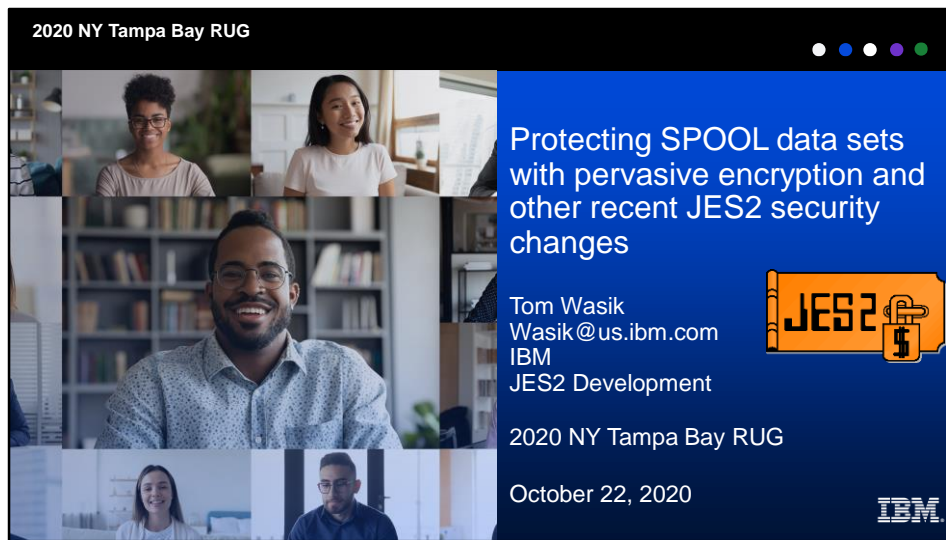


2020 NY Tampa Bay RUG




Protecting SPOOL data sets with pervasive encryption and other recent JES2 security changes

Tom Wasik
Wasik@us.ibm.com
IBM
JES2 Development

2020 NY Tampa Bay RUG

October 22, 2020



Welcome my name is Tom Wasik. I am the chief product owner for JES2 here at IBM. Today I want to talk to you about an enhancement that IBM has made to the product called JES2 SPOOL encryption. As this name implies, this function allows you to encrypt data that you write to your spool data set. But it's much more than just encryption. This enhancement changes the way that data is stored on the SPOOL. This enables other functions to be implemented such as compression. Compression will help you reduce the amount of data that you write to SPOOL saving you not only space, but also improving performance by reducing the number of I/Os. In this presentation I'll go into how to set up SPOOL encryption if this is what you need, and also will talk about compression and how to set up compression on your system.

JES2 SPOOL will never be the same

Data protection using the power of pervasive encryption to keep prying eyes from your instream data and SYSOUT

Compression using the integrated features of the z15 processor to save you significant space

No changes needed to your batch jobs or applications accessing SPOOL



(c) Copyright IBM Corporation, 2020

2

JES2 has been doing SPOOL for 50+ years. In all that time, the way data was stored on spool has not significantly changed. The most exciting thing that was done was blank truncation. Until now.

Pervasive encryption has come to JES data sets on SPOOL. This makes the data customers stored on SPOOL unreadable to anyone that does not have access to the appropriate encryption key. You use your security product to controls access to the keys and identifies what data to encrypt and what keys to used. This can be applied to each application SYSOUT or instream data sets securing data at rest on your SPOOL volume.

In addition, if you are running z15 hardware, JES2 will compress the data before encrypting it. This reduces the amount of data being encrypted and thus improves the performance. Depending on the nature of the data the savings can be significant. Standard character data is often compressed to less than 20% of its original size. This saves not only on the overhead of encryption, but overall overhead of reading and writing data to SPOOL, not to mention space savings storing the data. And for SYSOUT data set, you can get the savings of compression without doing encryption.

This was all done transparently. There are no changes needed to any application that writes or read data from SPOOL. In fact, the only way you can tell that a data set is compressed or encrypted is to check out the data set statistics in a product like SDSF.

SPOOL encryption

Data protection and compliance issues are becoming business imperatives

SPOOL data could contain sensitive data that falls under various compliance regulations

Encryption protects application data at rest on the JES2 SPOOL

Extends Data Set encryption paradigm to job's instream data and SYSOUT data



Designed to take advantage of the processing power of the z14

By managing key label access you can control who can access data across both regular data sets and spool data sets in a consistent manner

Controlling who can read (decrypt) the data is separate from who can control what happens to the data set

(c) Copyright IBM Corporation, 2020

3

Data breaches are becoming more and more of a problem in today's world. That's one of the reasons why z/OS developed pervasive encryption. It secures data without the need to alter the applications that create or use the data.

