FORUM SYSTEMS™

MANAGING & ACCELERATING XML WEB SERVICES SECURITY

Forum Systems Hands-on Training

Lab 2. Introduction to SOAPSonar

FORUM SYSTEMS

A Crosscheck Networks Company

Published: September 2014
Forum Systems Hands-on Training – Lab 2. Introduction to SOAPSonar
D-ASF-SE-010029

Contents

Introduction

Lab 2. Introduction to SOAPSonar

Skill Level

This lab is beginner skill level. Little to no prior experience with Forum Sentry or SOAPSonar is required.

Prerequisites

This lab requires the Forum Sentry Training Image, with a licensed copy Sentry and SOAPSonar.

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 a back-end SOAP service, and references the “Sample Web Service” that is built in Lab 1 of this FS Sentry Training Series.

Lab Overview

Understanding the basics of testing a SOAP Web service is critical for both Sentry developers and administrators.

For developers, an enterprise class tool is required to build and generate complex request messages that will invoke and test the service deployed through Sentry. Typically these secured services will require SSL, authentication, encryption, signatures, WS Security tokens, etc. The web services client testing application needs to be able to support all of these.

For administrators, the ability to test the policies deployed in Sentry, even if you are not intimately familiar with how they were built, is critical for troubleshooting and validating the status of Sentry policies in production.

This lab will cover the basic functional testing of a SOAP web service using SOAPSonar Enterprise Edition. As the SOAPSonar tool will be used throughout these labs for testing of SOAP and REST policies, an introduction to the tool is essential.

This lab will provide instructions for importing a WSDL into SOAPSonar and testing the SOAP Web service. Topics will include:

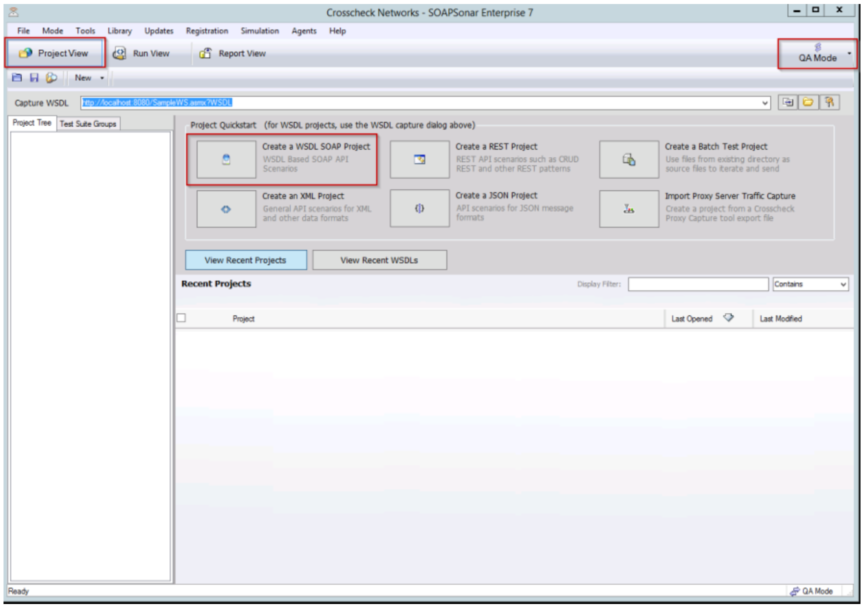
1. Basics of SOAPSonar
2. Functional Testing (QA Mode) in SOAPSonar
3. Success Criteria in SOAPSonar

Functional Testing – SOAPSonar QA Mode

Functional testing is the first pillar of SOA testing. Functional testing confirms if the service is working as expected.

Loading a WSDL

1. In SOAPSonar, test cases for functional testing are built in QA Mode under Project View.
2. After launching SOAPSonar, use the “Create a WSDL SOAP Project” option under the Project Quickstart menu.


