SPOTBUGS

SpotBugs es un programa que se encarga de realizar análisis estáticos para buscar errores de java.
Es un tipo de software libre distribuido bajo los términos de la Licencia Pública General Reducida de GNU y requiere JRE (o JDK) 1.8.0 o posterior para ejecutarse, aunque puede analizar programas para cualquier versión de Java desde la 1.0 a la 1.9

Descripción de errores:

SpotBugs busca más de 400 patrones de errores, como son:

-Mala práctica: Las violaciones de la práctica de codificación recomendada y esencial, como son problemas de igualdad, excepciones eliminadas, código hash, uso indebido al finalizar,ect.

Ejemplos: JUA,CNT,NP,SW,FI,AM…

-Corrección: Un error de codificación que resultó en un código distinto a lo que el desarrollador pretendía

Ejemplos: CN,NP,VR,IL,IO,FL,RpC…

-Experimental: Patrones de errores experimentales y no examinados por completo

Ejemplos: PRUEBA,OBL

-Internalización: Defectos de código que tienen que ver con la ubicación e internalización

Ejemplos: Dm,DP,FI,MS,EI,MS…

-Vulnerabilidad de código malicioso: Código que es vulnerable a ser atacado o no es muy confiable

Ejemplos: DP,FI,MS,EI,EI2…

-Corrección multiproceso: Defectos de código relacionados con subprocesos, bloqueos y volátiles

Ejemplos: AT,STCAL,NP,VO…

-Ruido aleatorio falso: Es útil para controlar experimentos de minería de datos

Ejemplos: NOISE

-Rendimiento: Código que aunque sea correcto no es eficiente

Ejemplos: HSC,Dm,Bx…

-Seguridad: Una entrada no confiable que podría crear una vulnerabilidad de seguridad que se puede explotar de forma remota

Ejemplos: XSS,PT,HRS,SQL…

-Código dudoso: Código confuso o anómalo el cual conduce a errores, como las tiendas locales inactivas, cambio fallido, conversiones no confirmadas, ect

Ejemplos: CAA,Dm,RP,RV,Eq,NS…

Uso de SpotBugs:

Se usa de forma independiente y a través de varias integraciones, que incluyen:

-SpotBugs Ant task:

Para instalar la tarea Ant debemos copiar `$SPOTBUGS_HOME/lib/spotbugs-ant.jar` en el subdirectorio de su instalación Ant

Para incorporar SpotBugs en build.xml (el script de compilación para Ant), primero debe agregar una definición de tarea. Esto debería aparecer de la siguiente manera:

```
<taskdef
    resource="edu/umd/cs/findbugs/anttask/tasks.properties"
    classpath="path/to/spotbugs-ant.jar" />
```

Una vez que haya agregado la definición de la tarea, puede definir un objetivo que utilice la tarea Spotbugs.

El elemento spotbugs debe tener el atributo de inicio establecido en el directorio en el que está instalado SpotBugs; en otras palabras, $SPOTBUGS_HOME.

-SpotBugs Maven Plugin:

Para integrar Find Security Bugs en el complemento SpotBugs, puede configurar su pom.xml gusto a continuación

```
<build>
    <plugins>
        [...]
        <plugin>
            <groupId>com.github.spotbugs</groupId>
            <artifactId>spotbugs-maven-plugin</artifactId>
            <version>4.7.2.1</version>
            <configuration>

<includeFilterFile>spotbugs-security-include.xml</includeFilterFile>

<excludeFilterFile>spotbugs-security-exclude.xml</excludeFilterFile>
                <plugins>
                    <plugin>
                        <groupId>com.h3xstream.findsecbugs</groupId>
                        <artifactId>findsecbugs-plugin</artifactId>
                        <version>1.12.0</version>
                    </plugin>
                </plugins>
            </configuration>
        </plugin>
    </plugins>
</build>
```

La <plugins>opción define una colección de PluginArtifact para trabajar. Especifique "Buscar errores de seguridad" agregando su ID de grupo, ID de artefacto, versión.

Los <includeFilterFile>y <excludeFilterFile>especifican los archivos de filtro para incluir y excluir informes de errores, respectivamente (consulte Archivo de filtro para obtener más detalles).

-SpotBugs Gradle Plugin

Se puede usar de dos formas:

-Usando los complementos DSL :

```
plugins {
  id "com.github.spotbugs" version "5.0.13"
}
```

-Usando la aplicación de complemento heredada :

```
buildscript {
  repositories {
    maven {
      url "https://plugins.gradle.org/m2/"
    }
  }
  dependencies {
    classpath
"com.github.spotbugs.snom:spotbugs-gradle-plugin:5.0.13"
  }
}

apply plugin: "com.github.spotbugs"
```

-SpotBugs Eclipse Plugin:

Para usarse require Eclipse Neon (4.6) o posterior

Para comenzar, haga clic con el botón derecho en un proyecto de Java en Package Explorer y seleccione la opción denominada "Spot Bugs". SpotBugs se ejecutará y los marcadores de problemas (que se muestran en las ventanas de origen y también en la vista de problemas de Eclipse) apuntarán a ubicaciones en su código que se han identificado como instancias potenciales de patrones de errores.

También puede ejecutar SpotBugs en archivos java existentes (jar, ear, zip, war, etc.). Simplemente cree un proyecto Java vacío y adjunte archivos a la ruta de clases del proyecto. Con eso, ahora puede hacer clic derecho en el nodo de archivo en Package Explorer y seleccionar la opción etiquetada como "Spot Bugs". Si además configura las ubicaciones del código fuente para los archivos binarios, SpotBugs también vinculará las advertencias generadas a los archivos fuente correctos.

Puede personalizar cómo se ejecuta SpotBugs abriendo el cuadro de diálogo Propiedades para un proyecto Java y eligiendo la página de propiedades "SpotBugs". Las opciones que puede elegir incluyen:

- Active o desactive la casilla de verificación "Ejecutar SpotBugs automáticamente". Cuando está habilitado, SpotBugs se ejecutará cada vez que modifique una clase Java dentro del proyecto.
- Elija la prioridad mínima de advertencia y las categorías de errores habilitadas. Estas opciones elegirán qué advertencias se muestran. Por ejemplo, si selecciona la prioridad de advertencia "Media", solo se mostrarán las advertencias de prioridad Media y Alta. De manera similar, si desmarca la casilla de verificación "Estilo", no se mostrarán advertencias en la categoría Estilo.

- Seleccionar detectores. La tabla le permite seleccionar qué detectores desea habilitar para su proyecto.

Link de instalación: https://spotbugs.github.io/eclipse/

Extensiones:

A SpotBugs se le pueden agregar nuevos detectores a través de complementos, los complementos populares incluyen:

fb-contrib:

Es un complemento detector adicional que se utiliza con el buscador de errores estáticos FindBugs. Descargue el archivo fb-contrib.jar y colóquelo en la ubicación adecuada según cómo desee usarlo

FindBugs recogerá automáticamente el archivo jar e incorporará estos detectores con los suyos.

- Para ejecutar fb-contrib desde la interfaz gráfica de usuario o ant, simplemente coloque el archivo jar fb-contrib en el directorio de complementos dentro del directorio Findbugs™.
-
- Para ejecutar fb-contrib desde eclipse, suponiendo que el complemento principal de FindBugs esté instalado, simplemente suelte fb-contrib.jar en el directorio de complementos de eclipse y reinicie eclipse.

Requiere FindBugs 3.0.1 o superior

find-sec-bugs:

-Puede detectar 141 tipos de vulnerabilidades

-Cubre frameworks populares como Spring-MVC, Tapestry,etc

-Complementos disponibles para Eclipse,Netbeans, etc

-Se puede usar con sistemas como Jenkins y SonarQube

-Es de código abierto y está abierto a contribuciones

-Proporciona amplias referencias para cada patrón de error