



SUBPROGRAMA AVANZA I+D 2010

VulneraNET

TSI-020100-2010-966

Guía para desarrollar plugins que permitan usar herramientas de seguridad desde VulneraNET

VulneraNET

Herramientas y Procesos Colaborativos de Detección, Predicción y Corrección de Vulnerabilidades de aplicaciones web para desarrolladores y auditores de seguridad

VULNERANET



Germinus XXI

Universidad Politécnica de Madrid

Universidad Carlos III de Madrid

Skyworks Media Group SL

Mántica Solutions



RESUMEN EJECUTIVO

Este documento es una guía del desarrollador para que cualquier desarrollador pueda realizar un plugin de una herramienta de seguridad para incluirla en VulneraNET.



Información del Documento

| | | | |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------------|
| Proyecto FIT Número | TSI-020100-2010-966 | Acrónimo | VulneraNET |
| Título completo | Herramientas y Procesos Colaborativos de Detección, Predicción y Corrección de Vulnerabilidades de aplicaciones web para desarrolladores y auditores de seguridad | | |
| URL | http://vulneranet.grupogesfor.com/ | | |
| URL del documento | | | |

| | | | | |
|---------------------------|---------------|------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Entregable | Número | D2.5 | Título | Guía para desarrollar plugins que permitan usar herramientas de seguridad desde VulneraNET. |
| Paquete de Trabajo | Número | PT2 | Título | Especificación de requisitos y definición de una arquitectura extensible de un panel de control para herramientas de seguridad para desarrolladores |
| Tarea | Número | T2.4 | Título | Diseño de la arquitectura de plugins para poder incluir distintas herramientas de seguridad en VulneraNET |



ÍNDICE

| | |
|---------------------------------------------------------------------------------|----|
| A. Introducción..... | 5 |
| B.Arquitectura..... | 6 |
| B.1 Introducción..... | 6 |
| B.2 Frameworks y bibliotecas utilizados..... | 6 |
| B.3 Visión general del sistema..... | 7 |
| B.4 Diseño de herramienta de seguridad..... | 9 |
| B.5 Diseño del entorno de ejecución de la herramientas de seguridad. | 14 |
| B.6 Diseño de clases de lectura y escritura de formato de intercambio | 16 |
| C.Integración de nuevas herramientas desde VulneraNET..... | 18 |
| C.1 Importación de informes desde VulneraNET..... | 18 |
| C.2 Integración de herramientas dentro de VulneraNET..... | 18 |
| C.2.1 Requisitos de las herramientas..... | 18 |
| C.2.2 Ejecución de las herramientas..... | 18 |
| C.2.3 Wapiti..... | 18 |
| C.2.4 Ejemplo de configuración del archivo de configuración para Wapiti..... | 25 |
| D.Índice de Ilustraciones..... | 30 |
| E.Referencias..... | 31 |



A. Introducción

Este documento tiene por objetivo servir como guía para los desarrolladores que deseen extender las funcionalidades de VulneraNET, bien sea añadiendo nuevas herramientas o mejorando el sistema.

En la primera parte del documento se explica la arquitectura de VulneraNET, detallando los paquetes y clases más importantes de la aplicación.

Posteriormente se explica de forma detallada como añadir una nueva herramienta de seguridad a VulneraNET con el ejemplo práctico de Wapiti.



B.Arquitectura

En esta sección se explica la arquitectura de la aplicación y las decisiones que se han tomado para realizar el diseño de la misma.

B.1 Introducción

El sistema ha sido concebido como una aplicación web, lo que quiere decir que la aplicación se aloja en un servidor web y es accesible a través de la red mediante un navegador web (Internet Explorer, Firefox, Chrome, etc.).

La plataforma elegida para la implementación y ejecución del sistema ha sido la plataforma Java. Para la ejecución de la aplicación es necesario su despliegue en un servidor de aplicaciones Java.

B.2 Frameworks y bibliotecas utilizados

La aplicación hace uso de diferentes *frameworks* que han facilitado la implementación de implementación.

- **Spring Framework [SprURL]**
 - Spring es un *framework* de desarrollo de aplicaciones para la plataforma Java.
 - Dentro de las facilidades que proporciona Spring, en este proyecto se ha utilizado la inyección de dependencias, de forma que los objetos en vez de crearlos la aplicación se inyectan en la propia aplicación a través de ficheros de configuración. Esto ha facilitado la creación de ficheros de configuración que posteriormente son mapeados como objetos Java dentro de VulneraNET.
 - Los patrones *factory* y *factory-method* han sido implementados usando archivos de Spring para configurar, según el caso, las diferentes factorías o clases disponibles para ser creadas.
- **Hibernate [HibURL]**
 - Hibernate es una herramienta para el mapeo objeto-relacional (ORM) que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación.
 - Este framework ha sido utilizado para realizar la capa de persistencia de VulneraNET, además permite usar casi cualquier sistema gestor de base de datos relacional sin cambiar el código de la aplicación.
- **Vaadin [VaaURL]**



- Vaadin es un *framework* Java para construir la interfaz gráfica de usuario de una aplicación web (Rich Internet Application).
- Este framework ha sido utilizado en VulneraNET para desarrollar la interfaz gráfica de usuario.

Otras bibliotecas utilizadas para el desarrollo de la aplicación fueron:

- ***XMLBeans [XMLURL]***
 - XMLBeans es una tecnología de acceso a XML mediante su unión a los tipos de Java.
 - Esta herramienta ha sido utilizada para leer y escribir el formato de intercambio de VulneraNET que se utiliza para intercambiar vulnerabilidades.
- ***Jasper Reports [JasURL]***
 - Es un motor de generación de informes, permite generar informes en formato HTML, PDF, Excel, etc con la información extraída de diferentes fuentes de datos.
 - Esta biblioteca ha sido usada para generar los informes en PDF que crea la aplicación con las vulnerabilidades detectadas.

B.3 Visión general del sistema

Estos los diferentes paquetes de VulneraNET. El diseño se ha realizado de forma modular y por capas.