Up until now, pervasive encryption did not apply to data that applications wrote to SPOOL even though that data can contain sensitive information, whether it be instream data or SYSOUT. SPOOL encryption extends pervasive encryption to these SPOOL data sets. With it, you can encrypt data at rest while it's sitting on your SPOOL. Like pervasive encryption, it takes advantage of the processing power of the z14 processor to perform the encryption.

SPOOL encryption truly is an extension of pervasive encryption. It uses the same paradigms that were used to encrypt DF/SMS datasets. This allows you to have a consistent scheme for managing access to key labels across not only DF/SMS data sets but also your data on SPOOL. By controlling who can access the key label you can control who can read (decrypt) the data. This can be done separately from who can manage the data sets.

Encrypting SPOOL data

Protection occurs on a JES2 spool data set level

Each SPOOL data set can have its own key label (and encryption key)

Usage of key labels is controlled through your security product

Applications access to data on SPOOL is unchanged

However, users must have access to the key label to decrypt the data

Uses same key labels concept as Data Set encryption to encrypt data before writing it

Accessing (decrypting) the data is done using the same key label

(c) Copyright IBM Corporation, 2020

4

SPOOL encryption provides the ability to protect individual JES2 data sets on SPOOL with different key labels and thus different encryption key. These are the same encryption key labels used by data set encryption to protect DF/SMS data sets. Access to the keys is controlled by the security product using the CSFKEYS class (as is done for data set encryption). Conditional access can be accomplished using `WHEN(CRITERIA(SMS(DSENCRYPTION)))`.

JES2 keeps track of the key labels in its internal control blocks associated with each data set. When the data set is read JES2 uses the saved key label to decrypt the data set. Applications that want to access a data set must have access to the key label in order to decrypt the data set. Other than this additional access requirement, applications do not need to change if SPOOL encryption is active.

Encryption keys


The required key labels are defined in the ICSF Cryptographic Key Data Set (CKDS)

Actual keys are associated with the key labels

Same type of key label setup as Data Set encryption

- Labels are up to 64 characters in length
- AES-256 bit encryption data key associated with the key label
- Set as a protected key in CSFKEYS
- Uses XTS encryption mode

(c) Copyright IBM Corporation, 2020



Loss of a key implies loss of access to the data the key encrypts

Key and key label management is critical for a robust security strategy

A consistent strategy should be developed across data set and spool encryption

However, JES2 data is generally short lived (days perhaps weeks) so processes like re-keying do not apply

5

Key labels must be defined in the ICSF cryptographic key data set (CKDS) in order to be used by SPOOL encryption. The actual keys are associated with the key label. Key labels can be up to 64 characters in length. The key label must be associated with an AES 256 bit keys and should be protected by the security product using the CSFKEYS class.

As with any encryption scheme the loss of a key or access to a key will prevent access to the data the key encrypts. Because of this, a robust key label management system must be used to ensure consistent access to the data. It is worth noting that data stored by JES2 differs from data stored using pervasive encryption in that JES2 data is generally short lived, days perhaps week versus years or more for regular datasets. Because of this JES2 does not have a scheme for re-keying existing SPOOL datasets.

Specifying a key label



To encrypt data, associate a key label with a data set

RACF JESJOBS profile `ENCRYPT.nodename.userid.jobname.dsname`

KEYLABEL field in the JES segment

Access list/UACC is not used

`DSKEYLBL=` keyword on the `DD *`, `DATA`, and `SYSOUT=` JCL statements, `TSO ALLOC`, or `DYNALLOC`

Job owner or submitter needs `READ` access to `FACILITY` class profile

`JES.ENCRYPT.OWNER`

or

`JES.ENCRYPT.SUBMITTER`

NJE `SYSOUT` receiver always uses the JESJOBS profile to assign key label to the `SYSOUT`

(c) Copyright IBM Corporation, 2020

6

To encrypt data on the jazz two spool you must associate the key label with the data set. There are two ways to do this. The first method uses the `KEYLABEL` field in the JES segment of a JESJOBS profile to assign a key label. The format of the JESJOBS profile is:

```
ENCRYPT.nodename.userid.jobname.dsname
```

Where the fields in the name are:

- `nodename` - NJE node name of the current JES2 JESplex
- `userid` - Owinging userid of the job. If this is an instream data set during `INPUT` phase processing, the `USER=` from the `JOB` card is used if specified, otherwise the submitting userid is used
- `jobname` - Current job name (TSO user, started task name) associated with the job
- `dsname` - The one to eight character `DSNAME=dsn` specified on the `DD JCL` statement defining the data set

The UACC and access list of this JESJOBS profile are not used.