Crosscheck Networks - SOAPSonar Enterprise 7
File
Mode
Tools
Library
Updates
Registration
Simulation
Agents
Help
Project View
Run View
Report View
QA Mode
New
Project Tree
Test Suite Groups
Project Quickstart (for WSDL projects, use the WSDL capture dialog above)
Create a WSDL SOAP Project
WSDL Based SOAP API Scenarios
Create a REST Project
REST API scenarios such as CRUD REST and other REST patterns
Create a Batch Test Project
Use files from existing directory as source files to iterate and send
Create an XML Project
General API scenarios for XML and other data formats
Create a JSON Project
API scenarios for JSON message formats
Import Proxy Server Traffic Capture
Create a project from a Crosscheck Proxy Capture tool export file
View Recent Projects
View Recent WSDLs
Recent Projects
Display Filter:
Contains
Project
Last Opened
Last Modified
Ready
QA Mode

Capture WSDL: http://localhost:8080/SampleWS.asmx?WSDL

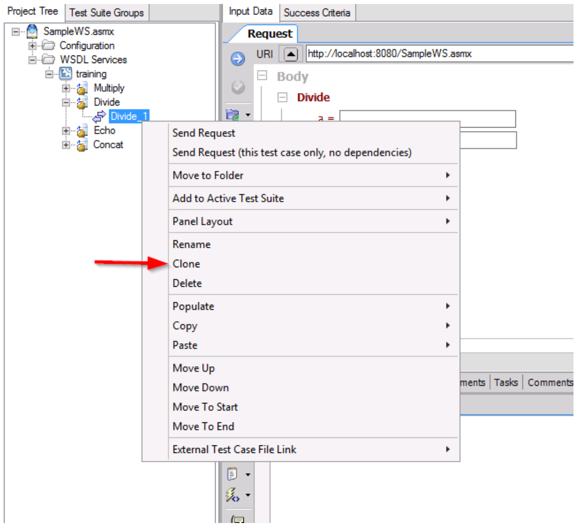
3. Enter the URI for the Sample web service WSDL file in the Capture WSDL field to load the WSDL into SOAPSonar.
4. After making any changes to the test cases, always be sure to click the ✓ button to commit the settings.
5. SOAPSonar projects are portable and can be saved using the File>>Save option.
6. When working with SOAP test cases, the data can be input on the Schema Fields tab, which provides a user friendly interface for the XML message being generated. You can view the actual XML along with HTTP Headers by clicking the XML tab.

Building Test Cases

As a part of this lab, we will set up and run a number of tests that make up a test suite in SOAPSonar. We will also learn how to configure success criteria to determine PASS/FAIL for a test case. Functional tests for the *Divide (int a, int b)* operation can easily be created in SOAPSonar as follows:

1. Configure two additional individual test cases in SOAPSonar for the operation as shown in the

Figure below. Right click on the Divide operation and setup three SOAP Tests using the Clone option. The result will be 3 Divide test cases.



2. Set the SOAP Request values for three test cases as listed below:
    a. *Divide_1: a=10, b=2*
    b. *Divide_2: a=10,b=100*
    c. *Divide_3: a=10,b=0*

Commit the values for Divide test cases by clicking ✓. Each request/operation can be executed by selecting the ➔ button within the request.

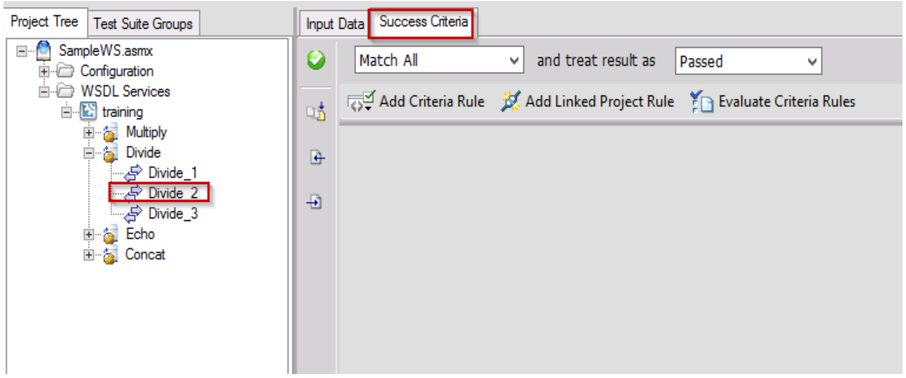
Success Criteria

For every SOAP request, the response from the web service must be evaluated. To determine whether a SOAP response is valid or invalid, a tester should setup pre-defined filters that examine HTTP return codes, SOAP faults, or any business specific content contained in a SOAP response.

