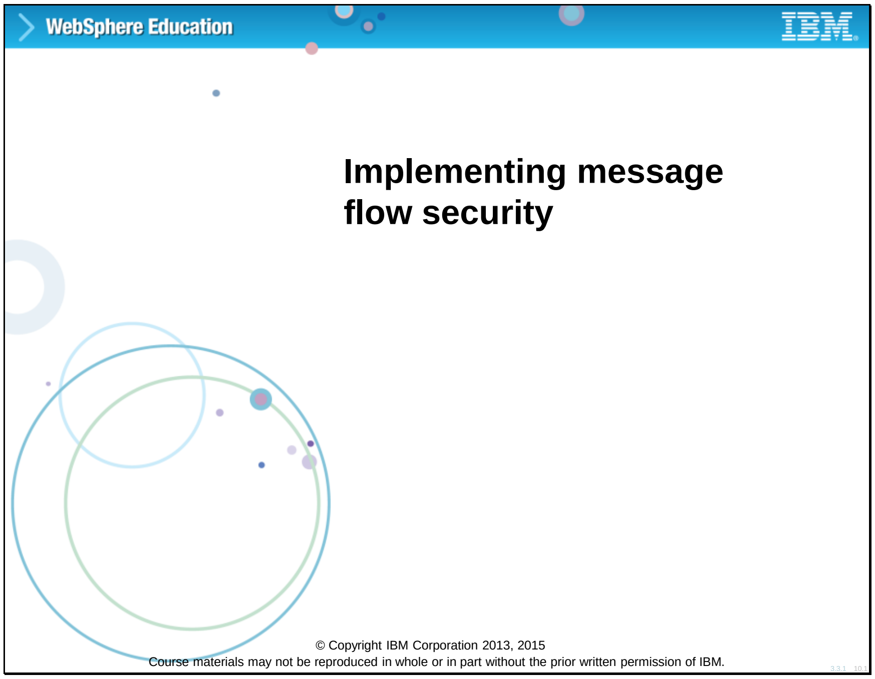
Slide 1



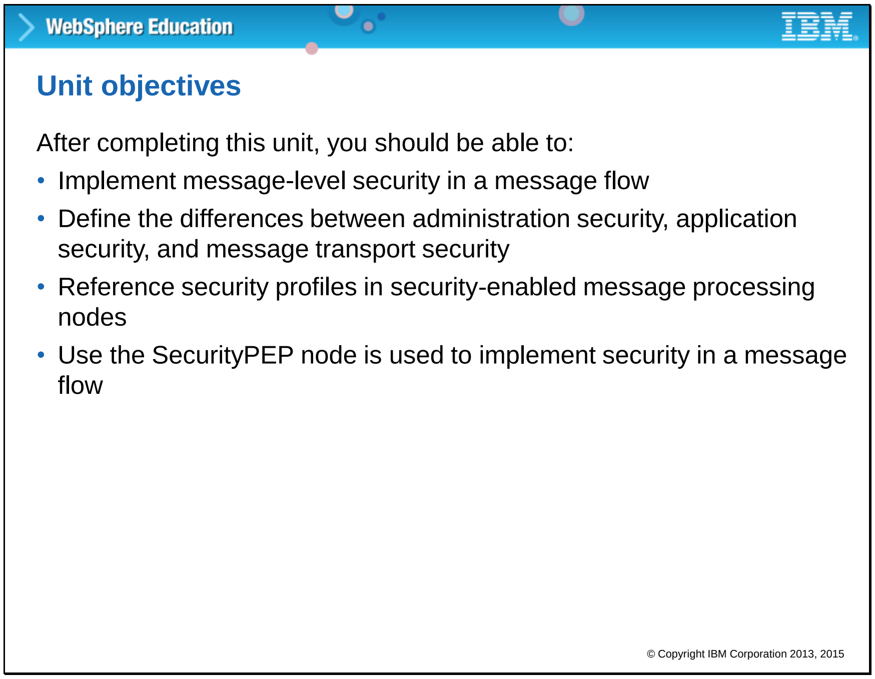
WebSphere Education

IBM

Implementing message flow security

3.3.1 10.1

Slide 2



WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Implement message-level security in a message flow
- Define the differences between administration security, application security, and message transport security
- Reference security profiles in security-enabled message processing nodes
- Use the SecurityPEP node is used to implement security in a message flow

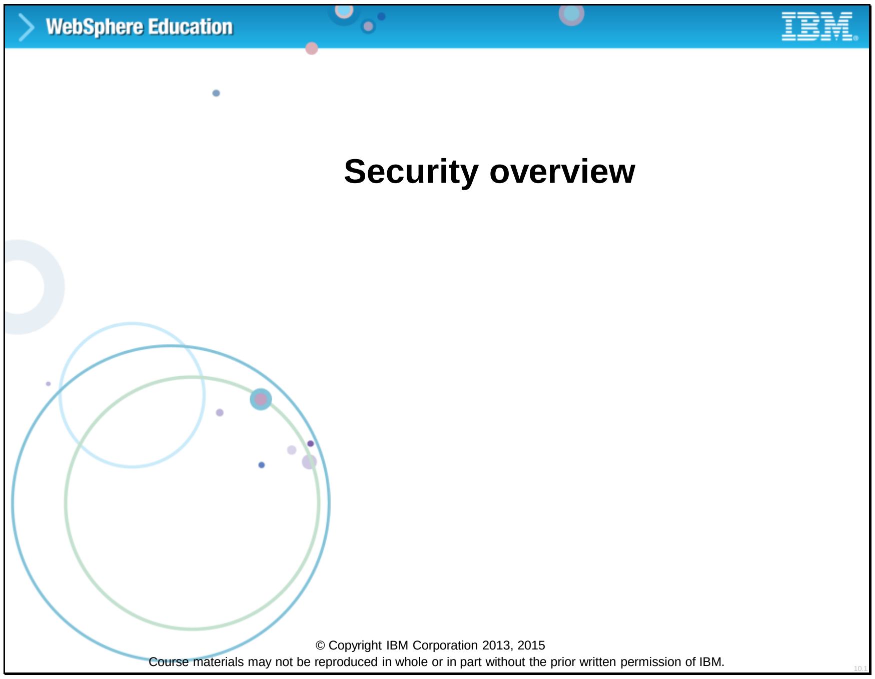
Unit objectives

This unit examines the different security considerations in IBM Integration Bus, and how to configure message flow security.

After completing this unit, you should be able to:

- Implement message-level security in a message flow
- Define the differences between administration security, application security, and message transport security
- Reference security profiles in security-enabled message processing nodes
- Use the SecurityPEP node is used to implement security in a message flow

Slide 4

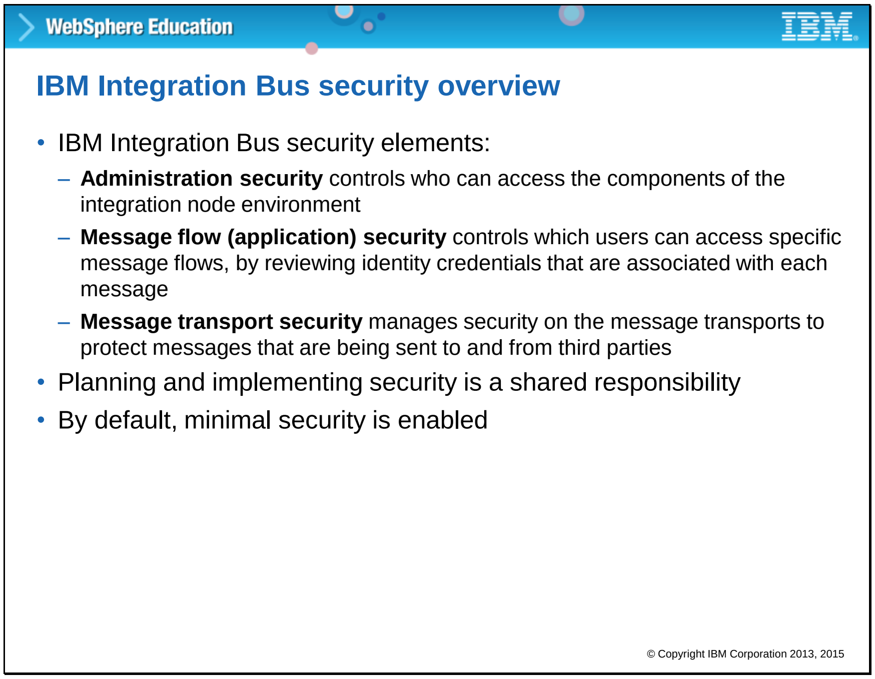


Security overview

Topic 1: Security overview

This topic provides a high-level overview of the security features in IBM Integration Bus V10 and defines the differences between administration, application, and message transport security.