The second method to assign a key label is to use the `DSKEYLBL=` keyword on the `DD *`, `DATA` or `SYSOUT` JCL statement, specify `DSKEYLBL` on the `TSO ALLOC` command, or as a text unit on `DYNALLOC`. This `DSKEYLBL` specification will override what was specified via the JESJOBS profile. However, to specify `DSKEYLBL`, a `FACILITY` class profile must pass.

The profile is either `JES.ENCRYPT.OWNER` or `JES.ENCRYPT.SUBMITTER` depending on when the data set being encrypted. The check is for `READ` access to a non-generic profile. For instream data set during `INPUT` phase, the `JES.ENCRYPT.SUBMITTER` profile is used. For all other cases (including instream data set in `PROCs` and `INCLUDEs`), the `JES.ENCRYPT.OWNER` profile is checked.

NJE `SYSOUT` receivers will perform a JESJOBS profile look up to determine if each the data set received should be encrypted. NJE job receivers will do the same checks that were done for normal batch jobs.

Other security checks

The following security check must pass to create or read data that is encrypted

Users of key labels must have read access to the label in the CSFKEYS class

Conditional access is supported using WHEN(CRITERIA(SMS(DSENC RYPTION)))

Access is needed to the ICSF CKDS Key Record Read2 (CSNBKRR2) service

This is controlled by read access to the CSFKRR2 entity in the CSFSERV class

CSFSERV access is not needed if ICSF is configured with CHECKAUTH(NO)

Security checks are not performed for system access to encrypt or decrypt data

- NJE/OFFLOAD job transmitters, SYSOUT transmitters, and SYSOUT receivers
- Local (non-FSS) and RJE printers and punches

(c) Copyright IBM Corporation, 2020

7

In addition to the security checks to access the key label in the CSFKEYS class applications wanting to create or access encrypted data sets must also have access to the ICSF services to do so . In particular, the CSNBKRR2 service access is controlled by the CSFKRR2 entity in the CSFSERV class. Note that if ICSF is configured with CHECKAUTH(NO) then the CSFSERV check is not done.

When JES2 is accessing SPOOL data sets for encryption or decryption it does not check access to the CSFKEYS class. This includes:

- NJE or SPOOL offload job transmitters
- SYSOUT transmitters
- SYSOUT receivers
- Local (non-FSS) printers and punches
- RJE printers and punches

Since SYSOUT receivers use JESJOBS profiles to assign key labels to incoming data sets without checking the CSDKEYS class, it is possible that the owner of the SYSOUT data set may not have access to the key label needed to decrypt the data.

Encryption example

```
//PRINTJOB JOB '','Test print job',MSGLEVEL=(1,1),NOTIFY=IBMUSER,
//          USER=IBMUSER,MSGCLASS=A,PRTY=3,CLASS=A
//STEP1 EXEC PGM=IEBDDG
//SYSPRINT DD SYSOUT=*,DSKEYLBL=ALPHA
//JOBOUT DD SYSOUT=*,DSN=4SECOUT
//SYSIN DD *,DSKEYLBL=INTERNAL
DSD OUTPUT=(JOBOUT)
FD NAME=A,STARTLOC=01,LENGTH=2, X
   PICTURE=2,'1'
FD NAME=B,STARTLOC=03,LENGTH=8,FORMAT=ZD,INDEX=1 X
FD NAME=C,STARTLOC=11,LENGTH=31, X
   PICTURE=31,'GENERATE MANY LINES'
FD NAME=D,STARTLOC=44,LENGTH=30, X
   PICTURE=30,'TEST CASE NAME = IBMUSERH'
CREATE FILL=' ',NAME=(A,B,C,D),QUANTITY=100
END
XX

RDEFINE JESJOBS ENCRYPT.ROCH.IBMUSER.PRINTJOB.SECOUT UACC(READ)
JES(KEYLABEL(OUTPUT))
```

(c) Copyright IBM Corporation, 2020

SYSPRINT DD is encrypted with key label ALPHA

- Explicitly specified DSKEYLBL

JOBOUT DD is encrypted with key label OUTPUT

- JESJOBS ENCRYPT profile

SYSIN DD is encrypted with key label INTERNAL

- Explicitly specified DSKEYLBL

8

This is an example of JCL that illustrates assigning key labels to various datasets.

