



FORUM SYSTEMS HANDS-ON TRAINING

LAB 10. TRANSACTION PRIVACY – SOAP/XML PAYLOAD ENCRYPTION AND DECRYPTION



FORUM SYSTEMS

A Crosscheck Networks Company

Legal Marks

No portion of this document may be reproduced or copied in any form, or by any means – graphic, electronic, or mechanical, including photocopying, taping, recording, or information retrieval system – without expressed permission from Forum Systems, Inc.

FORUMOST™ Firmware, Forum Systems XMLSec™ WebAdmin, Forum Systems XML Security Appliance™, Forum Sentry™, Forum Presidio™, Forum XWall™, Forum Sentry™ Web Services Gateway, Forum Presidio™ OpenPGP Gateway, Forum FIA Gateway™, Forum XWall Type-PCI™, Forum XWall® Web Services Firewall and Forum XRay™ are trademarks and registered trademarks of Forum Systems, Inc.

All other products are trademarks or registered trademarks of their respective companies.

Copyright © 2002-2014 Forum Systems, Inc. – All Rights Reserved.

Published: September 2014

Forum Systems Hands-on Training – Lab 10. Transaction Privacy – SOAP/XML Payload Encryption and Decryption
D-ASF-SE-010029

Contents

Introduction.....	4
<i>Skill Level</i>	4
<i>Prerequisites</i>	4
<i>Lab Overview</i>	4
Build a WSDL Policy in Sentry.....	5
Enable Encryption in SOAPSonar	7
XML Encryption and Decryption in Sentry	11
<i>Build the XML Encryption and Decryption Policies in Sentry</i>	11
<i>Configure Sentry for Decryption of the Request</i>	12
<i>Configure Sentry to Encrypt the Response</i>	15
Additional Testing and More Reading.....	19
<i>BACK IT UP!</i>	19
<i>Additional Tests</i>	19
<i>Additional Information</i>	19
About Forum Systems.....	20

Introduction

Lab 10. Transaction Privacy – SOAP/XML Payload Encryption and Decryption

Skill Level

This lab is Intermediate Skill Level. Some existing experience with Forum Sentry and SOAPSonar or completion of the Beginner Level labs is assumed.

Prerequisites

This lab requires the Forum Sentry Training Image, with a licensed copy of SOAPSonar Enterprise Edition.

Refer to the “FS_Training_Labs_v8-1_Introduction” document for information on the Forum Sentry Training Image and licensing SOAPSonar Enterprise Edition.

This lab requires the Sample SOAP Service built in Lab 1.

This lab requires the PKI infrastructure (keys) generated in Lab 7.

This lab assumes a basic knowledge of building WSDL policies in Sentry (Lab 5), testing a SOAP service with SOAPSonar (Lab 2), and building Task Lists in Sentry (Lab 10).

Lab Overview

This lab focuses on applying content layer encryption on SOAP/XML data processed by Sentry. In this lab we will build a new WSDL Policy for the Sample Web Service created in Lab 1 of this series. We will apply encryption to the SOAP request sent from SOAPSonar and Sentry will decrypt this data. The response from the back-end server will be encrypted by Sentry before being returned to SOAPSonar.

This use case will demonstrate both encryption and decryption in Sentry and SOAPSonar.

This lab will provide instructions for generating Encryption and Decryption Task Lists with Forum Sentry. Topics include:

1. Encrypting SOAP Messages with Sentry and SOAPSonar
2. Decrypting SOAP Messages with Sentry and SOAPSonar

Build a WSDL Policy in Sentry

In this step we will build a new WSDL Policy and a new HTTP Listener Policy. For full details on building a WSDL Policy in Sentry, please refer to Lab 5 of this series.

Follow the steps below to build a new WSDL Policy for the Sample Web Service created in Lab 1.

1. Navigate to the Gateway→WSDL Policies→WSDL Policies page and click New.
2. Create the WSDL Policy as follows:
 - a. Name: SampleWS_Encryption
 - b. Import the SampleWS WSDL file from the .NET WebMatrix service running on the Training VM. Import the WSDL via URL: <http://localhost:8080/SampleWS.asmx?WSDL>
3. Create the 2 virtual directories (for SOAP 1.1 and SOAP 1.2) using the following criteria:
 - a. Create a new HTTP Listener Policy and use for both virtual directories
 - i. Use Device IP: Enabled
 - ii. Listener Port: 84
 - b. Virtual Directory Paths
 - i. SOAP 1.1: /SampleWS.asmx
 - ii. SOAP 1.2: /SampleWS_SOAP12.asmx
 - c. Create a new HTTP Remote Policy to use for both virtual directories:
 - i. Name: SampleWS_Encryption-Remote
 - ii. Remote Policy Host: 127.0.0.1
 - iii. Remote Policy Port: 8080
 - d. The new WSDL Policy should be configured as follows:

WSDL POLICIES > WSDL POLICY

WSDL POLICY

Policy Name: SampleWS_Encryption

Upgrade Export WSDL Publish WSDL WSI Validation

Services Task Lists Settings IDP Rules Logging Documents

SERVICE	PORT	STATUS	VIRTUAL URI	PHYSICAL URI
training	trainingSoap	●	http://172.31.3.22:84/SampleWS.asmx	http://127.0.0.1:8080/SampleWS.asmx
training	trainingSoap12	●	http://172.31.3.22:84/SampleWS_SOAP12.asmx	http://127.0.0.1:8080/SampleWS.asmx

Enable Disable

Service: training — Port: trainingSoap

OPERATION	STATUS	ACL	INPUT MESSAGE	OUTPUT MESSAGE	IDP GROUP
Concat	●	[Allow All]	ConcatSoapIn	ConcatSoapOut	Default Operation Group (0)
Divide	●	[Allow All]	DivideSoapIn	DivideSoapOut	Default Operation Group (0)
Echo	●	[Allow All]	EchoSoapIn	EchoSoapOut	Default Operation Group (0)
Multiply	●	[Allow All]	MultiplySoapIn	MultiplySoapOut	Default Operation Group (0)

Enable Disable

Service: training — Port: trainingSoap12

OPERATION	STATUS	ACL	INPUT MESSAGE	OUTPUT MESSAGE	IDP GROUP
Concat	●	[Allow All]	ConcatSoapIn	ConcatSoapOut	Default Operation Group (0)
Divide	●	[Allow All]	DivideSoapIn	DivideSoapOut	Default Operation Group (0)
Echo	●	[Allow All]	EchoSoapIn	EchoSoapOut	Default Operation Group (0)
Multiply	●	[Allow All]	MultiplySoapIn	MultiplySoapOut	Default Operation Group (0)

- e. Click the trainingSOAP Port link to access the Virtual Directory page for the SOAP 1.1 port. Select the “Enable WSDL Access” option. Click Save.

WSDL POLICIES > WSDL POLICY

WSDL POLICY

Policy Name: SampleWS_Encryption

[Upgrade](#)
[Export WSDL](#)
[Publish WSDL](#)
[WSI Validation](#)

Services
Task Lists
Settings
IDP Rules
Logging
Documents

Process Response:
Off

IP ACL Policy:
Unrestricted
Edit

ACL Policy:
[Allow All]

XACML Policy:
[None]

Password Authentication:
[From Listener Policy]

Enable WSDL access:
☒

Redirect Policy:
[None]

Error Template:
SOAP 1.1 Fault Template
Edit

☐ Publish a different location in exported WSDL

Published Protocol:
http

Published Host:

Published Port:

Save

#	MESSAGE TYPE FILTER	FORMAT	DESCRIPTION	STATUS
1	MTOM Filter	MTOM	MTOM Filter	●
2	SOAP 1.1 Filter	Simple	WSDL 1.1 SOAP 1.1 Filter	●

[Restore Defaults](#)
[Enable](#)
[Disable](#)
[Delete](#)
[New](#)

6 www. forumsys.com

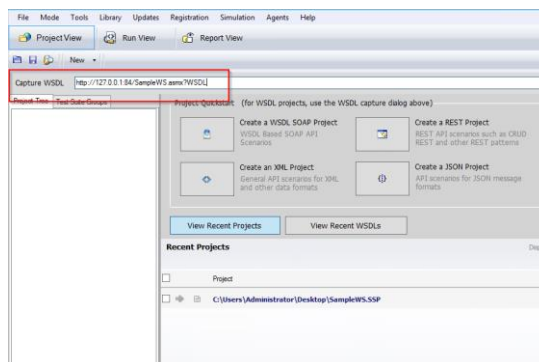
Enable Encryption in SOAPSonar



In this step we will load the WSDL from the new WSDL policy into SOAPSonar. We will send a clear text SOAP message through successfully, and lastly configure SOAPSonar to apply encryption.

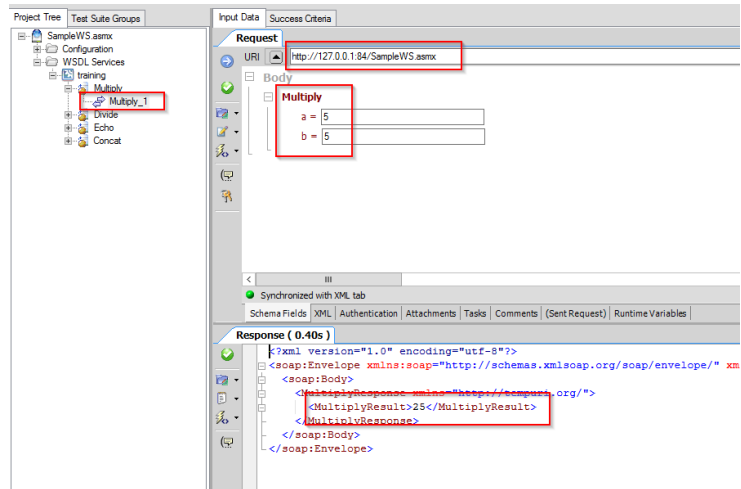
Hint – Be sure to use port 84 with this service as it is similar to the service build in Lab 5 on port 81.