Slide 4



WebSphere Education

IBM Integration Bus security overview

- IBM Integration Bus security elements:
  - **Administration security** controls who can access the components of the integration node environment
  - **Message flow (application) security** controls which users can access specific message flows, by reviewing identity credentials that are associated with each message
  - **Message transport security** manages security on the message transports to protect messages that are being sent to and from third parties
- Planning and implementing security is a shared responsibility
- By default, minimal security is enabled

IBM Integration Bus security overview

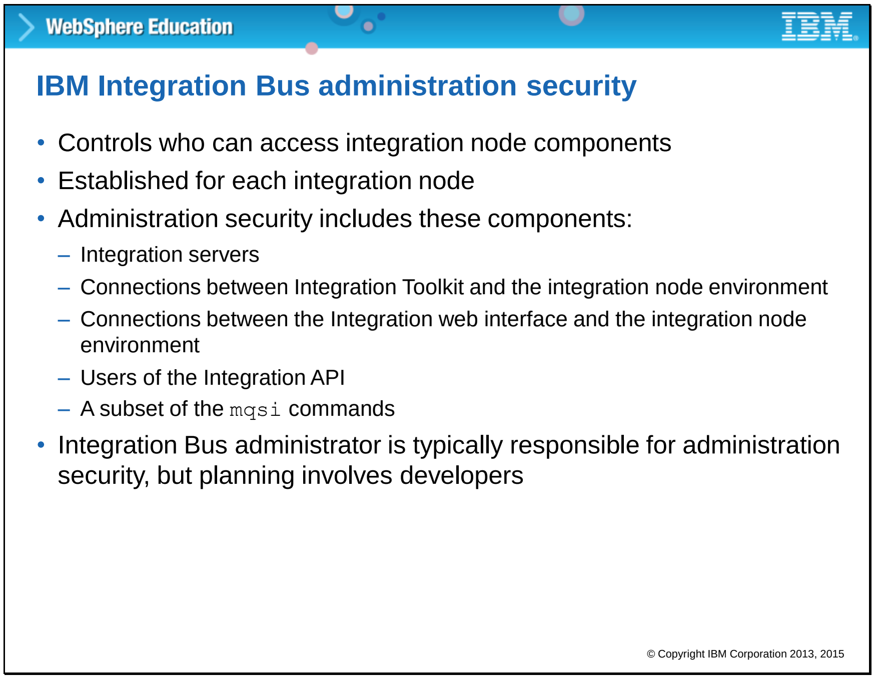
Integration Bus provides several elements for securing the Integration Bus environment:

- **Administration security** controls who can access the components of the integration node environment itself.
- **Message flow application security** controls which users and applications can access specific message flows, by reviewing identity credentials that are associated with each message. This unit concentrates on message flow application security after a brief review of the other security elements.
- **Message transport security** manages security on the message transport such as HTTP and FTP to protect messages that are being sent to and from third parties.

Planning for a secure Integration Bus environment is a joint effort and shared responsibility between the application developers and system administrators. It is important to understand how the various security elements of Integration Bus, MQ, and external messaging providers must work together to result in a coherent security framework.

By default, only minimal security is enabled during Integration Bus installation. In the exercise environment for this course, you run as a privileged user in a non-secure environment. This unsecured environment is likely to be different than the environment you use in your job.

Slide 5



WebSphere Education

IBM Integration Bus administration security

- Controls who can access integration node components
- Established for each integration node
- Administration security includes these components:
  - Integration servers
  - Connections between Integration Toolkit and the integration node environment
  - Connections between the Integration web interface and the integration node environment
  - Users of the Integration API
  - A subset of the `mqsi` commands
- Integration Bus administrator is typically responsible for administration security, but planning involves developers

IBM Integration Bus administration security

Integration Bus administration security regulates who can access the components of an integration node and its environment. The Integration Bus administrator enables administration security, and then grants privileges to individuals or groups or users to allow them the appropriate level of authority to complete the needed tasks.

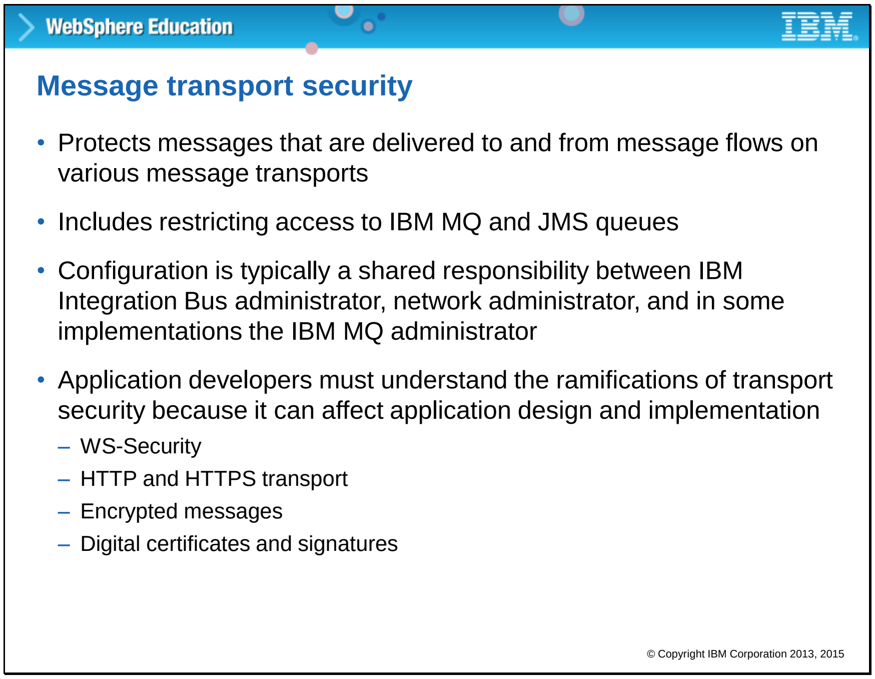
The components of an integration node environment include:

- Integration servers.
- The connections between the user of the Integration Toolkit and the integration node environment.
- The connections between the user of the Integration web user interface and the integration node environment.
- The users of the IBM Integration API.
- A subset of the MQSI commands, specifically those commands that you can use to review or alter the integration node configuration.

The Integration Bus administrator generally manages administration security. However, since administration security can affect what and how a developer must interact with the integration node environment, planning for administration security is often a shared responsibility during the planning phases.

The *IBM Integration Bus V10 System Administration* course explains planning and implementing administration security in detail.

Slide 6



WebSphere Education

Message transport security

- Protects messages that are delivered to and from message flows on various message transports
- Includes restricting access to IBM MQ and JMS queues
- Configuration is typically a shared responsibility between IBM Integration Bus administrator, network administrator, and in some implementations the IBM MQ administrator
- Application developers must understand the ramifications of transport security because it can affect application design and implementation
  - WS-Security
  - HTTP and HTTPS transport
  - Encrypted messages
  - Digital certificates and signatures

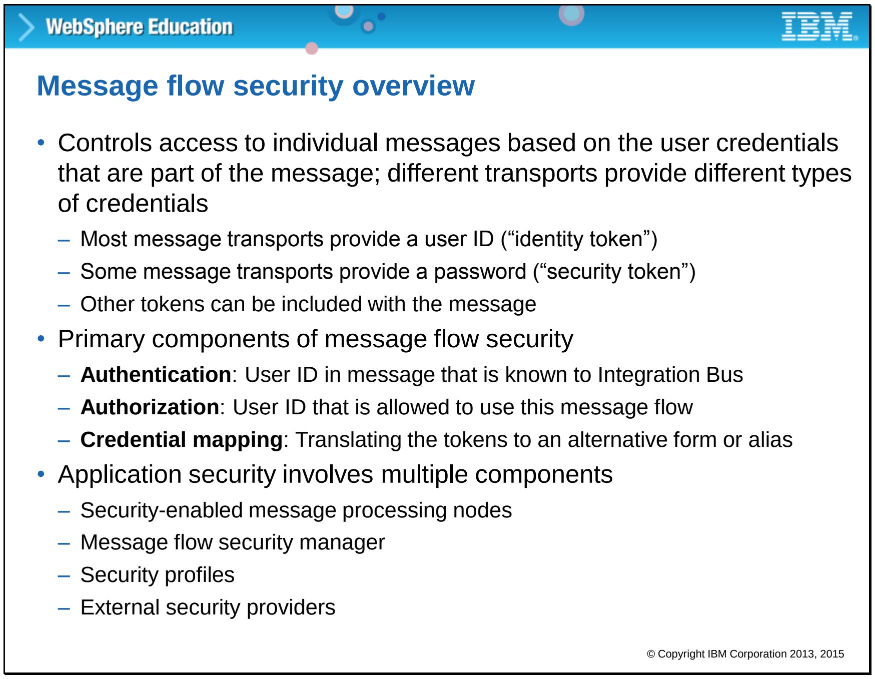
Message transport security

The messages that message flows use or produce can have their own protection requirements. It is normally the responsibility of the transport provider to secure those messages while in transit. This protection requirement can affect both the Integration Bus administrator and Integration Bus developers.

For example, it might be necessary for the administrator to configure the Integration Bus environment to match the message transport security requirements by securing MQ queues and JMS queues.

Or, it might be necessary for application developers to configure message flows to handle transport security features. For example, it might be necessary to configure HTTP nodes for HTTPS or SOAP nodes for WS-Security, or configure the message flow to manage encrypted messages, digital signatures, and other security elements that are related to message transport.

Slide 7



WebSphere Education

Message flow security overview

- Controls access to individual messages based on the user credentials that are part of the message; different transports provide different types of credentials
  - Most message transports provide a user ID (“identity token”)
  - Some message transports provide a password (“security token”)
  - Other tokens can be included with the message
- Primary components of message flow security
  - **Authentication**: User ID in message that is known to Integration Bus
  - **Authorization**: User ID that is allowed to use this message flow
  - **Credential mapping**: Translating the tokens to an alternative form or alias
- Application security involves multiple components
  - Security-enabled message processing nodes
  - Message flow security manager
  - Security profiles
  - External security providers

Message flow security overview

Integration Bus message flow application security primarily concerns the credentials that are associated with a message that is submitted for processing in a message flow. These credentials can include a user ID, password, or both. The credential information can be included within the message body, or as part of the message headers.

Message flow security determines whether the submitter of a message that the security token identifies is allowed to submit the message for processing. Determining this permission can involve a combination of steps in the following order:

1. **Authentication** establishes the identity of a user or entity and verifies that the identity is valid. For example, if the user ID for a message is **USER123**, authentication determines **that this user is** known to Integration Bus by verifying the identity token for the message. You use an *external security provider* to verify the token.

2. **Authorization** determines that the user or entity that is associated with the identity token is allowed to use the message flow.