The SYSPRINT DD is explicitly assigned the key label ALPHA using DSKEYLBL=. Since this is a SYSOUT dataset the owning userid IBMUSER must have READ access to the FACILITY class profile JES. ENCRYPT. OWNER.

The JOBOUT DD is assigned the key label OUTPUT by the JESJOBS ENCRYPT profile listed.

Finally, the SYSIN DD is explicitly assigned the key label OUTPUT using DSKEYLBL=. Since this is an instream data set processed during INPUT phase, the submitting userid must have READ access to the FACILITY class profile JES. ENCRYPT. SUBMITTER.

SPOOL data set compression

Significantly reduce the size of most SPOOL data sets

Stores more data on the same SPOOL volumes

Less data means less I/O to write and later read data

Less I/O means less CPU to write and later read the data

Designed to take advantage of the Integrated Accelerator for zEDC on the z15



Included feature of the z15 processor

No additional licensing fees

(c) Copyright IBM Corporation, 2020

9

In addition to encryption JES2 also supports data set compression for SPOOL data set. Compression takes advantage of the integrated accelerator for zEDC compression on the z15 processor. Using this feature to compress spool data significantly reduces the size of the data, allowing you to store more data on the same spool volumes using less I/O to both write and later read the data, and less CPU to perform the I/Os. The integrated accelerator for zEDC is an included feature on the z15 processor. There is no additional licensing fees to use this feature.

Setting up Compression

COMPRESS=YES on OUTCLASS statement

Always attempted for encrypted data set

Not available for instream data sets that are not encrypted

Compression ratios are VERY data dependent

Long LRECL data sets (CPDS) show largest benefits



CPU costs are not noticeable in most environments

- In single digit percent increase

I/O reductions is significant

Sample compression ratios

Assembler listing	517,887	82,696	84.03%
Test tool log	9,024,000	648,519	92.81%

Ratios available in SDSF JDS display

(c) Copyright IBM Corporation, 2020 10

To set up compression all that is needed is to set the COMPRESS=YES keyword on the corresponding OUTCLASS statement. If a data set is being encrypted JES2 will always attempt to compress the data set before encryption. Compression can be requested for any SYSOUT data set even if it is not being encrypted, however instream datasets will only be compressed if they are being encrypted.

CPU costs in testing done in house did not significantly change when using compression. In general, what was seen is a slight single digit percent increase in CPU with compression and encryption active. However the number of I/Os that occurred was significantly reduced.

Compression ratios vary greatly depending upon the data being compressed. Data that is already compressed will have a lower compression ratio than normal character data. In testing we found that character data would compress to less than 20% of its original size. Some of the best benefits from an overall performance standpoint we're seeing with long LRECL data sets such as CPDS data sets.

Compression ratios can be found in the JDS display in SDSF and are available in the extended status SSI.

Encryption and compression notes

JES system data sets not eligible

- JESJCLIN, JESMSGLG, JESJCL, JESYSMSG, \$INTTEXT, \$JOURNAL, \$SWABLKS, EVENTLOG

System jobs not eligible

- SYSLOG, Remote messages, EDS notify messages, \$TRCLOG

Data sets that use GET/PUT update not eligible

- Generic tracker to detect if you have any



(c) Copyright IBM Corporation, 2020

11

The spool encryption and compression functions of jazz two I designed to protect application data. JES2 system datasets, JESJCLIN, JESMSGLG, JESJCL, JESYSMSG, \$INTTEXT, \$JOURNAL, \$SWABLKS, and EVENTLOG) are currently not eligible for either function. Additionally system jobs, SYSLOG, Remote messages, EDS notify messages, and \$TRCLOG are not eligible for these functions.

One final note, JES2 supports a method of writing SYSOUT called PUT update. This allows an application to write a record to a SYSOUT data set And later point to that record an updated it. Unfortunately if the data is compressed changing the data could change the compressed record size. This updated size may no longer fit in the space allotted for the original record (a requirement for PUT update). Because of this put update is not allowed for compressed records. A generic tracker entry will be created if they put update is detected on your system. If compression or encryption is requested for these datasets the GET update will fail.