La aplicación está dividida en 3 niveles:

1. Compuesto por la interfaz gráfica de usuario (paquete gui).
2. Clases que realizan la lógica de negocio (paquetes *tracker*, *securityTool*, *report* y *admin*).
3. Base de datos (paquete db).

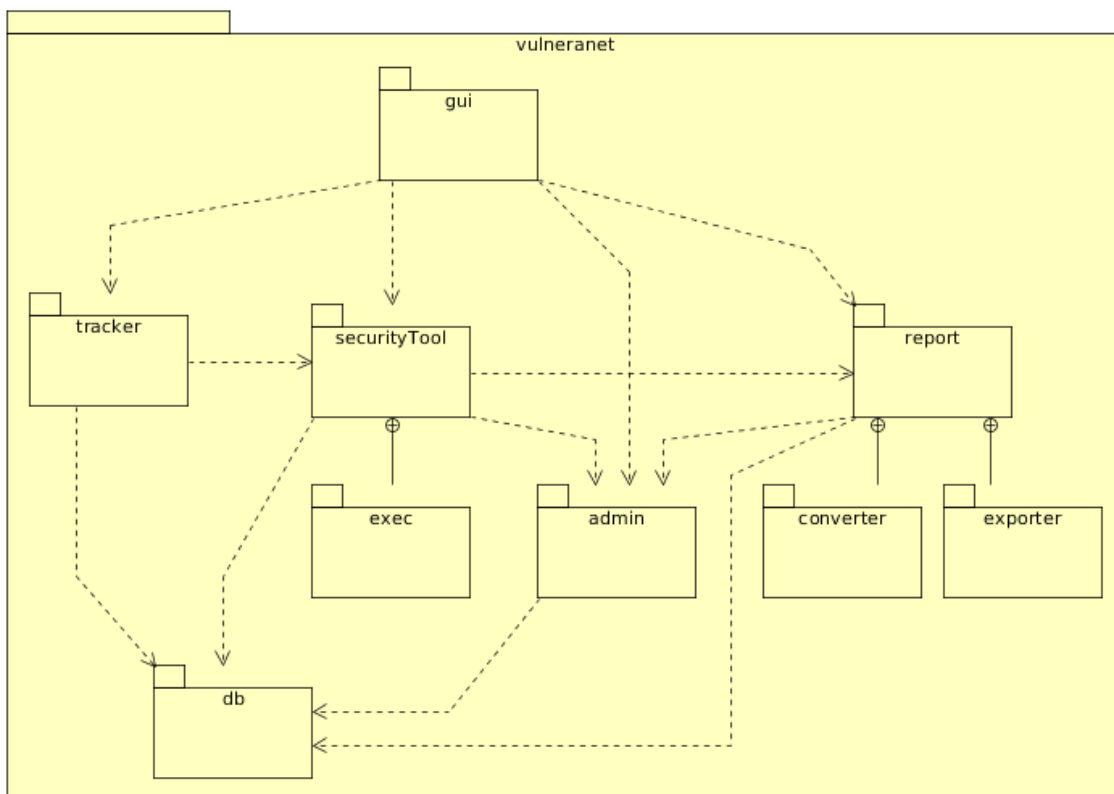


Ilustración 1: Diagrama de paquetes de VulneraNET

Descripción de los paquetes:

- **gui:** contiene las clases que genera la interfaz gráfica de usuario.
- **tracker:** contiene las clases relacionadas con el seguimiento de vulnerabilidades.
- **securityTool:** contiene las clases relacionadas con las herramientas de seguridad
 - **exec:** contiene las clases que se encargan de ejecutar una herramienta de seguridad.
- **report:** contiene las clases que representan un informe.
 - **converter:** contiene las clases que transforman un informe de objetos Java a XML y viceversa.
 - **exporter:** clases que generan un informe en PDF.
- **admin:** contiene los objetos relacionados con la administración de la aplicación, como por ejemplo la clase *Project* (proyecto) y *User* (usuario).
- **db:** clases relacionadas con la persistencia.

B.4 Diseño de herramienta de seguridad

El paquete securityTool de VulneraNET, contiene las clases que representan las herramientas de seguridad y está diseñado de la siguiente forma:

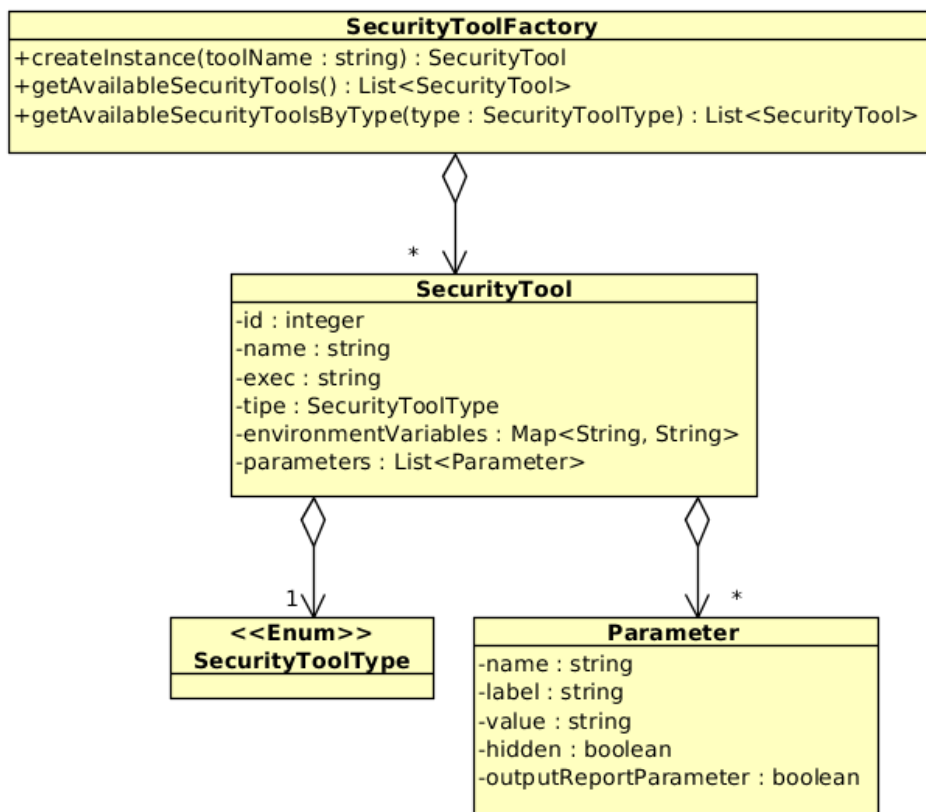


Ilustración 2: Diagrama de clases del paquete securityTool

Descripción de las clases:

