Bifurcación de software libre, RCWUI desde OSSEC WebUI

Alzate, Jorge.
Jalzate801@unab.edu.co
Universidad Autónoma de Bucaramanga

Resumen----La bifurcación o *forking* de proyectos, es una de las mayores ventajas que ofrece la adaptación entre las características (libertades) que defiende el software libre. Los pesimistas o quizá los más realistas, por el contrario, opinan que la bifurcación es, en gran mayoría, una senda hacia el debilitamiento del proyecto inicial y la pérdida del esfuerzo de una parte de la comunidad que finalmente se plasma en el abandono de la bifurcación.

Este trabajo sintetiza el proceso que encierra la bifurcación de la interfaz de usuario WebUI (*Web User Interface*) para la administración de la información generada por el sistema de detección de intrusos OSSEC (*Open Source Security)*, abordando el análisis funcional de otras alternativas con similar objetivo, la identificación de las especificaciones funcionales ausentes del proyecto original en comparación con las alternativas analizadas y finalmente una guía básica sobre las actividades del proceso ingenieril que se propone para implementar las contribuciones en futuro por parte de la comunidad.

Índice de Términos—Fork, WebUI, Ingeniería, Software

I. INTRODUCCIÓN

OSSEC es uno de los mayores exponentes IDS (*Intrusion Detection System*), analiza registros (*logs*) de las máquinas monitorizadas en busca de un algún evento que pueda representar un ataque. Para llevar a cabo esta operación utiliza herramientas para la detección de *rootkits*, los cuales son programas que permiten acceder con ciertos privilegios al sistema generalmente de forma oculta a los controles de los administradores. Esto con el fin de revisar la integridad de ficheros y programas del sistema.

La administración de este tipo de sistemas lleva consigo desde el monitoreo de las posibles alertas hasta las medidas preventivas y correctivas de cómo el sistema puede afrontar situaciones de riesgo para su seguridad. Estas dos actividades requieren de una dosis de arduo trabajo por parte de los administradores o personal encargado de la seguridad informática, quienes por lo general poseen otro conjunto de funciones a cargo. Por lo tanto es indispensable que pueda contar con herramientas que agilicen y faciliten estas tareas que en cierto momento pueden ser tediosas y repetitivas.

Ante este tipo de necesidad, surgen las interfaces de usuario, como elemento que permite al administrador interactuar con los aspectos propios de información resultante del trabajo que cumple OSSEC, brindando esa facilidad para interpretar y tomar decisión gracias a esa comunicación entre los elementos visuales de la interfaz y los datos de fondo a representar, que hoy en día se fusionan a través de un explorador web.

Dentro de estas alternativas se encuentra WebUI de OSSEC, un proyecto que de acuerdo al historial del repositorio en el sitio *Github*, presenta un avance poco significativo en la ampliación de funcionalidades con respecto al desarrollo de otras alternativas como el caso de Analogi, Splunk para OSSEC, Prewikka de Prelude y Kibana.

Esta situación sumada al control estricto y lentitud en el proceso de estudio para la aprobación de contribuciones, es propicia para generar una iniciativa de bifurcación al proyecto OSSEC WebUI, con el incentivo de equipararlo a través de la implementación de funcionalidades previamente identificadas en el análisis de las alternativas. Una bifurcación impulsada en primer instancia con el desarrollo de algunos componentes acompañados de la documentación del proceso ingenieril que esto conlleva, dando como resultado un guía base a

seguir por todo aquel miembro de la comunidad del software libre que esté dispuesto a contribuir con la evolución mucho más dinámica que la de proyecto originario.

II. ANÁLISIS FUNCIONAL DE LAS ALTERNATIVAS A OSSEC WEBUI

Cuando se trata de analizar interfaces gráficas se debe tener muy presente la finalidad que estas representan para el usuario, como un mecanismo de comunicación bidireccional, explotación de funcionalidades, adaptación y menor complejidad posible al momento de la interacción.

Es por ello que para esta actividad, se han seleccionado aspectos relevantes en cuanto a funcionalidad; la finalidad de analizar las alternativas se contempla como un punto de gran importancia para los resultados de la investigación, generando una comparativa de funcionalidades candidatas como requerimientos para la bifurcación.