Activation

Function shipped with OA57466 on JES2
z/OS 2.4

Not active by default

Can activate once all members completed
migration to z/OS 2.4

- **Once activated, pre-2.4 members cannot join the JESplex**

`$T SPOOLDEF,ADVANCED_FORMAT=ENABLED`

`$D SPOOLDEF`
displays current
setting



Can set `ADVANCED_FORMAT` to `DISABLED`
`ADVF=` is shorthand for `ADVANCED_FORMAT=`

Stop new data set compression and encryption

No effect on existing data sets

Does **NOT** allow pre-2.4 members in the JESplex

(c) Copyright IBM Corporation, 2020

12

The SPOOL encryption function was activated with APAR OA57466 which shipped against z/OS 2.4. Both spool encryption and compression required that records written to SPOOL are in a new enhanced format. The APAR ships with the advanced format not active by default. Before activating the new data format all members must have completed their migration to z/OS 2.4. Once this function is activated, pre z/OS 2.4 members will not be allowed to join the MAS. The command to activate advanced format is `$T SPOOLDEF,ADVANCED_FORMAT=ENABLED`.

Advanced format can be disabled by using the command `$T SPOOLDEF,ADVANCED_FORMAT=DISABLE`. Disabling advanced format does not affect existing data sets on SPOOL. It only prevents the encryption and compression of new datasets while the function is disabled. Even if advanced format is disabled pre-z/OS 2.4 members are still not allowed to join the JESplex.

`ADVANCED_FORMAT=` can be abbreviated `ADVF=`. To display the current setting, use the `$D SPOOLDEF` command.

Logging

No data set level SMF records for JES2 data sets

Section for compression and encryption in SMF 26 record at the job level



Compression statistics (job total compressed and uncompressed bytes)



Count of compressed and encrypted data sets

SDSF (via SSI) reports on individual data sets on the JDS panel

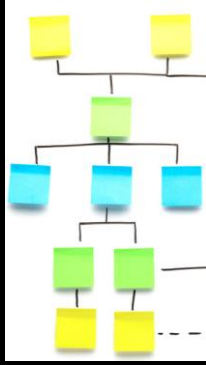
(c) Copyright IBM Corporation, 2020



13

JES2 does not create SMF records for SPOOL data sets. The only logging that compression or encryption are in use is in the SMF 26 record created when a job is purged. This record has been updated with a new section that contains a count of the number of datasets for the job that were compressed and the number that were encrypted. In addition it has accumulated byte counts for compressed datasets, with the bytes prior to compression and after compression. The other way to see if an individual data set has been compressed or encrypted is to use products like SDSF. The SSI used by SDSF to obtain data set information has new fields indicating a data set is encrypted with the key label and an indicator the data set was compressed with the pre and post compression byte counts.

Journey to SPOOL encryption



(c) Copyright IBM Corporation, 2020

14

1. Migrate JESplex to z/OS 2.4 and install all needed service (OA57466 et al)
2. Once fall back to an earlier release is no longer needed, activate advanced format (\$T SPOOLDEF,ADVANCED_FORMAT=ENABLED)
 - Enabling advanced format has no effect if no profiles exist
3. Develop (or extend existing) key label scheme for SPOOL encryption
 - Ensure needed RACF access to CSFKEYS and CSFSERV class profiles
4. Decide what data sets to encrypt and how you want to specify key labels
 - If using JCL, set up JES.ENCRYPT.OWNER and SUBMITTER profiles
5. Test encryption using JCL to ensure that all permissions are set up
6. Once ready, set up JESJOBS RACF policies to assign key labels as need

So let's summarize enabling SPOOL encryption on your system. The first step is to migrate all members of the JESplex to Z OS 2.4 and ensure all needed service is installed (OA57466 and any later APARs) on each system. Once you know that you will not need to fall back to an earlier release you can activate advanced format using the \$T SPOOLDEF,ADVANCED_FORMAT=ENABLED command. Remember enabling advanced format has no effect if there are no RACF profiles for encryption set up. At this point consider developing or extending the existing key label management scheme for SPOOL encryption. As part of developing the key label scheme you need to identify what user IDs will be doing encryption and decryption and ensure that those user IDs have access to the appropriate profiles in the CSFKEYS and CSFSERVE classes. At this point you need to decide what data sets you want to encrypt. You will probably want use JESJOBS profiles to assign keys to the data set you want to encrypt. However before doing this you may want to test using the JCL DSKEYLBL= on the DD cards to ensure that all your authorizations were set up correctly. Once you have completed testing with DSKEYLBL you can start defining the appropriate JESJOBS profiles.