In SOAPSonar you can build Success Criteria rules to provide a Pass or Fail grade based on the results of the web service call.

Success Criteria evaluation is typically done when running the multiple test cases in a Test Suite from Run View.

1. Click the *Success Criteria* Tab for test case *Divide_2*.



Project Tree | Test Suite Groups

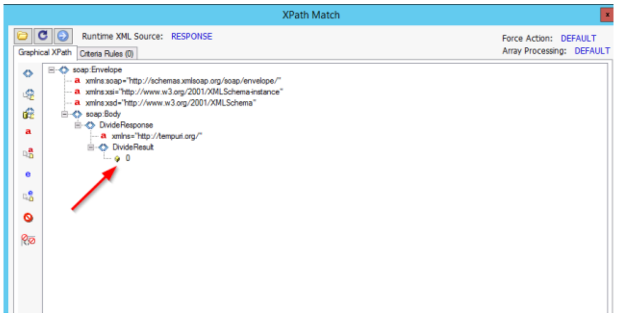
```
SampleWS.asmx
  Configuration
  WSDL Services
    training
      Multiply
      Divide
        Divide_1
        Divide_2
        Divide_3
      Echo
      Concat
```

Input Data | Success Criteria

Match All ▾ and treat result as Passed ▾

Add Criteria Rule | Add Linked Project Rule | Evaluate Criteria Rules

2. Select *Add Criteria Rule→Document →XPath Match*. Open the criteria rule and then right-click on the *DivideResult* value *0* node and select *Compare Element Value*. If the tree view shown below is not visible, click on to update the tree (this will send a request and display the response).



XPath Match

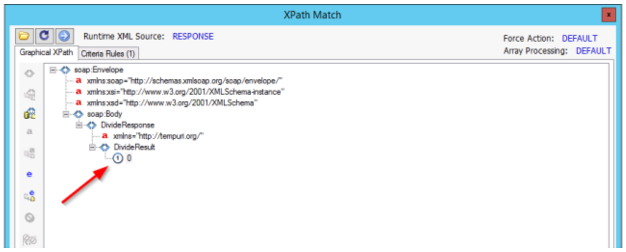
Runtime XML Source: RESPONSE

Force Action: DEFAULT
Array Processing: DEFAULT

Graphical XPath | Criteria Rules (0)

```
soap:Envelope
  a xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  a xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  a xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  soap:Body
    DivideResponse
      a xmlns="http://tempuri.org/"
      DivideResult
        0
```

After selecting *Compare Element Value*, the value will show up with a 1 next to it.



XPath Match

Runtime XML Source: RESPONSE

Force Action: DEFAULT
Array Processing: DEFAULT

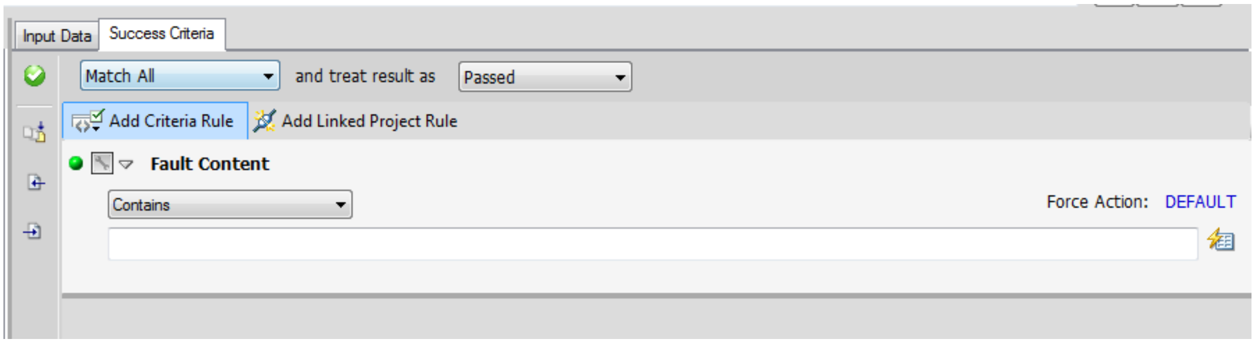
Graphical XPath | Criteria Rules (1)

```
soap:Envelope
  a xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  a xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  a xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  soap:Body
    DivideResponse
      a xmlns="http://tempuri.org/"
      DivideResult
        ① 0
```

3. For *Divide_3* test case, under the Success Criteria Tab, select *Add Criteria Rule→Document →*

SOAP Fault Match. By default this looks for SOAP Faults in the response.

