z/OS Communications Server

IBM Health Checker for SMTP, SNMP, and RSHD

Version 2 Release 1

Note:
Links to related publications are from original documents and might not work. The links to publications are included for reference purposes only.

Contents

Tables

Chapter 1. New Function Summary

V2R1 new function summary

IBM Health Checker for z/OS SMTPD MAIL RELAY

z/OS V2R1 Communications Server, with TCP/IP APAR PI51640 and SNA APAR OA50122, provides a new IBM Health Checker for z/OS application health check to help determine whether your SMTP server is configured as a mail relay. Specifying the INBOUNDOPENLIMIT statement to a valid non-zero value or allowing it to default to the value of 256 causes the SMTP server to open a listening port and implicitly become exploitable by remote users as a mail relay.

Dependency: You must install TCP/IP APAR PI51640 and SNA APAR OA50122 and start the IBM Health Checker for z/OS to use the new application health check.

Using the IBM Health Checker for z/OS SMTPD MAIL RELAY

To use the IBM Health Checker for z/OS SMTPD MAIL RELAY, perform the appropriate tasks in Table 1.

Table 1. IBM Health Checker for z/OS SMTPD MAIL RELAY

Task	Reference
To use the IBM Health Checker for z/OS application check support, take the following steps: 1. Configure and start the IBM Health Checker for z/OS. 2. Review the CSAPP_SMTPD_MAIL_RELAY health check output for potential migration actions.	See the following topics in IBM Health Checker for z/OS: User's Guide: • Setting up IBM Health Checker for z/OS • Working with check output • Managing checks

IBM Health Checker for z/OS SNMP agent public community name

z/OS V2R1 Communications Server, with TCP/IP APAR PI51640 and SNA APAR OA50122, provides a new IBM Health Checker for z/OS application health check to help determine whether your SNMP agent is configured with a community name of public. Because the SNMP community name of public is a well-known name, it should not be used with community-based security due to security considerations.

Dependency: You must install TCP/IP APAR PI51640 and SNA APAR OA50122 and start the IBM Health Checker for z/OS to use the new application health check.

Using the IBM Health Checker for z/OS SNMP agent public community name

To use the IBM Health Checker for z/OS SNMP agent public community name, perform the appropriate tasks in Table 2 on page 2.

Table 2. IBM Health Checker for z/OS SNMP agent public community name

Task	Reference
To use the IBM Health Checker for z/OS application check support, take the following steps: 1. Configure and start the IBM Health Checker for z/OS. 2. Review the CSAPP_SNMPAGENT_PUBLIC_COMMUNITY health check output for potential migration actions.	See the following topics in IBM Health Checker for z/OS: User's Guide: • Setting up IBM Health Checker for z/OS • Working with check output • Managing checks

IBM Health Checker for z/OS MVRSHD RHOSTS DATA

z/OS V2R1 Communications Server, with TCP/IP APAR PI51640 and SNA APAR OA50122, provides a new IBM Health Checker for z/OS application health check to help determine whether your MVRSHD server is active and whether RSH clients are using RHOSTS.DATA datasets for authentication. The MVRSHD server supports the RSH and REXEC protocols which transfer user ID and password information in the clear. There is also the potential of weak authentication for RSH clients using RHOSTS.DATA datasets. This authentication method allows remote command execution without requiring the RSH client to supply a password.

Dependency: You must install TCP/IP APAR PI51640 and SNA APAR OA50122 and start the IBM Health Checker for z/OS to use the new application health check.

Using the IBM Health Checker for z/OS MVRSHD RHOSTS DATA

To use the IBM Health Checker for z/OS MVRSHD RHOSTS DATA, perform the appropriate tasks in Table 3.

Table 3. IBM Health Checker for z/OS MVRSHD RHOSTS DATA

Task	Reference
To use the IBM Health Checker for z/OS application check support, take the following steps: 1. Configure and start the IBM Health Checker for z/OS. 2. Review the CSAPP_MVRSHD_RHOSTS_DATA health check output for potential migration actions.	See the following topics in IBM Health Checker for z/OS: User's Guide: • Setting up IBM Health Checker for z/OS • Working with check output • Managing checks

Chapter 2. IP Configuration Guide

Simple Network Management Protocol

This topic describes how to configure:

- Simple Network Management Protocol (SNMP) agent (osnmpd)
- z/OS® UNIX **snmp** or **osnmp** command
- NetView® SNMP command
- SNMP subagents
- Open Systems Adapter (OSA) support
- Trap forwarder daemon

Before you configure, read Understanding search orders of configuration information. It covers important information about data set naming and search sequences.

Provide community name information

SNMP agents are accessed by remote network management stations and by SNMP subagents. To allow network management stations to send inquiries to the SNMP agent, and SNMP subagents to connect to the SNMP agent, you can provide PW.SRC information that defines a list of community names and IP addresses that can use these community names. The community name operates as a password when accessing objects on, or connecting to, a destination SNMP agent. The subagents pass the community name to the agent on the connect request.

Guideline: Because the community name of public is a well-known name, it should not be configured to the agent due to security considerations. IBM Health Checker for z/OS can be used to check whether the community name of public has been configured to the SNMP agent. For more details about IBM Health Checker, see IBM Health Checker for z/OS: User's Guide.

All of the z/OS Communications Server SNMP subagents connect to the agent using the IPv4 primary interface IP address of the stack with which the subagent is associated, and a community name. As long as SOURCEVIPA is not in effect on the IPCONFIG profile statement, this IP address is the source IP address that the agent uses, along with the community name, to verify the subagent's authority to connect to the SNMP agent. The IPv4 primary interface IP address is either the first IP address in the HOME list or the IP address specified on a PRIMARYINTERFACE TCP/IP profile statement. If SOURCEVIPA is in effect, the IP address used by the agent to verify the subagent's authority is the virtual IP address associated with the IPv4 primary interface IP address. For information on determining which virtual IPv4 address is associated with a physical IPv4 address, see the HOME statement in z/OS Communications Server: IP Configuration Reference. Check the Netstat HOME/-h output to verify the IPv4 primary interface address of the stack.

The PW.SRC information is optional. If no PW.SRC information is found and no community name is specified for the **-c** parameter at agent invocation, then the SNMP agent will accept requests with a community name of 'public' from any IP address. If a PW.SRC file exists, but is empty, and if no community name is specified on the **-c** parameter at the agent invocation, then no requests will be accepted by the agent.

Note: Verify that there is no SNMPD.CONF file because this file can be used only with SNMPv3. If an SNMPD.CONF file is found, the PW.SRC file will not be used.

If creating a data set, you can specify a sequential data set with the following attributes: RECFM=FB, LRECL=80, and BLKSZ=3120. Other data set attributes might also work, depending on your installation parameters.

Provide trap destination information

Traps are unsolicited messages that are sent by an SNMP agent to an SNMP network management station. An SNMP trap contains information about a significant network event. The management application running at the management station interprets the trap information sent by the SNMP agent.

Note: When the SNMP agent starts, it retrieves an IP address for itself. If it retrieves an IPv6 colon-hexadecimal address, when it sends traps the source IP address in each trap will be 0.0.0.0.

For a detailed description of the SNMP trap types provided by z/OS CS, see z/OS Communications Server: IP System Administrator's Commands.

The SNMP agent Distributed Protocol Interface allows subagents other than those shipped with z/OS Communications Server (which might be running on another host) to generate SNMP traps. This can allow for support of other types of traps. For more information about SNMP DPI, see the z/OS Communications Server: IP Programmer's Guide and Reference.