The easiest way to verify the data sets you wanted to be encrypted were indeed encrypted is to use the SDSF JDS panel 2 display the encryption state and key label used for the data set. Long term you can use SMF 26 to track that jobs that should have encrypted data sets indeed are using encryption

Something completely different NJE

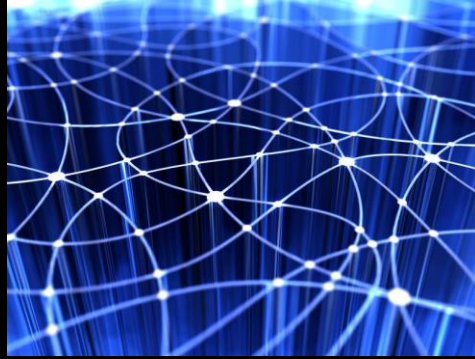
NJE security is based on clear text data in NJE headers

Based on assumption that the network is secure

Only properly authenticated NJE nodes can inject objects into the network

THIS IS NJE so this is the JES guys

This is not TCP/IP so not the network guys



(c) Copyright IBM Corporation, 2020

15

Now for something completely different, lets talk NJE. NJE sits on top of the various networking protocols (TCP/IP, SNA, BSC) and sends jobs and SYSOUT between nodes. Security in NJE is based on data that is passed in the NJE headers associated with data object. These contain information about where the object came from, the user, group, and SECLABEL associated with the object at the origin node. But the bottom line is that the assumption is the only way something can get onto an NJE network is a properly authenticated NJE node put it there.

Note we are talking NJE which is a layer on top of a protocol like TCP/IP. Typically, NJE is the domain of the JES guys and they deal with the concepts. The network guy deal with the actual connections that NJE uses. I say this since many of the concepts and terms of NJE also exist in the networking world (like SUBNETs) but have a very different meaning and usage.

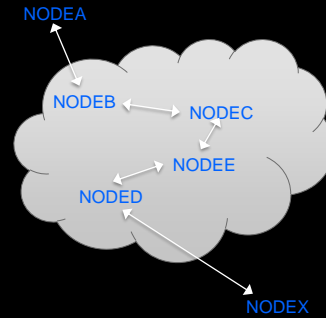
High level NJE concepts

Store and forward

NJE uses a concept of store and forward

Data moving from NODEA to NODEX can travel through intermediate nodes

At the intermediate nodes, the data is stored locally until it is time to move it to the next node in the path



(c) Copyright IBM Corporation, 2020

16

One of the things NJE does is something called store and forward. Sending something from NODEA to NODEX can involve sending it to intermediate nodes on the way to the destination. The data is received and stored on the intermediate node until it can be sent to the next node. Think of it this way, if you are taking a subway to a destination, there may be stations you pass through, but if you are lucky, you can take the one train to your destination. However, if you are not lucky you may have to transfer to another train or perhaps even to a bus to get to your destination. NJE is like this. Store and forward is like having to transferring trains. Due to the infrastructure it may be necessary, but it leaves you exposed to the outside world (OK, in my world, the subway car is safety and standing in a station is unsafe. Not a perfect analogy).

Strange new node in the network

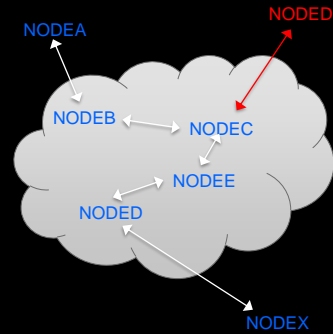
A new node has joined our network

Strange, it has the same name as another node in the network

This is an imposter, allowed in through a single unprotected node (perhaps it is that zVM guy)

So what harm can this imposter do?

Ever GOOGLE "NJE python"?



(c) Copyright IBM Corporation, 2020

17

So one day you look at your NJE cloud and notice an unexpected red node sitting out there. And it has the same name as another node in your network. This is an imposter. Perhaps a pen tester? Perhaps something worse? He got in through a node that did not set up strong sign on security for signing on NJE nodes. Perhaps it is that VM guy that says he does not need security because they are a test environment. But they do hook into your NJE network. But what is this imposter node? Someone with a mainframe? Perhaps running on mainframe emulator on a PC? Or something simpler. Ever GOOGLE "NJE python"? There are python programs to do NJE. They can run on any PC that is in your TCP/IP network. They could run in python on your mainframe.

But what harm can this imposter do?

Imposter in the network

So our imposter has placed a job on the network

It claims to be from NODEX

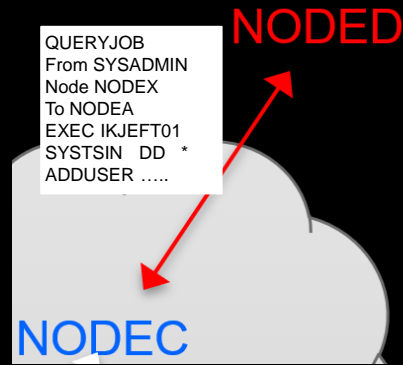
It is going to NODEA

Can NODEA detect that this is from the imposter?