For the *Divide_3* test case, we expect a SOAP Fault caused by a divide-by-zero exception. We set the result as a PASS because while a fault is returned, this is the expected behavior.



Match All — and treat result as — Passed

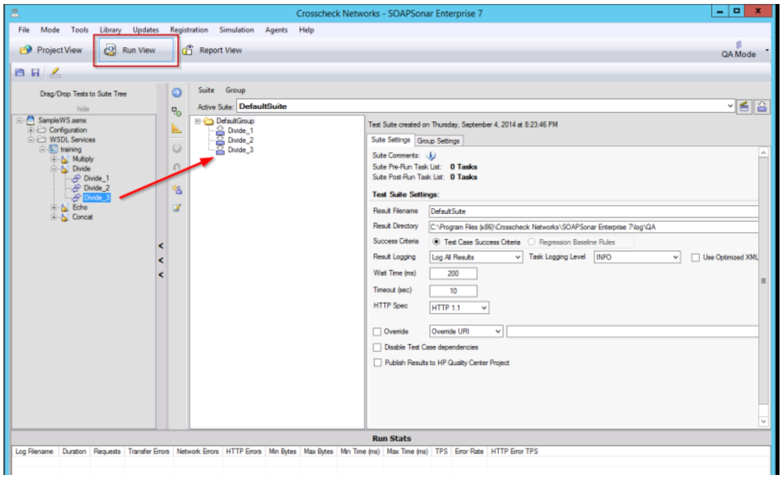
Fault Content

Contains

Force Action: DEFAULT

Note we are not setting a Success Criteria rule for the Divide_1 test case. The default success criteria is HTTP 200 = PASS and this will be used with the Divide_1 test case.

4. Once the test cases are defined, the user can select any combination of test cases and build a Test Suite. To build a test suite, click on the *Run View* as shown in the Figure below. From the left-most navigation panel, Drag-and-Drop test cases into the Default Test Suite. Test cases that appear under *DefaultGroup* are selected to run as the Default Suite.



Active Suite: DefaultSuite

Test Suite created on Thursday, September 4, 2014 at 8:23:46 PM

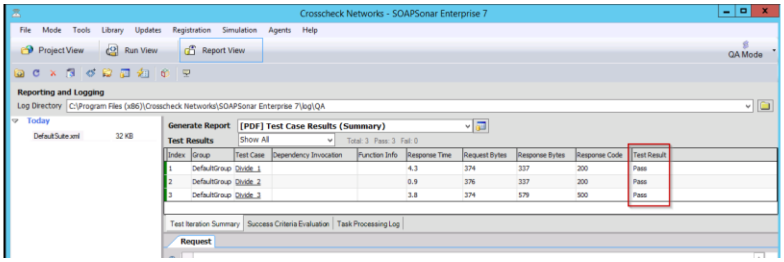
Suite Comments:
Suite Pre-Run Task List: **0 Tasks**
Suite Post-Run Task List: **0 Tasks**

Test Suite Settings:

Result Filename: DefaultSuite
Result Directory: C:\Program Files (x86)\Crosscheck Networks\SOAPSonar Enterprise 7\log\QA
Success Criteria: ● Test Case Success Criteria ○ Regression Baseline Rules
Result Logging: Log All Results — Task Logging Level: INFO — □ Use Optimized XML
Wait Time (ms): 200
Timeout (sec): 10
HTTP Spec: HTTP 1.1
□ Override — Override URI
□ Disable Test Case dependencies
□ Publish Results to HP Quality Center Project

Log Filename	Duration	Requests	Transfer Errors	Network Errors	HTTP Errors	Min Bytes	Max Bytes	Min Time (ms)	Max Time (ms)	TPS	Error Rate	HTTP Error TPS

5. Run the test cases in Run View by clicking ➔. When the tests finish, select *Analyze Results in Report View* on *the Real-time Run Monitor*. As seen in the *Report View* Panel below, all tests PASS (as seen in the Test Result Column) with the responses from the Divide operations as expected.



