Documentation updates for Communications Server Socket Directory Permissions
July 14, 2026

The following book updates are added to z/OS 3.2 and are also applicable to z/OS 3.1 and 2.5.

z/OS Communications Server: IP Configuration Guide

A new topic "Network management interfaces (NMI) sockets directory required permissions" is added to Chapter 3. Security under TCP/IP resource protection.

z/OS Communications Server provides information about TCP/IP and VTAM network operations by using network management interfaces (NMIs).

The /var/sock directory is used to store AF_UNIX socket files that are created by NMI services. The following NMI services use this directory:

- Real-time TCP/IP network monitoring NMI
- Local IPsec NMI
- Network security services NMI
- SNA network monitoring NMI

For more information about NMIs, see Network management interfaces in the z/OS Communications Server: IP Programmer's Guide and Reference.

If the /var/sock directory does not exist, use a superuser ID to create it, set the access to all, and set the sticky bit on.

```
mkdir /var/sock
chown 0 /var/sock
chmod 777 /var/sock
chmod +t /var/sock
```

The /var/sock directory must meet the following requirements:

- The NMI service user ID (see the table below) must have permission to create, delete, read, and write files to this directory.
- The /var/sock directory must not be a symbolic link to another directory.
- The sticky bit must be set on the /var/sock directory.

If /var/sock does not meet the above requirements, the associated NMI service will fail – see the table below to determine the scope of the failure.

Network management interface	**NMI service user ID**	**Scope of failure**	**Failure message issued with reason**
Real-time TCP/IP network monitoring NMI	TCP/IP user ID	TCP/IP initialization continues but the real-time TCP/IP NMI is not available	EZZ4251I
Local IPsec NMI	IKED user ID	IKED initialization continues but the local IPsec NMI is not available	EZD2105I and EZD1122I
Network Security Services NMI	NSSD user ID	NSSD fails to start	EZD1396I
SNA network monitoring NMI	VTAM user ID	VTAM initialization continues, but the SNA NMI is not available	IST1927I

Topic "**Steps for authorizing resources for NSS**" in Chapter 19. Network security services -> Preparing to provide network security services is updated to provide new information on the /var/sock directory.

Perform the following steps to authorize access to the appropriate resources:

1) Define and authorize the NSSD user ID.

Documentation updates for Communications Server Socket Directory Permissions
July 14, 2026

The NSS server is a z/OS® UNIX application that you can start from the z/OS UNIX shell or from an MVS started procedure. Before starting the NSS server, you must define the NSSD user ID to the external security manager with UID 0. If you start the NSS server from an MVS started procedure, the NSSD user ID must also be authorized to the STARTED class.

Issue the following commands:

```
ADDUSER  NSSD     DFLTGRP(OMVSGRP) NOPASSWORD OMVS(UID(0)  HOME('/'))
RDEFINE  STARTED  NSSD.*            STDATA(USER(NSSD))
SETROPTS RACLIST(STARTED) REFRESH
SETROPTS GENERIC(STARTED) REFRESH
```

If the /var/sock directory does not already exist, see the new topic described above 'Network management interfaces (NMI) sockets directory required permissions' for the steps to create the directory and set the required permissions.

Topic "Steps for authorizing the IKE daemon to RACF" in Appendix E. Steps for preparing to run IP security is updated to provide additional information on creating the directories IKED needs when the IKED user ID is not a superuser.

- If IKED is defined using a nonzero UID (for example, 300) and IKE GID (for example, 931), then IKED needs the following access:

  ```
  ADDGROUP IKE OMVS(GID(931))
  ADDUSER  IKED DFLTGRP(IKE) OMVS(UID(300) HOME('/var/ike/'))
  NOPASSWORD
  CONNECT  IKED GROUP(IKE) UACC(READ)
  RDEFINE  STARTED  IKED.*         STDATA(USER(IKED))
  PERMIT   BPX.DAEMON CLASS(FACILITY) ID(IKED)    ACCESS(READ)
  SETROPTS RACLIST(STARTED) REFRESH
  SETROPTS GENERIC(STARTED) REFRESH
  ```

  a. Set the /var directory access to all using the following command:

     chmod 777 /var

  b. If the /var/ike directory does not already exist, use a superuser ID to create it. The IKED user ID must have permission to create, delete, read, and write files to this directory. Set the IKED user ID as the owner of /var/ike. The z/OS UNIX

ipsec command user ID must have access to the /var/ike directory for the -f default and reload commands. See Steps for creating group access control over the path for the ipsec command (updated in this PDF) in the z/OS Communications Server: IP System Administrator's Commands for information on setting up a group to provide ipsec command users access to /var/ike. In the example below, group ownership of /var/ike is set to the group IKE.

```
mkdir /var/ike
chown IKED /var/ike
chgrp IKE /var/ike
chmod 775 /var/ike
```

If /var/ike does not meet the following requirements, IKED fails to start and message EZD2105I is written indicating the reason.