Follow the Steps below to load the WSDL into SOAPSonar and test the new WSDL Policy.

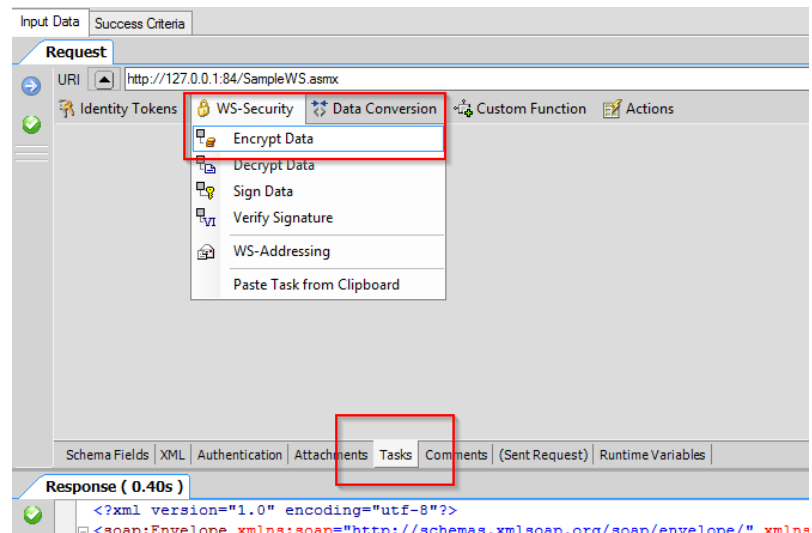
1. Load the Sentry WSDL into SOAPSonar.
 - a. Launch SOAPSonar
 - b. In the Capture WSDL field, enter the URL to the Sentry virtual directory and add ?WSDL to the end. This should work: <http://127.0.0.1:84/SampleWS.asmx?WSDL>



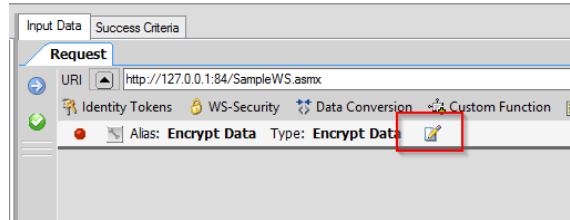
- c. Click Enter (or click the icon to the right of the bar) to retrieve and load the WSDL from Sentry into SOAPSonar.
 - d. SOAPSonar will parse the WSDL and list out the 4 operations for this service (Multiply, Divide, Echo, Concat). SOAPSonar will also build sample test cases for each operation.
 - e. Expand the 'Multiply' operation and click the test case named "Multiply_1". In the request pane, notice the URL is the Sentry Virtual Directory.
 - f. Enter values in the Multiple 'a' and 'b' fields, then click  to commit the settings and  to send the request. The request will be processed by Sentry and then sent to the WebMatrix service. The response will be generated by WebMatrix and then returned to the client through Sentry.




- g. You have successfully tested the WSDL policy with clear text SOAP.
2. Import the Sentry_Server_Key certificate, generated in Lab 7, into the Windows Key Store via the SOAPSonar PKI Tools interface.
 - a. In Sentry, go to the Resources → PKI → Keys page. Located the Sentry_Server_Key cert and download it in PEM format
 - b. In SOAPSonar, open the Tools → PKI Management window
 - c. Browse to Win Storages → Current User → My
 - d. Right click on My and choose Load Certificate
 - e. Browse to the downloaded cert and import it
 3. Build a Task in SOAPSonar to encrypt the SOAP request. Click on the Tasks tab on the bottom of the Request pane. From the WS Security drop down, choose Encrypt Data.

