OWASP
The Open Web Application Security Project

OWASP ESAPI for Java EE 2.0a

Installation Guide

alpha

This page is intentionally blank

Foreword

This document provides instructions for installing version 2.0a of the Java EE language version of the OWASP Enterprise Security API (ESAPI). OWASP ESAPI toolkits help software developers guard against security-related design and implementation flaws. Just as web applications and web services can be Public Key Infrastructure (PKI) enabled (PK-enabled) to perform for example certificate-based authentication, applications and services can be OWASP ESAPI-enabled (ES-enabled) to enable applications and services to protect themselves from attackers.

We'd Like to Hear from You

Further development of ESAPI occurs through mailing list discussions and occasional workshops, and suggestions for improvement are welcome. Please address comments and questions concerning the API and this document to the ESAPI mail list, owasp-esapi@lists.owasp.org

Copyright and License

This page is intentionally blank

Table of Contents

This page is intentionally blank

1 About ESAPI for Java EE

ESAPI for Java EE can be installed and integrated with your application code in a number of ways, depending on your existing workflow. Approaches covered in this guide are:

- Option 1: Using Maven2
- Option 2: Using Ant
- Option 3: Using an IDE
    - Eclipse 3.2 or newer
    - NetBeans 6.TODO or newer
    - IntelliJ Idea TODO or newer

The ESAPI for Java EE 2.0a distribution can be obtained from the following sources:

Pre-Built Jar	The current version of ESAPI for Java is available in the "Featured Downloads" section of the owasp-esapi-java project on Google Code: http://code.google.com/p/owasp-esapi-java/ As of this writing, the latest version is 2.0rc2 (ESAPI-2.0rc2.jar), with the official 2.0 release to come approximately in January 2010 (TODO: verify).
Maven Repository	ESAPI for Java is not yet available from a public maven repository. TODO: Eventually at http://oss.sonatype.org/content/repositories/googlecode-snapshots/org/owasp/
Building From Source	Building ESAPI is beyond the scope of this guide, but information is available at: http://www.owasp.org/index.php/ESAPI-Building

The ESAPI for Java EE 2.0a distribution media contains the following:

- The Java archive (.jar) files comprising the ESAPI for Java EE toolkit.
- Sample code.
- Product documentation consisting of:
    - This document, the *OWASP ESAPI for JavaEE Installation Guide*, in PDF, with instructions on how to install and build ESAPI for Java EE.
    - The *OWASP ESAPI for JavaEE Release Notes*, in PDF, with the latest information on ESAPI for Java EE.
    - The *OWASP ESAPI for JavaEE Javadoc*, in HTML format.

2 Prerequisites

Before you start the installation, ensure that:

- You have read these installation instructions.
- You have installed Java 1.5 SDK or above.
- You have installed Java EE jar files compatible with your Java SDK (e.g., Java EE 5 for Java 1.5 SDK), or a Java EE-enabled version of your IDE

This page is intentionally blank

3 Installation

3.1 Distribution Directory Structure

The following describes the ESAPI for Java EE distribution structure.

Directory	Content
`<root>/`	
`JavaEE-ESAPI_2.0a_install. pdf`	ESAPI install guide
`JavaEE-ESAPI_2.0a_ReleaseNotes.pdf`	ESAPI release notes
`Readme.txt`	ESAPI readme
`License.txt`	ESAPI license
`esapi.jar`	ESAPI JAR
`esapi.properties`	ESAPI configuration file
`log4j.properties`	Log4j configuration file
`doc/`	ESAPI documentation
`java/`	ESAPI source code
`src/`	
`lib/`	ESAPI dependencies

Todo – add sample code to the above – swingset?

The ESAPI JAR contains the following:

- The Java binary (`.class`) files of the ESAPI interfaces
- The Java binary (`.class`) files of the ESAPI provider reference implementations
- A configuration file (`ESAPI.properties`) file that controls which implementation classes will provide functionality for an ESAPI installation as well as many other configuration parameters. This file comes configured to use the default ESAPI reference implementations, which can be extended or replaced by custom implementations as needed.
- A Maven 2 Project Object Model (`pom.xml`) file indicating the dependencies of ESAPI for Java

3.2 Installation Using Maven2

1. Add the following stanza to your POM file:

```
<dependencies>
    …
    <dependency>
        <groupId>OWASP</groupId>
        <artifactId>ESAPI</artifactId>
        <version>2.0</version>
    </dependency>
    …
</dependencies>
```

2. ESAPI is not yet available from a standard public repository (TODO, ETA?), so you will need to add the ESAPI jar to your local machine or site repository.
    a. Get an ESAPI jar using directions in Section 3.
    b. Run the following command to add the ESAPI jar to your local developer maven2 repository:

```
mvn install:install-file -DgroupId=OWASP -DartifactId=ESAPI -Dversion=2.0 -
Dpackaging=jar -Dfile=ESAPI-2.0rc2.jar
```

    c. Additionally, if you host your own internal repository, you can add ESAPI to it using:

```
mvn deploy:deploy-file -DgroupId=OWASP -DartifactId=AntiSamy -Dversion=1.2 -
Dpackaging=jar -Dfile= ESAPI-2.0rc2.jar -Durl=your_repo_url -
DrepositoryId=[your_repo_id]
```

3. Extract ESAPI.properties and validation.properties from the ESAPI jar and copy them both in the the directories `src/main/resources` and `src/test/resources`. (Note: this will create two separate copies.) If you prefer and are able to use the same versions for development and testing, you can copy them to one directory and then link them to the other directory. In this way, the two copies will not become out-of-sync.)

3.3 Installation Using Ant

TODO

3.4 Installation Using Eclipse

Step 1 Add the ESAPI Jar to the classpath. In Project > Properties > Java Build Path > Libraries use "Add JARS…" if the ESAPI jar is part of your project directory structure (e.g., checked into source control with your project) or "Add External JARS" if you maintain a separate directory of jar dependencies.

Step 2 Extract ESAPI.properties and validation.properties from the ESAPI jar and copy them somewhere that will be available to Run and Debug Configurations

Installation Tip:

- A reasonable default location during development is inside a "`.esapi`"folder in your user directory.

Step 3 If you elected to place the ESAPI.properties and validation.properties somewhere other than your user home directory, you will need to provide the directory via a VM argument.

Installation Tips:

- In Run > Run Configuration (or Debug Configuration), on the Arguments Tab, add to VM Arguments: -Dorg.owasp.esapi.resources=".esapi" Where ".esapi" is the absolute or relative path of the directory containing ESAPI.properties and validation.properties.
- To include ESAPI in all run configurations: in Preferences > Java > Installed JREs > Edit, add: -Dorg.owasp.esapi.resources=".esapi" Where ".esapi" is the absolute or relative path of the directory containing ESAPI.properties and validation.properties

3.5 Installation Using NetBeans

Step 1 Add the ESAPI Jar to the classpath: right-click the project, choose Properties, then under Categories choose Libraries.

Installation Tips:

- If you use a shared Libraries Folder, simply make copy the ESAPI jar into the directory specified by Libraries Folder.
- Otherwise on the Compile tab, click AddJAR/Folder and navigate to the ESAPI jar.

Step 2 Extract ESAPI.properties and validation.properties from the ESAPI jar and copy them somewhere that will be available to Run

and Debug Configurations.

Installation Tips:

- A reasonable default location during development is inside a ".esapi"folder in your user directory.
- See Section TODO for information on how ESAPI locates its configuration file.

Step 3 If you elected to place the ESAPI.properties and validation.properties somewhere other than your user home directory, you will need to provide the directory via a VM argument.

Installation Tips:

- In Run > Set Project Configuration > Customize, in the VM Options field: -Dorg.owasp.esapi.resources=".esapi" Where ".esapi" is the absolute or relative path of the directory containing ESAPI.properties and validation.properties.

3.6 Installation Using IDEA

TODO

4 Configuration

4.1 Initial Configuration

There is initial configuration that should be done regardless of application or deployed environment. <more details summarizing>

Step 1 The default logging facility in ESAPI can use either log4j or Java logging (i.e.,the classes in java.util.logging). By default, ESAPI.properties is configured to use log4j. If you do not use log4j, locate the the two "ESAPI.Logger" lines in ESAPI.properties and comment out the ESAPI reference logger that uses log4j and uncomment out the one for JavaLogFactory. That section of your ESAPI.properties should look like this:
Log4JFactory Requires log4j.xml or log4j.properties in classpath - http://www.lalilуna.de/log4j-tutorial.html
#ESAPI.Logger=org.owasp.esapi.reference.Log4JLogFactory
ESAPI.Logger=org.owasp.esapi.reference.JavaLogFactory

Step 2 You MUST replace the ESAPI Encryptor.MasterKey and Encryptor.MasterSalt in ESAPI.properties with ones you personally generate. By default, the ESAPI.properties file has neither of these set and therefore any many encryption related things will fail until you properly set them. Change them now by using:
cd <directory containing ESAPI jar>
java -classpath ESAPI-2.0rc2.jar
org.owasp.esapi.reference.JavaEncryptor

The final lines of output from this will look something like:
Copy and paste this into ESAPI.properties

```
Encryptor.MasterKey=<something here>
Encryptor.MasterSalt=<something here>
```

Simply take the two generated entries and paste them into your ESAPI.properties, replacing the empty ones already there. These are the unique key and salt for your ESAPI installation.

Step 3 In any deployed context you should make sure to restrict file permissions on the ESAPI.properties file. Since tampering with or unauthorized read access of this file could subvert the choice of security implementation, the ESAPI.properties file becomes a key part of your security stance. You and your team can share a common ESAPI.properties file for development and testing, but your team

should insist on generating new Encryptor.MasterKey and Encryptor.MasterSalt values using the same manual steps described above once your application that is using ESAPI goes into production. From that point, make sure that you use your operating system protection (especially in your production environment) to restrict read and write access only to your application and possibly to your production support personnel on a need-to-know basis. Details of how to do this are beyond the scope of this installation document.

Step 4 If you will be using the reference implementations provided with ESAPI, there are additional dependencies you must provide in your project. (For Maven users, the ESAPI pom.xml will include them automatically as transitive dependencies)

For DefaultAccessController:

commons-configuration.jar:

```
http://www.ibiblio.org/maven/commons-configuration/jars/commons-
configuration-1.5.jar
```

commons-lang.jar:

```
 http://commons.apache.org/downloads/download_lang.cgi
```

commons-collections.jar

```
http://www.ibiblio.org/maven/commons-collections/jars/commons-
collections-3.2.jar
```

ESAPI-AccessControlPolicy.xml

TODO

For DefaultValidator:

AntiSamy 1.3:

```
http://owaspantisamy.googlecode.com/files/antisamy-bin.1.3.jar
```

NekoHTML 0.9.5:

```
http://sourceforge.net/projects/nekohtml/files/nekohtml/nekohtml-
1.9.13/nekohtml-1.9.13.zip/download
```

Xerces 2.9.1:

```
http://mirror.atlanticmetro.net/apache/xerces/j/Xerces-J-bin.2.9.1.zip
```

For Log4JLogFactory logger:

Log4j 1.2.12:

```
http://logging.apache.org/log4j/1.2/download.html
```

For DefaultHTTPUtilities:

Commons-FileUpload 1.2:

```
http://commons.apache.org/downloads/download_fileupload.cgi
```

Step 5 To test if ESAPI has been successfully integrated and configured, create a file called EsapiIntegrationTest.java and paste in:

```
import org.owasp.esapi.ESAPI;

public class EsapiTest {

        public static void main(String[] args)
        {
                System.out.println("ESAPI.accessController found: "
                                + ESAPI.accessController());
        }
}
```

If you can run this file and see the println output, then ESAPI has been successfully installed and configured! You can now begin using ESAPI functionality to secure your web applications!

4.2 Configuration Checklists

There is additional configuration that should be as ESAPI security controls are added into your application. <more details summarizing>

4.2.1 ESAPI.properties Checklist

Property	Setting
`ESAPI.AccessControl`	The default is org.owasp.esapi.reference.DefaultAccessController. This should be changed when <todo>

Todo
Todo
Todo
Todo

This page is intentionally blank

5 Where to Go From Here

OWASP is the premier site for Web application security. The OWASP site hosts many projects, forums, blogs, presentations, tools, and papers. Additionally, OWASP hosts two major Web application security conferences per year, and has over 80 local chapters. The OWASP ESAPI project page can be found here http://www.owasp.org/index.php/ESAPI

The following OWASP projects are most likely to be useful to users/adopters of ESAPI:

- OWASP Application Security Verification Standard (ASVS) Project - http://www.owasp.org/index.php/ASVS
- OWASP Top Ten Project - http://www.owasp.org/index.php/Top_10
- OWASP Code Review Guide - http://www.owasp.org/index.php/Category:OWASP_Code_Review_Project
- OWASP Testing Guide - http://www.owasp.org/index.php/Testing_Guide
- OWASP Legal Project - http://www.owasp.org/index.php/Category:OWASP_Legal_Project

Similarly, the following Web sites are most likely to be useful to users/adopters of ESAPI:

- OWASP - http://www.owasp.org
- MITRE - Common Weakness Enumeration – Vulnerability Trends, http://cwe.mitre.org/documents/vuln-trends.html
- PCI Security Standards Council - publishers of the PCI standards, relevant to all organizations processing or holding credit card data, https://www.pcisecuritystandards.org
- PCI Data Security Standard (DSS) v1.1 - https://www.pcisecuritystandards.org/pdfs/pci_dss_v1-1.pdf

This page is intentionally blank

This page is intentionally blank

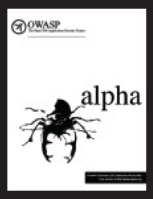
This page is intentionally blank

THE BELOW ICONS REPRESENT WHAT OTHER VERSIONS ARE AVAILABLE IN PRINT FOR THIS TITLE BOOK.

ALPHA: "Alpha Quality" book content is a working draft. Content is very rough and in development until the next level of publication.

BETA: "Beta Quality" book content is the next highest level. Content is still in development until the next publishing.

RELEASE: "Release Quality" book content is the highest level of quality in a books title's lifecycle, and is a final product.



ALPHA
PUBLISHED

YOU ARE FREE:

to share - to copy, distribute and transmit the work

to Remix - to adapt the work

UNDER THE FOLLOWING CONDITIONS:

Attribution. You must aatribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Share Alike. - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

OWASP
The Open Web Application Security Project

The Open Web Application Security Project (OWASP) is a worldwide free and open community focused on improving the security of application software. Our mission is to make application security "visible," so that people and organizations can make informed decisions about application security risks. Everyone is free to participate in OWASP and all of our materials are available under a free and open software license. The OWASP Foundation is a 501c3 not-for-profit charitable organization that ensures the ongoing availability and support for our work.