To use traps, you must provide SNMPTRAP.DEST information defining a list of managers to which traps are sent. The SNMPTRAP.DEST information is optional. If no trap destination file is found, then the SNMP agent sends traps to the IP address of the SNMP agent and issues a warning message indicating that defaults are in effect. If a trap destination file exists, but is empty, no traps are sent.

Guideline: If you use SNMPTRAP.DEST to configure trap information, the agent uses the hardcoded community name of public in the outbound traps. Because the community name of public is a well-known name, it should not be used in SNMP traps due to security considerations. To configure specific community names for trap destinations, you must convert your SNMPTRAP.DEST information to a SNMPD.CONF configuration file format. For more information about how to accomplish this conversion, see Migrating the PW.SRC file and SNMPTRAP.DEST file to the SNMPD.CONF file in z/OS Communications Server: IP Configuration Reference.

Note: Verify that there is no SNMPD.CONF file. If an SNMPD.CONF file is found, the SNMPTRAP.DEST file will not be used.

If creating a data set, you can specify a sequential data set with the following attributes: RECFM=FB, LRECL=80, and BLKSZ=3120. Other data set attributes might also work, depending on your installation parameters.

Provide community-based and user-based security and notification destination information

If you want to use user-based security, either concurrently with or instead of community-based security, you must configure security definitions and notification destinations. To allow SNMP subagents to connect to the SNMP agent using user-based security, you must configure community-based security definitions. The

SNMP subagents pass the community name to the agent on the connect request. The community name operates as a password when the SNMP subagents connect to the SNMP agent.

Guideline: Because the community name of public is a well-known name, it should not be configured to the agent due to security considerations. IBM Health Checker for z/OS can be used to check whether the community name of public has been configured to the SNMP agent. For more details about IBM Health Checker, see IBM Health Checker for z/OS: User's Guide.

All of the z/OS Communications Server SNMP subagents connect to the agent using the IPv4 primary interface IP address of the stack with which the subagent is associated, and a community name. As long as SOURCEVIPA is not in effect on the IPCONFIG profile statement, this IP address is the source IP address that the agent uses, along with the community name, to verify the subagent's authority to connect to the SNMP agent. The IPv4 primary interface IP address is either the first IP address in the HOME list or the IP address specified on a PRIMARYINTERFACE TCP/IP profile statement. If SOURCEVIPA is in effect, the IP address used by the agent to verify the subagent's authority is the virtual IP address associated with the IPv4 primary interface IP address. For information on determining which virtual IPv4 address is associated with a physical IPv4 address, see the HOME statement in z/OS Communications Server: IP Configuration Reference. Check the Netstat HOME/-h output to verify the IPv4 primary interface address of the stack.

SNMPv3 provides the ability to configure the agent dynamically, from either a local or remote host, and to make changes in the configuration while the SNMP agent is running. Doing SNMP agent configuration dynamically requires a good understanding of how the SNMP SET commands can be issued to create new rows or to change or delete existing rows, as well as familiarity with the SNMP engine configuration tables defined in RFCs 3584 and 3411 through 3415. For information about accessing RFCs, see Related protocol specifications.

As an alternative to dynamically configuring the SNMP agent, z/OS Communications Server supports a configuration file to be read at agent initialization called the SNMPD.CONF file. Dynamic configuration changes made with SNMP SET commands to the SNMP agent configuration entries will be written out to the SNMPD.CONF file, so they will continue to be in effect even after the SNMP agent is restarted.

Configuring the TSO Remote Execution server

Procedure

Perform the following steps to configure the TSO Remote Execution server:

1. Update AUTOLOG and PORT statements in the PROFILE.TCPIP data set.
2. Determine whether the Remote Execution client will send a Remote Execution (REXEC) command or Remote Shell (RSH) command.
3. Permit remote users to access MVS™ resources. (Required only if the client is not sending a password.)
4. Update the Remote Execution cataloged procedure.
5. Create a user exit routine (optional).
6. Permit access to JESSPOOL files.

Step 3: Permit remote users to access MVS resources (optional)

This step is necessary only if your installation allows users to issue remote execution commands without the requirement of specifying a password on the remote execution client.

Procedure

Use the following steps to ensure that the server can correctly access necessary MVS resources. You can use z/OS Security Server (RACF®) or an equivalent security program.

1. Verify that your system has been configured for allowing surrogate job submission as described in z/OS Security Server RACF Security Administrator's Guide (SC28-1915) or by using an equivalent security program.
2. Authorize the TSO Remote Execution server to submit jobs for the MVS user ID specified with the **-l** option of the RSH command. This can be done with the RACF facility as described in z/OS Security Server RACF Security Administrator's Guide (SC28-1915), or by using an equivalent security program.
3. Define an *mvs_userid*.RHOSTS.DATA data set and authorize the TSO Remote Execution server user ID permission to read this data set. This can be done with the RACF facility as described in z/OS Security Server RACF Security Administrator's Guide (SC28-1915), or by using an equivalent security program.

   Note: This is the user ID used to start the RXSERVE address space.

   This data set identifies the Remote Execution clients that can execute MVS commands remotely by sending an RSH command.

   When a Remote Execution client sends an RSH request to the TSO Remote Execution server, the request includes the local user ID of the client user (*local_userid*) and, if the client user specified the **-l** option of the RSH command, the request also contains the user ID to use on the remote host (*mvs_userid*). If the client does not specify the **-l** option, the user ID to be used on the remote host is assumed to be the same as the *local_userid*.

   When the TSO Remote Execution server receives an RSH command without a password, the server looks for a data set called *mvs_userid*.RHOSTS.DATA. The *mvs_userid*.RHOST.DATA data set contains one or more entries. Each entry consists of two parts, a fully qualified name of the client user's host and a *local_userid* associated with that host. The *local_userid* is case sensitive. If the data set exists, the server reads it and looks for an entry with a host name that matches the client user's host. If the user ID specified on this entry in the RHOSTS.DATA data set matches the *local_userid* passed on the RSH command, the RSH command continues processing. If the entry does not exist, the server responds to the client with message `EZA4386E Permission denied`.

   Tip: If the client connected to this host through a link-local address, the client's host name generated by the resolver can be in the format *hostname%scope*. Adding *%scope* information to the appropriate RHOSTS.DATA client host definitions results in a more efficient search for a matching client host name. For details on the support for including scope information on configured host names, see z/OS Communications Server: IPv6 Network and Application Design Guide.

   In the following example of an RHOSTS.DATA data set, the MVS client user `mvsuser` is allowed to issue the RSH command without a password from host `rs60007` with a local AIX® user ID of `aixuser`.

Example of mvsuser.RHOSTS.DATA data set:

```
rs60007.itso.ral.ibm.com aixuser
```

Guideline: It is suggested not enabling the MVRSHD server. The MVRSHD server supports the RSH and REXEC protocols which transfer user ID and password information in the clear. There is also the potential of weak authentication for RSH clients using RHOSTS.DATA datasets. This authentication method allows remote command execution without requiring the RSH client to supply a password. IBM Health Checker for z/OS can be used to check whether the MVRSHD server is active and detect an RSH client attempting to use an RHOSTS.DATA dataset for authentication. For more details about IBM Health Checker, see IBM Health Checker for z/OS: User's Guide.

4. Users may be authenticated using Kerberos or GSS. If the username in the Kerberos or GSS credentials matches the local user ID (*local_userid*) of the client supplied by the RSH client, then no password is required.

Chapter 3. IP Configuration Reference

OSNMPD parameters

The SNMP agent (OSNMPD) runs in a separate address space that executes load module EZASNMPD. OSNMPD can be started without parameters or you can add any of the parameters in this topic.