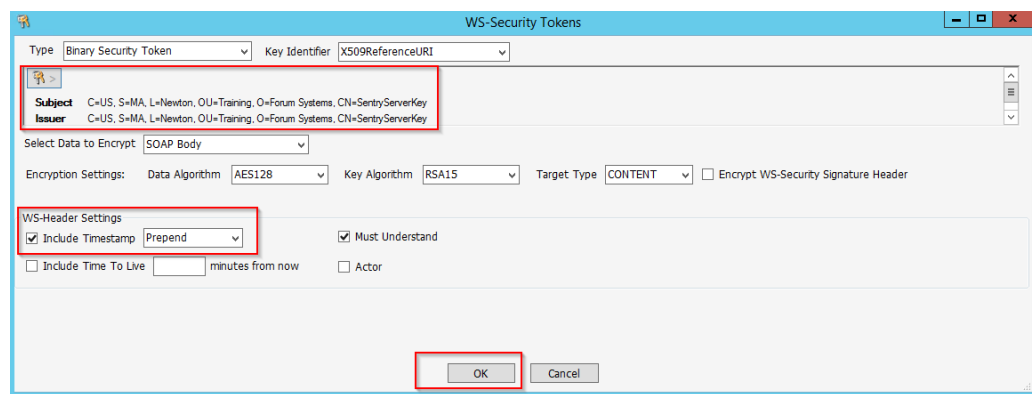


4. Click the Pencil&Paper icon to configure the Encrypt Data task.

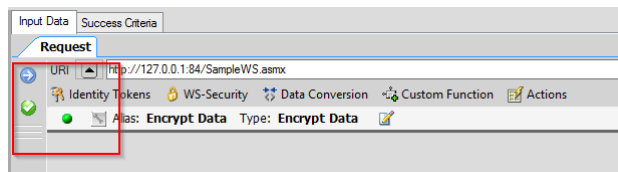


5. On this screen we will configure the Encrypt Data task as follows:

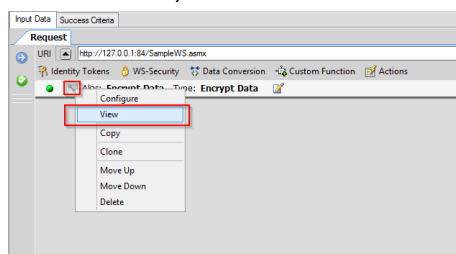
- Type: Binary Security Token (leave default)
- Key Identifier: X509ReferenceURI (leave default)
- Click the  icon and Browse PKI – select the SentryServer cert under the My folder (the cert downloaded from Sentry and imported in step 2 above)
- Select Data to Encrypt: SOAP Body (leave default)
- Encryption Settings: leave all defaults
- WS-Header Settings: Enable the Include Timestamp (prepend) option, leave all other defaults.
- Click OK.

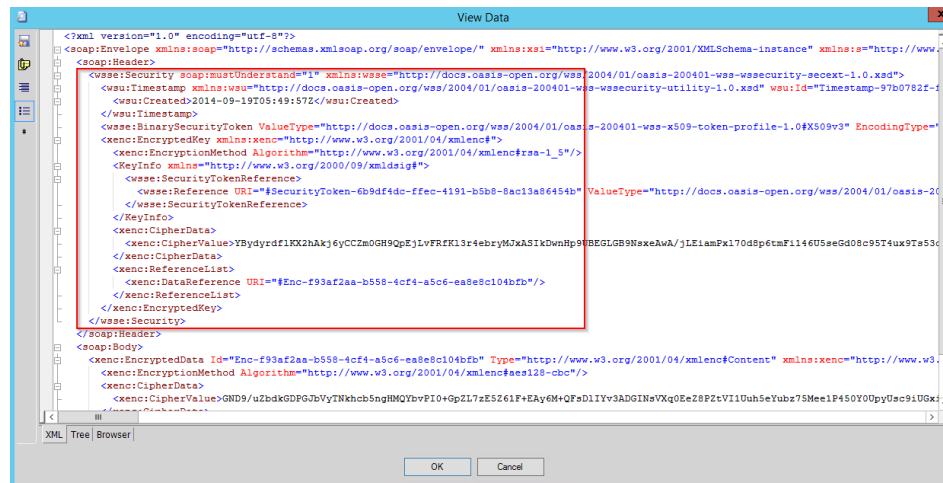


6. The Encrypt Data task is now enabled, as indicated by the green circle.



7. To test the Encrypt Data task in SOAPSonar, click the wrench icon and choose View.







10. Commit all settings and save the SOAPSonar project file.
11. If you send the request into Sentry now, you'll receive an error from Sentry. The "Invalid WSDL Message" IDP rule is triggered because Sentry fails schema validation of this encrypted document. This is expected behavior.

XML Encryption and Decryption in Sentry

We will now configure Sentry to decrypt an encrypted request and then encrypt the response that is sent back to SOAPSonar. This includes building the XML Encryption / XML Decryption policies and the Encrypt Elements / Decrypt Elements tasks that will process the runtime traffic. Lastly, the Task Lists will be grouped and associated to the new WSDL Policy built in this lab.

Build the XML Encryption and Decryption Policies in Sentry

In this step we will build the XML Encryption and XML Decryption policies that associate the keys to this process. These "reusable policy objects" will later be referenced in the tasks that process the runtime traffic.

With the XML Encryption Policy, you will specify a public certificate that Sentry will use to encrypt data for the client, who will use the corresponding private key for decryption.

With the XML Decryption Policy, you will specify a private key that will be used to decrypt the data encrypted by the client using the corresponding public certificate.

Follow the steps below to build the XML Encryption Policy.

1. Navigate to the Resources→Security Policies→XML Encryption page.
2. Click New to build a new XML Encryption Policy.
3. Build the XML Encryption policy with the following criteria:
 - a. Name: ClientCert_Encryption_Policy
 - b. Algorithm: 3DES
 - c. Key Wrap Algorithm: RSA
 - d. Encryption Mode: Use the "ClientCert" pre-stored peer certificate
 - e. Validate against Signer Group: Do not validate
 - f. Click Create

The screenshot shows the 'XML ENCRYPTION POLICY' configuration form. The 'Policy Name' field is set to 'ClientCert_Encryption_Policy'. The 'Algorithm' dropdown is set to '3DES'. The 'Key Wrap Algorithm' dropdown is set to 'RSA'. The 'Encryption Mode' section has three radio buttons: 'Use the certificate from an identified user', 'Use the same certificate used for client signature verification', and 'Use this pre-stored peer certificate'. The third option is selected. Below it, a dropdown menu is set to 'ClientCert', with an 'Edit' link next to it. The 'Validate against Signer Group' dropdown is set to 'Do not validate'. The 'Symmetric Key' field is empty. A red box highlights the 'Encryption Mode' section and the 'ClientCert' dropdown. A 'Create' button is at the bottom right.