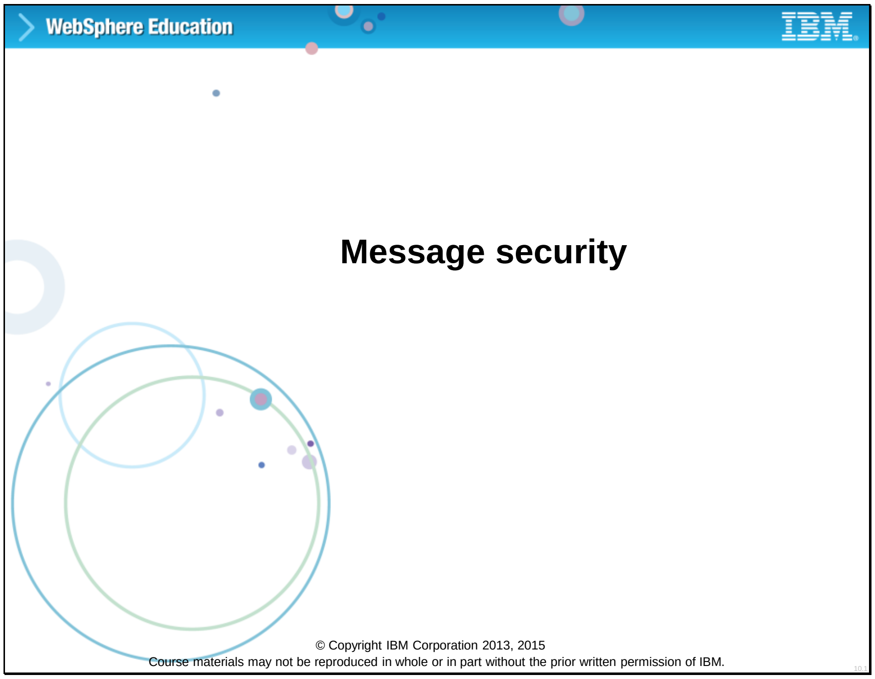
3. **Credential mapping** is the process of converting the credentials in the incoming security token to a different format or to an alternative identity. Identity mapping, sometimes called *identity federation*, can also map identity information in one realm to an equivalent identity in a different realm. For example, a user can be identified with user ID **USER123** in the Sales

application, but **FCST821** in the forecasting system. Credential mapping can convert the various identities that are associated with the same user.

In Integration Bus, the following components are important for application security:

- **Security-enabled message processing nodes** that can query the security information in a message and then act based on that information.
- The **Message Flow Security Manager** that works with the integration node to act on security information, as the security profiles dictate.
- **Security profiles** that control the actions to take when a security-enabled message processing node receives a message.
- When the security profiles so direct, the Message Flow Security Manager calls **external security providers** to authenticate, authorize, or map credentials.

Slide 8



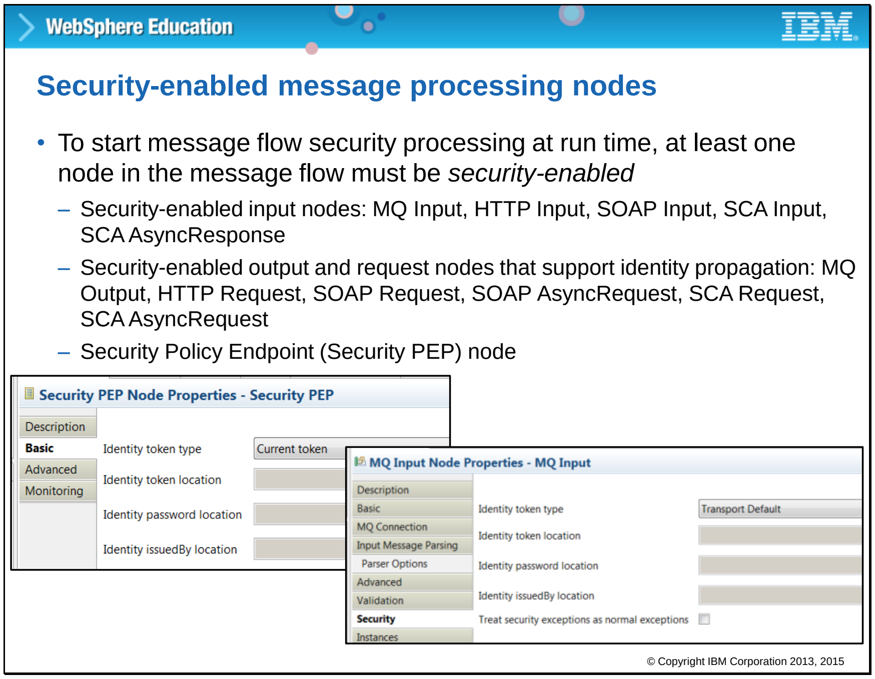
WebSphere Education

Message security

Topic 2: Message security

In this topic, you learn how to implement message-level security in a message flow, reference security profiles in security-enabled message processing nodes, and use the Security PEP node to implement security in a message flow.

Slide 10



Security-enabled message processing nodes

- To start message flow security processing at run time, at least one node in the message flow must be *security-enabled*
  - Security-enabled input nodes: MQ Input, HTTP Input, SOAP Input, SCA Input, SCA AsyncResponse
  - Security-enabled output and request nodes that support identity propagation: MQ Output, HTTP Request, SOAP Request, SOAP AsyncRequest, SCA Request, SCA AsyncRequest
  - Security Policy Endpoint (Security PEP) node

Security PEP Node Properties - Security PEP

Identity token type: Current token
Identity token location
Identity password location
Identity issuedBy location

MQ Input Node Properties - MQ Input

Identity token type: Transport Default
Identity token location
Identity password location
Identity issuedBy location
Treat security exceptions as normal exceptions □

Security-enabled message processing nodes

Several message processing nodes can be used in managing message flow security. These nodes are used in combination with the Message Flow Security Manager and security profiles to authenticate, authorize, and map credentials.

The message processing nodes on which you can enable security are:

- MQ, HTTP, SOAP, and SCA input nodes
- MQ output node
- HTTP, SOAP, and SOAP Async request nodes
- Security PEP node

To use message flow security, at least one message processing node in a message flow must be *security-enabled*. When a message encounters a security-enabled message processing node, security processing begins.

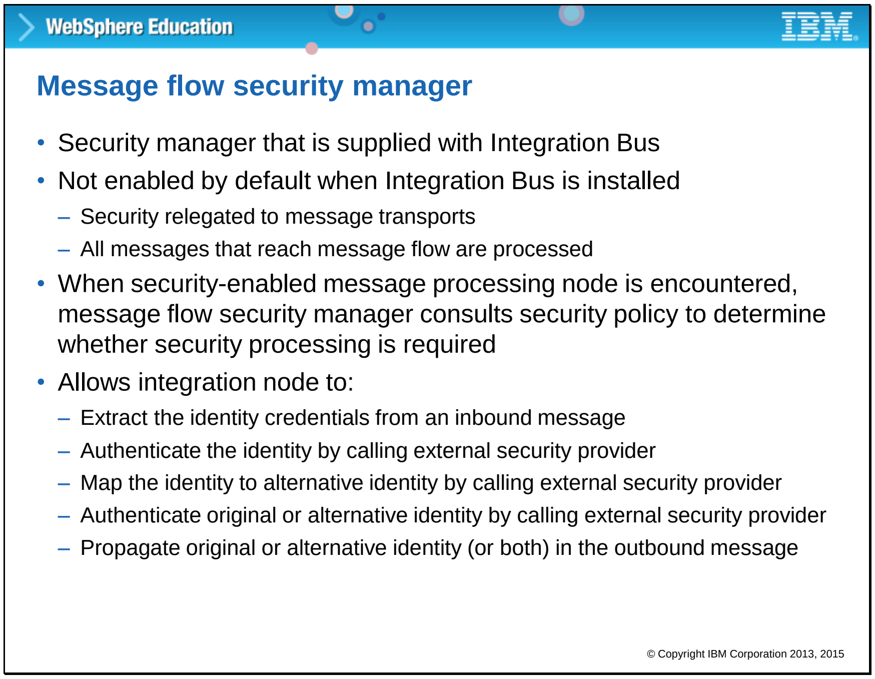
In general, the *input* node that starts the message flow is where you do the initial security checking (including authenticate, authorize, or credential mapping). The *output* nodes are used to propagate the security credentials through the message transport to the message destination. You can include one or more SecurityPEP nodes anywhere in the message flow that you want to use an external security provider to authenticate, authorize, or map credentials.

You enable security by configuring the node properties. For example, the MQ Input node Security properties include:

- **Identity token type** specifies what type of identity token is present in the message.
- **Identity token location** specifies where in the message to find the identity (user ID).
- **Identity password location** specifies where in the message to find the password.
- **Identity issuedBy location** specifies who sent the identity.
- **Treat security exceptions as normal exceptions**: If this property is enabled, any security exception causes the message assembly to propagate through the **Failure** terminal of the node (if it is wired). The default value is “disabled”, which ensures that a security exception forces a rollback, even if the **Failure** terminal is wired.

The security properties vary with each node. For example, some security properties on the HTTPInput node do not exist on the MQInput node because of the differences in the transport security. Be sure that you understand the security properties that are specific to the type of message processing node that you are using.