- **SecurityToolFactory:** implementa el patrón Factory-method, es la clase sobre la que recae la responsabilidad de instanciar las diferentes herramientas de seguridad (SecurityTool) configuradas dentro de VulneraNET.
- **SecurityTool:** Contiene la configuración de una herramienta de seguridad. La responsabilidad de la creación de esta clase recae sobre **SecurityToolFactory** que es quien instancia las diferentes herramientas en base a su configuración. Para instalar una nueva herramienta en VuneraNET es necesario crear un fichero XML en el directorio *conf/tools* con el siguiente formato (es un fichero de definición de *beans* de SpringFramework):

La herramienta de seguridad tiene que exportar el resultado del análisis en el formato de intercambio de Wapiti (ver documento D2.2).



```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-
3.0.xsd">

  <bean id="wapiti"
class="es.gesfor.vulneranet.securitytool.SecurityTool"
scope="prototype">
  <property name="name" value="Wapiti" />
  <property name="exec" value="python" />
  <property name="type">
    <bean
class="es.gesfor.vulneranet.securitytool.SecurityToolType" factory-
method="valueOf">
      <constructor-arg>
        <value>BLACK_BOX</value>
      </constructor-arg>
    </bean>
  </property>
  <property name="environmentVariables">
    <map>
      <entry key="LANG">
        <value>es_ES.UTF-8</value>
      </entry>
      <entry key="HOME">
        <value>/home/wapity/</value>
      </entry>
      <entry key="PYTHONIOENCODING">
        <value>UTF-8</value>
      </entry>
    </map>
  </property>
  <property name="parameters">
    <list>
      <bean class="es.gesfor.vulneranet.securitytool.Parameter">
        <property name="value" value="/home/wapiti/src/wapiti.py"/>
        <property name="hidden" value="true" />
      </bean>
      <bean class="es.gesfor.vulneranet.securitytool.Parameter">
        <property name="label" value="URL" />
        <property name="mandatory" value="true" />
      </bean>
      <bean class="es.gesfor.vulneranet.securitytool.Parameter">
        <property name="name" value="-s" />
        <property name="label" value="Start URL" />
      </bean>
      <bean class="es.gesfor.vulneranet.securitytool.Parameter">
        <property name="name" value="-x" />
        <property name="label" value="Exclude URL" />
      </bean>
    </list>
  </property>
</bean>
</beans>
```



```
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-p" />
  <property name="label" value="Proxy" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-c" />
  <property name="label" value="Cookie File" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-t" />
  <property name="label" value="Timeout" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-a" />
  <property name="label" value="HTTP Authentication" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-r" />
  <property name="label" value="Remove URL Parameter" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-m" />
  <property name="label" value="Modules" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-v" />
  <property name="label" value="Verbosity" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-b" />
  <property name="label" value="Scope" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-f" />
  <property name="label" value="XML" />
  <property name="value" value="xml" />
  <property name="hidden" value="true" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-o" />
  <property name="label" value="File Name" />
  <property name="outputReportParameter" value="true" />
  <property name="hidden" value="true" />
</bean>
</list>
</property>
</bean>
</beans>
```



A continuación se explica cada aspecto a configurar, el texto que debe ser rellenado por el desarrollador se especifica en **negrita**. Es necesario configurar:

- **El nombre de la herramienta:** en el ejemplo "Wapiti"

```
<property name="name" value="Wapiti" />
```

- **El comando a ejecutar para la herramienta:** en el ejemplo "python"

```
<property name="exec" value="python" />
```

- **El tipo de herramienta:** caja negra (introducir BLACK_BOX), caja blanca (introducir WHITE_BOX). En el ejemplo BLACK_BOX (caja negra)

```
<property name="exec" value="python" />
  <property name="type">
    <bean class="es.gesfor.vulneranet.securitytool.SecurityToolType"
factory-method="valueOf">
      <constructor-arg>
        <value>BLACK_BOX</value>
      </constructor-arg>
    </bean>
  </property>
</property>
```

- **Variables de entorno:** Se debe incluir una entrada (entry) por cada variable de entorno que se quiera especificar como atributo *key* es el nombre de la variable de entorno y el texto del tag *value* es el valor de la variable de entorno.

```
<property name="environmentVariables">
  <map>
    <entry key="LANG">
      <value>es_ES.UTF-8</value>
    </entry>
    <entry key="HOME">
      <value>/home/wapity/</value>
    </entry>
    <entry key="PYTHONIOENCODING">
      <value>UTF-8</value>
    </entry>
  </map>
</property>
```