Follow the steps below to build the XML Decryption Policy

1. Navigate to the Resources→Security Policies→XML Decryption page.
2. Click New to build a new XML Decryption Policy.
3. Build the XML Decryption policy with the following criteria:
 - a. Name: SentryServer_Decryption_Policy
 - b. Algorithm: Any
 - c. Key Wrap Algorithm: Any RSA

d. Key Pair: Sentry_Server_Key

DECRYPT XML > XML DECRYPTION POLICY

XML DECRYPTION POLICY

Policy Name: SentryServer_Decryption_Policy

Algorithm: [Any]

Key Wrap Algorithm: [Any RSA]

Key Pair: Sentry_Server_Key Edit

Symmetric Key:

Create

Configure Sentry for Decryption of the Request

In this step we will build a Task List (and Task List Group) that will be used on the request processing of the WSDL policy to decrypt the encrypted request. This step also requires importing a sample encrypted SOAP message.

Follow the steps below to configure Sentry to decrypt the incoming encrypted SOAP message.

1. Navigate to the Resources→Documents→Documents page.
2. Import the “EncryptedMultipleRequest.xml” document that was saved from SOAPSonar earlier in this lab. This document should be in the Samples folder of the Desktop of the Training Image.

DOCUMENTS > NEW DOCUMENT

SAMPLE DOCUMENT

Name: EncryptedMultipleRequest.xml

Document: Create From WSDL WSDL Policy: SampleWS Edit

Service: training Edit

Port: trainingSoap Edit

Operation: Concat Edit

Message: ConcatSoapIn Edit

File URL Paste

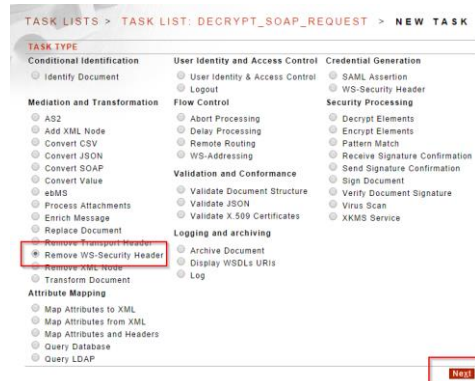
Choose File No file chosen

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:su="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://tempuri.org/">
  <soap:Header>
    <wsse:Security soap:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="Timestamp-d66e5b30-4d74-49c7-9b84-5b42355f644">
        <wsu:Created>2014-09-19T06:18:33Z</wsu:Created>
      </wsu:Timestamp>
      <wsse:BinarySecurityToken ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-"/>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <tns:ConcatSoapIn />
  </soap:Body>
</soap:Envelope>
```

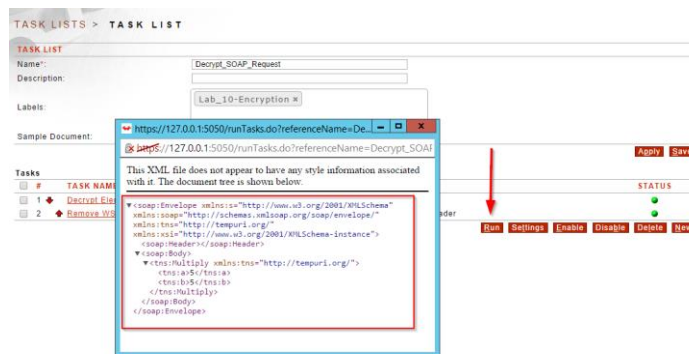
Apply Save

3. Navigate to the Gateway→Task Policies→Task Lists page. Click New to create a new Task List.
4. Configure the Task List with the following criteria:
 - a. Name: Decrypt_SOAP_Request
 - b. Label: Lab_10-Encryption
 - c. Sample Document: EncryptedMultipleRequest.xml
 - d. Click Apply

9. To remove the WS Security header after decryption, add a Remove WS Security Header task found under the Mediation category. This task has no configuration options, simply add it and click Save.



10. Your Task List now has two tasks. Run the Task List again, notice the sample document is decrypted and the WS Security Header is removed. This is representative of the document that Sentry will send along to the back-end service after the Task List Processing.



11. Save the Task List which will return you to the Task Lists page.
12. Select the Task List and click the Add to New Task List Group button. This adds the Task List to a Task List Group.
13. Associate the new Task List Group "Decrypt_SOAP_Request" to the WSDL Policy.
 - a. Navigate to the Gateway→WSDL Policies→WSDL Policies page.
 - b. Open the "SampleWS_Encryption" WSDL Policy.
 - c. Go to the Settings tab and scroll down to the Processing Settings.
 - d. Check the "Pre-Process request" box and associate the "Decrypt_SOAP_Request" Task List Group as the "Request Pre-Process Task List Group"
 - e. Click Save

WSDL POLICIES > WSDL POLICY

WSDL POLICY

Policy Name: SampleWS_Encryption Upgrade Export WSDL PA

Services Task Lists Settings IDP Rules Logging Documents

☐ Validate SOAP body from WSDL schema

☐ Validate message using WSI Basic Profile [Configure Tests](#)

PROCESSING SETTINGS

☒ Pre-process requests

Request Pre-Process Task List Group: Decrypt_SOAP_Request [Edit](#)

☐ Post-process requests

Request Post-Process Task List Group: Decrypt_SOAP_Request [Edit](#)

☐ Pre-process responses (Response processing must be enabled.)

Response Pre-Process Task List Group: Decrypt_SOAP_Request [Edit](#)