La metodología propuesta para el análisis funcional es la siguiente:

1. Delimitación de funcionalidades y/o características a analizar en cada alternativa.

Estas funcionalidades pueden categorizarse en 2 tipos: a) Aquellas que se soportan en la administración de la información generada por el sistema de detección intrusiones OSSEC y b) las que permiten manipular la configuración del sistema.

-Tableros de información o cuadros de mando: Este punto es coyuntural para cada alternativa, ya que a través de esta funcionalidad permite monitorear, controlar y gestionar registros relacionados con las actividades de OSSEC.

-Funciones de búsqueda: Determinante para una interfaz web ya que permite filtrar la información a partir de un patrón.

-Estadísticas: La interpretación cuantitativa de los datos permite tomar decisiones. Al igual que la anterior, son funcionalidades de casi obligado uso ante los volúmenes de información recopilada.

-Gráficos: Funcionalidad de gran apoyo para la representación de la información y toma de decisiones de forma rápida ante eventualidades que reporte OSSEC a través de sus alertas.

-Configuración de HIDS (*Host Based Intrusion Detection System*)-OSSEC: Parametrización y/o modificación de configuraciones propias del sistema de detección de intrusos OSSEC. Todas aquellas funcionalidades y/o características que permitan facilitar tareas relacionadas con la administración del sistema a través de la interfaz web de usuario.

2. Análisis de las alternativas a través de criterios que permitan corroborar el cumplimiento de las funcionalidades y/o características definidas en el paso anterior. De igual forma la identificación de funcionalidades y/o características adicionales.

A través de propiciar una plataforma para su experimentación, se documentarán los aspectos técnicos sobre la instalación y configuración para cada alternativa. De esta manera se puede emitir un criterio sobre las funcionalidades y/o características analizadas y aquellas que son valor agregado propias de la herramienta.

3. Agrupación de las funcionalidades y/o características que serán candidatas a requerimientos para la bifurcación.

Una vez analizadas las alternativas se procede a seleccionar aquellas funcionalidades y/o características ausentes o sujetas a mejoras de la interfaz gráfica de usuario OSSEC WebUI. Esto con el fin de incentivar la participación de programadores de la comunidad de software libre con la bifurcación de este proyecto.

TABLA I
RESUMEN ANÁLISIS FUNCIONAL DE ALTERNATIVAS A OSSEC WEBUI

Herramienta	Funcionalidad / Característica	Cumple
Analogi	Tablero de información	SI
	Funciones de búsqueda o filtro	SI
	Estadísticas	SI
	Gráficos	SI
	Configuración HIDS OSSEC	NO
Splunk para OSSEC	Tablero de información	SI
	Funciones de búsqueda o filtro	SI
	Estadísticas	SI
	Gráficos	SI
	Configuración HIDS OSSEC	SI
Prewikka	Tablero de información	SI
	Funciones de búsqueda o filtro	SI
	Estadísticas	SI
	Gráficos	SI
	Configuración HIDS OSSEC	SI
Kibana	Tablero de información	SI
	Funciones de búsqueda o filtro	SI
	Estadísticas	SI
	Gráficos	SI
	Configuración HIDS OSSEC	NO

III. INGENIERIA DE SOFTWARE EN ENTORNOS DE SOFTWARE LIBRE APLICADA A LA BIFURCACION

La tarea de bifurcar el proyecto recae principalmente en un aspecto legal, concretamente la licencia a la cual se acoge. En este caso el proyecto OSSEC WebUI se encuentra liberado bajo la licencia GPLv3 (*General Public License version* 3)[1] la cual garantiza la prevalencia de las libertades del software libre, entre las que se permite la modificación de su código fuente.

Al bifurcar un proyecto se puede optar por solicitar la aprobación e integración de las contribuciones al proyecto principal o para el caso del presente trabajo, liderar un nuevo proyecto que cuente con una gran base en su estructura soportada a través del código fuente del proyecto originario.

Publicado el proyecto como obra derivada, que por cuestiones legales y de acuerdo a la licencia del código fuente originario, en el caso de estudio, GPLv3, el nuevo proyecto deberá liberarse bajo esta misma licencia.

METODOLOGÍA ÁGIL DE SOFTWARE LIBRE.