NO

Does NODEA trust jobs from NODEX?

?



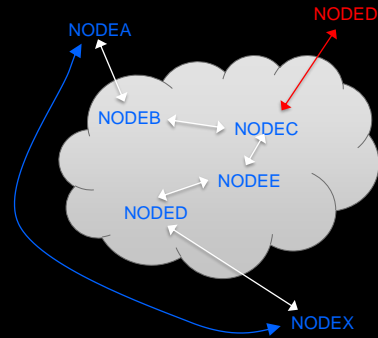
(c) Copyright IBM Corporation, 2020

18

So here is a job that the imposter on the red NODED sends to NODEC. It claims to come from the SYSADMIN user on NODEX and is going to NODEA. The job runs some RACF commands in TSO batch. It could have easily done other things (FTP?). But the question is, can NODEA detect that this job is not really from NODEX? In general, NO it cannot. The rules of NJE store and forward do not allow this. So the real question is, does NODEA trust jobs from NODEX?

Routing based NJE security

Routing based security adds path to destination to the mix
Jobs that come from unexpected places are marked dubious
Dubious jobs are not to be trusted
Connect NODEA directly to NODEX via a TCP/IP connection
On NODEA define NODEX as DIRECT=YES and on
NODEX define NODEA as DIRECT=YES
Now NODEA and NODEX will only send data over the blue
direct connection
Anything arriving on NODEA that does not come on the
direct connection is dubious



(c) Copyright IBM Corporation, 2020

19

There are some things that can help the situation. The one that can be of the most help is routing based security. This is using information about how an object got to the destination node. There are a number of checks that can be made but the most obvious is to look at the node that gave me the object. The easiest way to think of it is to make connection directly to any node you want to trust. And then use the DIRECT=YES keywords to define the nodes (NODEA defines NODEX as DIRECT=YES and NODEX defines NODEA as DIRECT=YES). This tells JES2 not to use store and forward to get objects to the other node. If store and forward is not used, then any object claiming to be from NODEX that arrives via another node is unexpected and the object is marked dubious. Dubious jobs are not trusted.

Advanced routing based security (NJE SUBNETs)

NJE subnets are a way of defining a set of NJE nodes

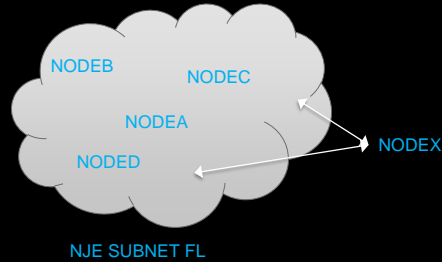
Typically an NJE subnet is a campus or perhaps a company

Rule is when sending an object to a node in the same NJE subnet as you, do not send it outside the NJE subnet

So an object going from NODED to NODEC would never be sent to NODEX

So if NODEX sends an object claiming to be from NODEA, NODEB, NODEC, or NODED, then something is wrong

This is another case when a job is marked dubious



(c) Copyright IBM Corporation, 2020

20

In NJE we have a concept of SUBNETs. THIS IS NOT WHAT THE NETWORK GUYS THINKS OF WHEN YOU SAY SUBNET. These are NJE subnet.

A subnet is a collection of nodes that have 2 properties that are of interest

- If a node is not the local SUBNET, then if path determination can reach one node in a subnet, then it can reach all nodes in the subnet through that node.
- When discovering the path to a node, a path out of a subnet can never be used to return into the subnet. What that means is that an object on a node inside a subnet, that is destined for another node inside the subnet, can never be sent outside the subnet to get to its destination.

A node can only be in at most one subnet.

It is the last property that is interesting to this discussion (what originates in a subnet that is destined for a node in the subnet, never leaves the subnet). And we care about that for the local subnet (ie the subnet the local node is in, if any).

A job is dubious if...

Arrives with origin being local node and destination is ...