☐ Post-process responses (Response processing must be enabled.)

Response Post-Process Task List Group: Decrypt_SOAP_Request [Edit](#)

WEB SERVICES RELIABLE MESSAGING

☐ Enable reliable messaging

Reliable Messaging Policy: ▼

Save

14. If you now send the encrypted SOAP request from SOAPSonar again, you should no longer receive an error. Instead, Sentry should successfully decrypt the document before schema validation, then send the decrypted document to the remote server. Check the Sentry System log to confirm the processing being done by Sentry.

X0000D1	09003	D	Connecting to back end server at URL 'http://127.0.0.1:8080/SampleWS.asmx'
X0000D1	0E10B	D	No TaskListGroup configured, document will not be processed
X0000D1	0E20A	D	Document left WSDL validation
X0000D1	0E208	D	WSDL message: MultiplySoapIn
X0000D1	0E221	D	ACL check skipped - no ACL associated with operation 'Multiply'.
X0000D1	0E209	D	Document entered WSDL validation
X0000D1	0E207	D	Matched WSDL operation 'Multiply(MultiplySoapIn)'
X0000D1	0E603	D	Document left Process Manager
X0000D1	0E600	I	Successfully processed task list 'Decrypt_SOAP_Request'
X0000D1	0E010	D	Document left 'Remove WS-Security Header' task: 'Remove WS-Security Header'
X0000D1	0E00F	D	Document entered 'Remove WS-Security Header' task: 'Remove WS-Security Header'
X0000D1	0E010	D	Document left 'Decrypt Elements' task: 'Decrypt Elements'
X0000D1	0E00F	D	Document entered 'Decrypt Elements' task: 'Decrypt Elements'
X0000D1	0E602	D	Document entered Process Manager for task list 'Decrypt_SOAP_Request'
X0000D1	0E10D	D	<input type="checkbox"/> Incoming document identified to TaskList 'Decrypt_SOAP_Request' using TaskListGroup 'Decrypt_SOAP_Request'
X0000D1	08407	D	<input type="checkbox"/> Request document: <pre><?xml version="1.0" encoding="utf-8"?> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://tempuri.org/"><soap:Header><wsse:Security soap:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-2004-01-wss-wssecurity-secext-1.0.xsd"><wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-2004-01-wss-wssecurity-utility-1.0.xsd" wsu:Id="Ttimestamp-8a2elc19-5fe3-48e8-baf9-fdc0831730a2"><wsu:Created>2014-09-19T07:01:58Z</wsu:Created></wsu:Timestamp><wsse:BinarySecurityToken ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0-x509v3" EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0" Value=""/></pre>

Configure Sentry to Encrypt the Response

In this step we will build a Task List (and Task List Group) that will be used on the response processing of the WSDL policy to encrypt the SOAP response returned to the client.

This step also requires enabling Response Processing on the HTTP Remote Policy.

Follow the steps below to configure Sentry to encrypt the SOAP response message.

1. Navigate to the Gateway→Network Policies→Network Policies page.

2. Modify the HTTP Remote Policy “SampleWS_Encryption-Remote” which is used by the WSDL policy built in this lab. Click the Process Response link and set this to On.

NETWORK POLICIES > HTTP REMOTE POLICY

RESPONSE PROCESSING

☒ Process Response

POLICY SELECTIONS

Policy Name: SampleWS_Encryption-Remote

Outbound Protocol: HTTP

Remote Server: 127.0.0.1:8080

TCP Timeouts: Connect: 10 Read: 600 Connection Limit: Unlimited

Process Response: On

Finish

3. Navigate to the Gateway→Task Policies→Task Lists page. Click New to create a new Task List.
4. Configure the Task List with the following criteria:
 - a. Name: Encrypt_SOAP_Response
 - b. Label: Lab_10-Encryption
 - c. Sample Document: SoapDocument.xml
 - d. Click Apply

TASK LISTS > TASK LIST

TASK LIST

Name: Encrypt_SOAP_Response

Description:

Labels: Lab_10-Encryption

Sample Document: SoapDocument.xml

Apply Save

5. Click New to add a new task.
6. Under the Security Processing category, select Encrypt Elements and click Next.
7. Configure the Encrypt Elements task as follows:
 - a. Name: Encrypt Elements
 - b. Type: Encrypt Content
 - c. Encryption Policy: ClientCert_Encryption_Policy (built earlier in this lab)
 - d. Key Identifier: Serial Number
 - e. Encrypt Attachments: Unchecked
 - f. Canonicalize base64Binary data (MTOM-compatible): Unchecked