El software libre tiene como fortaleza la participación de la comunidad para su constante evolución, es por ello que la metodología de desarrollo de software seleccionada permitirá obtener diversas versiones del producto de acuerdo a las funcionalidades que se vayan implementando; Según afirma Jacobson, el modelo de proceso más adecuado para el desarrollo de software es un proceso iterativo e incremental [2]. Esto facilitará los procesos de depuración y validación soportados en un plan de pruebas que finiquitará en un software satisfactorio para los usuarios. Adicional a ello el modelo permite añadir o modificar requerimientos que no se hallan contemplado inicialmente; esto último se ajusta a la participación de otros colaboradores a través de sus contribuciones, ya sea en cualquiera de las etapas que encierra el proceso ingenieril: Análisis, Diseño , Desarrollo y Pruebas.

Como base para el desarrollo de software libre se encuentra la construcción colaborativa, es por ello que sumado a la justificación anterior se opta por orientar el proyecto a una metodología ágil de software libre o también conocida como *open source*, la cual ha dado fruto a grandes desarrollos como el *kernel* de *Linux* entre otros. Se detalla a continuación:

-Adaptativa a los requerimientos que surgen. No predictiva.

-Se apuesta por la distribución de trabajo entre diferentes colaboradores o equipos de colaboradores.

-Se define el repositorio para el software.

-Se designan supervisores de código. Los cuáles serán las personas encargadas de autorizar cambios del fuente en el repositorio. Además debe coordinar y mantener la consistencia del diseño del software. Para la presente investigación se contará con un líder o supervisor de código que inicialmente implementará algunas requerimientos y liberará el proyecto para su respectiva adhesión de colaboradores.

-El resto de colaboradores puede realizar cambios en su código base. Entendiendo código base como una copia que puede manipular localmente cada colaborador.

Como factor adicional a la metodología anterior, desde esta investigación se propone que otra función de los supervisores de código sea mantener actualizada la diagramación funcional del software con el fin de soportar el proceso ingenieril en cuanto a especificación, visualización y documentación del proyecto.

De acuerdo a lo postulado en el modelo incremental las 4 fases o etapas (análisis, diseño, desarrollo, pruebas) se aplican de manera iterativa para ir generando entregas que representan los avances al proyecto. La investigación propone diferentes actividades de ingeniería de software apoyadas en herramientas de software libre:

-Determinación de requerimientos.

-La diagramación empleando el lenguaje de modelado unificado a través de diagramas de casos de uso, diagramas de actividades y diagramas de componentes.

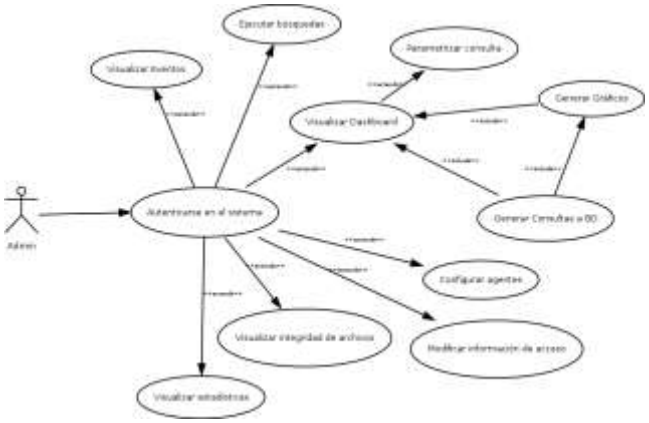

Fig. 1 Diagrama de casos de uso, diseñado con el software libre DIA.

Fig. 2 Diagrama de actividades, diseñado con el software libre DIA

-Definición de interfaces para unificar criterios de diseño visual.

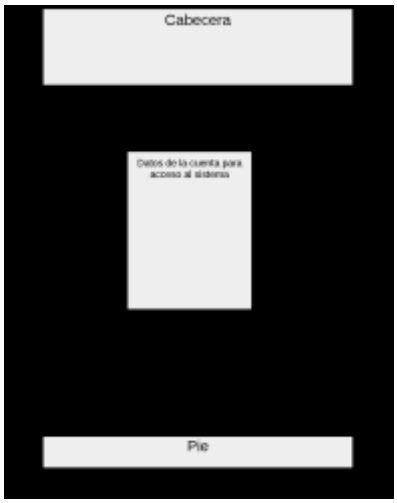

Fig. 3 Definición de interfaces, diseñado con el software libre Pencil.