- **Parámetros de la aplicación:** estos parámetros se añadirán al comando de ejecución `exec`. El orden de los parámetros es importante será el orden con el que se deben pasar a la herramienta para su ejecución. Para añadir un parámetro es necesario crear un bean del tipo `es.gesfor.vulneranet.securitytool.Parameter` con las propiedades:
 - *name*, nombre del parámetro, se debe configurar para aquellos parámetros de la forma: “-f formatoDelFichero”, el nombre del parámetro será “-f” y el valor “formatoDelFichero” para parámetros que se pasen directamente sin indicar su nombre no es necesario configurar nombre alguno.
 - *value*: valor por defecto del parámetro
 - *label*: etiqueta del parámetro, será la forma de referirse al parámetro desde la interfaz gráfica.
 - *mandatory*: puede ser *true* o *false*, dependiendo de si el parámetro es obligatorio u opcional, *true* indica que es obligatorio y *false* que no.
 - *hidden*: puede ser *true* o *false*, dependiendo de si se quiere que se muestre o no el parámetro para que lo rellene el usuario, *true* indica que no se mostrará y *false* que si se mostrará.
 - *outputReportParameter*: propiedad especial, que si se especifica a *true* indica que es el parámetro al que se le indica el fichero que contendrá el formato de intercambio de VulneraNET para el análisis realizado por la herramienta.

```
<property name="parameters">
  <list>
    <bean class="es.gesfor.vulneranet.securitytool.Parameter">
      <property name="value" value="/home/wapiti/src/wapiti.py" />
      <property name="hidden" value="true" />
    </bean>
    <bean class="es.gesfor.vulneranet.securitytool.Parameter">
      <property name="label" value="URL" />
      <property name="mandatory" value="true" />
    </bean>
    <bean class="es.gesfor.vulneranet.securitytool.Parameter">
      <property name="name" value="-f" />
      <property name="label" value="XML" />
      <property name="value" value="xml" />
      <property name="hidden" value="true" />
    </bean>
    <bean class="es.gesfor.vulneranet.securitytool.Parameter">
      <property name="name" value="-o" />
      <property name="label" value="File Name" />
      <property name="outputReportParameter" value="true" />
      <property name="hidden" value="true" />
    </bean>
  </list>
</property>
```



```

</bean>
</list>
</property>
    
```

B.5 Diseño del entorno de ejecución de la herramientas de seguridad

Este paquete tiene el nombre es.vulneranet.securityTool.exec y se encarga de la ejecución de las herramientas de seguridad.

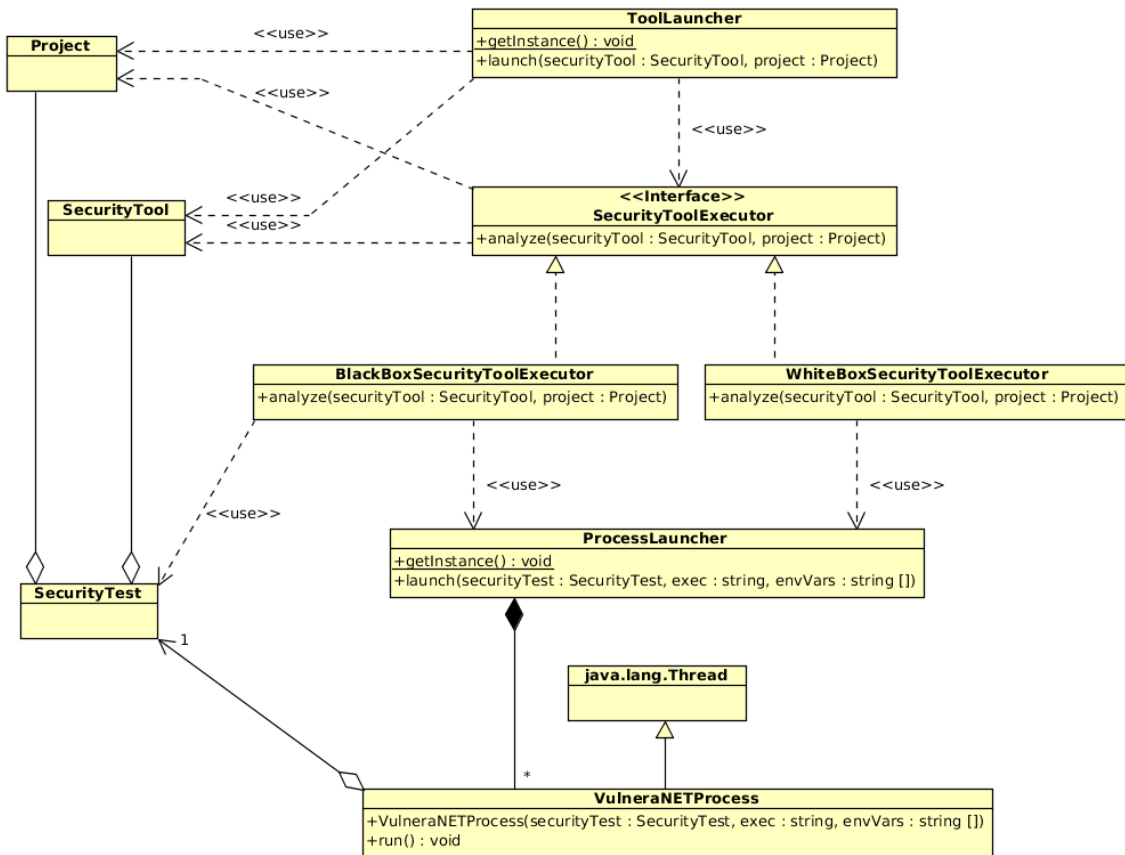


Ilustración 3: Diagrama de clases del paquete securityTool.exec

Resumen de las clases:

- ProcessLauncher:** Clase que implementa el patrón Singleton, su función es lanzar procesos a nivel de sistema operativo, para ello hace uso de la clase **VulneraNETProcess** que extiende la clase **Thread** y por ende implementa la interfaz **Runnable**. Esta clase lanza los procesos cuando se invoca al método **launch** que necesita los argumentos: **securityTest**, que es un objeto que contiene los detalles del test que se está realizando; **exec**, el comando a ejecutar y



envVars, las variables de entorno que se deseen establecer para la ejecución del comando.

- **ToolLauncher:** Clase que implementa el patrón *Singleton* y el patrón *Facade* (es la única clase con visibilidad pública del paquete). Mediante el método *launch* lee el fichero de configuración *conf/tools/executors.xml* y crea el *executor* que está configurado para la herramienta que se le pasa como parámetro (*securityTool*), sino hay ninguno configurado en dicho fichero de configuración se creará el de por defecto de caja negra (*BlackBoxSecurityToolExecutor*) o de caja blanca (*WhiteBoxSecurityToolExecutor*).