Reporting and Logging

Log Directory: C:\Program Files (x86)\Crosscheck Networks\SOAPSonar Enterprise 7\log\QA

DefaultSuite.xml 32 KB

Generate Report [PDF] Test Case Results (Summary)

Test Results Show All — Total: 3 Pass: 3 Fail: 0

Index	Group	Test Case	Dependency Invocation	Function Info	Response Time	Request Bytes	Response Bytes	Response Code	Test Result
1	DefaultGroup	Divide_1			4.3	374	337	200	Pass
2	DefaultGroup	Divide_2			0.9	376	337	200	Pass
3	DefaultGroup	Divide_3			3.8	374	579	500	Pass

6. Trigger a failure following the steps below.

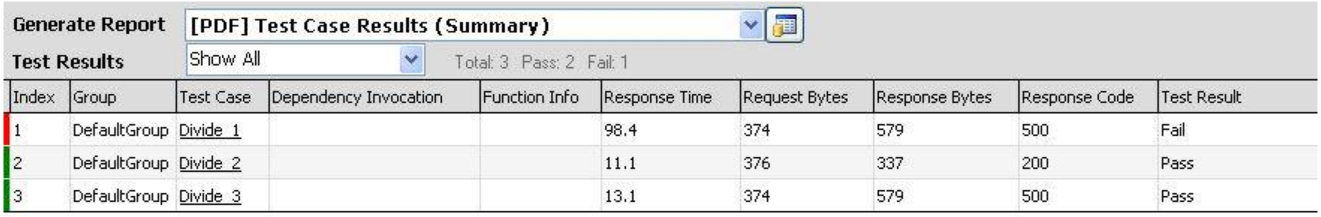
a. Go back to the *Project View* and change the value for the *Divide_1* test case to *a=10* and *b=0*. Make sure you commit the changes by clicking ✓ button.

b. Go to the *Run View* Panel and run the Default Test Suite again. Notice 1 failure in the Realtime Run Monitor.

c. Click *Analyze Results in Report View* on the Real-time Run Monitor.

As shown in the Figure below, you should see two *Pass* and one *Fail* value in the *Test Result column* with *the Divide_1* Test failing.

For Divide_1, we did not set a Success Criteria. The default success criterion is HTTP 200 = PASS, however for a SOAP Fault, the server returns HTTP 500, therefore resulting in a FAIL for the *Divide_1* test case. For the same inputs, *Divide_3* results in PASS.



Generate Report [PDF] Test Case Results (Summary)

Test Results Show All Total: 3 Pass: 2 Fail: 1

Index	Group	Test Case	Dependency Invocation	Function Info	Response Time	Request Bytes	Response Bytes	Response Code	Test Result
🟥 1	DefaultGroup	Divide_1			98.4	374	579	500	Fail
🟩 2	DefaultGroup	Divide_2			11.1	376	337	200	Pass
🟩 3	DefaultGroup	Divide_3			13.1	374	579	500	Pass

This concludes Lab 2. You should now be familiar with loading a WSDL into SOAPSonar and building functional test cases, running them in Project View and Run View and building Success Criteria Rules to provide a pass or fail grade for each test case.

Unlike web-site testing, you cannot rely on HTTP codes alone for web services testing. Deep content inspection is critical for evaluating whether a service is behaving as expected.

Later labs in this series will cover using SSL, authentication, and security parameters with the SOAP requests generated in SOAPSonar.

END

Additional Testing and More Reading

BACK IT UP!

It is recommended that you save SOAPSonar project after completing this lab.

To export the SOAPSonar configuration, use the File→Save As option. The SOAPSonar configuration is saved as an .SSP file. Saving the configuration file(s) after each lab is recommended. In some cases you may choose to use new SOAPSonar project files for new labs.

We recommend including the lab number in the name of the saved project files.

Additional Tests and Discussion Topics

1. SOAPSonar can be used for automated "end to end" regression testing providing a valuable tool for both service and gateway monitoring.
2. Explore the various modes in SOAPSonar including Vulnerability Mode for SOAP services.

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.