-Aplicación de ingeniería inversa para la extracción de modelos de datos.

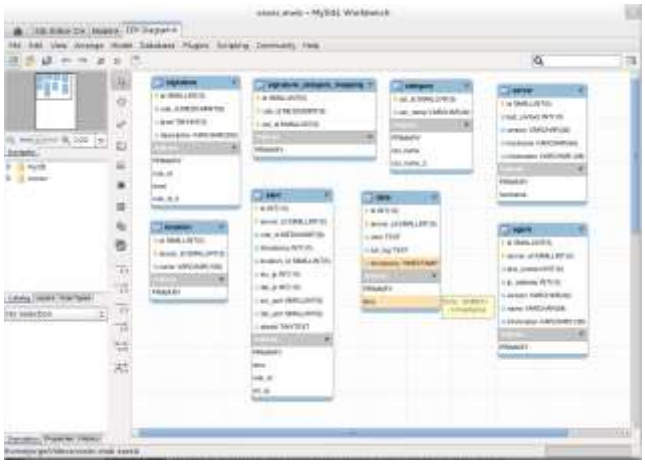

Fig. 4 Ingeniería inversa a una base de datos usando software libre MySQL Workbench.

-Desarrollo basado en componentes que permite explotar la reutilización de código.

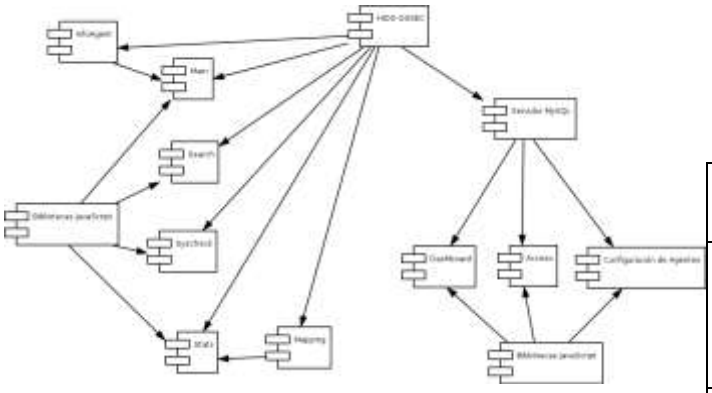

Fig. 5 Diagrama de componentes, diseñado con software libre DIA

-Pruebas de unidad e integración empleando guiones de prueba y pruebas automatizadas.

IV. Conclusiones

-Predomina la noción de conflicto, reflejada en las notorias manifestaciones de una parte de la comunidad de software libre, cuya representación principal es la bifurcación de proyectos. Bifurcaciones que son generadas en gran medida por la misma esencia que caracteriza una comunidad en línea, puesto que los conflictos forman parte de los seres humanos y por ende son proclives a diferencias de criterio, criterio que puede ser técnico, organizacional, ideológico con gran énfasis en lo político, cultural y de entorno.

-La importancia de optar por una bifurcación externa que sea potencializada por la comunidad frente al estancamiento y/o abandono del proyecto OSSEC-WebUI. Esta acción debe considerarse como una alternativa propiciada con el fin de equiparar funcionalmente el proyecto en relación con el estado actual de otras iniciativas, incluso de aquellas de software privativo, obteniendo un resurgimiento en su actividad.

-La ingeniería de software en los proyectos de software libre debe ser implementada con mayor presencia en actividades que permitan representar el sistema en forma comprensible para todo aquel interesado en el proyecto.

-La documentación técnica de carácter libre como debilidad en los proyectos de software libre y la necesidad de la misma para acelerar la evolución del producto con la integración de nuevos desarrolladores.