 - g. Select Elements to Encrypt: Select the ‘soap:Body’ node and click Apply. This will build an XPath expression and put a lock icon next to this node.
 - h. Click Apply and Save.

TASK LISTS > TASK LIST: ENCRYPT_SOAP_RESPONSE > TASK: ENCRYPT ELEMENTS

Configuration saved

ENCRYPT

Task Type: Encrypt Elements
 Task Name*:
 On Error: ☒ Log & Halt Processing ☐ Log & Continue

ENCRYPTION PROPERTIES

Type: ☐ Encrypt Element ☒ Encrypt Content
 Method: ☒ WSS 1.1 ☐ WSS 2004 ☐ XML Encryption
 Encryption policy:
 Key Identifier: ☒ SerialNumber ☐ X.509 ☐ SubjectKeyIdentifier ☐ Subject
☐ Encrypt attachments
☐ Canonicalize base64Binary data (MTOM-compatible)

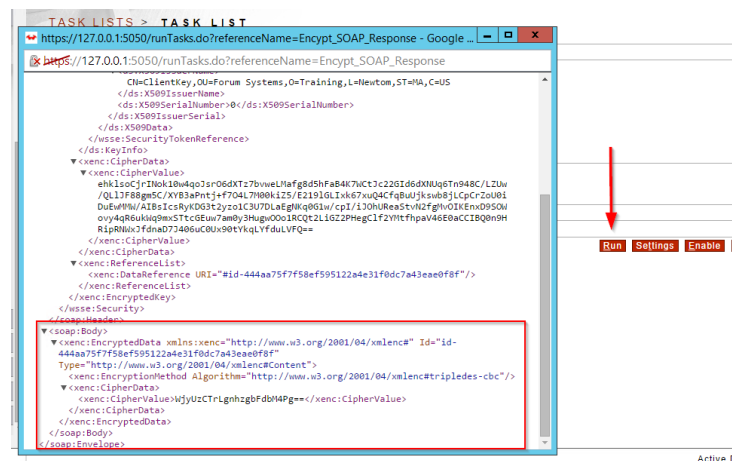
SELECT ELEMENTS TO ENCRYPT

☐ soap:Envelope
☒ soap:Body

Elements to Encrypt

☐ PATH
☐

8. Test the Task List with the Run button. A new browser window should pop-up displaying an encrypted SOAP Body node and a WS Security Header. This is representative of the response document Sentry will return to the client after successful task processing.



9. Save the Task List which will take you back to the Task Lists page.
10. Select the Task List and click the Add to New Task List Group button. This adds the Task List to a Task List Group.
11. Associate the new Task List Group “Encrypt_SOAP_Response” to the WSDL Policy.
 - a. Navigate to the Gateway→WSDL Policies→WSDL Policies page.
 - b. Open the “SampleWS_Encryption” WSDL Policy.
 - c. Go to the Settings tab and scroll down to the Processing Settings.
 - d. Check the “Post-Process response” box and associate the “Encrypt_SOAP_Response” Task List Group as the “Response Post-Process Task List Group”

e. Click Save

WSDL POLICIES > WSDL POLICY

WSDL POLICY

Policy Name: SampleWS_Encryption

Upgrade

Services Task Lists Settings IDP Rules Logging Documents

☐ Validate SOAP body from WSDL schema

☐ Validate message using WSI Basic Profile [Configure Tests](#)

PROCESSING SETTINGS

☒ Pre-process requests

Request Pre-Process Task List Group: Decrypt_SOAP_Request [Edit](#)

☐ Post-process requests

Request Post-Process Task List Group: Decrypt_SOAP_Request [Edit](#)

☐ Pre-process responses (Response processing must be enabled.)

Response Pre-Process Task List Group: Decrypt_SOAP_Request [Edit](#)

☒ Post-process responses (Response processing must be enabled.)

Response Post-Process Task List Group: Encrypt_SOAP_Response [Edit](#)

WEB SERVICES RELIABLE MESSAGING

☒ Enable reliable messaging

Reliable Messaging Policy: [Edit](#)

Save

12. If you now send the encrypted SOAP request from SOAPSonar again, Sentry will:
- decrypt the request
 - send the decrypted message to the back-end .NET Web Matrix service
 - receive a SOAP response
 - encrypt the SOAP Body of the response and return it to SOAPSonar
 - check the Sentry System log and confirm both the request and response processing
 - SOAPSonar is now sending an encrypted request and receiving an encrypted response

Input Data Success Criteria

Request

URI: http://127.0.0.1:84/SampleWS.asmx

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Header>

<wsse:Security soap:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-2004-01-wss-wssecurity-secext-2004-01-02.xsd">

<wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-2004-01-wss-wssecurity-secext-2004-01-02.xsd">

<wsu:Created>2014-09-19T07:25:22Z</wsu:Created>

</wsu:Timestamp>

<wsse:BinarySecurityToken Value="http://docs.oasis-open.org/wss/2004/01/oasis-2004-01-wss-wssecurity-secext-2004-01-02.xsd#BinarySecurityToken" ValueURI="http://www.w3.org/2001/04/xmlenc#uri">

<xenc:EncryptedKey xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">

</wsse:Security>

</soap:Header>

<soap:Body>

<xenc:EncryptedData Id="id-b96c6e38fb94625184caa3caecc4d6be90c4f808" Type="http://www.w3.org/2001/04/xmlenc#data">

<xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc">

<xenc:CipherData>

<xenc:CipherValue>4MEZCe0xnmHfUsC805qRuWfXp51etFCntVc3oF++jca/Te9DDnPRJ

sB2Io4DP+ukJdcDrNW/5xIBzLBaV6dBos9tD1MWZ2wZfLytPfyCJ14cSy92+DScQtKy3hISQw==</xenc:CipherValue>

</xenc:CipherData>

</xenc:EncryptedData>

</soap:Body>