When starting OSNMPD from MVS, add the parameters to the `PARMS=` keyword on the EXEC statement of the OSNMPD cataloged procedure. When starting OSNMPD from z/OS UNIX System Services, specify the desired parameters on the osnmpd command.

Rule: The parameters must be entered in lowercase because they are case sensitive.

Parameter
Description

-A Forces the SNMP Agent to obtain an IPv4 address for itself when it initializes. If no IPv4 addresses are available, then the IPv4 loopback address (127.0.0.1) is used. If this parameter is not specified, the SNMP Agent uses an IP address that might be IPv6 or IPv4.

-a Specifies that the packets sent by the SNMP Agent for responses and notifications should be sent using the physical interface address, rather than a VIPA address (if SOURCEVIPA is configured).

-C *class*

Permits you to control SNMP subagent connections to the SNMP agent via a z/OS UNIX connection. For z/OS UNIX connections, a z/OS UNIX path name is used. This path name can be specified on the agent's -s parameter. See the description of the -s parameter in this topic for more information. All of the z/OS Communications Server SNMP subagents use a z/OS UNIX connection to connect to the agent.

In order for subagents to successfully connect to the agent, either the subagents must be defined with superuser authority or the path name's read and write file access permission bits must be set for the class associated with the subagent's user ID. For more detailed information about file access permission bits, see the information about handling security for your files in z/OS UNIX System Services User's Guide.

This parameter's *class* value specifies the class or classes of the subagent user IDs, for those subagents that you want to permit to connect to the SNMP agent using a z/OS UNIX connection. This parameter causes the path name's read and write file access permission bits to be set for the specified classes.

The valid values for the *class* variable are 1 - 4 and are defined as follows:

1 Group class
Specify this value if you want only those subagents whose user IDs are associated with the same security product group as the agent to be able to connect. The resulting file access permission bit value in octal is 660.

2 Other class
Specify this value if you want only those subagents whose user IDs

are not associated with the same security product group as the agent to be able to connect. The resulting file access permission bit value in octal is 606.

3 Both Group and Other class
Specify this value if you want all subagents to be able to connect to the agent. The resulting file access permission bit value in octal is 666.

4 Only User class
Specify this value if you do not want any subagents to be able to connect to the agent using a z/OS UNIX connection.

If the -C parameter is not specified, default level 1 is used. This means that the group read and write permission bits for the path name are set and that subagents whose user IDs are associated with the same security product group as the SNMP agent are able to connect.

-c *community*

This parameter can be used to dynamically configure a community name to the agent instead of defining it in the SNMP agent configuration file. A community name is a password that can accompany an SNMP request that the agent receives. Use community names with community-based security to restrict access to SNMP management data. See Configure the SNMP agent and Configure the SNMP agent (OSNMPD) in z/OS Communications Server: IP Configuration Guide for more information about community-based security.

The value that you specify for *community* is configured as both an ASCII and an EBCDIC community name. This parameter should be specified only if you are using community-based security with the SNMP agent, and you want to dynamically define a community name that any requestor can use to retrieve SNMP management data. Specifying a community name on this parameter when starting the agent causes the community to be defined to the agent with a mask and an IP address of zeros. Therefore, any request received with this *community* value would be authenticated (for example, the request would be accepted from any IP address). If the specified community name is also defined in the SNMP agent configuration file, the definition for this community name in the configuration file is overridden by specifying the community name on the -c parameter. This parameter is case sensitive.

The default value for this parameter is the community name *public*, but this default community name is dynamically defined to the agent only if no agent configuration file is found.

Guideline: Because the community name of public is a well-known name, it should not be configured to the agent due to security considerations. IBM Health Checker for z/OS can be used to check whether the community name of public has been configured to the SNMP agent. For more details about IBM Health Checker, see IBM Health Checker for z/OS: User's Guide.

-d *level*

Specifies the level of tracing to be started. The valid values for *level* are 0–255. If the -d parameter is not specified, then the default level of 0 is used, meaning no tracing is done. If the -d parameter is specified without a *level*, then a level of 31 is used, meaning all SNMP requests/responses/traps and DPI activity is traced.

There are eight levels of tracing provided. Each level selected has a corresponding number. The sum of the numbers associated with each level of tracing selected is the value which should be specified as *level*. If the agent is started, tracing options can be dynamically changed using the MVS MODIFY command. For more information about agent tracing, see the z/OS Communications Server: IP Diagnosis Guide.

The numbers for the trace levels are:

0 No tracing (default)

1 Trace SNMP requests

2 Trace SNMP responses

4 Trace SNMP traps

8 Trace DPI packets

16 Trace DPI internals (currently, no specific traces are recorded for this trace level)

32 Agent internal trace

64 Agent internal trace plus extended storage dump traces

128 Agent internal trace plus extended storage dump traces and additional information

-i *interval*

Specifies the interval (in minutes) at which dynamic configuration changes to the SNMP agent should be written out to the SNMPD.CONF configuration file. Valid values are 0–10. The default value is 5. This parameter is only relevant when the SNMPD.CONF file is used for SNMPv3 configuration.

Guideline: Configuration updates made dynamically (by SNMP SET requests) cause the SNMPD.CONF file to be overwritten by the SNMP agent.

-p *port* Listens for SNMP packets on this port. The default is port 161. If you change the port to something other than 161, you must also configure any subagents and managers, such as osnmp, to use the new port.

-s *socketname*

Specifies the path name of the z/OS UNIX file to be used in accepting requests from subagents that communicate with the agent by way of z/OS UNIX connections. This value can be configured either by specifying it on the `-s` parameter or by specifying it as the value of the dpiPathNameForUnixStream MIB object in OSNMPD.DATA. The default is /var/dpi_socket. All of the z/OS Communications Server SNMP subagents use a z/OS UNIX connection to connect to the agent.

The SNMP agent creates this path name every time it initializes. In order for subagents to successfully connect to the agent using this path name, either the subagents must be defined with superuser authority or, the file access permission bits for this path name must be set to read and write.

- If the subagent's user ID is associated with the same security product group as the SNMP agent, the Group read and write permission bits must be set.
- If the subagent's user ID is not associated with the same security product group as the SNMP agent, the Other read and write permission bits must be set.

You can use the agent's -C parameter to ensure that the agent sets the appropriate permission bits when the agent creates the path name.

For more detailed information about file access permission bits, see information about handling security for your files in z/OS UNIX System Services User's Guide. If you need to change the path name's file access permission bits after the agent has initialized, you can use the z/OS UNIX chmod command. For more information about the chmod command, see z/OS UNIX System Services Command Reference.

? | -help
Displays the usage statement for the command. If this parameter is specified, all other parameters are ignored. If OSNMPD was started from MVS, the usage information is written to syslogd. If OSNMPD was started from z/OS UNIX System Services, the usage information is displayed to the invoker of the command.

PW.SRC statement syntax

The PW.SRC statements specify community names and hosts that can use each community name. The format of a statement is:

```
community_name  desired_network  snmp_mask
```

The *community_name* can be up to 32 characters in length. This value can contain both uppercase and lowercase letters; however, it is case sensitive. In any requests received by the SNMP agent, the *community_name* must match the *community_name* specified in PW.SRC exactly, including the correct case.

Guideline: Because the community name of public is a well-known name, it should not be configured to the agent due to security considerations. IBM Health Checker for z/OS can be used to check whether the community name of public has been configured to the SNMP agent. For more details about IBM Health Checker, see IBM Health Checker for z/OS: User's Guide.

The *desired_network* is the IPv4 address in dotted decimal notation or IPv6 address in colon hexadecimal notation representing the range of addresses for which this *community_name* can be used. The *desired_network* must be specified; there is no default value.

If *desired_network* is an IPv6 address, then *snmp_mask* is either an IPv6 address mask in colon hexadecimal notation or an integer from 0 to 128 specifying the number of IPv6 address prefix bits used to construct an IPv6 address mask. The IP address mask is logically ANDed with the origin address of the incoming SNMP message.

If *desired_network* is an IPv4 address, then *snmp_mask* is either an IPv4 address mask (for example, 255.255.255.0) or an integer from 0 to 32 specifying the number of IPv4 address prefix bits used to construct an IPv4 address mask. The IP address mask is logically ANDed with the origin address of the incoming SNMP message.

Restriction: Scope information cannot be specified as part of the *desired_network* value.

If the value resulting from this logical ANDing equals the value of *desired_network*, the incoming message is accepted. *snmp_mask* must be specified; there is no default value.

- All parameters for each community must be on the same statement.
- Sequence numbers are not allowed on the statements.
- Comments begin with either an asterisk (*) or a # character.

Guidelines:

- If an error is detected in processing an entry and no appropriate default value can be assumed, the entry is discarded and an error message is written to the syslog daemon.

SNMPTRAP.DEST statement syntax

The SNMPTRAP.DEST statements list managers who are to receive the traps, and the protocol used to send traps. The format of a statement is:

```
manager UDP
```

The *manager* is the host to which the trap is to be sent. This can be a host name, or it can be the IP address of the host in IPv4 dotted decimal or IPv6 colon hexadecimal notation. If a host name is specified, the value can contain both uppercase and lowercase letters and it is not case sensitive. The protocol must be UDP. There should be one entry in the data set for each host to which you want to send traps.

- All parameters for each host must be on the same statement.
- Sequence numbers are not allowed on the statements.
- Comments begin with an asterisk (*) or a # character.

Restriction: Scope information cannot be specified as part of the *manager* value.

Guidelines:

- If an error is detected in processing an entry and no appropriate default value can be assumed, the entry is discarded and an error message is written to the syslog daemon.
- If you use SNMPTRAP.DEST to configure trap information, the agent uses the hardcoded community name of public in the outbound traps. Because the community name of public is a well-known name, it should not be used in SNMP traps due to security considerations. To configure specific community names for trap destinations, you must convert your SNMPTRAP.DEST information to a SNMPD.CONF configuration file format. For more information about how to accomplish this conversion, see Migrating the PW.SRC file and SNMPTRAP.DEST file to the SNMPD.CONF file.

COMMUNITY entry

```
COMMUNITY communityName securityName securityLevel netAddr netMask storageType
```

Field definitions

communityName

The community name for community-based security (SNMPv1 or SNMPv2c). The *netAddr* must be specified; there is no default value. It must be a 1 - 32 character string.

Guideline: Because the community name of public is a well-known name, it should not be configured to the agent due to security considerations. IBM Health Checker for z/OS can be used to check whether the

community name of public has been configured to the SNMP agent. For more details about IBM Health Checker, see IBM Health Checker for z/OS: User's Guide.

securityName
The *securityName* defined for this communityName. The *securityName* is the more generic term for the principal (user or community) for which other entries, such as VACM_GROUP and TARGET_PARAMETERS, are defined. The community name must match the *securityName* exactly. The *securityName* is 1 - 32 characters. A dash (-) indicates the default value; a *securityName* equal to the specified communityName.

securityLevel
Indicates the security level to be applied when processing incoming or outgoing messages with this community name. Valid values are noAuthNoPriv or none to indicate no authentication or privacy protocols are applied, or dash (-) to indicate the default value of noAuthNoPriv. Encryption is not supported on SNMPv1 and SNMPv2c messages.

netAddr
The host IP address or network address, in IPv4 dotted decimal or IPv6 colon hexadecimal notation, representing the range of addresses for which this community name can be used. When an SNMP request is received, the *netMask* value is logically ANDed with the origin address of the request. The resulting value must equal the value of *netAddr* for the SNMP request to be permitted to the community name. The netAddr must be specified; there is no default value.

netMask
The IP address mask or IP address prefix defined for this communityName. If *netAddr* is an IPv6 colon hexadecimal address, then *netMask* is either an IPv6 colon hexadecimal address mask (for example, FFFF:FFFF::) or an integer from 0 to 128 specifying the number of IPv6 address prefix bits used to construct an IPv6 address mask. If *netAddr* is an IPv4 dotted decimal address, then *netMask* is either an IPv4 address mask (for example, 255.255.255.0) or an integer from 0 to 32 specifying the number of IPv4 address prefix bits used to construct an IPv4 address mask. The mask is logically ANDed with the origin address of the incoming SNMP message. If the resulting value equals the value of *netAddr*, the incoming message is accepted. *netMask* must be specified; there is no default value.

storageType
Indicates the type of storage in which this definition is to be maintained. Storage types are defined in RFC 1903. Note that the value of volatile is not supported in the SNMPD.CONF file. Valid values are:

nonVolatile
Indicates the entry definition persists across reboots of the SNMP agent; it can, however, be changed or even deleted by dynamic configuration requests.

permanent
Indicates the entry definition persists across reboots of the SNMP agent; it can be changed but not deleted by dynamic configuration requests.

readonly
Indicates the entry definition persists across reboots of the SNMP agent; it cannot be changed or deleted by dynamic configuration requests.

dash (-)
Indicates the default value of nonVolatile.

SNMP_COMMUNITY entry

```
SNMP_COMMUNITY communityIndex communityName securityName contextEngineID
contextName transportTag storageType
```

Field definitions

communityIndex
Indicates a locally unique identifier for this SNMP_Community definition. Valid values are 1 - 32 characters in length. There is no default value.

communityName
The community name for community-based security (SNMPv1 or SNMPv2c) . Valid values are 1 - 32 characters in length. There is no default value. The agent assumes that community names are encoded in ASCII. If an EBCDIC-encoded community name is needed, it must be specified as a UTF-8 name. See the Usage Note topic for an explanation about how to configure a UTF-8 name.

Guideline: Because the community name of public is a well-known name, it should not be configured to the agent due to security considerations. IBM Health Checker for z/OS can be used to check whether the community name of public has been configured to the SNMP agent. For more details about IBM Health Checker, see IBM Health Checker for z/OS: User's Guide.

securityName
The securityName defined for this *communityName*. The *securityName* is the more generic term for the principal (user or community) for which other entries, such as VACM_GROUP and TARGET_PARAMETERS, are defined. Valid values are 1 - 32 characters in length. There is no default value.

contextEngineID
Indicates the location of the context in which information is accessed. A dash (-) indicates the default value of the local SNMP agent's engine ID. Only the default value is supported.

contextName
The corresponding context value. Valid values are 1-32 characters in length. Only a dash (-) indicating the default, which is an empty string, is supported.

transportTag
Indicates a tag value to be compared with the values in the tagLists defined in the snmpTargetAddrTable (either on TARGET_ADDRESS entries or by way of dynamic configuration). Those target addresses (whose tag value match this tag) identify the transport endpoints from which a request containing this community are accepted. Valid values are 1 - 255 characters. No delimiters are allowed. A dash (-) indicates the default, which is no tag value.

storageType
Indicates the type of storage in which this definition is to be maintained. Valid values are:

nonVolatile
Indicates the entry definition persists across reboots of the SNMP agent; it can, however, be changed or even deleted by dynamic configuration requests.

permanent
Indicates the entry definition persists across reboots of the SNMP agent; it can be changed but not deleted by dynamic configuration requests.

readonly
Indicates the entry definition persists across reboots of the SNMP agent; it cannot be changed or deleted by dynamic configuration requests.

dash (-)
Indicates the default value of nonVolatile.

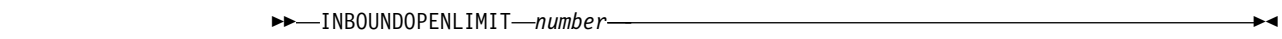
INBOUNDOPENLIMIT statement

Use the INBOUNDOPENLIMIT statement to specify the maximum number of simultaneous TCP connections over which SMTP server receives mail. This number can be in the range 2 - 256 connections. The default, if this statement is not valid, is 256.

Guidelines:

- This statement is optional. If it is not coded, the maximum number of TCP connections used by SMTP is limited to 256 (because it uses the PASCAL API).
- Specifying the INBOUNDOPENLIMIT statement to a valid non-zero value or allowing it to default to the value of 256 causes the SMTP server to open a listening port and implicitly become exploitable by remote users as a mail relay. IBM Health Checker for z/OS can be used to check whether the SMTP server is configured as a mail relay. For more details about IBM Health Checker, see IBM Health Checker for z/OS: User's Guide.

Syntax



Parameters

number
A value in the range 2 - 256 can be coded to reflect the maximum number of simultaneous TCP connections used by the SMTP server for inbound mail.

Restriction:A value of 0 can be used and is a special case. If 0 is coded, then the SMTP server does not open a listening connection. Also, if 0 is coded, you cannot use AUTOLOG to monitor and restart the SMTP started procedure, because there is no listening connection to monitor. If this number is coded incorrectly, the default value of 256 is used.

Examples

Use the following code to set the maximum number of simultaneous TCP connections that are used by the SMTP server to 10:

```
INBOUNDOPENLIMIT   10
```

Usage notes

If 0 is coded, you cannot use AUTOLOG to monitor and restart SMTP.

Related topics

OUTBOUNDOPENLIMIT statement

Remote execution server parameters

The system parameters required by the Remote Execution server are passed by the PARM operand of the EXEC statement in the Remote Execution cataloged procedure. Update the following parameters as required by your installation:

EX= or EXIT=
Name of a user exit routine to inspect and alter JOB and EXEC parameters prior to submission of TSO batch jobs initiated by remote commands.

IPV6=
Y or N, indicating whether the server should attempt communication over an IPv6 network. If this option is not specified, the server attempts IPv6 communication. Specifying N for this option prevents IPv6-only clients from communicating with this server.

Tip: This option is useful for installations that have not migrated user exits to accommodate IPv6 addresses.

PRE= or PREFIX=
A four-character value used as the first four characters in the job name of jobs that are submitted. The remaining characters of the job name is a sequential number in the range of 1 - 9999.

PUR= or PURGE=
Y or N, indicating whether a job submitted by the server should be purged immediately after execution or held in the output queue.

TSO= or TSOPROC=
The name of the TSO batch procedure. The default is IKJACCNT. The name IKJACCNT can be modified in the exit routine specified with the EXIT parameter.

MSG= or MSGCLASS=
The MSGCLASS parameter for TSO batch jobs submitted to execute remote commands. The default is H, which points to a HELD output class.

Restrictions:

- This parameter must not be altered by the exit routine.
- For JES3 users, the HELD output class needs to be defined as a HELD output class for external writer.

TSC= or TSCLASS=
The SYSOUT class for the SYSTSPRT DD statement for submitted jobs. The default is A.

Restriction: For JES3 users, the HELD output class needs to be defined as a HELD output class for external writer.

MAX= or MAXCONN=
The maximum number of open sockets at any one time. Usually, each client requires 2 sockets while the command is being processed and the output is being returned. The minimum acceptable value is 512. This is also the default.

TR= or TRACE=
The trace options that are to be in effect for the Remote Execution server.

Rule: If more than one trace parameter is specified, enclose the parameters in parentheses.

LOG
Specifies trace records written to SYSPRINT.

NOL= or NOLOG
Specifies no trace records written to SYSPRINT.

SEN= or SEND
Specifies trace records sent to the client.

NOS= or NOSEND
Specifies no trace records sent to the client.

CLI= or CLIENT=*client*
Specifies a specific client host for which trace records are to be produced.

ALLC= or ALLCLIENTS
Specifies that trace records are to be produced for all clients.

RE= or RESET
Sets the trace options to NOLOG, NOSEND, ALLCLIENTS.

SL= or SECLABEL=
Y or N, indicating whether the server should attempt to add a security label to the job card. If this option is not specified, the server attempts to add a security label to the job card. If Y is specified for this option, the server adds a security label (if one exists) to the job card following the message class parameter. For more information about the multilevel security environment and configuring z/OS Communications Server in that environment, see the multilevel security information in the z/OS Communications Server: IP Configuration Guide.

Use the MODIFY command to dynamically change all but the following parameters:

- MAXCONN
- PREFIX
- IPv6
- SECLABEL

Tip: All parameters can now be abbreviated. For example, EXIT can be abbreviated to EX.

Guideline: It is suggested not enabling the MVRSHD server. The MVRSHD server supports the RSH and REXEC protocols which transfer user ID and password information in the clear. There is also the potential of weak authentication for RSH clients using RHOSTS.DATA datasets. This authentication method allows remote command execution without requiring the RSH client to supply a password. IBM

Health Checker for z/OS can be used to check whether the MVRSHD server is active and detect an RSH client attempting to use an RHOSTS.DATA dataset for authentication. For more details about IBM Health Checker, see IBM Health Checker for z/OS: User's Guide.

Chapter 4. IP Diagnosis Guide

IBM Health Checker for z/OS

IBM® Health Checker for z/OS is a z/OS component that installations can use to gather information about their system environment and system parameters to help identify potential configuration problems before they impact availability or cause outages. Individual products, z/OS components, or ISV software can provide checks that take advantage of the IBM Health Checker for z/OS framework.

For more information about IBM Health Checker for z/OS, see IBM Health Checker for z/OS: User's Guide.

z/OS Communications Server TCP/IP provides the following checks:

CSAPP_FTPD_ANONYMOUS_JES
Checks whether the following statements have been configured for an FTP server:

- ANONYMOUS
- ANONYMOUSLEVEL 3
- ANONYMOUSFILETYPEJES FALSE

When ANONYMOUS FTP is allowed on the FTP server, it is recommended that the value specified for ANONYMOUSLEVEL be set to 3 and that the value specified for ANONYMOUSFILETYPEJES be set to FALSE. Otherwise, anonymous users can submit jobs to run on the system.

CSAPP_MVRSHD_RHOSTS_DATA
Checks whether the MVRSHD server is active and if an RSH client has been detected using RHOSTS.DATA datasets for authentication. The MVRSHD server supports the RSH and REXEC protocols which transfer user ID and password information in the clear. There is also the potential of weak authentication for RSH clients that use RHOSTS.DATA datasets. This authentication method allows remote command execution without requiring the RSH client to supply a password.

CSAPP_SMTPD_MAIL_RELAY
Checks whether the INBOUNDOPENLIMIT statement has been set to 0 in the SMTP configuration file. The SMTP server is implicitly exploitable by remote users as a mail relay when the INBOUNDOPENLIMIT statement is explicitly configured with a value other than 0 or is allowed to default to the value of 256.

CSAPP_SNMPAGENT_PUBLIC_COMMUNITY
Checks whether the SNMP agent has been configured with a community name of public. The community name of public is a well-known name and should not be used with community-based security due to security considerations.

CSRES_AUTOQ_GLOBALTCPIPDATA
Checks whether the AUTOQUIESCE operand has been specified on the UNRESPONSIVETHRESHOLD resolver setup statement and that the GLOBALTCPIPDATA resolver setup statement has not been specified in the resolver setup file.

CSRES_AUTOQ_RESOLVEVIA
Checks whether the RESOLVEVIA statement has been specified with the value TCP in the global TCPIP.DATA file when the autonomic quiescing of unresponsive name servers function is active.

CSRES_AUTOQ_TIMEOUT
Checks whether the configured resolver timeout value in the global TCPIP.DATA file exceeds the optimal setting when the autonomic quiescing of unresponsive name servers function is active. By default, this check is performed once when the resolver is initialized and whenever a MODIFY REFRESH command is issued. This default value can be overridden on either a POLICY statement in the HZSPRM*xx* parmlib member or on a MODIFY command.

CSTCP_CINET_PORTRNG_RSV_*TCPIPstackname*
Checks whether the port range specified by INADDRANYPORT and INADDRANYCOUNT in the BPXPRMxx parmlib member is reserved for OMVS on this stack, when operating in a CINET environment. A port range is reserved on a TCP/IP stack using the PORTRANGE TCP/IP profile statement. By default, this check is performed once at stack initialization. This default can be overridden on either a POLICY statement in the HZSPRMxx parmlib member or on a MODIFY command. The check name is suffixed by *TCPIPstackname*, which is the job name of each TCP/IP stack that is started, to define a separate check for each stack.

CSTCP_IPMAXRT4_*TCPIPstackname*
Checks whether the total number of IPv4 indirect routes in the TCP/IP stack routing table has exceeded the maximum threshold. When this threshold is exceeded, OMPROUTE and the TCP/IP stack can potentially experience high CPU consumption from routing changes. A large routing table is considered to be inefficient in network design and operation. By default, this check is performed at the following times:

- Whenever the total number of indirect routes exceeds the maximum threshold (default 2000)
- 30 minutes after stack initialization (provided that the maximum threshold has not been exceeded)
- Specified interval (default 168 hours for weekly)

The defaults for the maximum threshold and interval can be overridden on either a POLICY statement in the HZSPRMxx parmlib member or on a MODIFY command. The check name is suffixed by *TCPIPstackname*, which is the job name of each TCP/IP stack that is started, to define a separate check for each stack.

CSTCP_IPMAXRT6_*TCPIPstackname*
Checks whether the total number of IPv6 indirect routes in the TCP/IP stack routing table has exceeded the maximum threshold. When this threshold is exceeded, OMPROUTE and the TCP/IP stack can potentially experience high CPU consumption from routing changes. A large routing table is considered to be inefficient in network design and operation. By default, this check is performed at the following times:

- Whenever the total number of indirect routes exceeds the maximum threshold (default 2000)
- 30 minutes after stack initialization (provided that the maximum threshold has not been exceeded)
- Specified interval (default 168 hours for weekly)

The defaults for the maximum threshold and interval can be overridden on either a POLICY statement in the HZSPRMxx parmlib member or on a MODIFY command. The check name is suffixed by *TCPIPstackname*, which is the job name of each TCP/IP stack that is started, to define a separate check for each stack.

CSTCP_SYSTCPIP_CTRACE_*TCPIPstackname*
Checks whether TCP/IP Event Trace (SYSTCPIP) is active with options other than the default options (MINIMUM, INIT, OPCMDS, or OPMSGS). By default, this check will be performed once at stack initialization and then will be repeated once every 24 hours. This default can be overridden on either a POLICY statement in the HZSPRMxx parmlib member or on a MODIFY command. The check name is suffixed by *TCPIPstackname*, which is the job name of each TCP stack that is started, to define a separate check for each stack.

CSTCP_SYSPLEXMON_RECOV_*TCPIPstackname*
Checks whether the IPCONFIG DYNAMICXCF or IPCONFIG6 DYNAMICXCF parameters have been specified and the GLOBALCONFIG SYSPLEXMONITOR RECOVERY parameter has been specified. This check produces an exception message if the IPCONFIG DYNAMICXCF or IPCONFIG6 DYNAMICXCF parameters were specified, but the GLOBALCONFIG SYSPLEXMONITOR NORECOVERY parameter is in effect. By default, this check is performed once at stack initialization. This default can be overridden on either a POLICY statement in the HZSPRMxx parmlib member or on a MODIFY command. The check name is suffixed by *TCPIPstackname*, which is the job name of each TCP stack that is started, to define a separate check for each stack.

CSTCP_TCPMAXRCVBUFRSIZE_*TCPIPstackname*
Checks whether the configured TCP maximum receive buffer size is sufficient to provide optimal support to the z/OS Communications Server FTP Server. By default, this check is performed once at stack initialization and whenever a VARY TCPIP,,OBEYFILE command changes the TCPMAXRCVBUFRSIZE parameter. By default, it checks that TCPMAXRCVBUFRSIZE is at least 180K. These defaults can be overridden on either a POLICY statement in the HZSPRMxx parmlib member or on a MODIFY command. The check name is suffixed by *TCPIPstackname*, which is the job name of each TCP stack that is started, to define a separate check for each stack.

ZOSMIGV2R1_CS_GATEWAY
Checks whether GATEWAY statements are in use on the system. By default, this check is inactive. This default can be overridden on either a POLICY statement in the HZSPRM*xx* parmlib member or on a MODIFY command. If an IBM Health Checker for z/OS exception message is generated, migration must be performed.

ZOSMIGV2R1_CS_LEGACYDEVICE
Checks whether any TCP/IP profile statements for legacy device types have been configured on this system. The check pertains to the following profile statements:

- DEVICE and LINK profile statements for the following device types: ATM, CDLC, CLAW, HYPERchannel, SNALINK (LU0 and LU6.2), and X.25.
- Profile statements associated with ATM device types: ATMARPSV, ATMLIS, and ATMPVC.

By default, this check is inactive. This default can be overridden on either a POLICY statement in the HZSPRM *xx* parmlib member or on a MODIFY command. If an IBM Health Checker for z/OS exception message is generated, migration must be performed.

Chapter 5. IP Messages: Volume 1 (EZA)

EZA4443I Attempt to open *userid*.**RHOSTS.DATA requested by** *user* **on** *host*

Explanation: MVRSHD server detected an RSH client attempting to use *userid*.RHOSTS.DATA for authentication.

In the message text:

userid
: The remote user ID specified by the RSH client to use for execution of the remote command

user
: The local user ID of the RSH client

host
: Fully qualified local host name of the RSH client

System action: The system continues processing.

Operator response: Not applicable.

System programmer response: Not applicable.

User response: Not applicable.

Problem determination: Not applicable.

Source: z/OS CS TCP/IP: REXEC

Module: MVRSHD

Routing code: Not applicable.

Descriptor code: Not applicable.

Automation: Not applicable.

Example:

```
socket 2: EZA4443I Attempt to open USER1.RHOSTS.DATA requested by user2 on sys209.pok.im.com
```

Chapter 6. SNA Messages

ISTH029I No MVRSHD servers are active

Explanation: Check CSAPP_MVRSHD_RHOSTS_DATA ran successfully and found no exceptions. The check determined that no MVRSHD servers are active.

IBM suggests avoiding the use of MVRSHD servers. The MVRSHD server supports the RSH and REXEC protocols which transfer user ID and password information in the clear. There is also the potential of weak authentication for RSH clients using RHOSTS.DATA datasets. This authentication method allows remote command execution without requiring the RSH client to supply a password.

System action: The system continues processing.

Operator response: Not applicable.

System programmer response: Not applicable.

User response: Not applicable.