Slide 10



WebSphere Education

Message flow security manager

- Security manager that is supplied with Integration Bus
- Not enabled by default when Integration Bus is installed
  - Security relegated to message transports
  - All messages that reach message flow are processed
- When security-enabled message processing node is encountered, message flow security manager consults security policy to determine whether security processing is required
- Allows integration node to:
  - Extract the identity credentials from an inbound message
  - Authenticate the identity by calling external security provider
  - Map the identity to alternative identity by calling external security provider
  - Authenticate original or alternative identity by calling external security provider
  - Propagate original or alternative identity (or both) in the outbound message

Message flow security manager

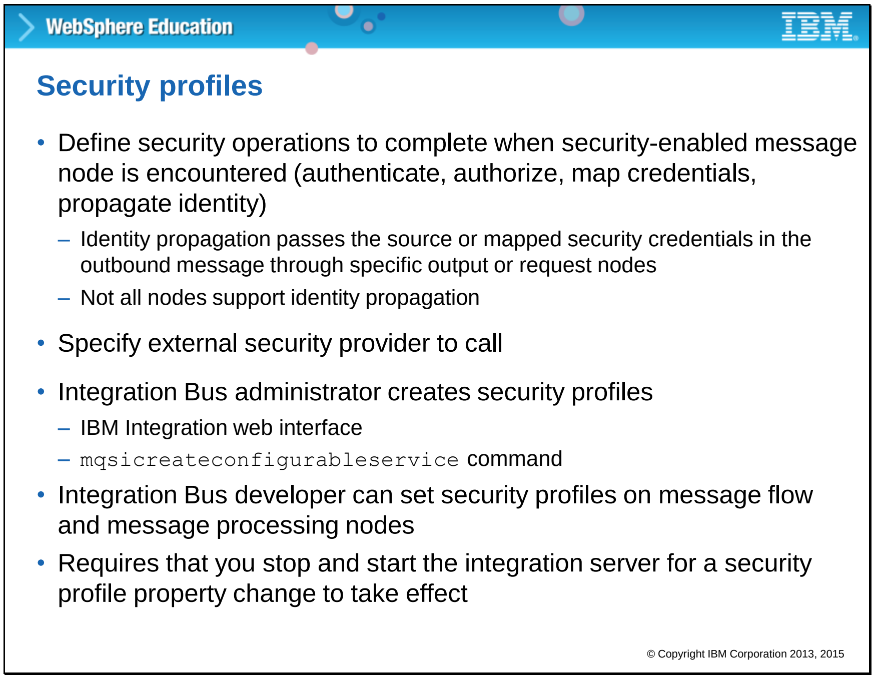
If the Integration Bus administrator does not enable message flow security, Integration Bus relies on the security that the message transport mechanism provides. In this case, the integration node processes all messages that are delivered to it, using the integration node service identity (the user ID the integration node runs under) as a proxy identity for all message instances. Any identity that is present in the incoming message is ignored.

When the Message Flow Security Manager is enabled, it does not delegate security to the message transport mechanism. Instead, the Message Flow Security Manager enables the integration node to provide end-to-end secure processing of message in Integration Bus by:

- Extracting the identity from an inbound message
- Authenticating the identity
- Mapping the identity to an alternative identity
- Checking that either the alternative identity or the original identity is authorized to access the message flow.
- Propagating either the alternative identity or the original identity with an outbound message

Authentication, identity mapping, and alternative identify checking require an external security provider. You learn more about external security providers later in this unit.

Slide 11



WebSphere Education

Security profiles

- Define security operations to complete when security-enabled message node is encountered (authenticate, authorize, map credentials, propagate identity)
  - Identity propagation passes the source or mapped security credentials in the outbound message through specific output or request nodes
  - Not all nodes support identity propagation
- Specify external security provider to call
- Integration Bus administrator creates security profiles
  - IBM Integration web interface
  - `mqsicreateconfigurableservice` command
- Integration Bus developer can set security profiles on message flow and message processing nodes
- Requires that you stop and start the integration server for a security profile property change to take effect

Security profiles

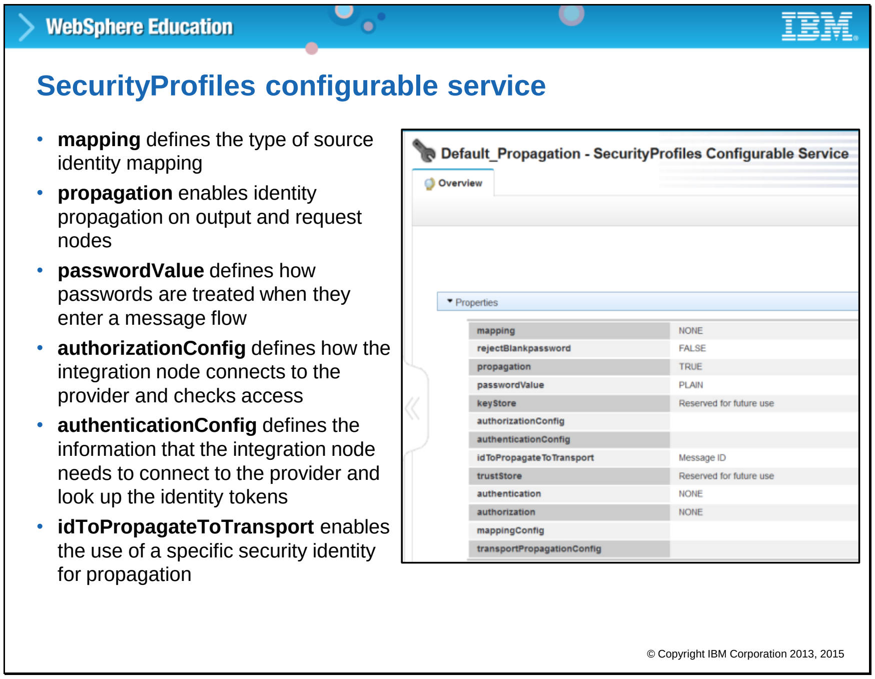
In Integration Bus, a security profile controls access to a message flow based on the identity that is associated with the message. A security profile provides a security mechanism that is independent of both the transport type and message format.

A security profile allows an administrator to specify whether identity and security token propagation, authentication, authorization, and mapping are done on the identity or security tokens that are associated with messages in the message flow, and if so, which external security is used.

You can create the security profile by using the mqsicreateconfigurableservice command or the Integration web user interface.

You must stop and start the integration server for a security profile property change to take effect.

Slide 12



WebSphere Education

SecurityProfiles configurable service

- **mapping** defines the type of source identity mapping
- **propagation** enables identity propagation on output and request nodes
- **passwordValue** defines how passwords are treated when they enter a message flow
- **authorizationConfig** defines how the integration node connects to the provider and checks access
- **authenticationConfig** defines the information that the integration node needs to connect to the provider and look up the identity tokens
- **idToPropagateToTransport** enables the use of a specific security identity for propagation

Default_Propagation - SecurityProfiles Configurable Service

Overview

▾ Properties

mapping	NONE
rejectBlankpassword	FALSE
propagation	TRUE
passwordValue	PLAIN
keyStore	Reserved for future use
authorizationConfig	
authenticationConfig	
idToPropagateToTransport	Message ID
trustStore	Reserved for future use
authentication	NONE
authorization	NONE
mappingConfig	
transportPropagationConfig	

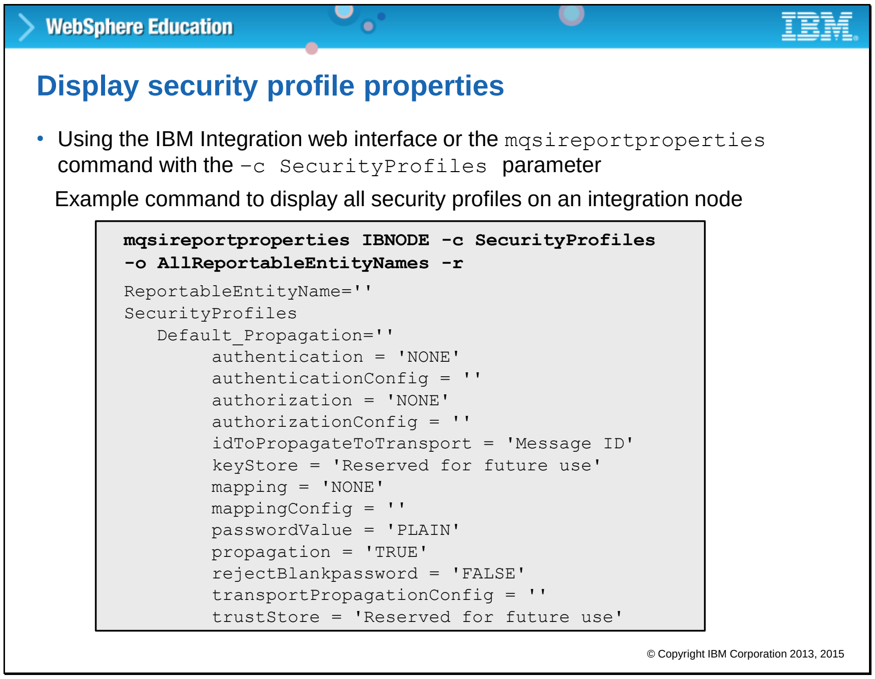
SecurityProfiles configurable service

Security profiles contain values for the following properties:

- **mapping** defines the type of mapping that is done on the source identity.
- **propagation** enables or disables identity propagation on output and request nodes.
- **passwordValue** defines how passwords are treated when they enter a message flow.
- **authorizationConfig** defines how the integration node connects to the provider, and contains additional information that can be used to check access (for example, a group that can be checked for membership).
- **authenticationConfig** defines the provider-specific information that the integration node needs to connect to the provider, and the information that is needed to look up the identity tokens.
- **idToPropagateToTransport** enables the use of a specific security identity for propagation.
- **authentication** defines the type of authentication that is done on the source identity.
- **authorization** defines the types of authorization checks that are done on the mapped or source identity.
- **mappingConfig** defines how the integration node connects to the provider, and contains additional information that is required to look up the mapping routine.
- **transportPropagationConfig** provides a specific security identity to propagate when **idToPropagateToTransport** is set to STATIC ID.

For a detailed description of each property, see the IBM Knowledge Center for IBM Integration Bus V10.

Slide 13



WebSphere Education

Display security profile properties

- Using the IBM Integration web interface or the `mqsireportproperties` command with the `–c SecurityProfiles` parameter

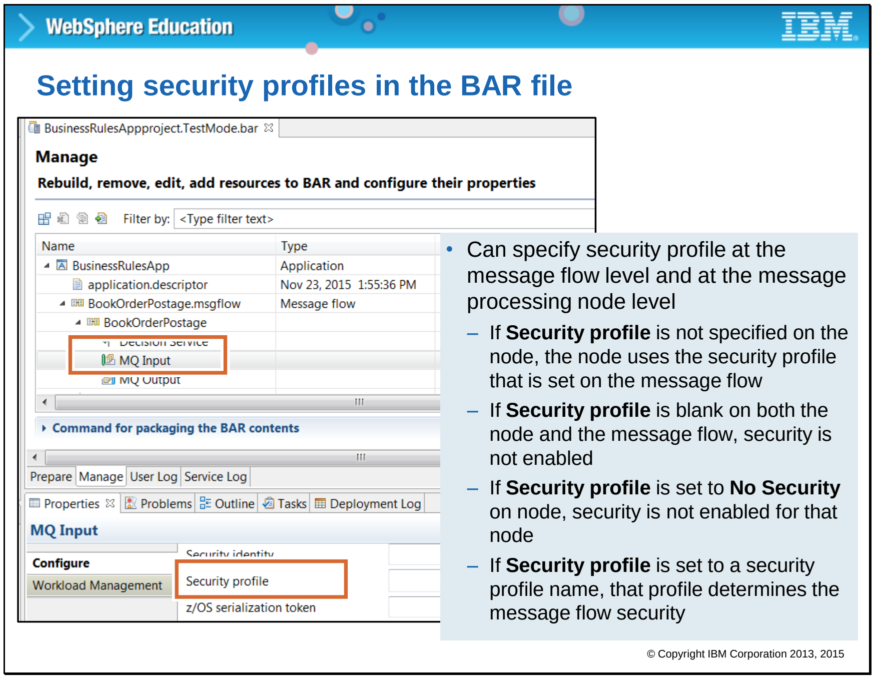
Example command to display all security profiles on an integration node

```
mqsireportproperties IBNODE -c SecurityProfiles
-o AllReportableEntityNames -r
ReportableEntityName=''
SecurityProfiles
   Default_Propagation=''
        authentication = 'NONE'
        authenticationConfig = ''
        authorization = 'NONE'
        authorizationConfig = ''
        idToPropagateToTransport = 'Message ID'
        keyStore = 'Reserved for future use'
        mapping = 'NONE'
        mappingConfig = ''
        passwordValue = 'PLAIN'
        propagation = 'TRUE'
        rejectBlankpassword = 'FALSE'
        transportPropagationConfig = ''
        trustStore = 'Reserved for future use'
```

Display security profile properties

You can use the Integration web interface and the mqsireportproperties command with the –c SecurityProfiles parameter to view security profile properties, which are shown in this example.

Slide 14


WebSphere Education
IBM
BusinessRulesAppproject.TestMode.bar
Manage
Rebuild, remove, edit, add resources to BAR and configure their properties
Filter by: <Type filter text>
Command for packaging the BAR contents
Prepare Manage User Log Service Log
Properties Problems Outline Tasks Deployment Log
MQ Input
Configure
Workload Management
Security identity
Security profile
z/OS serialization token

Setting security profiles in the BAR file

Name	Type
BusinessRulesApp	Application
application.descriptor	Nov 23, 2015 1:55:36 PM
BookOrderPostage.msgflow	Message flow
BookOrderPostage	
Decision Service	
MQ Input	
MQ Output	

- Can specify security profile at the message flow level and at the message processing node level
  - If **Security profile** is not specified on the node, the node uses the security profile that is set on the message flow
  - If **Security profile** is blank on both the node and the message flow, security is not enabled
  - If **Security profile** is set to **No Security** on node, security is not enabled for that node
  - If **Security profile** is set to a security profile name, that profile determines the message flow security

Setting security profiles in the BAR file

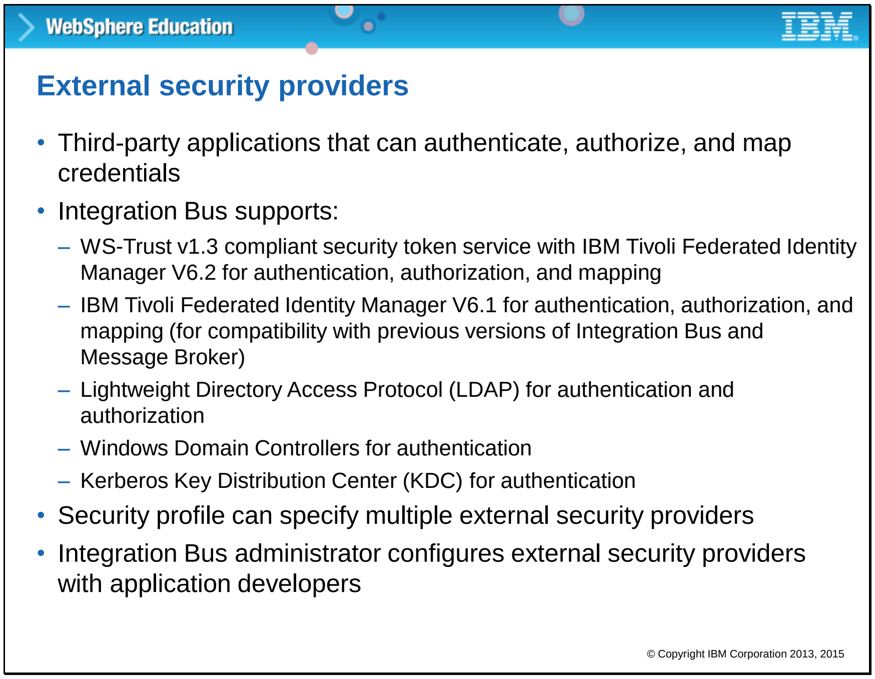
Security profiles are associated with an Integration Bus application in the BAR file.

The message flow, the security-enabled input and output message processing nodes, and the SecurityPEP node have a **Security profile** property that you can set with the BAR File editor.

You can specify security profile at the message flow level and at the message processing node level.

- If a security profile is not specified on the node, the node uses the security profile that is set on the message flow.
- If the **Security Profile** property is blank on both the node and the message flow, security is not enabled.
- If the **Security profile** is set to **No Security** on a message processing node, security is not enabled for that node.
- If the **Security profile** property is set to a security profile name, that profile determines the message flow security.

Slide 15



WebSphere Education

External security providers

- Third-party applications that can authenticate, authorize, and map credentials
- Integration Bus supports:
  - WS-Trust v1.3 compliant security token service with IBM Tivoli Federated Identity Manager V6.2 for authentication, authorization, and mapping
  - IBM Tivoli Federated Identity Manager V6.1 for authentication, authorization, and mapping (for compatibility with previous versions of Integration Bus and Message Broker)
  - Lightweight Directory Access Protocol (LDAP) for authentication and authorization
  - Windows Domain Controllers for authentication
  - Kerberos Key Distribution Center (KDC) for authentication
- Security profile can specify multiple external security providers
- Integration Bus administrator configures external security providers with application developers

External security providers

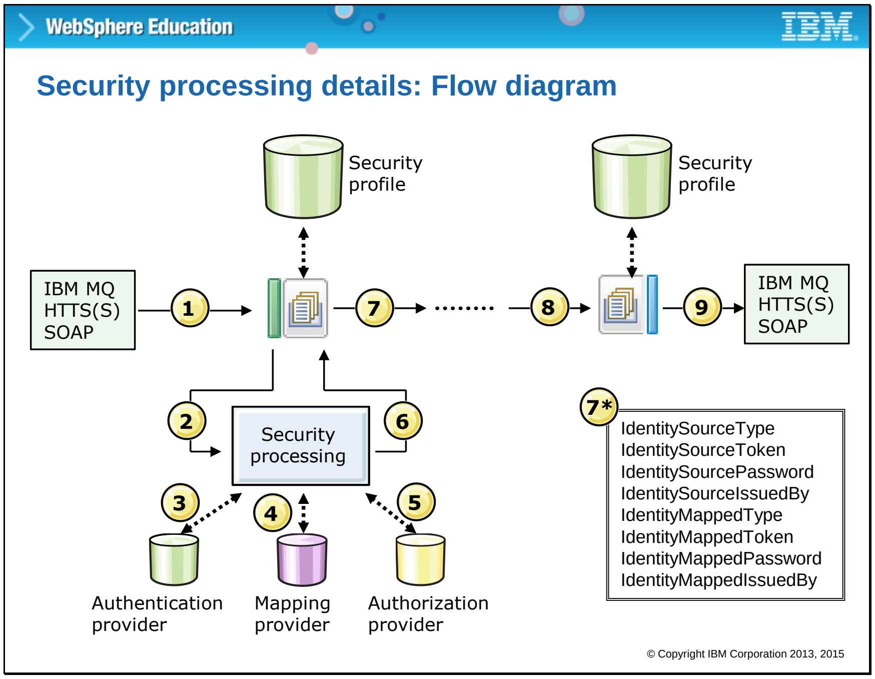
If an identity token or security token must be authenticated, authorized, or mapped to an alternative identity, Integration Bus calls an external security provider.

Integration Bus supports the external security providers listed here.

The Integration Bus administrator can configure the security profile to use different security providers for different security functions. For example, you might use LDAP for authentication and WS-Trust V1.3 STS for mapping and authorization.

The Integration Bus administrator is responsible for configuring the external security providers but might require coordination with application developer requirements.

Slide 16



WebSphere Education

IBM

Security processing details: Flow diagram

Security profile

Security profile

IBM MQ
HTTS(S)
SOAP

1

7

8

9

IBM MQ
HTTS(S)
SOAP

2

Security processing

6

3

4

5

Authentication provider

Mapping provider

Authorization provider

7*

IdentitySourceType
IdentitySourceToken
IdentitySourcePassword
IdentitySourceIssuedBy
IdentityMappedType
IdentityMappedToken
IdentityMappedPassword
IdentityMappedIssuedBy

Security processing details: Flow diagram

This slide summarizes the sequence of events that occur when security enabled input node in the message flow receives an input message.

1. When a message arrives at a security enabled input node, the presence of a security profile that is associated with the node indicates whether message flow security is configured.
2. If a security profile is associated with the node or message flow, the security enabled input node extracts the identity information from the input message.
3. If authentication is specified in the security profile, the security manager calls the configured security provider to authenticate the identity. If the security cache did not expire, subsequent messages with the same credentials that arrive at the message flow are completed with the cached result.
4. If identity mapping is specified in the security profile, the security manager calls the configured security provider to map the identity to an alternative identity.
5. If authorization is specified in the security profile, the security manager calls the configured security provider to authorize that the identity has access to this message flow.

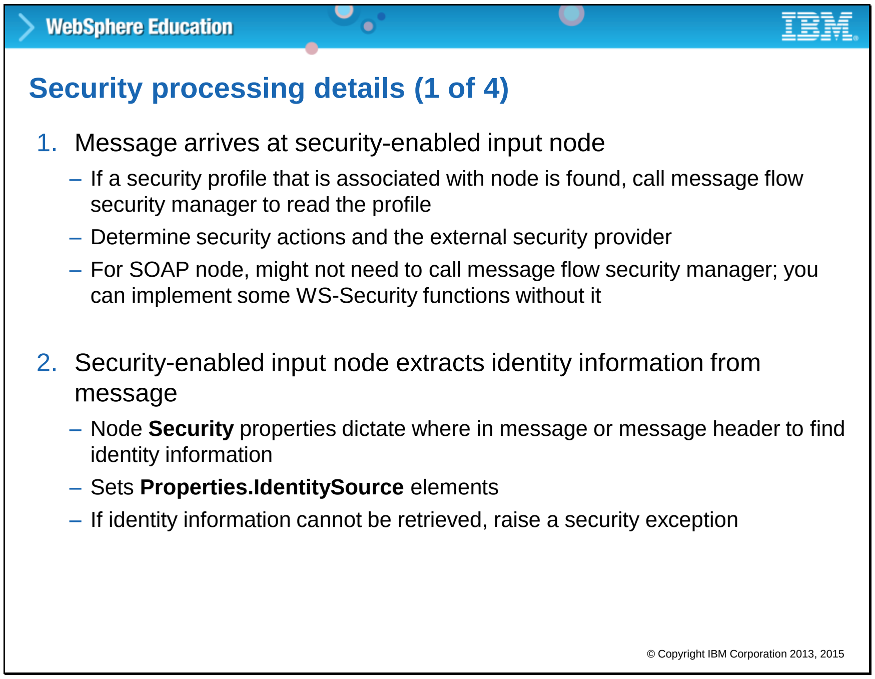
6. When all security processing is complete, or when the message flow security manager raises a security exception, control returns to the input node.

7. The message and its source and mapped identity information is propagated down the message flow.

8. When the message reaches a security enabled output node, a security profile that is associated with the node indicates that the current identity token is to be propagated when the message is sent.
9. The propagated identity is forwarded to the target application.

The next series of slides examine these steps in more detail.

Slide 17



WebSphere Education

Security processing details (1 of 4)

1. Message arrives at security-enabled input node
   - If a security profile that is associated with node is found, call message flow security manager to read the profile
   - Determine security actions and the external security provider
   - For SOAP node, might not need to call message flow security manager; you can implement some WS-Security functions without it

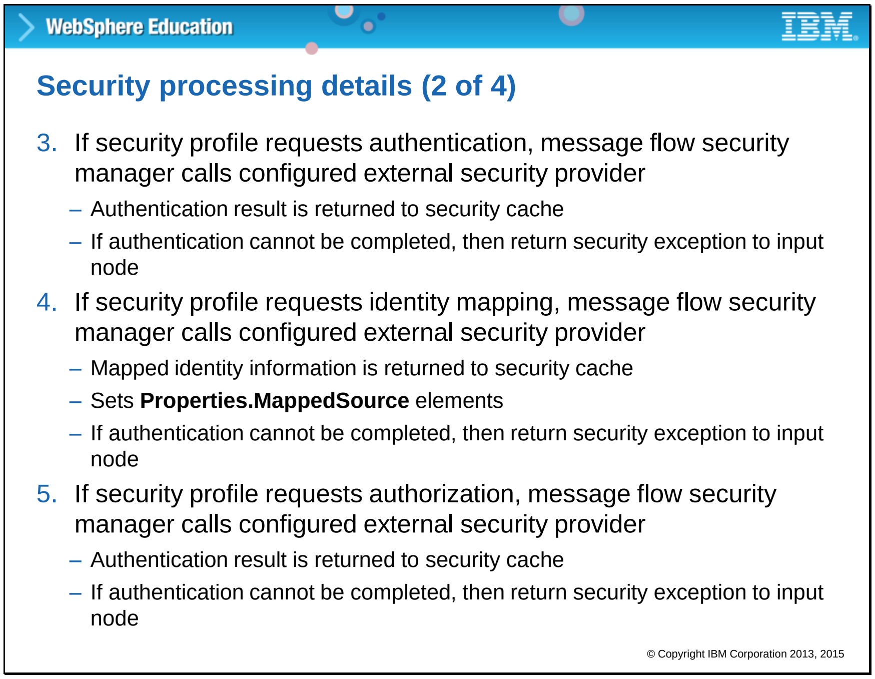
2. Security-enabled input node extracts identity information from message
   - Node **Security** properties dictate where in message or message header to find identity information
   - Sets **Properties.IdentitySource** elements
   - If identity information cannot be retrieved, raise a security exception

Security processing details (1 of 4)

1. When a message arrives at a security enabled input node, the presence of a security profile that is associated with the node indicates whether message flow security is configured. The integration node's security manager is called to read the profile, which specifies the combination of propagation, authentication, authorization, and mapping for the identity of the message. It also specifies the external security provider.

2. If a security profile is associated with the node or message flow, the security-enabled input node extracts the identity information from the input message based on the configuration on the node's **Security** properties page. It sets the IdentitySource elements in the IdentitySource folder. If the security tokens cannot be successfully extracted, a security exception is raised.

Slide 18



WebSphere Education

Security processing details (2 of 4)

3. If security profile requests authentication, message flow security manager calls configured external security provider
   - Authentication result is returned to security cache
   - If authentication cannot be completed, then return security exception to input node
4. If security profile requests identity mapping, message flow security manager calls configured external security provider
   - Mapped identity information is returned to security cache
   - Sets **Properties.MappedSource** elements
   - If authentication cannot be completed, then return security exception to input node
5. If security profile requests authorization, message flow security manager calls configured external security provider
   - Authentication result is returned to security cache
   - If authentication cannot be completed, then return security exception to input node

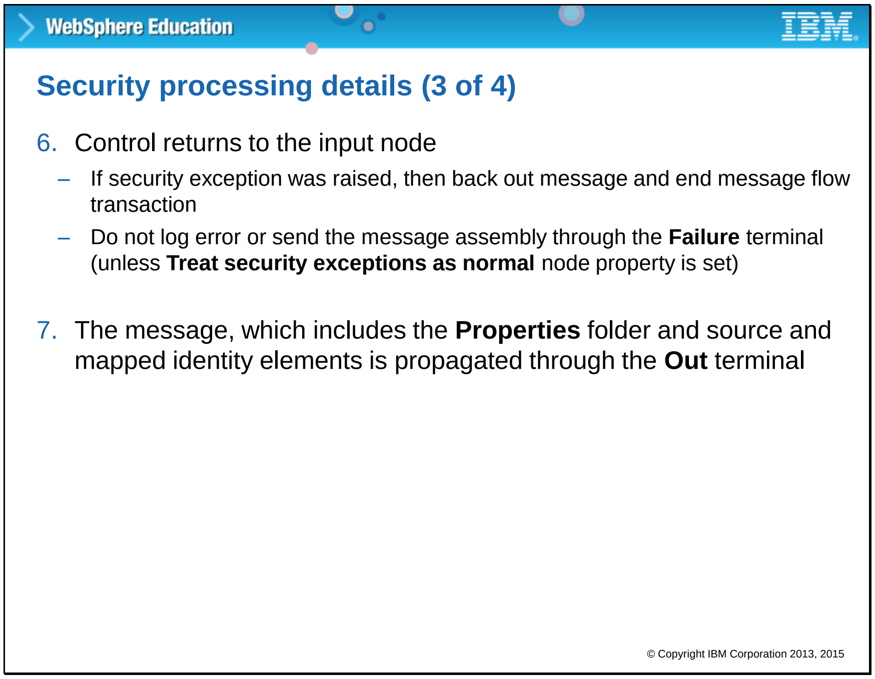
Security processing details (2 of 4)

3. If authentication is specified in the security profile, the security manager calls the configured security provider to authenticate the identity. A failure returns a security exception to the node. A security cache is provided for the authentication result, which enables subsequent messages with the same credential that arrive at the message flow to be completed with the cached result, unless it expired.

4. If identity mapping is specified in the security profile, the security manager calls the configured security provider to map the identity to an alternative identity. A failure returns an exception to the node. Otherwise, the mapped identity information is set in the IdentityMapped elements in the **Properties** folder. A security cache is provided for the result of the identity mapping.

5. If authorization is specified in the security profile, the security manager calls the configured security provider to authorize that the identity (either mapped or source) has access to this message flow. A failure returns an exception to the node. A security cache is provided for the authorization result.

Slide 19



WebSphere Education

Security processing details (3 of 4)

6. Control returns to the input node
   - If security exception was raised, then back out message and end message flow transaction
   - Do not log error or send the message assembly through the **Failure** terminal (unless **Treat security exceptions as normal** node property is set)

7. The message, which includes the **Properties** folder and source and mapped identity elements is propagated through the **Out** terminal

Security processing details (3 of 4)

6. When all security processing is complete, or when the message flow security manager raises an exception, control returns to the input node.

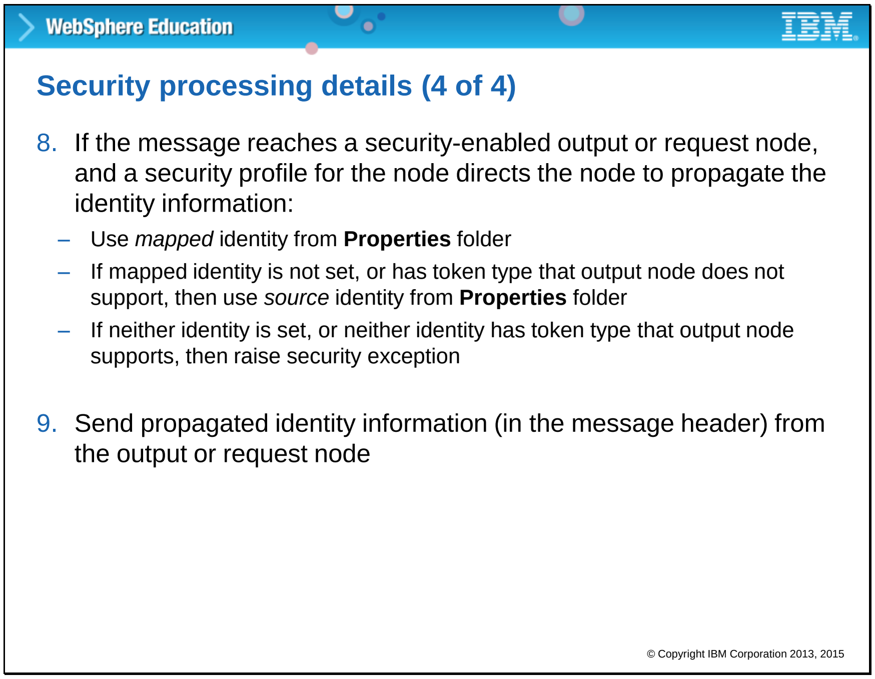
   If security exception was raised, the message is backed out and message flow transaction ends.

   By default, no message is written to the system log, and the message assembly is not propagated through the Failure terminal. Security exceptions in security-enabled input nodes are managed in this way to prevent a security denial of service attack from filling the logs and destabilizing the system.

   If the input node Treat security exceptions as normal property is set (on the nodes, which have this property), then a security exception is handled like any other exception. When this property is set, you can write an error handler for the Failure terminal.

7. If all security processing is successful, the message, including the **Properties** folder and its source and mapped identity information, is propagated down the message flow through the *Out* terminal of the input node.

Slide 20



WebSphere Education

Security processing details (4 of 4)

8. If the message reaches a security-enabled output or request node, and a security profile for the node directs the node to propagate the identity information:
   - Use *mapped* identity from **Properties** folder
   - If mapped identity is not set, or has token type that output node does not support, then use *source* identity from **Properties** folder
   - If neither identity is set, or neither identity has token type that output node supports, then raise security exception

9. Send propagated identity information (in the message header) from the output or request node

Security processing details (4 of 4)

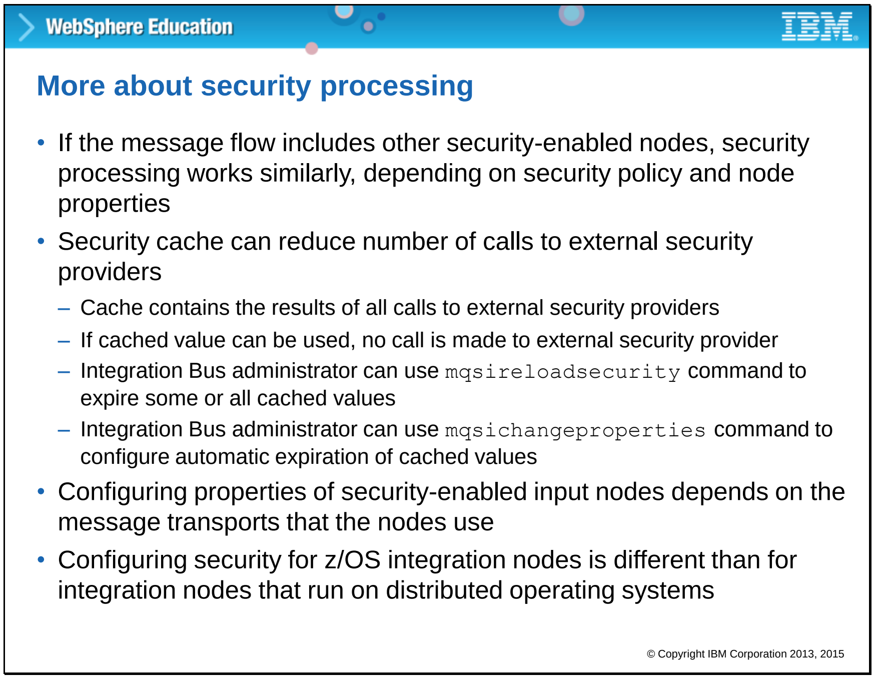
8. When the message reaches a security enabled output or request node, a security profile that is associated with the node can indicate that the current identity token is propagated when the message is sent.

   If the security profile indicates that propagation is required, the mapped identity is used. If the mapped identity is not set, or if it has a token type that the node does not support, the source identity is used. If no identity is set, or if the mapped and the source identity has a token type that the node does not support, a security exception is returned to the node.

   Some output and request nodes can automatically propagate only the identity token. If you use one of those nodes and you want to include the security token in the outbound message, use a Compute node in the message flow before the output or request node. Save the security token in the message header or message body, depending on what the receiving application expects.

9. The propagated identity is included in the appropriate message header when it is sent.

Slide 21



WebSphere Education
IBM

More about security processing

- If the message flow includes other security-enabled nodes, security processing works similarly, depending on security policy and node properties
- Security cache can reduce number of calls to external security providers
  - Cache contains the results of all calls to external security providers
  - If cached value can be used, no call is made to external security provider
  - Integration Bus administrator can use `mqsireloadsecurity` command to expire some or all cached values
  - Integration Bus administrator can use `mqsichangeproperties` command to configure automatic expiration of cached values
- Configuring properties of security-enabled input nodes depends on the message transports that the nodes use
- Configuring security for z/OS integration nodes is different than for integration nodes that run on distributed operating systems

More about security processing

If you add other security-enabled nodes or SecurityPEP nodes to the message flow, the basic processing remains the same. That is, if you added an HTTPRequest node and a SecurityPEP node, each of those nodes would follow steps 3-7, depending on how the security policy was configured for those nodes.

Each message that a message flow processes can make many calls to external security providers. To reduce the number of calls, a security cache holds the results of all calls to external security providers.

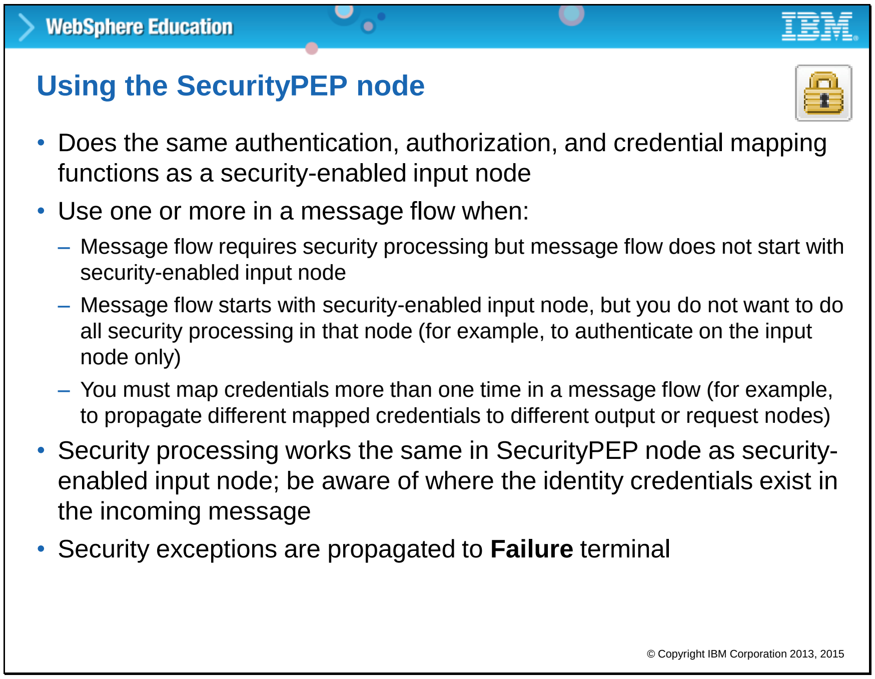
Instead of immediately calling the external security provider, the integration node checks to see whether the lookup values are already cached for the security identity and security token. If the values are in cache and are still valid, those cached values are used.

You can use the mqsireloadsecurity command to force some or all of the cached values to be marked as ‘expired’. You can also use the mqsichangeproperties command to configure the automatic expiration.

How you configure security-enabled input nodes depends on the message transport that is being used. For example, SOAP nodes do not have **Security** properties that you configure. Security for SOAP nodes is configured by using WS-Security (or Kerberos) policy sets and bindings.

Configuring security is different for integration nodes that are running in a z/OS environment as compared with integration nodes that are running in other environments. For specific details on z/OS security configuration, see the IBM Knowledge Center.

Slide 22



WebSphere Education

Using the SecurityPEP node

- Does the same authentication, authorization, and credential mapping functions as a security-enabled input node
- Use one or more in a message flow when:
  - Message flow requires security processing but message flow does not start with security-enabled input node
  - Message flow starts with security-enabled input node, but you do not want to do all security processing in that node (for example, to authenticate on the input node only)
  - You must map credentials more than one time in a message flow (for example, to propagate different mapped credentials to different output or request nodes)
- Security processing works the same in SecurityPEP node as security-enabled input node; be aware of where the identity credentials exist in the incoming message
- Security exceptions are propagated to **Failure** terminal

Using the SecurityPEP node

When you use the Security Policy Endpoint (SecurityPEP) message processing node in a message flow, it can do the same authentication, authorization, and credential mapping that a security-enabled input node does. It calls the Message Flow Security Manager just as an input node does.

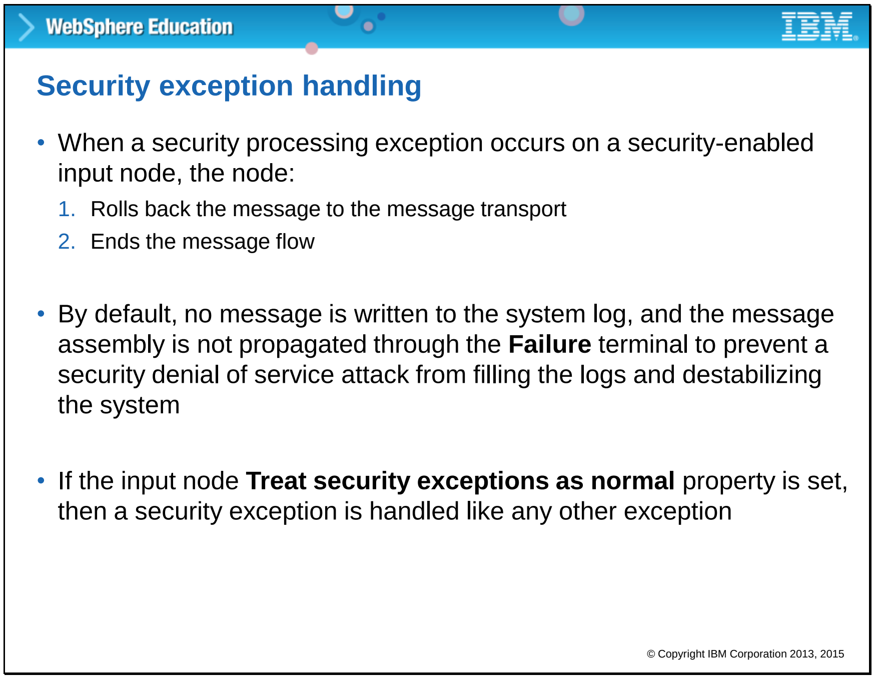
You might want to use one or more SecurityPEP nodes in a message flow for several reasons:

- If the message flow does not start with a security-enabled input node; for example, if the message flow starts with a JMSInput or FileInput node.
- If the message flow starts with a security-enabled input node, but you want only to authenticate or authorize the credentials on the input node, and not map credentials at that point.
- If the message flow starts with a security-enabled input node, but you want only to authenticate or authorize the credentials on the input node. Later in the flow you must map credentials multiple times. This scenario might be the case if the message flow contains multiple security-enabled output or request nodes and you must map credentials before each of those nodes. This capability is useful if you are sending a message to multiple destinations by using multiple message transports, for example. Use a SecurityPEP node that is configured before each output or request node.

The authentication, authorization, and credential mapping work in the SecurityPEP node as it does in a security-enabled input node. However, you must understand where the identity token (and security token, if applicable) is in the inbound message. These locations can affect how you configure the SecurityPEP node. For example, if the message flow does not contain a security-enabled input node, the *IdentitySource* and *IdentityMapped* elements of the Properties tree of the inbound message are empty.

You saw earlier that security exceptions that are raised in security-enabled input nodes are handled differently than most exceptions. Such is not the case with the SecurityPEP node. A security exception in the SecurityPEP node propagates the message assembly to the Failure terminal of the node, and writes a message to the system log.

Slide 23



WebSphere Education

Security exception handling

- When a security processing exception occurs on a security-enabled input node, the node:
  1. Rolls back the message to the message transport
  2. Ends the message flow
- By default, no message is written to the system log, and the message assembly is not propagated through the **Failure** terminal to prevent a security denial of service attack from filling the logs and destabilizing the system
- If the input node **Treat security exceptions as normal** property is set, then a security exception is handled like any other exception

Security exception handling

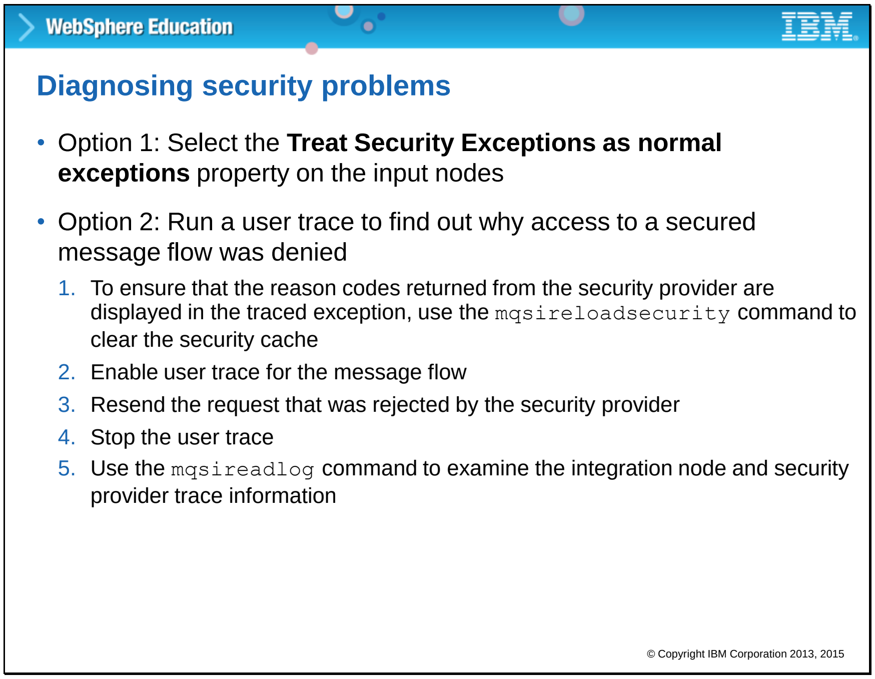
Exception handling for security exceptions is not managed the same as most other input node errors. Most exceptions that occur on input nodes cause the node to roll back the message to the message transport, write an error message to the system log, and then propagate the message assembly to the input node Failure terminal.

When a security processing exception occurs on a security-enabled input node, the node rolls back the message to the message transport and ends the message flow.

By default, no message is written to the system log, and the message assembly is not propagated through the Failure terminal. Security exceptions in security-enabled input nodes are managed in this way to prevent a security denial of service attack from filling the logs and destabilizing the system.

If the input node **Treat security exceptions as normal** property is set, then a security exception is handled like any other exception.

Slide 24



WebSphere Education

IBM

Diagnosing security problems

- Option 1: Select the **Treat Security Exceptions as normal exceptions** property on the input nodes
- Option 2: Run a user trace to find out why access to a secured message flow was denied
  1. To ensure that the reason codes returned from the security provider are displayed in the traced exception, use the `mqsireloadsecurity` command to clear the security cache
  2. Enable user trace for the message flow
  3. Resend the request that was rejected by the security provider
  4. Stop the user trace
  5. Use the `mqsireadlog` command to examine the integration node and security provider trace information

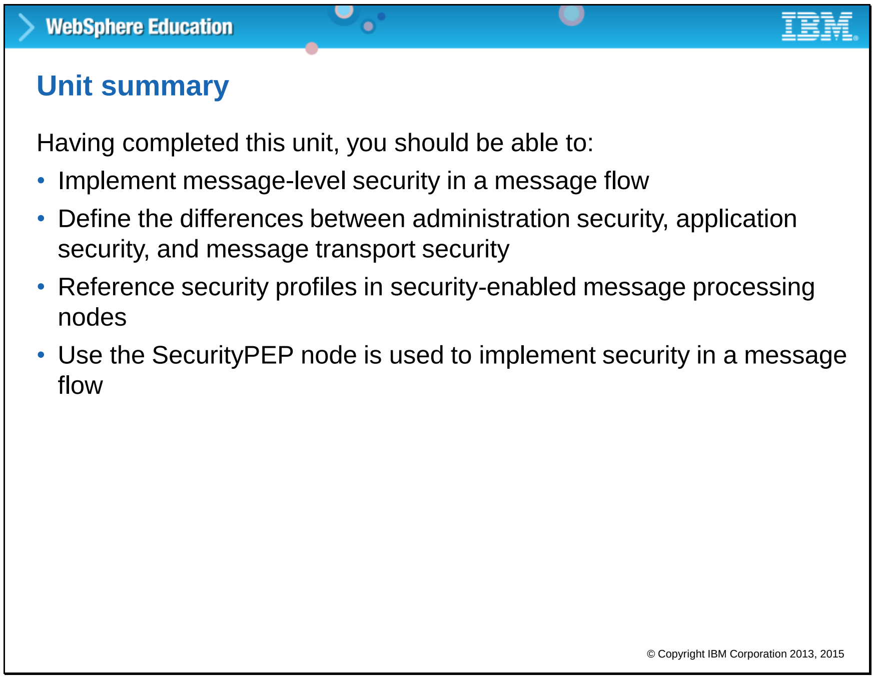
Diagnosing security problems

How do you diagnose security problems?

If you selected the option to **Treat security exceptions as normal**, then a security exception is handled like any other exception and you can check the system log for error information.

If you did not select the option to **Treat security exceptions as normal**, then run a user trace to find out why access to a secured message flow failed. If you are going to run a user trace, be sure to run the mqsireloadsecurity command first to ensure that the reason codes that the security provider returns are displayed in the trace exception. You learned about user trace in the prerequisite course, *IBM Integration Bus V10 Application Development I*.

Slide 25


WebSphere Education
IBM

Unit summary

Having completed this unit, you should be able to:

- Implement message-level security in a message flow
- Define the differences between administration security, application security, and message transport security
- Reference security profiles in security-enabled message processing nodes
- Use the SecurityPEP node is used to implement security in a message flow

Unit summary

This unit examined the different security considerations in IBM Integration Bus, and how to configure message flow security.

Having completed this unit, you should be able to:

- Implement message-level security in a message flow
- Define the differences between administration security, application security, and message transport security
- Reference security profiles in security-enabled message processing nodes
- Use the SecurityPEP node is used to implement security in a message flow