</soap:Envelope>

Schema Fields XML Authentication Attachments Tasks (1) Comments (Sent Request) Runtime Variables

Response (0.40s)

<xenc:DataReference URI="#id-b96c6e38fb94625184caa3caecc4d6be90c4f808"/>

</xenc:ReferenceList>

</xenc:EncryptedKey>

</wsse:Security>

</soap:Header>

<soap:Body>

<xenc:EncryptedData Id="id-b96c6e38fb94625184caa3caecc4d6be90c4f808" Type="http://www.w3.org/2001/04/xmlenc#data">

<xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc">

<xenc:CipherData>

<xenc:CipherValue>4MEZCe0xnmHfUsC805qRuWfXp51etFCntVc3oF++jca/Te9DDnPRJ

sB2Io4DP+ukJdcDrNW/5xIBzLBaV6dBos9tD1MWZ2wZfLytPfyCJ14cSy92+DScQtKy3hISQw==</xenc:CipherValue>

</xenc:CipherData>

</xenc:EncryptedData>

</soap:Body>

</soap:Envelope>

END

Additional Testing and More Reading

BACK IT UP!

It is recommended that you export your WSDL Policy and/or your full Sentry configuration after completing this lab. To export the WSDL policy, navigate to the WSDL Policies page, select the WSDL policy and use the GDM Export option to export the policy (and all dependencies) as a password encrypted FSG file. This can later be imported on the System→Configuration→Import/Export screen.

To export your full Sentry configuration, navigate to the System→Configuration→Import/Export screen and use the Export option in the center of the page to export the full Sentry configuration file as a password encrypted FSX file. This can later be imported on the same screen.

Backup your SOAPSonar project file by using the File→Save As option. All of your test cases will be saved in an .SSP file.

Additional Tests

1. What happens if you use encrypt on either side with a different key?
2. Configure different elements to be encrypted / decrypted by SOAPSonar and Sentry.
3. How does the .NET WebMatrix back-end service respond if Sentry does not remove the WS Security Header?
4. Add SSL to this policy to ensure data level and network layer privacy.

Additional Information

For more information on Task Lists see the following Helpdesk FAQs:

<https://helpdesk.forumsys.com.com/entries/95448796-FAQ-How-to-Workaround-Design-Time-Task-List-Errors>

<https://helpdesk.forumsys.com/entries/70324767-How-To-Configure-Identify-Document-Task-to-Match-Multiple-Values>

<https://helpdesk.forumsys.com/entries/39364583-How-To-Configuring-Case-Sensitivity-with-the-Identify-Document-Task-in-Forum-Sentry>

<https://helpdesk.forumsys.com/entries/76439018-How-To-Modifying-or-Filtering-on-XML-Element-Attribute-Values>

For more information, review the following Forum Sentry Admin Guide:

1. Network Policies Guide
2. WSDL Policies Guide
3. Task Management Guide
4. Security Policies and PKI Guide

About Forum Systems

Forum Systems is the global leader in API and Cloud Security technology with industry-certified, patented, and proven products deployed in the most rigorous and demanding customer environments worldwide. Forum Systems has been an industry leader for over 12 years and has built the core architecture of its technology on the foundation of FIPS 140-2 and NDPP. Forum Systems security-first mindset enables trusted, network edge deployments of its technology for protecting critical enterprise transactions.

Our product technology is purpose-built and designed for mission-critical, enterprise-class scalable solutions where business solutions require the modern day security and identity enforcement protection, while enabling a scalable architecture and low-latency, high-volume throughput.

Forum Systems supports global enterprise customers across industries in commercial, government, and military sectors. Forum Systems technology provides the leading-edge of modern-day cyber-security innovation with integrated identity and SSO features that enable out-of-the box business solutions with point-and-click technology.

Forum's patented; FIPS 140-2 and NDPP certified hardware and virtual products make modern-day business communications secure by actively protecting and accelerating data exchange and API service access across networks and business boundaries. For more information, please visit www.forumsys.com.