Problem determination: Not applicable.

Source: z/OS CS Health Checker

Module: ISTHCCK2

Routing code: Not applicable.

Descriptor code: Not applicable.

Automation: Not applicable.

Example:

```
ISTH029I  There are no MVRSHD servers active
```

ISTH030E One or more MVRSHD servers are active

Explanation: Check CSAPP_MVRSHD_RHOSTS_DATA determined that one or more MVRSHD servers are active.

IBM suggests avoiding the use of MVRSHD servers. The MVRSHD server supports the RSH and REXEC protocols which transfer user ID and password information in the clear. There is also the potential of weak authentication for RSH clients using RHOSTS.DATA datasets. This authentication method allows remote command execution without requiring the RSH client to supply a password.

System action: The system continues processing.

Operator response: Contact the system programmer.

System programmer response: Examine the report that was produced by check CSAPP_MVRSHD_RHOSTS_DATA.

This report lists all the active MVRHSD server address spaces. To assist in identifying the server instance, the report contains the MVRSHD server job name in the first column, the ASID value in hexadecimal format in the second column, and a data value in hexadecimal format in the third column. The following indicates the meaning of the data value:

0 – This MVRSHD server is active and no attempts to authenticate using RHOSTS.DATA have occurred.

1 – This MVRSHD server is active and there have been one or more attempts to authenticate using RHOSTS.DATA.

To obtain more information regarding future attempts of RSH clients attempting to use RHOSTS.DATA datasets for authentication, take the following action:

Enable internal RHOSTS level MVRSHD server tracing by issuing MODIFY *proc_name*,TRACE=(LOG,RHOSTS). When a new RSH client using a RHOSTS.DATA dataset for authentication is detected, an EZA4443I message will be recorded in the MVRSHD server joblog. The message will read:

```
EZA4443I Attempt to open USER1.RHOSTS.DATA requested by user2 on sys209.pok.im.com
```

For more information on the EZA4443I message, see EZA4443I in z/OS Communications Server: IP Messages Volume 1 (EZA).

To disable the RHOSTS level MVRSHD server tracing, issue the MODIFY *proc_name*,TRACE=(LOG,NORHOSTS) command. To disable all tracing in addition to the RHOSTS level MVRSHD server tracing, issue the MODIFY *proc_name*,TRACE=(NOLOG,NORHOSTS) command.

To disable the ability for RSH clients to use RHOSTS.DATA datasets for authentication, take the following action:

Identify and delete all specified *userid*.RHOSTS.DATA datasets.

The EZA4443I message can help identify which datasets are in use.

To disable the support for RSH and REXEC protocols, take the following action:

Stop all active instances of MVRSHD server.

For more information on the RHOSTS.DATA dataset, see Step 3: Permit remote users to access MVS resources (optional) in z/OS Communications Server: IP Configuration Guide.

User response: Not applicable.

Problem determination: Not applicable.

Source: z/OS CS Health Checker

Module: ISTHCCK2

Routing code: Not applicable.

Descriptor code: Not applicable.

Automation: Not applicable.

Example:

```
ISTH030E One or more MVRSHD servers are active
```

ISTH031I No active SMTP servers are configured as mail relays

Explanation: Check CSAPP_SMTPD_MAIL_RELAY ran successfully and found no exceptions. The check determined that no active SMTP servers are configured as mail relays.

IBM suggests that the INBOUNDOPENLIMIT configuration statement be set to 0 for SMTP servers. Specifying the INBOUNDOPENLIMIT statement with a valid non-zero value or allowing it to default to the value of 256 causes the SMTP server to open a listening port and implicitly become exploitable by remote users as a mail relay.

System action: The system continues processing.

Operator response: Contact the system programmer.

System programmer response: Not applicable.

User response: Not applicable.

Problem determination: Not applicable.

Source: z/OS CS Health Checker

Module: ISTHCCK2

Routing code: Not applicable.

Descriptor code: Not applicable.

Automation: Not applicable.

Example:

```
ISTH031I  No active SMTP servers are configured as mail relays
```

ISTH032E One or more active SMTP servers are configured as mail relays

Explanation: Check CSAPP_SMTPD_MAIL_RELAY determined that one or more active SMTP servers are configured as mail relays.

IBM suggests that the INBOUNDOPENLIMIT configuration statement be set to 0 for SMTP servers. Specifying the INBOUNDOPENLIMIT statement with a valid non-zero value or allowing it to default to the value of 256 causes the SMTP server to open a listening port and implicitly become exploitable by remote users as a mail relay.

System action: The system continues processing.

Operator response: Contact the system programmer.

System programmer response: Examine the report that was produced by check CSAPP_SMTPD_MAIL_RELAY.

This report lists all the SMTP server address spaces configured as mail relays. To assist in identifying the server instance, the report contains the SMTP server job name in the first column and the ASID value in hexadecimal format in the second column. The SMTP server configuration file includes the INBOUNDOPENLIMIT configuration statement. The INBOUNDOPENLIMIT statement is explicitly configured with a value other than 0 or was allowed to default to the value 256.

To disable the ability for the SMTP server to be used as a mail relay, take the following action:

Disable the listener port on the SMTP server.

To do this, set INBOUNDOPENLIMIT 0 statement in the SMTP server configuration file.

For more information, see INBOUNDOPENLIMIT statement in z/OS Communications Server: IP Configuration Reference.

User response: Not applicable.

Problem determination: Not applicable.

Source: z/OS CS Health Checker

Module: ISTHCCK2

Routing code: Not applicable.

Descriptor code: Not applicable.

Automation: Not applicable.

Example:

```
ISTH032E  One or more active SMTP servers are configured as mail relays
```

ISTH033I No active SNMP agents are configured with a community name of public

Explanation: Check CSAPP_SNMPAGENT_PUBLIC_COMMUNITY ran successfully and found no exceptions. The check determined that no active SNMP agents are configured with a community name of public.

IBM suggests not configuring a community name of public, nor permitting the SNMP agent to use the default community name of public. Because the SNMP community name of public is a well-known name, it should not be used with community-based security due to security considerations.

System action: The system continues processing.

Operator response: Not applicable.

System programmer response: Not applicable.

User response: Not applicable.

Problem determination: Not applicable.

Source: z/OS CS Health Checker

Module: ISTHCCK2

Routing code: Not applicable.

Descriptor code: Not applicable.

Automation: Not applicable.

Example:

```
ISTH033I  No active SNMP agents are configured with a community name of public
```

ISTH034E One or more active SNMP agents are configured with a community name of public

Explanation: Check CSAPP_SNMPAGENT_PUBLIC_COMMUNITY determined that one or more active SNMP agents are configured with a community name of public.

IBM suggests not configuring a community name of public, nor permitting the SNMP agent to use the default community name of public. Because the SNMP community name of public is a well-known name, it should not be used with community-based security due to security considerations.

System action: The system continues processing.

Operator response: Not applicable.

System programmer response: Examine the report that was produced by check CSAPP_SNMPAGENT_PUBLIC_COMMUNITY. This report lists all the SNMP agent address spaces for which a community name of public is configured. To assist in identifying the server instance, the report contains the SNMP agent job name in the first column and the ASID value in hexadecimal format in the second column.

To remove the community name of public associated with the SNMP agent, take one of the following actions:

- If no SNMP agent configuration files are being used, explicitly define a community name other than public as the value of the -c start parameter.

  For more information on the -c start parameter, see OSNMPD parameters in z/OS Communications Server: IP Configuration Reference.
- If you are using a PW.SRC configuration file, rename the community named public to something else by specifying the associated PW.SRC statement.

  For more information on defining a community via the PW.SRC configuration file, see PW.SRC statement syntax in z/OS Communications Server: IP Configuration Reference.
- If you are using a SNMPD.CONF configuration file, rename the community named public by specifying the COMMUNITY or SNMP_COMMUNITY statements.

  For more information on the COMMUNITY statement and the SNMP_Community statement, see SNMPD.CONF statements in z/OS Communications Server: IP Configuration Reference.

You can also migrate your SNMP configuration from community-based security to user-based security using SNMPv3 protocols. SNMPv3 provides a more powerful and flexible framework for message security and access control. SNMPv3 user-based security can be used only when all your SNMP manager and network management applications support it. For an introduction to user-based security using SNMPv3, see User-based security in z/OS Communications Server: IP Configuration Guide.

User response: Not applicable.

Problem determination: Not applicable.

Source: z/OS CS Health Checker

Module: ISTHCCK2

Routing code: Not applicable.

Descriptor code: Not applicable.

Automation: Not applicable.

Example:

```
ISTH034E  One or more active SNMP agents are configured with a community name of public
```

Chapter 7. IBM Health Checker for z/OS User's Guide

Communications Server checks (IBMCS)

CSAPP_FTPD_ANONYMOUS_JES

Description:
Checks to see if ANONYMOUS users can submit jobs.

Reason for check:
When ANONYMOUS FTP is allowed on the FTP server, it is critical that ANONYMOUSLEVEL 3 and ANONYMOUSFILETYPEJES FALSE are set, or anonymous users can submit jobs to run on the system.

z/OS releases the check applies to:
z/OS V2R1 and later.

User override of IBM values:
The following shows the default keywords for the check, which you can override on either a POLICY statement in the HZSPRMxx parmlib member or on a MODIFY command. This statement can be copied and modified to override the check defaults:

```
UPDATE
CHECK(IBMCS,CSAPP_FTPD_ANONYMOUS_JES)
DATE('date_of_the_change')
REASON('Your reason for making the update.')
ACTIVE
SEVERITY(MEDIUM)
INTERVAL(ONETIME)
```

Debug support:
No

Verbose support:
No

Parameters accepted:
No

Reference:
For more information on ANONYMOUS, ANONYMOUSLEVEL, and ANONYMOUSFILETYPEJES, see the following sections in *z/OS Communications Server: IP Configuration Reference*:

- The ANONYMOUS (FTP server) statement
- The ANONYMOUSLEVEL (FTP server) statement
- The ANONYMOUSFILETYPEJES (FTP server) statement

Messages:
This check issues the following exception messages:

- ISTH021E

See *z/OS Communications Server: SNA Messages*.

SECLABEL recommended for multilevel security users:
SYSLOW - see *z/OS Planning for Multilevel Security and the Common Criteria* for information on using SECLABELs.

CSAPP_MVRSHD_RHOSTS_DATA

Description:
Checks to see if the MVRSHD server is active and detects if a RSH client uses a RHOSTS.DATA dataset for authentication.

Reason for check:
The MVRSHD server supports the RSH and REXEC protocols that transfer user ID and password information in the clear. There is also the potential of weak authentication for RSH clients that use RHOSTS.DATA datasets. This authentication method allows remote command execution without requiring the RSH client to supply a password.

z/OS releases the check applies to:
z/OS V2R1 and later.

User override of IBM values:
The following shows the default keywords for the check, which you can override on either a POLICY statement in the HZSPRMxx parmlib member or on a MODIFY command. This statement can be copied and modified to override the check defaults:

```
UPDATE
CHECK(IBMCS,CSAPP_MVRSHD_RHOSTS_DATA)
DATE('date_of_the_change')
REASON('Your reason for making the update.')
ACTIVE
SEVERITY(MEDIUM)
INTERVAL(ONETIME)
```

Debug support:
No

Verbose support:
No

Parameters accepted:
No

Reference:
For more information on the RHOSTS.DATA dataset, see the Step 3: Permit remote users to access MVS resources (optional) section in *z/OS Communications Server: IP Configuration Guide*.

Messages:
This check issues the following exception messages:

- ISTH030E

See *z/OS Communications Server: SNA Messages*.

SECLABEL recommended for multilevel security users:
SYSLOW - see *z/OS Planning for Multilevel Security and the Common Criteria* for information on using SECLABELs.

CSAPP_SMTPD_MAIL_RELAY

Description:
Checks to see if the INBOUNDOPENLIMIT statement is set to 0.

Reason for check:
Specifying the INBOUNDOPENLIMIT statement to a valid non-zero value or

allowing it to default to the value of 256 causes the SMTP server to open a listening port and implicitly become exploitable by remote users as a mail relay.

z/OS releases the check applies to:
z/OS V2R1 and V2R2.

User override of IBM values:
The following shows the default keywords for the check, which you can override on either a POLICY statement in the HZSPRMxx parmlib member or on a MODIFY command. This statement can be copied and modified to override the check defaults:

```
UPDATE
CHECK(IBMCS,CSAPP_SMTPD_MAIL_RELAY)
DATE('date_of_the_change')
REASON('Your reason for making the update.')
ACTIVE
SEVERITY(MEDIUM)
INTERVAL(ONETIME)
```

Debug support:
No

Verbose support:
No

Parameters accepted:
No

Reference:
For more information on the INBOUNDOPENLIMIT statement, see the INBOUNDOPENLIMIT statement section in *z/OS Communications Server: IP Configuration Reference*.

Messages:
This check issues the following exception messages:

- ISTH032E

See *z/OS Communications Server: SNA Messages*.

SECLABEL recommended for multilevel security users:
SYSLOW - see *z/OS Planning for Multilevel Security and the Common Criteria* for information on using SECLABELs.

CSAPP_SNMPAGENT_PUBLIC_COMMUNITY

Description:
Checks to see if the SNMP agent has been configured with a community name of public.

Reason for check:
The community name of public is a well-known name and should not be used with community-based security because of security considerations. The community name can be defined by one of the following methods:

- Specify the -c start parameter.
- Configure a PW.SRC configuration file.
- Configure the COMMUNITY or SNMP_COMMUNITY statements in the SNMPD.CONF configuration file.

If you use SNMPTRAP.DEST to configure trap information, the agent uses the hardcoded community name of public in the outbound traps. To configure

specific community names for trap destinations, you must convert your SNMPTRAP.DEST information to a SNMPD.CONF configuration file format.

z/OS releases the check applies to:
z/OS V2R1 and later.

User override of IBM values:
The following shows the default keywords for the check, which you can override on either a POLICY statement in the HZSPRMxx parmlib member or on a MODIFY command. This statement can be copied and modified to override the check defaults:

```
UPDATE
CHECK(IBMCS,CSAPP_SNMPAGENT_PUBLIC_COMMUNITY)
DATE('date_of_the_change')
REASON('Your reason for making the update.')
ACTIVE
SEVERITY(MEDIUM)
INTERVAL(ONETIME)
```

Debug support:
No

Verbose support:
No

Parameters accepted:
No

Reference:
For more information on configuring community names, see the following sections in *z/OS Communications Server: IP Configuration Reference*:

- OSNMPD parameters
- PW.SRC statement syntax
- COMMUNITY entry
- SNMP_COMMUNITY entry
- Migrating the PW.SRC file and SNMPTRAP.DEST file to the SNMPD.CONF file

Messages:
This check issues the following exception messages:

- ISTH034E

See *z/OS Communications Server: SNA Messages*.

SECLABEL recommended for multilevel security users:
SYSLOW - see *z/OS Planning for Multilevel Security and the Common Criteria* for information on using SECLABELs.

Index

Printed in USA