- Local node
- A direct node

Destination is local node, origin is a direct node, but arrived via some other node

Job arrived via a node not in the local subnet but the origin and destination is within the local subnet

Marked dubious before it arrived

Once marked dubious, setting is forwarded with job

Special cases:

- Origin is unknown node
- Destination is node it arrived via
- Origin node in NJE header does not match RACF token

These are the cases that trigger a job to be marked dubious. When a job arriving is marked dubious, the following message is issued:

`$HASP541 Routing for job named jobname received on device (adjnode)`

from userid at node is inconsistent with information in the NJE header

- Header indicates inconsistency detected on store and forward node
- Job indicates it originates from local node
- Job arrived from unknown origin node
- Job indicates it originates from within the local subnet but arrived from outside the local subnet
- Job from direct node arrived via store and forward
- Job from adjacent node specifies to execute on that adjacent node
- Node name in SAF token does not match NJE job header

Dubious Jobs Externals

VFYPATH= on NODE for local and DIRECT=YES nodes

- Activates dubious check for nodes

VERIFY_SUBNET= (or VFYSUBNET=) on NJEDEF

- Activates subnet dubious check

PRECHECK= on NJEDEF

- Activates RACF non-NJE pre check for dubious jobs

DUBIOUS= on \$D JOB/JOBGROUP commands

- Displays dubious setting for job (pre and post execution)



(c) Copyright IBM Corporation, 2020

22

The following options control this processing:

VFYPATH= on the NODE statement associated with an arriving job's origin node affects how that job is processed. A setting of YES can only be specified for the local node and nodes defined as DIRECT=YES. When set to YES, a job will be marked dubious if:

- For a NODE that also specifies DIRECT=YES, the job arrives from some other adjacent node (jobs must arrive from the directly connected node).
- For the NODE statement for the local node, the job claims to originate at the local node and indicates it will execute on the local node.
- For the NODE statement for the local node, the job claims to originate at the local node and indicates it will execute on a directly connected node.

VERIFY_SUBNET= (or VFYSUBNET=) on the NJEDEF statement will mark as dubious jobs that claim to be from a node within the local subnet and indicate they will execute on a node within the local subnet, but arrived from an adjacent node that is outside the local subnet. The local subnet is the SUBNET= value specified on the NODE statement for the local node.

PRECHECK= on the NJEDEF statement controls whether or not jobs that are marked dubious will be pre-validated to ensure they are allowed to run without depending on the settings in the NODES class (as if the NODES class profile specifies READ). This is an extra call to the security product before the normal verification of the job.

NJE Security Health Check

Checks the following for trusted nodes:

- Ensure directly connected or in local subnet
- Ensure NJE password protection
- Ensure using TLS
- Ensure VFYPATH is set

Checks for the local node:

- Ensure VFYPATH set for the local node
- Ensure if local node is in a subnet, VERIFY_SUBNET set
- Ensure that PRECHECK is set

Checks for non-trusted nodes:

- If node can be connected, ensure password protection

(c) Copyright IBM Corporation, 2020



An JES NJE security health check ensures that trusted nodes are configured using best practices. There is one instance of the health check per active JES subsystem. The name of the check reflects the subsystem it is running against. If it is the primary subsystem, the name is JES_NJE_SECURITY. If it is a secondary subsystem the subsystem name is added to the check name (so JES_NJE_SECURITY_JESA). The output of the check is a list of the trusted nodes and any problems with how they are defined. The check results in medium and high severity exceptions.

To perform its functions, the check must know what nodes are considered trusted. This information is only known by the security product. An interface was defined that allows the check to pass a list of nodes into the security product and get back a list of which of those nodes are trusted. This list is returned by the exec and used to perform configuration testing. RACF shipped an exec to perform this function.

The check is set to run every 6 hours (3 if there is a problem detected).

There are 3 sections to this health check:

- Trusted nodes checks (including a list of trusted nodes)
- Local node check
- Non-trusted nodes that can be or are connected

Each section returns a list of nodes and the exceptions found, followed by a list of exception messages based on what exceptions are found.

Pervasive encryption survey

Please take a minutes to complete
the IBM Z pervasive encryption
survey at

[Pervasive Encryption Survey](#)

Thank You!

(c) Copyright IBM Corporation, 2020



IBM has some questions about how you are using or are planning to use pervasive encryption. Please take a minute to answer a few simple questions. Your responses are greatly appreciated

Thank you!

Questions?

Tom Wasik
JES Chief Product Owner

wasik@us.ibm.com
+1-507-253-3870
ibm.com

(c) Copyright IBM Corporation, 2020

25

Thank you for your time and attention. If you have any questions, please feel free to email me.

Notices and disclaimers

© 2020 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed “as is” without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.*

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

(c) Copyright IBM Corporation, 2020

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those

customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

26

Notices and disclaimers

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml