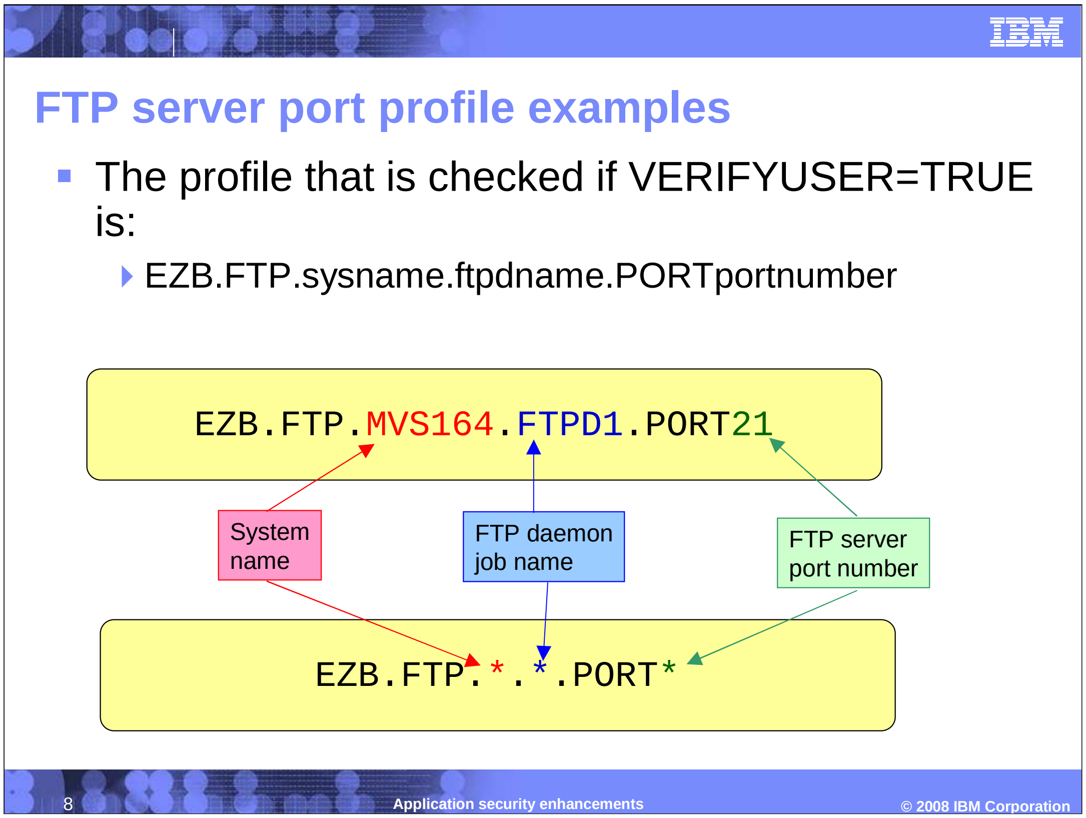
FTP verifies that each user ID logging in has read access to the server port profile used for TLS level 3 client authentication. Default value is FALSE (previous behavior).

TLS is **not** required!

7 Application security enhancements © 2008 IBM Corporation

In V1R10, z/OS FTP has provided the capability for you to restrict the ability to log into the FTP server only to those user IDs that have read access to the TLS server port profile. You do not need to be using TLS to exploit this additional check.

To implement this function on FTP server host, you first code VERIFYUSER TRUE in server FTP.DATA. You then define the SAF server port profile in class SERVAUTH and Permit user IDs to the profile. Grant READ access or greater to authorize. Finally, you activate the SERVAUTH class.



IBM

FTP server port profile examples

- The profile that is checked if VERIFYUSER=TRUE is:
  - EZB.FTP.sysname.ftpdname.PORTportnumber

```
EZB.FTP.MVS164.FTPD1.PORT21
```

System name

FTP daemon job name

FTP server port number

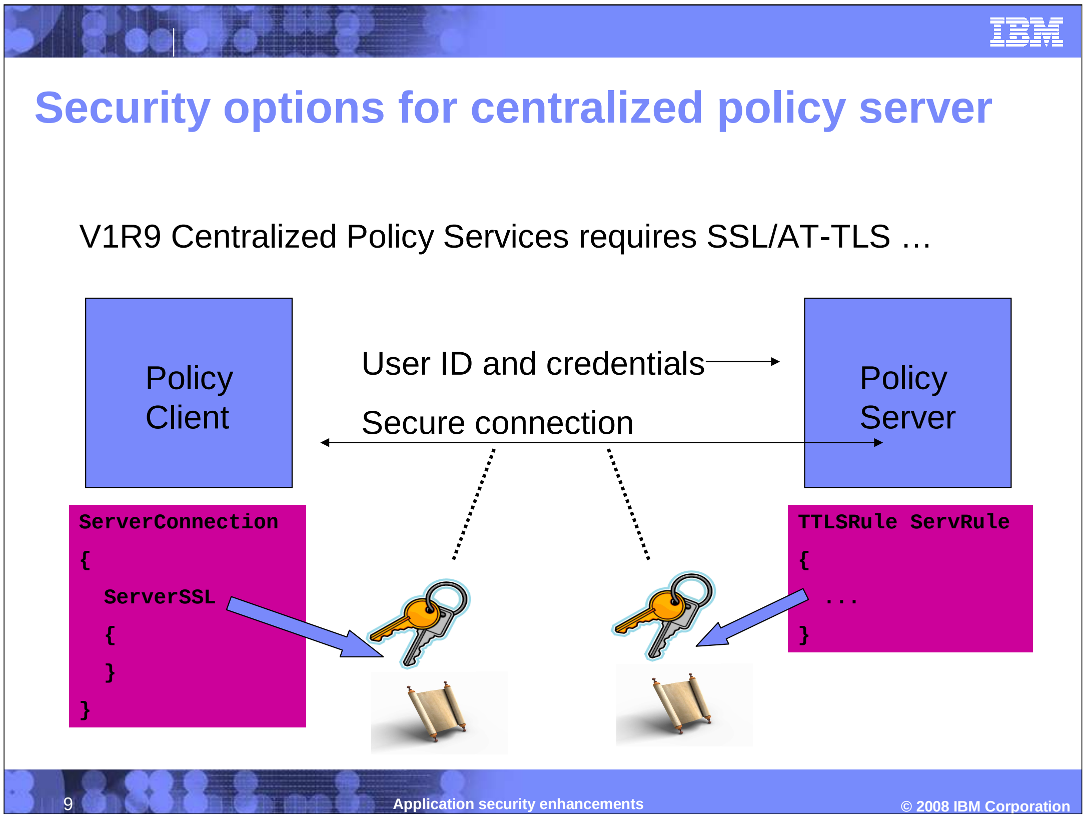
```
EZB.FTP.*.*.PORT*
```

8 Application security enhancements © 2008 IBM Corporation

As a reminder, the format of the FTP server port profile is shown here with two examples. Wildcards are allowed to the extent your security product allows wildcards.

In the first example, the system name is MVS164, and the FTP daemon job name is FTPD1. This is typical if you use the FTP sample start procedure, EZAFTPAP in SEZAINST, to start FTP). The FTP port number is 21. You can define this profile to control which user Ids can log into the FTP server listening on port 21.

In the second example, wildcards are used to protect all the FTP server ports with a single profile.



Security options for centralized policy server

V1R9 Centralized Policy Services requires SSL/AT-TLS …

Policy Client

User ID and credentials

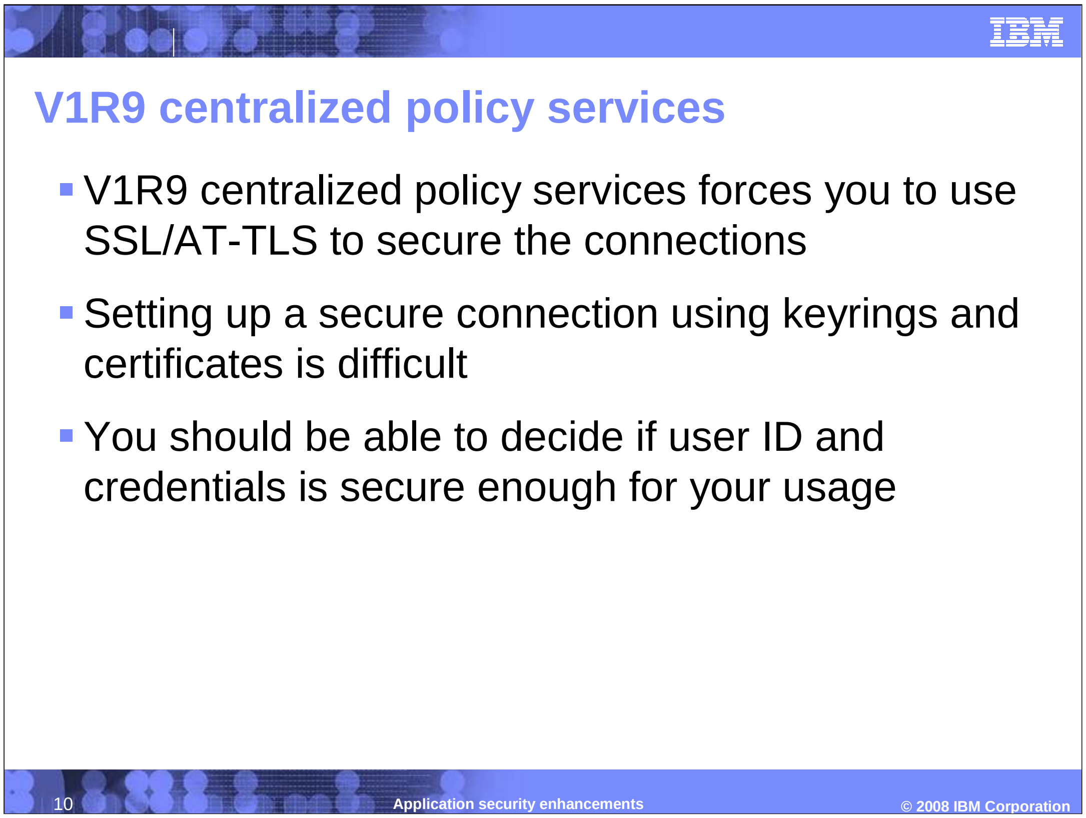
Secure connection

Policy Server

```
ServerConnection
{
  ServerSSL
  {
  }
}
```

```
TTLSRule ServRule
{
  ...
}
```

In V1R9 you are required to set up secure connections between policy clients and the policy server for the Centralized Policy Services function. You are also required to set up a user ID and credentials (password or PassTicket) for each policy client. The secure connections use SSL on the policy client and AT-TLS policy on the policy server to point to the appropriate keyrings containing the certificates used to establish secure connections.



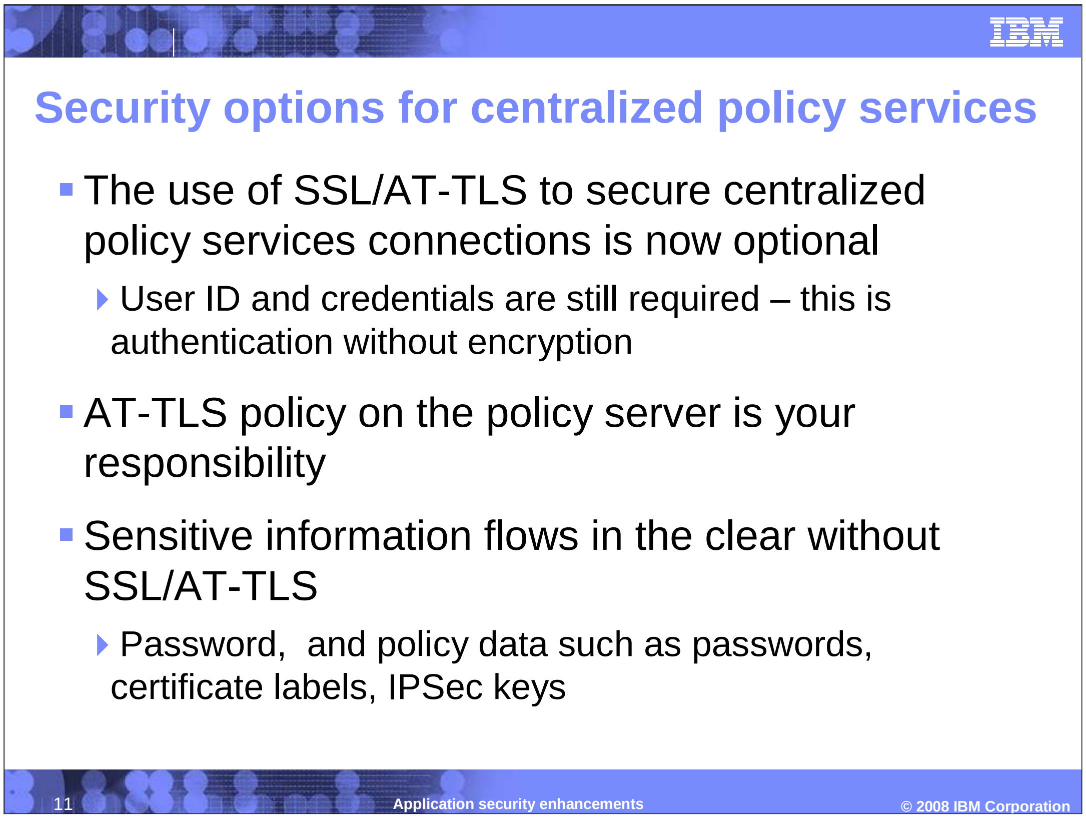
V1R9 centralized policy services

- V1R9 centralized policy services forces you to use SSL/AT-TLS to secure the connections
- Setting up a secure connection using keyrings and certificates is difficult
- You should be able to decide if user ID and credentials is secure enough for your usage

10 Application security enhancements © 2008 IBM Corporation

The requirement to use SSL/AT-TLS to set up secure connections can be seen as a roadblock to implementing centralized policy services. Setting up the required keyrings and certificates is difficult, and adds a substantial amount of work to the total configuration effort.

You should be allowed to decide for themselves if secure connections are required, or if security using user ID and credentials is good enough.



IBM

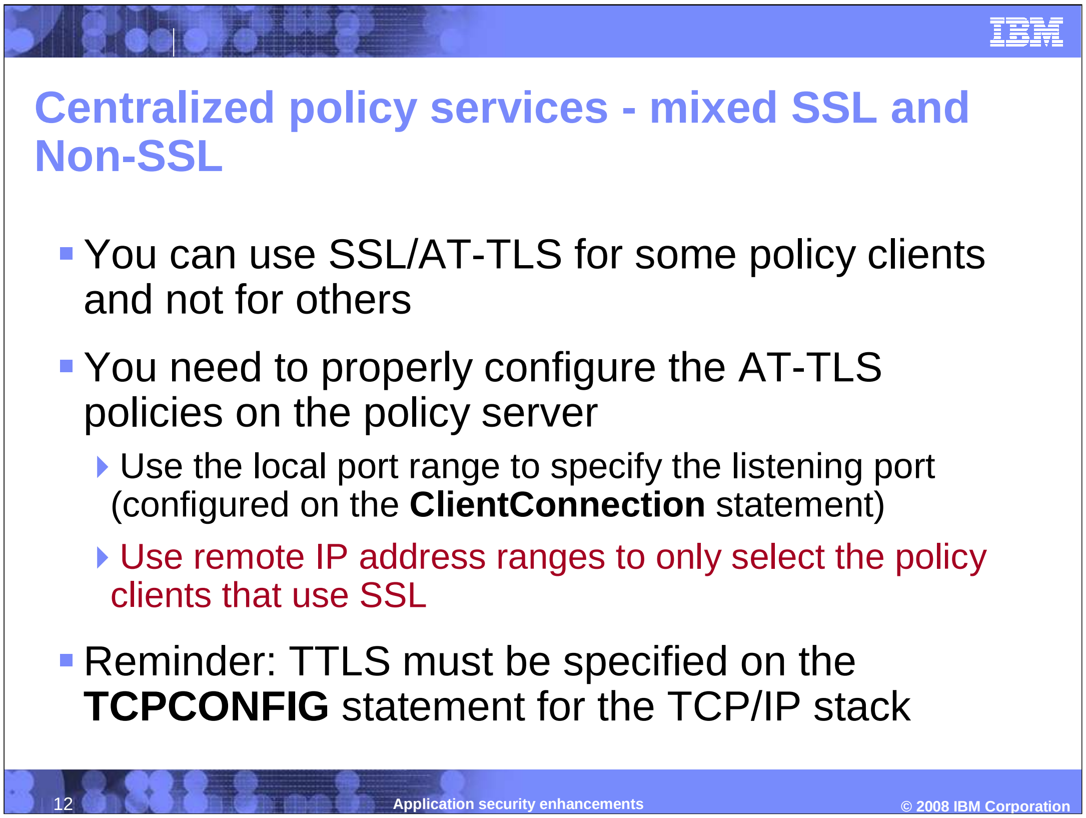
Security options for centralized policy services

- The use of SSL/AT-TLS to secure centralized policy services connections is now optional
  - User ID and credentials are still required – this is authentication without encryption
- AT-TLS policy on the policy server is your responsibility
- Sensitive information flows in the clear without SSL/AT-TLS
  - Password, and policy data such as passwords, certificate labels, IPSec keys

SSL configuration on the policy client is now optional. Note that a user ID and credentials are still required. The use of user ID and credentials without SSL provides authentication without encryption.

You are still responsible for setting up (or not) the AT-TLS policies on the policy server. If none of the policy clients are using SSL you don't need any AT-TLS policies to protect any policy client connections. If you are not using AT-TLS for anything else you also don't need to configure the TCP/IP stack for AT-TLS.

Be aware that sensitive information such as the password used to authenticate with the policy server, and any policy information retrieved from the policy server, flows in the clear without the use of SSL/AT-TLS.

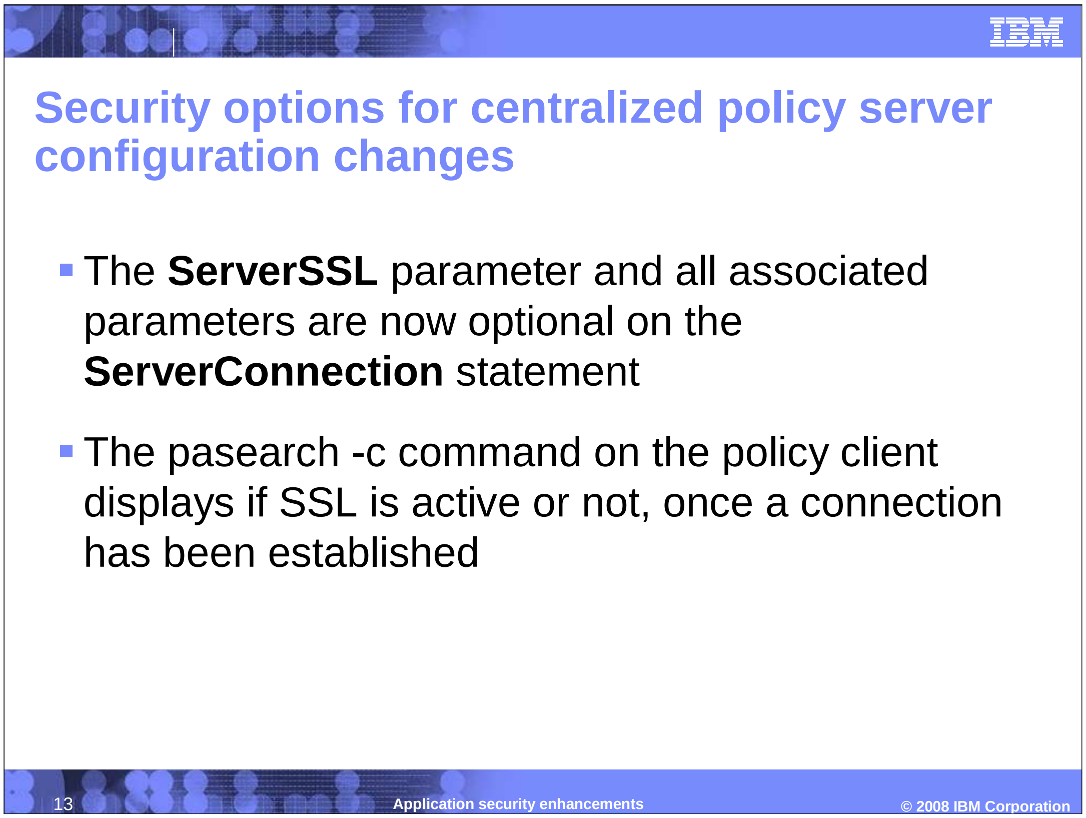


Centralized policy services - mixed SSL and Non-SSL

- You can use SSL/AT-TLS for some policy clients and not for others
- You need to properly configure the AT-TLS policies on the policy server
  - Use the local port range to specify the listening port (configured on the **ClientConnection** statement)
  - Use remote IP address ranges to only select the policy clients that use SSL
- Reminder: TTLS must be specified on the **TCPCONFIG** statement for the TCP/IP stack

You can choose to use SSL/AT-TLS for a subset of your policy clients, while not using it for the rest. In this case you must still define one or more AT-TLS policies on the policy server, but you have to ensure that the policies only select the policy clients that use SSL. One way to do this is to specify remote IP addresses, ranges, or groups on the AT-TLS rules to specify the set of policy clients using SSL.

If you are using any AT-TLS policies, the TCP/IP stack must be configured with the TTLS parameter on the TCPCONFIG statement.

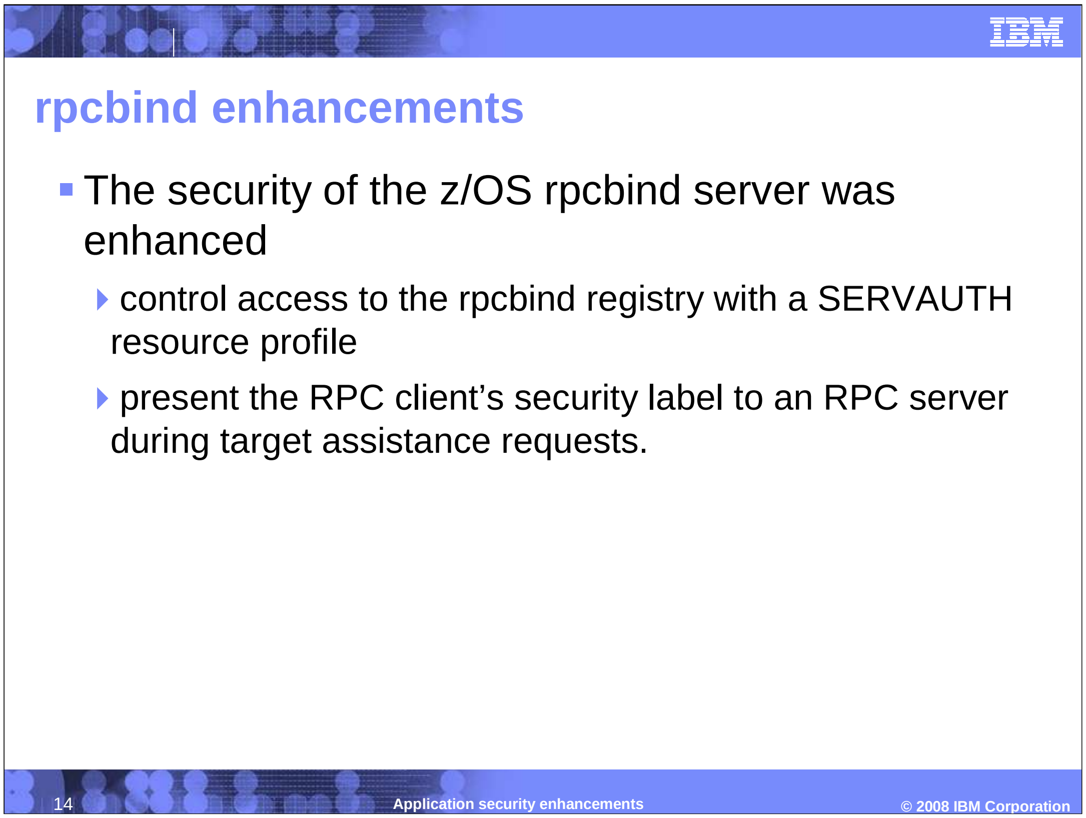


Security options for centralized policy server configuration changes

- The **ServerSSL** parameter and all associated parameters are now optional on the **ServerConnection** statement
- The pasearch -c command on the policy client displays if SSL is active or not, once a connection has been established

The **ServerConnection** statement is used on the policy client to configure parameters used to connect to the policy server. The **ServerSSL** parameter and all other associated SSL parameters are now optional.

Once a connection to the policy server is established, the pasearch -c command on the policy client shows whether SSL is active.



rpcbind enhancements

- The security of the z/OS rpcbind server was enhanced
  - control access to the rpcbind registry with a SERVAUTH resource profile
  - present the RPC client’s security label to an RPC server during target assistance requests.

14 Application security enhancements

The security of the z/OS rpcbind server was enhanced.

rpcbind now controls access to the rpcbind registry with a SERVAUTH resource profile. The profile to use is EZB.RPCBIND.sysname.rpcbindname.REGISTRY.

rpcbind will also now present the RPC client’s security label to an RPC server during target assistance requests. In V1R9, rpcbind presents its own security label on requests from clients, so servers can not trust the security label on an rpcbind request. By passing through the client’s security label, V1R10 rpcbind requests can be trusted.

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:

z/OS

A current list of other IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

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.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.