- The IKED user ID must have permission to create, delete, read, and write files to /var/ike.
- Write access to /var/ike must be restricted to the owning UID or group, for example --w--w--- permissions.
- /var/ike directory must not be a symbolic link to another directory.

c. If the /var/sock directory does not already exist, see the new topic described above 'Network management interfaces (NMI) sockets directory required permissions' for the steps to create the directory and set the required permissions.

Topic "**Steps for configuring the DMD**" in Chapter 20. Defensive filtering is updated to provide additional information on creating the directories that the DMD needs.

Step 2. Create the directories that the DMD needs.

a. Use a superuser ID to create the directory /var/dm for use by the DMD. The DMD user ID must have permission to create, delete, read, and write files to this directory. Set the DMD user ID as the owner of /var/dm. Restrict write access to the directory to the owning UID. Set the sticky bit for the directory.

```
mkdir /var/dm
chown DMD /var/dm
chmod 755 /var/dm
chmod +t /var/dm
```

If /var/dm does not meet the following requirements, DMD fails to start and message EZD1736I is written indicating the reason.

- The DMD user ID must have permission to create, delete, read, and write files to /var/dm.
- Write access to /var/dm must be restricted to the owning UID or group, for example --w--w--- permissions.
- /var/dm must not be a symbolic link to another directory.
- The sticky bit must be set on the /var/dm directory.

b. If you set the PID file location with the DMD_PIDFILE environment variable, ensure that the path portion of the file name exists and that the DMD user ID has permission to create and write files to that directory. If you use the default PID file location, /var/dm/dmd.pid, you have already created the directory and given the DMD user ID the appropriate access in the previous step.

   Restrictions:

   - If the DMD PID file is a symbolic link, the target file must have an owning UID or GID that matches the EUID or EGID that is assigned to the DMD.
   - If the DMD PID file is a hard link or the target of a hard link, users that are outside the owner or group of the directory in which the PID file (for example, var/dm) is stored cannot have write access to the directory. Additionally, write access to the PID file must be limited to the owning UID or group, for example, --w--w----permissions.

Documentation updates for Communications Server Socket Directory Permissions
July 14, 2026

Topic "Other considerations when starting the Policy Agent" in Chapter 15. Policy-based networking -> Starting and stopping the Policy Agent is updated to provide new information on the /var/pagent directory.

If you start the Policy Agent with a user ID that does not have superuser authority [UID(0)], then ensure that the PAGENT user ID has the following permissions:

Directories

- /tmp/ - read and write permission is required for the PAGENT user ID
- /var/tmp/ - read and write permission is required for the PAGENT user ID
- /var/pagent – read, write, and execute permission is required for the PAGENT user ID.

  Policy agent creates socket file pagent.sock in /var/pagent. This socket is used by the policy api (papi) to access policy agent. For example, the pasearch command connects to the socket file to retrieve policies for display.

  Use a superuser ID to create the directory /var/pagent for use by Policy Agent (pagent). The pagent user ID must have permission to create, delete, read and write files to this directory. Set the pagent user ID as the owner of /var/pagent. Restrict write access to the directory to the owning UID. Set the sticky bit on for the directory.

  ```
  mkdir /var/pagent
  chown PAGENT /var/pagent
  chmod 755 /var/pagent
  chmod +t /var/pagent
  ```

  If /var/pagent does not meet the following requirements, pagent fails to start and message EZD1593I is written indicating the reason.

  - The pagent user ID must have permission to create, delete, read, and write files to /var/pagent.
  - Write access to /var/pagent must be restricted to the owning UID or group, for example --w--w--- permissions.
  - /var/pagent must not be a symbolic link to another directory.
  - The sticky bit must be set on the /var/pagent directory.

z/OS Communications Server: IP System Administrator's Commands

Topic "Group access control for local host stacks" in Chapter 3. Managing network security -> ipsec command->ipsec command security is updated to:

- Remove the requirement for group access control for the ipsec -k and ipsec -y commands.
- Provide additional information on requirements for the /var/ike directory.

A user does not require root authority to use the ipsec command, but to avoid erroneous or malicious manipulations of files that are used by the ipsec command, the administrator must perform the following steps to require group access control. Group access control is required for the following commands:

- To change filter sets in the stack on the local system (ipsec -f default or reload), the ipsec command creates or deletes a specific marker file that the stack accesses.

Steps for creating group access control over the path for the ipsec command

1. Create a supplementary RACF® group, assign it a group ID (gid), and ensure that the primary administrator is a member of the group.

   ```
   ADDGROUP IKE OMVS(GID(931))
   CONNECT user-special GROUP(IKE) UACC(READ)
   ```

   ---

2. Issue the following UNIX System Services commands to set file management at the group level. The path for the ipsec command files is /var/ike. If the directory /var/ike does not exist, first use a superuser ID to create it.