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-
3.0.xsd">

  <bean id="securityToolExecutors" class="java.util.HashMap">
    <constructor-arg>
      <map>
        <entry value-ref="executorId">
          <key>Wapiti</key>
        </entry>
      </map>
    </constructor-arg>
  </bean>

  <bean id="executorId"
class="es.gesfor.vulneranet.securitytool.exec.WapitiExecutor"/>
</beans>
```

- **SecurityToolExecutor:** Interfaz que deben implementar los *executor* de herramientas de seguridad. El método *analyze* debe generar hacer uso de la clase *ProcessLauncher*, dentro de ella se debe generar el comando de ejecución (*exec*) leyendo los parámetros de la herramienta de seguridad (clase *SecurityTool*) y las variables de entorno apropiadas (*envVars*). Además creará un objeto de tipo *SecurityTest* que tendrá una referencia al objeto *SecurityTool* que se le pasa.
- **BlackBoxSecurityToolExecutor:** Implementa el método *analyze* de la interfaz *SecurityToolExecutor* y genera el comando a ejecutar en base a la configuración de *SecurityTool*.
- **SecurityTool:** (ver B.4 Diseño de herramienta de seguridad).



B.6 Diseño de clases de lectura y escritura de formato de intercambio

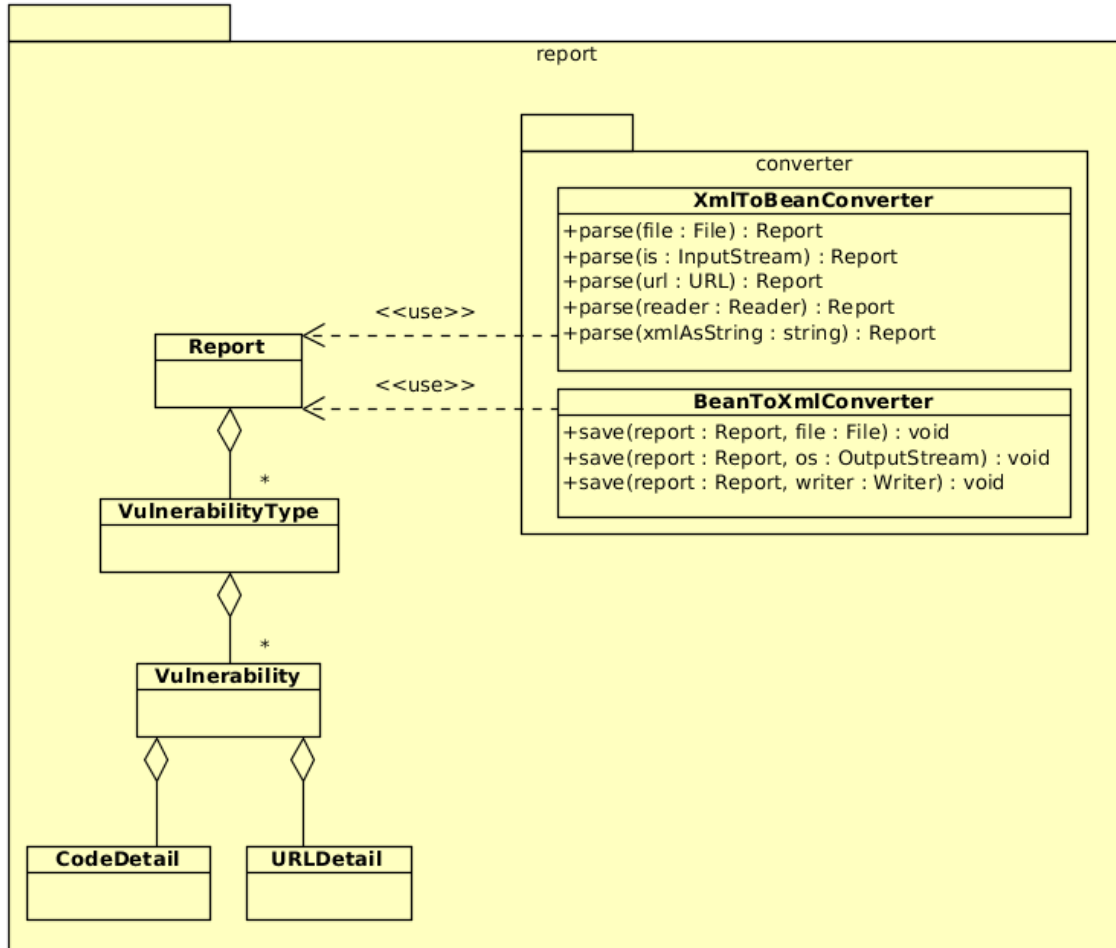
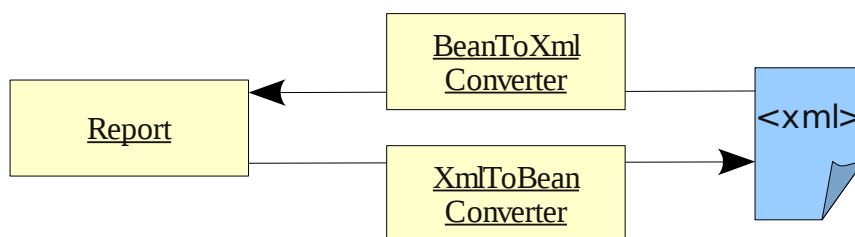


Ilustración 4: Paquete report.converter

Las clases que hacen la transformación de objetos a XML y viceversa son:

- es.gesfor.vulneranet.report.converter.XmlToBeanConverter
- es.gesfor.vulneranet.report.converter.BeanToXmlConverter





XmlToBeanConverter transforma el XML en un objeto de la clase Report, tiene métodos que admiten como entrada del XML un fichero, un input stream, una URL, un reader o que el XML esté directamente en un String:

- public Report parse(File file)
- public Report parse(InputStream is)
- public Report parse(URL url)
- public Report parse(Reader reader)
- public Report parse(String xmlAsString)

BeanToXmlConverter transforma un objeto de tipo Report en XML y lo guarda en un fichero, lo escribe en un output stream o un writer según el método:

- public void save(Report report, File file)
- public void save(Report report, OutputStream outputStream)
- public void save(Report report, Writer writer)

Las clases que representan el XML ya transformado están en el paquete es.gesfor.vulneranet.report.*

Un objeto **Report** contiene una lista de objetos **VulnerabilityType**, a su vez este objeto **VulnerabilityType** contiene objetos de tipo **Vulnerability**. **Vulnerability** (este objeto contiene los datos de una vulnerabilidad), a su vez este objeto **Vulnerability** contiene un objeto de tipo **CodeDetail** y otro de tipo **UriDetail**.

C. Integración de nuevas herramientas desde VulneraNET

C.1 Importación de informes desde VulneraNET

VulneraNET permite importar informes de otras herramientas siempre y cuando estén en el formato de intercambio de VulneraNET.

C.2 Integración de herramientas dentro de VulneraNET

La integración de herramientas permite que dichas aplicaciones sean lanzadas desde VulneraNET, realicen las pruebas de seguridad y posteriormente VulneraNET leerá el informe siempre que éste esté en el formato de intercambio de VulneraNET (definido en el documento D2.2)

C.2.1 Requisitos de las herramientas

Las herramientas que deseen ser integradas en VulneraNET deben tener los siguientes requisitos:

- Generar el informe en formato de intercambio de VulneraNET.
- Se debe poder especificar la ruta donde se quiere generar dicho informe.

C.2.2 Ejecución de las herramientas

La forma por defecto de lanzar las herramientas de caja negra y caja blanca se realiza mediante consola, con una llamada de Java `Runtime.getRuntime().exec()`.

Si se desea realizar otro tipo de invocación de la herramienta de forma diferente, por ejemplo, si la aplicación se ha realizado mediante Java y se puede integrar importando sus clases, se necesita implementar un nuevo *executor* que es la clase de la arquitectura de VulneraNET que debe invocar la herramienta (ver B.5 Diseño del entorno de ejecución de la herramientas de seguridad).

C.2.3 Wapiti

La herramienta Wapiti es un ejemplo de herramienta de caja negra que ha sido integrada en VulneraNET.

Wapiti se lanza desde consola de la siguiente forma:

```
python wapiti.py http://server.com/base/url/ [opciones]
```

Los parámetros que contiene son:

- `wapiti.py`:



- Primer parámetro, como está implementado en python el primer parámetro es el nombre del archivo .py
- URL
 - Segundo parámetro La URL sobre la que realizar el test.

Después de los parámetros vienen las opciones que son variables y cuyo orden no afecta a la ejecución del programa. Cada parámetro va junto a una clave para el parámetro que debe ser colocado antes de valor del parámetro. Por ejemplo, para especificar un *proxy* se debe colocar la clave para el parámetro “-p” y después poner un espacio y después el valor del *proxy*. Todos los parámetros van separados por un espacio entre si.

Los parámetros aceptados por Wapiti son los siguientes:

- -s <url>
 - Para especificar una URL con la que empezar
- -x <url>
 - Para excluir una URL del análisis (por ejemplo scripts de logout)
- -p <url_proxy>
 - Especifica un proxy
- -c <cookie_file>
 - Para usar una cookie
- -t <timeout>
 - Establece el tiempo del timeout (en segundos)
- -a <login%password>
 - Establece credenciales para autenticación HTTP
- -r <parameter_name>
 - Borra un parámetro de las URL
- -n <limit>
 - Define el límite de URL a leer con el mismo patrón
- -m <module_options>
 - Indica los módulos y métodos HTTP que van a ser usados en los ataques.
- -v <level>
 - Establece el nivel de logs por pantalla
- -b <scope>
 - Establece el ámbito del escaneo de Wapiti.



- -f <type_file>
 - Establece el tipo de informe
- -o <output>
 - Establece el nombre del informe

La forma del fichero de configuración es de la siguiente forma:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-
3.0.xsd">
  <bean id="wapiti"
class="es.gesfor.vulneranet.securitytool.SecurityTool"
scope="prototype">
  <property name="name" value="Wapiti" />
  <property name="exec" value="python" />
  <property name="type">
    <bean
class="es.gesfor.vulneranet.securitytool.SecurityToolType" factory-
method="valueOf">
      <constructor-arg>
        <value>BLACK_BOX</value>
      </constructor-arg>
    </bean>
  </property>
  <property name="environmentVariables">
    <map>
      <entry key="LANG">
        <value>es_ES.UTF-8</value>
      </entry>
      <entry key="HOME">
        <value>/home/wapiti/</value>
      </entry>
      <entry key="PYTHONIOENCODING">
        <value>UTF-8</value>
      </entry>
    </map>
  </property>
  <property name="parameters">
    <list>
      <bean class="es.gesfor.vulneranet.securitytool.Parameter">
        <property name="value" value="/home/wapiti/src/wapiti.py" />
        <property name="hidden" value="true" />
      </bean>
      <bean class="es.gesfor.vulneranet.securitytool.Parameter">
        <property name="label" value="URL" />
        <property name="mandatory" value="true" />
      </bean>
    </list>
  </property>
</bean>
```



```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-s" />
  <property name="label" value="Start URL" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-x" />
  <property name="label" value="Exclude URL" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-p" />
  <property name="label" value="Proxy" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-c" />
  <property name="label" value="Cookie File" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-t" />
  <property name="label" value="Timeout" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-a" />
  <property name="label" value="HTTP Authentication" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-r" />
  <property name="label" value="Remove URL Parameter" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-m" />
  <property name="label" value="Modules" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-v" />
  <property name="label" value="Verbosity" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-b" />
  <property name="label" value="Scope" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-f" />
  <property name="label" value="XML" />
  <property name="value" value="xml" />
  <property name="hidden" value="true" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-o" />
  <property name="label" value="File Name" />
  <property name="outputReportParameter" value="true" />
  <property name="hidden" value="true" />
</bean>
```



```

    </bean>
  </list>
</property>
</bean>
</beans>

```

A continuación se explica cada aspecto a configurar en el fichero, el texto que debe ser rellenado por el desarrollador se especifica en **negrita**. Es necesario configurar:

- **El nombre de la herramienta:** en el ejemplo "Wapiti"

```
<property name="name" value="Wapiti" />
```

- **El comando a ejecutar para la herramienta:** en el ejemplo "python"

```
<property name="exec" value="python" />
```

- **El tipo de herramienta:** caja negra (introducir BLACK_BOX), caja blanca (introducir WHITE_BOX). En el ejemplo BLACK_BOX (caja negra)

```

<property name="exec" value="python" />
  <property name="type">
    <bean class="es.gesfor.vulneranet.securitytool.SecurityToolType"
factory-method="valueOf">
      <constructor-arg>
        <value>BLACK_BOX</value>
      </constructor-arg>
    </bean>
  </property>
</property>

```

- **Variables de entorno:** Se debe incluir una entrada (entry) por cada variable de entorno que se quiera especificar como atributo *key* es el nombre de la variable de entorno y el texto del tag *value* es el valor de la variable de entorno. En el siguiente ejemplo se incluyen las variables de entorno:

- Variable *LANG* con valor *es_ES.UTF-8*, variable *HOME* con valor */home/wapiti* y la variable *PYTHONIOENCODING* con valor *UTF-8*.

```

<property name="environmentVariables">
  <map>
    <entry key="LANG">
      <value>es_ES.UTF-8</value>
    </entry>
  </map>
</property>

```



```
<entry key="HOME">
  <value>/home/wapiti/</value>
</entry>
<entry key="PYTHONIOENCODING">
  <value>UTF-8</value>
</entry>
</map>
</property>
```

- **Parámetros de la aplicación:** estos parámetros se añadirán al comando de ejecución `exec`. El orden de los parámetros es importante será el orden con el que se deben pasar a la herramienta para su ejecución. Para añadir un parámetro es necesario crear un bean del tipo `es.gesfor.vulneranet.securitytool.Parameter` con las propiedades:
 - `name`: nombre del parámetro, se debe configurar para aquellos parámetros de la forma: “-f formatoDelFichero”, el nombre del parámetro será “-f” y el valor “formatoDelFichero” para parámetros que se pasen directamente sin indicar su nombre no es necesario configurar nombre alguno.
 - `value`: valor por defecto del parámetro
 - `label`: etiqueta del parámetro, será la forma de referirse al parámetro desde la interfaz gráfica.
 - `mandatory`: puede ser `true` o `false`, dependiendo de si el parámetro es obligatorio u opcional, `true` indica que es obligatorio y `false` que no. Por defecto es `false`, y no es necesario definirlo a menos que se desee declarar con valor `true`.
 - `hidden`: puede ser `true` o `false`, dependiendo de si se quiere que se muestre o no el parámetro para que lo rellene el usuario, `true` indica que no se mostrará y `false` que si se mostrará. Por defecto es `false`, y no es necesario definirlo a menos que se desee declarar con valor `true`.
 - `outputReportParameter`: propiedad especial, que si se especifica a `true` indica que es el parámetro al que se le indica el fichero que contendrá el formato de intercambio de VulneraNET para el análisis realizado por la herramienta.

```
<property name="parameters">
  <list>
    <bean class="es.gesfor.vulneranet.securitytool.Parameter">
      <property name="value" value="/home/wapiti/src/wapiti.py" />
      <property name="hidden" value="true" />
    </bean>
    <bean class="es.gesfor.vulneranet.securitytool.Parameter">
      <property name="label" value="URL" />
    </bean>
  </list>
</property>
```



```
<property name="mandatory" value="true" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-f" />
  <property name="label" value="XML" />
  <property name="value" value="xml" />
  <property name="hidden" value="true" />
</bean>
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-o" />
  <property name="label" value="File Name" />
  <property name="outputReportParameter" value="true" />
  <property name="hidden" value="true" />
</bean>
</list>
</property>
```

La interfaz gráfica para rellenar los parámetros de Wapiti se mostrará de la siguiente forma:

Configuración herramienta caja negra: Wapiti

Parámetros

URL *

Start URL

Exclude URL

Proxy

Cookie File

Timeout

HTTP Authentication

Remove URL Parameter

Modules

Verbosity

Scope

Realizar

Ilustración 5: Pantalla para rellenar los parámetros de Wapiti



C.2.4 Ejemplo de configuración del archivo de configuración para Wapiti

Recordamos que Wapiti se lanza desde consola de la siguiente forma:

```
python wapiti.py http://server.com/base/url/ [opciones]
```

Los parámetros que contiene son:

- `wapiti.py`: Primer parámetro de la aplicación con lo cual se especificará el primero en la lista de parámetros. Este parámetro es fijo y no debe ser rellenado por los usuarios con lo cual se especifica `hidden=true` y no hace falta especificarle un `label` (no se mostrará en la interfaz de usuario).

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="value" value="/home/wapiti/src/wapiti.py" />
  <property name="hidden" value="true" />
</bean>
```

- URL: Segundo parámetro, es la URL sobre la que se realiza el test. Este parámetro debe ser rellenado por el usuario, con lo cual no se establece a `hidden` y se le pone el `label` URL para que el usuario que es el parámetro que está rellanando. Además como es obligatorio pues si no se rellena no funciona la aplicación (`mandatory=true`)

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="label" value="URL" />
  <property name="mandatory" value="true" />
</bean>
```

Además las opciones que Wapiti acepta son:

- `-s <url>` (
 - Especifica una URL con la que empezar. Lleva nombre (`name`) y etiqueta (`label`) porque es un campo que debe rellenar el usuario mediante la interfaz gráfica con lo cual no es oculto (`hidden` por defecto es `false` no hace falta especificarlo). No es obligatorio (`mandatory` por defecto es `false` no hace falta especificarlo).

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-s" />
  <property name="label" value="Start URL" />
</bean>
```

- `-x <url>`



- Excluye una URL del análisis (por ejemplo scripts de logout). Lleva nombre (*name*) y etiqueta (*label*) porque es un campo que debe rellenar el usuario mediante la interfaz gráfica con lo cual no es oculto (*hidden* por defecto es *false* no hace falta especificarlo). No es obligatorio (*mandatory* por defecto es *false* no hace falta especificarlo).

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">  
  <property name="name" value="-x" />  
  <property name="label" value="Exclude URL" />  
</bean>
```

- -p <url_proxy>
 - Especifica un proxy. Lleva nombre (*name*) y etiqueta (*label*) porque es un campo que debe rellenar el usuario mediante la interfaz gráfica con lo cual no es oculto (*hidden* por defecto es *false* no hace falta especificarlo). No es obligatorio (*mandatory* por defecto es *false* no hace falta especificarlo).

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">  
  <property name="name" value="-p" />  
  <property name="label" value="Proxy" />  
</bean>
```

- -c <cookie_file>
 - Para usar una cookie. Lleva nombre (*name*) y etiqueta (*label*) porque es un campo que debe rellenar el usuario mediante la interfaz gráfica con lo cual no es oculto (*hidden* por defecto es *false* no hace falta especificarlo). No es obligatorio (*mandatory* por defecto es *false* no hace falta especificarlo).

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">  
  <property name="name" value="-c" />  
  <property name="label" value="Cookie File" />  
</bean>
```

- -t <timeout>
 - Establece el tiempo del timeout (en segundos). Lleva nombre (*name*) y etiqueta (*label*) porque es un campo que debe rellenar el usuario mediante la interfaz gráfica con lo cual no es oculto (*hidden* por defecto es *false* no hace falta especificarlo). No es obligatorio (*mandatory* por defecto es *false* no hace falta especificarlo).

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">  
  <property name="name" value="-t" />
```



```
<property name="label" value="Timeout" />
</bean>
```

- -a <login%password>
 - Establece credenciales para autenticación HTTP. Lleva nombre (*name*) y etiqueta (*label*) porque es un campo que debe rellenar el usuario mediante la interfaz gráfica con lo cual no es oculto (*hidden* por defecto es *false* no hace falta especificarlo). No es obligatorio (*mandatory* por defecto es *false* no hace falta especificarlo).

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-a" />
  <property name="label" value="HTTP Authentication" />
</bean>
```

- -r <parameter_name>
 - Borra un parámetro de las URL. Lleva nombre (*name*) y etiqueta (*label*) porque es un campo que debe rellenar el usuario mediante la interfaz gráfica con lo cual no es oculto (*hidden* por defecto es *false* no hace falta especificarlo). No es obligatorio (*mandatory* por defecto es *false* no hace falta especificarlo).

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-r" />
  <property name="label" value="Remove URL Parameter" />
</bean>
```

- -n <limit>
 - Define el límite de URL a leer con el mismo patrón. Lleva nombre (*name*) y etiqueta (*label*) porque es un campo que debe rellenar el usuario mediante la interfaz gráfica con lo cual no es oculto (*hidden* por defecto es *false* no hace falta especificarlo). No es obligatorio (*mandatory* por defecto es *false* no hace falta especificarlo).

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-n" />
  <property name="label" value="Limit" />
</bean>
```

- -m <module_options>



- Indica los módulos y métodos HTTP que van a ser usados en los ataques. Lleva nombre (*name*) y etiqueta (*label*) porque es un campo que debe rellenar el usuario mediante la interfaz gráfica con lo cual no es oculto (*hidden* por defecto es *false* no hace falta especificarlo). No es obligatorio (*mandatory* por defecto es *false* no hace falta especificarlo).

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">  
  <property name="name" value="-m" />  
  <property name="label" value="Modules" />  
</bean>
```

- -v <level>
 - Establece el nivel de logs por pantalla. Lleva nombre (*name*) y etiqueta (*label*) porque es un campo que debe rellenar el usuario mediante la interfaz gráfica con lo cual no es oculto (*hidden* por defecto es *false* no hace falta especificarlo). No es obligatorio (*mandatory* por defecto es *false* no hace falta especificarlo).

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">  
  <property name="name" value="-v" />  
  <property name="label" value="Verbosity" />  
</bean>
```

- -b <scope>
 - Establece el ámbito del escaneo de Wapiti. Lleva nombre (*name*) y etiqueta (*label*) porque es un campo que debe rellenar el usuario mediante la interfaz gráfica con lo cual no es oculto (*hidden* por defecto es *false* no hace falta especificarlo). No es obligatorio (*mandatory* por defecto es *false* no hace falta especificarlo).

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">  
  <property name="name" value="-b" />  
  <property name="label" value="Scope" />  
</bean>
```

- -f <type_file>
 - Establece el tipo de informe. Lleva nombre (*name*) y valor (*value*) pero no se desea mostrar para ser rellenado por el usuario porque siempre tiene el mismo valor por tanto se debe poner *hidden* a verdadero (*true*).

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">  
  <property name="name" value="-f" />
```



```
<property name="value" value="xml" />
<property name="hidden" value="true" />
</bean>
```

- -o <output>
 - Establece el nombre del informe. Lleva nombre (*name*) y no se desea mostrar para ser rellenado por el usuario por lo que se pone *hidden* a verdadero (*true*). Este parámetro es especial, porque es el que usa VulneranET para indicar la ubicación donde Wapiti dejará el informe de vulnerabilidades por ello se rellena la propiedad *outputReportParameter* con *true*.

```
<bean class="es.gesfor.vulneranet.securitytool.Parameter">
  <property name="name" value="-o" />
  <property name="outputReportParameter" value="true" />
  <property name="hidden" value="true" />
</bean>
```

D.Índice de Ilustraciones

| | |
|----------------------------------------------------------------------|----|
| Ilustración 1: Diagrama de paquetes de VulneraNET..... | 8 |
| Ilustración 2: Diagrama de clases del paquete securityTool..... | 9 |
| Ilustración 3: Diagrama de clases del paquete securityTool.exec..... | 14 |
| Ilustración 4: Paquete report.converter..... | 16 |
| Ilustración 5: Pantalla para rellenar los parámetros de Wapiti..... | 24 |



E.Referencias

[HibURL] Sitio web de Hibernate,

[JasURL] Sitio web de Jasper Reports,
<http://jasperforge.org/projects/jasperreports>

[SprURL] Sitio web de Spring Framework, <http://www.springsource.org/about>

[VaaURL] Sitio web de Vaadin, <http://vaadin.com>

[XMLURL] Sitio web de XMLBeans, <http://xmlbeans.apache.org/>