   ```
   mkdir /var/ike
   chgrp IKE /var/ike
   chmod 775 /var/ike
   ```

   ---

3. Use RACF commands to control which users can manipulate files.

```
CONNECT USER5 GROUP(IKE) UACC(READ)
REMOVE USER5 GROUP(IKE)
```

Requirements for the /var/ike directory for use by the ipsec command:

- The user ID issuing the ipsec -f default or reload command must have permission to create, delete, read, and write files to this directory.
- Write access to /var/ike must be restricted to the owning UID and group, for example --w--w--- permissions.
- The /var/ike directory must not be a symbolic link to another directory.

If the requirements are not met, the ipsec -f default or reload command fails with message EZD1555I which indicates the reason.

If the IKE daemon is in use on the system, there are also requirements for the /var/ike directory for the IKED user ID. See **Steps for authorizing the IKE daemon to RACF** (updated in this PDF) in the z/OS Communications Server: IP Configuration Guide.

z/OS Communications Server: IP Messages Volume 2 (EZB, EZD)

The following new messages are added:

EZD1396I THE NSS SOCKET DIRECTORY *directory* DOES NOT MEET THE PERMISSION RESTRICTIONS – *reason*

This message is issued by the Network Security Services (NSS) server when the socket directory exists but does not meet the required directory permissions.

In the message text:

directory
: The name of the NSS NMI socket directory.

reason
: The reason that the NSS NMI failed to start.

***reason* values**	**Explanation**
symbolic link detected	Indicates that *directory* is a symbolic link. This is not permitted.

not a directory	Indicates that *directory* exists and is not a valid directory.
sticky bit is not on	Indicates that the sticky bit is not set on *directory*. This is required.
directory attributes were modified	Indicates that the *directory* was modified after it was verified by the NSSD user ID and before the NSSD user ID could create the /var/sock/nss socket file.

System action: The NSS server does not start.

System programmer response:

The system programmer response depends on the *reason* value.

***reason* value(s)**	**System programmer response**
symbolic link detected	1. Remove the link with the `unlink /var/sock` command 2. Create the socket directory and set the correct permissions. See the new topic described above 'Network management interfaces (NMI) sockets directory required permissions' in the z/OS Communications Server: IP Configuration Guide.
not a directory	1. Delete /var/sock file with the `rm /var/sock` command 2. Create the socket directory and set the correct permissions. See the new topic described above 'Network management interfaces (NMI) sockets directory required permissions' in the z/OS Communications Server: IP Configuration Guide.
sticky bit is not on	Set the sticky bit on the NSS NMI directory with the `chmod +t /var/sock command`
directory attributes were modified	Use a superuser ID to issue the `ls -la` command to ensure the following:

	• /var/sock is not a symbolic link. If it is a symbolic link, follow the steps for 'symbolic link detected' in this table. • Directory /var/sock has the sticky bit set. If not, follow the steps for 'sticky bit is not on' in this table. • /var/sock is a valid directory. If not, follow the steps for 'not a directory' in this table.

After the problem is resolved, restart the NSS server.

EZD1397I *syscall* COMMAND FAILED FOR NSS SOCKET DIRECTORY *directory* – ERRNO *errno* ERRNOJR *errnojr* ERRNO DESCRIPTION *description*

A system call issued by the network security services (NSS) server to access or create the directory failed. This message is issued when NSS attempts to establish an AF_UNIX socket for the NSS network monitoring interface (NMI).

In the message text:

syscall
A string representation of the system call NSS issued. System calls are described in the Library functions information in the z/OS C/C++ Runtime Library Reference.

directory
The socket directory that cannot be accessed or created by NSS.

errno
The z/OS® UNIX System Services return code. These return codes are listed and described in the Return codes (errnos) information in z/OS UNIX System Services Messages and Codes.

errnojr
The hexadecimal z/OS UNIX System Services reason code. The format of the 4-byte reason code is explained in the introduction to the Reason codes information in the z/OS UNIX System Services Messages and codes, where the reason codes are listed.

description
Describes the meaning of the *errno* value.

System action: NSS server initialization fails.

Documentation updates for Communications Server Socket Directory Permissions
July 14, 2026

System programmer response:

Correct the error indicated by the *errno*, *errnojr*, and *description* values. There are several possible causes of this error, such as directory permissions that do not allow read, write, or execute access. See z/OS C/C++ Runtime Library Reference for more information about the *syscall* function call.

Ensure that the /var/sock directory exists or that the NSS user ID has the correct permissions to create it. If the NSS server needs to create the directory, all elements of the directory path must exist, except for the last element. For example, at least the /var directory must exist. See the new topic described above 'Network management interfaces (NMI) sockets directory required permissions' in the z/OS Communications Server: IP Configuration Guide.

After the problem is resolved, restart NSSD.

EZD1554I *syscall* command failed for directory *directory* - errno *errno* (*description*) errnojr *errnojr*

A system call issued by ipsec -f default or reload command processing failed when attempting to access *directory*.

In the message text:

syscall

A string representation of the system call the ipsec command issued. System calls are described in the Library functions information in the z/OS C/C++ Runtime Library Reference.

directory

The name of the directory the ipsec -f default or reload processing was unable to access.

errno

The z/OS UNIX System Services return code. These return codes are listed and described in the Return codes (errnos) information in the z/OS UNIX System Services Messages and Codes.

errnojr

The hexadecimal z/OS UNIX System Services reason code. The format of the 4-byte reason code is explained in the introduction to the Reason codes (errnojrs) information in the z/OS UNIX System Services Messages and Codes, where the reason codes are listed.

System action: The ipsec command processing ends.

System programmer response:

Correct the error indicated by the errno, errnojr, and description values. There are several possible causes of this error, such as directory permissions that do not allow read, write, or execute access. See z/OS XL C/C++ Runtime Library Reference for more information about the syscall function call.

The ipsec command user must be able to create, delete, read, and write files to the directory. See Steps for creating group access control over the path for the ipsec command (updated in this PDF) in the z/OS Communications Server: IP System Administrator's Commands for more details.

Once the error is corrected, reissue the ipsec command.

EZD1555I The directory *directory* does not meet the permission restrictions - *reason*

This message is issued by the ipsec command when the /var/ike directory exists but does not meet the required directory permissions. See 'Steps for creating group access control over the path for the ipsec command' (updated in this PDF) in the z/OS Communications Server: IP System Administrator's Commands for /var/ike directory requirements for the ipsec command.

In the message text:

Directory
: The name of the existing directory.

reason
: The reason the directory does not meet the required permissions

reason values	Explanation
symbolic link detected	Indicates that *directory* is a symbolic link. This is not permitted.
not a directory	Indicates that *directory* is not a valid directory.

ipsec command user ID *userID* has insufficient permissions	Indicates that the user ID issuing the ipsec -f default or reload command does not have sufficient access to *directory* to create or delete marker files. The ipsec command user ID is not defined as UID 0, does not own *directory*, and is not a member of the owning group defined on the *directory*.
write other permitted	Indicates that write access on *directory* is not restricted to owning UID and group.
directory attributes were modified	Indicates that *directory* was modified after it was verified by the ipsec command and before the ipsec command could create or delete marker files.

System action: The ipsec command processing ends.

System programmer response:

The system programmer response depends on the *reason* value.

***reason* value(s)**	**System programmer response**
symbolic link detected	1. Remove the link with the `unlink /var/ike` command 2. If the ipsec command user ID is not a superuser: • Create the /var/ike directory restricting write access to the owning UID and GID and permit access for the appropriate users to create, delete, read and write files to this directory. See **Steps for creating group access control over the path for the ipsec command** (updated in this PDF) in the z/OS Communications Server: IP System Administrator's Commands. 3. If the linked directory contained a file in the format *stackname*.default, use the `ipsec -f default -p stackname` command to recreate the file

not a directory	1. Delete the /var/ike file with the `rm /var/ike` command 2. If the ipsec command user ID is not a superuser: • Create the /var/ike directory restricting write access to the owning UID and GID and permit access for the appropriate users to create, delete, read and write files to this directory. See **Steps for creating group access control over the path for the ipsec command** (updated in this PDF) in the z/OS Communications Server: IP System Administrator's Commands.
ipsec command user ID *userID* has insufficient permissions	Permit the ipsec command user ID (e.g., IPSECUSR) access to create, delete, read, and write files to the directory. If the /var/ike directory is owned by group IKE: • `CONNECT IPSECUSR GROUP(IKE) UACC(READ)` See **Steps for creating group access control over the path for the ipsec command** (updated in this PDF) in the z/OS Communications Server: IP System Administrator's Commands.
write other permitted	Ensure write access on the directory is restricted to the owning UID and owning group • `chmod 775 /var/ike`
directory attributes were modified	Use a superuser ID to issue the `ls -la` command to ensure the following: • The directory is not a symbolic link. If it is a symbolic link, follow the steps for 'symbolic link detected' in this table. • The directory has write access restricted to the owning UID and group. If not, follow the steps for 'write other permitted' in this table. • The permissions on the directory allow the ipsec command user ID access to create, delete, read and writes files. If not, follow the steps for 'ipsec command user ID has insufficient permissions' in this table.

	• The directory is a valid directory. If not, follow the steps for 'not a directory' in this table.

Once the error is corrected, reissue the ipsec command.

EZD1593I THE POLICY AGENT SOCKET DIRECTORY *directory* DOES NOT MEET PERMISSION RESTRICTIONS - *reason*

This message is issued by Policy Agent (pagent) when its socket directory exists but does not meet the required directory permissions. See 'Other considerations when starting the Policy Agent' (updated in this PDF) in the z/OS Communications Server: IP Configuration Guide for pagent socket directory requirements.

In the message text:

d*irectory*
: The name of the pagent socket directory

reason
: The reason the socket directory does not meet the required permissions

***reason* value**	**Explanation**
symbolic link detected	Indicates that *directory* is a symbolic link. This is not permitted.
not a directory	Indicates that *directory* exists and is not a valid directory.
calling process user ID has insufficient permissions	Indicates that the pagent user ID does not have sufficient access to *directory* to create its AF_UNIX socket. The pagent user ID is not defined as UID0 and is not part of the owning uid and group defined on *directory*.
write other permitted	Indicates that write access on *directory* is not restricted to the owning uid and group.
sticky bit is not on	Indicates that the sticky bit is not set on *directory*. This is required.

socket directory attributes have been modified	Indicates that *directory* was modified after it was verified by pagent and before pagent could create its AF_UNIX socket.

System action: Policy Agent initialization fails and policy agent ends.

System programmer response:

The system programmer response depends on the *reason* value.

***reason* value(s)**	**System programmer response**
symbolic link detected	3. Remove the link with the `unlink /var/pagent` command 4. If the PAGENT user ID is not a superuser, create the /var/pagent directory and set the correct permissions. See Other considerations when starting the Policy Agent (updated in this PDF) in the z/OS Communications Server: IP Configuration Guide.
not a directory	1. Delete the /var/pagent file with the `rm /var/pagent` command 2. Create the /var/pagent directory and set the correct permissions. See Other considerations when starting the Policy Agent (updated in this PDF) in the z/OS Communications Server: IP Configuration Guide.
calling process user ID has insufficient permissions	Permit the pagent user ID access to create, delete, read, and write files to the socket directory • `chown PAGENT /var/pagent`
write other permitted	Ensure write access on the socket directory is restricted to the owning UID • `chmod 755 /var/pagent`
sticky bit is not on	Ensure the sticky bit is set on the socket directory • `chmod +t /var/pagent`

socket directory attributes have been modified	Use a superuser ID to issue the `ls -la` command to ensure the following: • The socket directory is not a symbolic link. If it is a symbolic link, follow the steps for 'symbolic link detected' in this table. • The socket directory has write access restricted to the owning UID or group. If not, follow the steps for 'write other permitted' in this table. • The socket directory has the sticky bit set. If not, follow the steps for 'sticky bit is not on' in this table. • The permissions on the socket directory allow the pagent user ID access to create, delete, read and writes files. If not, follow the steps for 'calling process user ID has insufficient permissions' in this table. • The socket directory is a valid directory. If not, follow the steps for 'not a directory' in this table.

Once the error is corrected, restart pagent.

EZD1594I *syscall* COMMAND FAILED FOR POLICY AGENT SOCKET DIRECTORY *directory* - ERRNO *errno* ERRNOJR *errnojr* ERRNO DESCRIPTION *description*

A system call issued by Policy Agent (pagent) to create or access the pagent socket directory failed.

In the message text:

syscall

A string representation of the system call Pagent issued. System calls are described in the Library functions information in the z/OS C/C++ Runtime Library Reference.

directory

The socket directory that cannot be accessed or created by the pagent user ID.

errno
The z/OS UNIX System Services return code. These return codes are listed and described in the Return codes (errnos) information in z/OS UNIX System Services Messages and Codes.

errnojr
The hexadecimal z/OS UNIX System Services reason code. The format of the 4-byte reason code is explained in the introduction to the Reason codes (errnojrs) information in z/OS UNIX System Services Messages and Codes, where the reason codes are listed.

description
Describes the meaning of the errno value.

System action: Policy Agent initialization fails and policy agent ends.

System programmer response:

Correct the error indicated by the errno, errnojr, and description values. There are several possible causes of this error, such as directory permissions that do not allow read, write, or execute access. See z/OS XL C/C++ Runtime Library Reference for more information about the syscall function call.

Ensure that the socket directory exists or that pagent has the correct permissions to create it. If pagent needs to create the socket directory, all elements of the directory path must exist, except for the last element. For example, at least the /var directory must exist. If the pagent user ID is defined with a nonzero UID, create the /var/pagent directory before starting pagent. The socket directory should be owned by the pagent user ID and pagent should be able to create, delete, read, and write files to the directory. See Other considerations when starting the Policy Agent (updated in this PDF) in the z/OS Communications Server: IP Configuration Guide for more details.

Once the error is corrected, restart pagent.

EZD1735I *syscall* COMMAND FAILED FOR DMD SOCKET DIRECTORY *directory* – ERRNO *errno* ERRNOJR *errnojr* ERRNO DESCRIPTION *description*

A system call issued by the Defense Manager Daemon (DMD) to access the socket directory failed.

In the message text:

syscall

A string representation of the system call DMD issued. System calls are described in the Library functions information in the z/OS C/C++ Runtime Library Reference.

directory

The name of the DMD socket directory.

errno

The z/OS® UNIX System Services return code. These return codes are listed and described in the return codes (errnos) information in z/OS UNIX System Services Messages and Codes.

errnojr

The hexadecimal z/OS UNIX System Services reason code. The format of the 4-byte reason code is explained in the introduction to the reason codes (errnojrs) information in z/OS UNIX System Services Messages and Codes, where the reason codes are listed.

description

Describes the meaning of *errno*.

System action: DMD initial startup processing fails and the DMD ends.

System programmer response:

Correct the error indicated by the *errno*, *errnojr*, and *description* values. There are several possible causes of this error, such as directory permissions that do not allow read, write, or execute access. See z/OS XL C/C++ Runtime Library Reference for more information about the *syscall* function call.

The /var/dm directory should be owned by the DMD user ID and DMD should be able to create, delete, read, and write files to the directory. See the 'Steps for configuring the DMD' (updated in this PDF) in the z/OS Communications Server: IP Configuration Guide for more details.

Once the error is corrected, restart DMD.

EZD1736I THE DEFENSE MANAGER DAEMON SOCKET DIRECTORY *directory* DOES NOT MEET THE PERMISSION RESTRICTIONS - *reason*

This message is issued by the Defense Manager daemon (DMD) when its socket directory exists but does not meet the required directory permissions. See 'Steps for configuring the DMD' (updated in this PDF) in the z/OS Communications Server: IP Configuration Guide for DMD socket directory requirements.

Documentation updates for Communications Server Socket Directory Permissions
July 14, 2026

In the message text:

d*irectory*

The name of the DMD socket directory

reason

The reason the socket directory does not meet the required permissions

***reason* value**	**Explanation**
symbolic link detected	Indicates that *directory* is a symbolic link. This is not permitted.
not a directory	Indicates that *directory* exists and is not a valid directory.
DMD user ID has insufficient permissions	Indicates that the DMD user ID does not have sufficient access to *directory* to create its AF_UNIX socket. The DMD user ID is not defined as UID0 and is not part of the owning uid and group defined on *directory*.
write other permitted	Indicates that write access on *directory* is not restricted to the owning uid and group.
sticky bit is not on	Indicates that the sticky bit is not set on *directory*. This is required.
socket directory attributes were modified	Indicates that *directory* was modified after it was verified by DMD and before DMD could create its AF_UNIX socket.

System action: DMD initial startup processing fails and the DMD ends

System programmer response:

The system programmer response depends on the *reason* value.

***reason* value(s)**	**System programmer response**
symbolic link detected	1. Remove the link with the `unlink /var/dm` command 2. Create the directory and set the correct permissions. See 'Steps for configuring the DMD' (updated in this PDF) in the z/OS Communications Server: IP Configuration Guide. 3. Preserve any files in the DMD filters directory

	If the DefensiveFilterDirectory statement in the DMD configuration file specifies a filters directory that does not use the /var/dm symbolic link, no further action is needed to preserve the files. If either the DefensiveFilterDirectory parameter uses the /var/dm symbolic link or the default defensive filter directory /var/dm/filters is being used, any files in the filters directory should be preserved. There are 2 options: Option 1: Update the DefensiveFilterDirectory parameter to specify the target of the /var/dm symbolic link. For more information on the DefensiveFilterDirectory parameter, see DmConfig statement in the z/OS Communications Server: IP Configuration Reference. Option 2: Create the filters directory in /var/dm and copy the files from the previous directory to the new directory.
not a directory	3. Delete the /var/dm file with the `rm /var/dm` command 4. Create the directory and set the correct permissions. See 'Steps for configuring the DMD' (updated in this PDF) in the z/OS Communications Server: IP Configuration Guide.
DMD user ID has insufficient permissions	Permit the DMD user ID access to create, delete, read, and write files to the socket directory • `chown DMD /var/dm`
write other permitted	Ensure write access on the socket directory is restricted to the owning UID • `chmod 755 /var/dm`
sticky bit is not on	Ensure the sticky bit is set on the socket directory • `chmod +t /var/dm`
socket directory attributes were modified	Use a superuser ID to issue the `ls -la` command to ensure the following:

	• The socket directory is not a symbolic link. If it is a symbolic link, follow the steps for 'symbolic link detected' in this table. • The socket directory has write access restricted to the owning UID or group. If not, follow the steps for 'write other permitted' in this table. • The socket directory has the sticky bit set. If not, follow the steps for 'sticky bit is not on' in this table. • The permissions on the socket directory allow the DMD user ID access to create, delete, read and writes files. If not, follow the steps for 'DMD user ID has insufficient permissions' in this table. • The socket directory is a valid directory. If not, follow the steps for 'not a directory' in this table.

Once the error is corrected, restart the DMD.

EZD2103I *syscall* COMMAND FAILED FOR IKE DAEMON HOME DIRECTORY *directory* – ERRNO *errno* (*description*) ERRNOJR *errnojr*

A system call issued by the IKE daemon to access or create the IKED home directory failed.

In the message text:

syscall

A string representation of the system call the IKE daemon issued. System calls are described in the Library functions information in the z/OS C/C++ Runtime Library Reference.

directory

The name of the existing home directory that cannot be accessed or created by the IKED user ID.

errno

The z/OS UNIX System Services return code. These return codes are listed and described in the Return codes (errnos) information in z/OS UNIX System Services Messages and Codes.

description

Describes the meaning of the errno value.

errnojr

The hexadecimal z/OS UNIX System Services reason code. The format of the 4-byte reason code is explained in the introduction to the Reason codes (errnojrs) information in z/OS UNIX System Services Messages and Codes, where the reason codes are listed.

System action: IKE daemon initialization stops and the daemon ends.

System programmer response:

Correct the error indicated by the errno, errnojr, and description values. There are several possible causes of this error, such as directory permissions that do not allow read, write, or execute access. See z/OS XL C/C++ Runtime Library Reference for more information about the syscall function call.

Ensure that the home directory exists or that the IKE daemon has the correct permissions to create it. If the IKE daemon needs to create the home directory, all elements of the directory path must exist, except for the last element. For example, at least the /var directory must exist. If the IKED user ID is defined with a nonzero UID, create the /var/ike directory before starting the IKE daemon. The home directory should be owned by the IKED user ID and the IKE daemon should be able to create, delete, read, and write files to the directory. See the 'Steps for authorizing the IKE daemon to RACF' (updated in this PDF) in the z/OS Communications Server: IP Configuration Guide for more details.

Once the error is corrected, restart the IKE daemon.

EZD2104I THE IKE DAEMON HOME DIRECTORY *directory* DOES NOT MEET THE PERMISSION RESTRICTIONS - *reason*

This message is issued by the IKE daemon when its home directory exists but does not meet the required directory permissions. See 'Steps for authorizing the IKE daemon to RACF' (updated in this PDF) in the z/OS Communications Server: IP Configuration Guide for IKED home directory requirements.

In the message text:

directory

The name of the IKED home directory.

reason

The reason the IKED home directory does not meet the required permissions

reason values	Explanation
symbolic link detected	Indicates that *directory* is a symbolic link. This is not permitted.
not a directory	Indicates that *directory* is not a valid directory.
IKED user ID has insufficient permissions	Indicates that the IKED user ID does not have sufficient access to *directory* to create its AF_UNIX socket. The IKED user ID is not defined as UID 0, does not own the *directory*, and is not part of the owning group defined on the *directory*.
write other permitted	Indicates that write access on *directory* is not restricted to owning UID and group.
home directory attributes were modified	Indicates that *directory* was modified after it was verified by the IKE daemon and before the IKE daemon could create its socket and PID file.

System action: IKE daemon initialization stops and the daemon ends.

System programmer response:

The system programmer response depends on the *reason* value.

***reason* value(s)**	**System programmer response**
symbolic link detected	1. Remove the link with the unlink /var/ike command 2. Create the directory and set the correct permissions. See Steps for authorizing the IKE daemon to RACF (updated in this PDF) in the IP Configuration Guide. 3. If the linked directory contains a file in the format *stackname*.default, use the ipsec -f default -p *stackname* command to recreate the file
not a directory	1. Delete the /var/ike file with the rm /var/ike command 2. Create the directory and set the correct permissions. See Steps for authorizing the IKE daemon to RACF (updated in this PDF) in the IP Configuration Guide.
IKED user ID has insufficient	Permit the IKED user ID access to create, delete, read, and write files to the directory

permissions	• chown IKED /var/ike
write other permitted	Ensure write access on the directory is restricted to the owning UID and owning group • chmod 775 /var/ike
home directory attributes were modified	Use a superuser ID to issue the ls -la command to ensure the following: • The directory is not a symbolic link. If it is a symbolic link, follow the steps for 'symbolic link detected' in this table. • The directory has write access restricted to the owning UID and group. If not, follow the steps for 'write other permitted' in this table. • The permissions on the directory allow the IKED user ID access to create, delete, read and write files. If not, follow the steps for 'IKED user ID has insufficient permissions' in this table. • The directory is a valid directory. If not, follow the steps for 'not a directory' in this table.

Once the error is corrected, restart the IKE daemon.

EZD2105I The IPsec NMI socket directory *directory* does not meet the permission restrictions - *reason*

This message is issued by the IKE daemon when the socket directory exists but does not meet the required directory permissions.

In the message text:

directory
: The name of the IPsec NMI socket directory.

reason
: The reason that the IPsec NMI failed to start.

reason values	Explanation
symbolic link detected	Indicates that *directory* is a symbolic link. This is not permitted
not a directory	Indicates that *directory* exists and is not a valid directory

sticky bit is not on	Indicates that the sticky bit is not set on *directory*. This is required.
directory attributes were modified	Indicates that the *directory* was modified after it was verified by the IKED user ID and before the IKED user ID could create the /var/sock/ipsecmgmt socket file.

System action: The local IPsec NMI is not available. IKE initialization continues.

System programmer response:

If the local IPsec NMI service is not used, no action is needed.

To activate the local IPsec NMI service, the system programmer response depends on the *reason* value.

***reason* value**	**System programmer response**
symbolic link detected	1. Remove the link with the `unlink /var/sock` command 2. Create the socket directory and set the correct permissions. See the new topic described above 'Network management interfaces (NMI) sockets directory required permissions' in the z/OS Communications Server: IP Configuration Guide.
not a directory	1. Delete /var/sock file with the `rm /var/sock` command 2. Create the socket directory and set the correct permissions. See the new topic described above 'Network management interfaces (NMI) sockets directory required permissions' in the z/OS Communications Server: IP Configuration Guide.
sticky bit is not on	Set the sticky bit on the IPsec NMI directory `chmod +t /var/sock`

directory attributes were modified	Use a superuser ID to issue the `ls -la` command to ensure the following: • /var/sock is not a symbolic link. If it is a symbolic link, follow the steps for 'symbolic link detected' in this table. • Directory /var/sock has the sticky bit set. If not, follow the steps for 'sticky bit is not on' in this table. • /var/sock is a valid directory. If not, follow the steps for 'not a directory' in this table.

To activate the local IPsec NMI service, stop and restart IKED after resolving the problem. For more details on the local IPsec NMI service, see Local IPSec NMI in the IP Programmer's Guide and Reference.

EZD2106I *syscall* command failed for IPsec NMI socket directory *directory* – errno *errno* (*description*) errnojr *errnojr*

A system call issued by IKED to access or create the directory failed. This message is issued during initialization of the IKE daemon when it attempts to establish an AF_UNIX socket for the local IPsec network monitoring interface (NMI). If the socket directory exists but does not meet the required directory permissions this message is written to syslogd.

In the message text:

syscall
A string representation of the system call IKED issued. System calls are described in the Library functions information in the z/OS C/C++ Runtime Library Reference.

directory
The socket directory that cannot be accessed or created by the IKED user ID.

errno
The z/OS® UNIX System Services return code. These return codes are listed and described in the Return codes (errnos) information in z/OS UNIX System Services Messages and Codes.

description
Describes the meaning of the *errno* value.

errnojr
The hexadecimal z/OS UNIX System Services reason code. The format of the 4-byte reason code is explained in the introduction to the Reason codes information in the z/OS UNIX System Services Messages and codes, where the reason codes are listed.

System action: The local IPsec NMI is not available. IKE initialization continues.

System programmer response:

Correct the error indicated by the *errno*, *errnojr*, and *description* values. There are several possible causes of this error, such as directory permissions that do not allow read, write, or execute access. See z/OS C/C++ Runtime Library Reference for more information about the *syscall* function call.

Ensure that the /var/sock directory exists or that the IKE user ID has the correct permissions to create it. If the IKE daemon needs to create the directory, all elements of the directory path must exist, except for the last element. For example, at least the /var directory must exist. If the IKED user ID is defined with a nonzero UID, create the directory before starting the IKE daemon. See the new topic described above 'Network management interfaces (NMI) sockets directory required permissions' in the z/OS Communications Server: IP Configuration Guide.

To activate the local IPsec NMI service, stop and restart IKED after resolving the problem. For more details on the local IPsec NMI service, see Local IPSec NMI in the IP Programmer's Guide and Reference.

z/OS Communications Server: IP Messages Volume 4 (EZZ, SNM)

The following new message is added:

Documentation updates for Communications Server Socket Directory Permissions
July 14, 2026

EZZ4251I TCP/IP NMI DIRECTORY *directory* DID NOT MEET PERMISSION RESTRICTIONS FOR *appl_name* – *reason*

This message is issued by the TCP/IP stack when it attempts to establish the AF_UNIX socket for a real-time network monitoring interface (NMI) application. If the socket directory exists but does not meet the required directory permissions, this message is written.

In the message text:

directory
: The name of the TCP/IP real-time NMI socket directory.

appl_name
: The name of the associated application server. Possible values are: SYSTCPDA, SYSTCPCN, SYSTCPOT, SYSTCPSM, SYSTCPER, SYSTCPES

reason
: The reason the TCP/IP real-time NMI application failed.

reason values	Explanation
symbolic link detected	Indicates that *directory* is a symbolic link. This is not permitted.
not a directory	Indicates that *directory* is not a valid directory.
sticky bit is not on	Indicates that the sticky bit is not set on the *directory*. This is required.
directory attributes were modified	Indicates that the *directory* was modified after it was verified by the TCP/IP stack user ID and before the TCP/IP stack user ID could create the socket files.

System action: TCP/IP real-time NMI processing ends for the specified application. TCP/IP stack processing continues.

System programmer response:

The system programmer response depends on the *reason* value.

***reason* value(s)**	**System programmer response**
symbolic link detected	Remove the link with the `unlink /var/sock` command.

not a directory	Delete /var/sock file with the `rm /var/sock` command
sticky bit is not on	Set the sticky bit on the TCP/IP real-time NMI directory `chmod +t /var/sock`
directory attributes were modified	Use a superuser ID to issue the `ls -la` command to ensure the following: • /var/sock is not a symbolic link. If it is a symbolic link, follow the steps for 'symbolic link detected' in this table. • Directory /var/sock has the sticky bit set. If not, follow the steps for 'sticky bit is not on' in this table. • /var/sock is a valid directory. If not, follow the steps for 'not a directory' in this table.

After the problem is resolved, issue a VARY TCPIP,,OBEYFILE with the NETMONITOR configuration to start the TCP/IP real-time NMI.