Apéndice

TABLA II

ANÁLISIS FUNCIONAL DE LA ALTERNATIVA ANALOGI

Funcionalidades y/o Características	Observaciones
Tableros de información o cuadros de mando	El despliegue de información es básico, propicia un entorno para representar la información requerida. Gracias a su conectividad con Mysql posee una fuente amplia de datos.
Funciones de búsqueda o Filtros	Permite paramétrizar la información visualizada a través de tablas y gráficos al instante. Filtrar de acuerdo a criterios de ID de regla, categoría, entre niveles de alerta, entre fechas, host de origen, inclusión y exclusión de datos.
Estadísticas	Totalizar alertas de acuerdo a su ID y nivel.
Gráficos	Posee su fuerte en la librería de gráficos amCharts, bastante robusta. La licencia de esta librería es restrictiva en cuanto a uso no comercial, donde se impone la visualización del enlace al sitio amcharts.com en un lugar estratégico en cada gráfico que se emplee.
Configuración HIDS-OSSEC	N/A
Adicionales	No relevantes.

TABLA III

ANÁLISIS FUNCIONAL DE LA ALTERNATIVA SPLUNK PARA OSSEC

Funcionalidades y/o Características	Observaciones
Tableros de información o cuadros de mando	Predeterminadamente posee dos tipos de dashboard para desplegar la información: detallada o resumida (OSSEC Dashboard y Summarized). Cada uno de estos tableros o cuadros de mando puede paramétrizar la consulta en períodos de tiempo(minutos, horas, días, desde el inicio de los tiempos o en tiempo real). El dashboard de splunk para OSSEC esta distribuido en cuadros de información dinámica y gráficos que permiten tener un amplio panorama de la información generada por OSSEC.
Funciones de búsqueda o Filtros	La sección de búsqueda permite desde una área de texto o barra de búsqueda, teclear los términos por los cuales se desea filtrar los eventos a consultar. Se pueden emplear: conjunto de términos, frases entre doble comilla, usar operadores booleanos, comodines y valores de campo. Como característica importante en la sección de búsqueda, se cuenta con la opción de búsqueda avanzada, la cual consiste en el uso de comandos para transformar, filtrar y desplegar eventos. Por ejemplo el uso de la barra vertical o carácter de canalización.
Estadísticas	A la totalización de eventos de acuerdo a su tipo ó severidad y la característica de Top que permite agrupar los eventos más relevantes de acuerdo a la anterior clasificación; se cuenta también con una sección de estadísticas para construir tablas, visualizaciones, rápidos reportes y comandos para transformar búsquedas en diagramas de tiempo y resumir datos.
Gráficos	Los gráficos son uno de los componentes de mayor importancia en los dashboard que ofrece splunk. La mayor cantidad de información que despliega esta herramienta se encuentra soportada con la opción de graficación. La principal librería que usa splunk para sus gráficos es JSChart, que puede ser descargada bajo licencia que tiene como restricción que su uso no sea con fines comerciales.
Configuración HIDS-OSSEC	Una de las funcionalidades relacionadas con la configuración del sistema HIDS OSSEC ofrecida por splunk para OSSEC es es administrador de agentes (Agent Management), el cual permite ejecutar acciones como adicionar, listar, remover agentes y extraer las claves de agente.
Adicionales	Splunk para OSSEC permite entre otras funcionalidades: Covertura de agentes: A través de una interfaz se puede chequear el estado de conexión de los agentes al sistema OSSEC, el tiempo que ha trascurrido desde su conexión, el total de máquinas, el porcentaje de cobertura en la relación número de máquinas / agentes trabajando o conectados, agente inactivos, el estado actual del agente, su estado en un período de tiempo, visor para desplegar información sobre los cambios e integridad de archivos. Finalmente esta herramienta proporciona un conjunto de utilidades para explorar información sobre reportes de malware, eventos de alta severidad, todo esto acompañado de un conjunto de alternativas para generar reportes detallados y resumidos con la opción de ser exportados a formatos como CSV, Raw Events, PDF, XML, JSON.

TABLA IV
ANÁLISIS FUNCIONAL DE LA ALTERNATIVA PREWIKKA

Funcionalidades y/o Características	Observaciones
Tableros de información o cuadros de mando	El despliegue de la información de alertas a través de cuadros de mando que permiten navegar hacia el detalle con la característica de actualizar automáticamente los avisos de alertas. Adicionalmente es posible gestionar las alertas a través de la operación de borrado.
Funciones de búsqueda o Filtros	La creación de filtros sobre cada unas de las columnas (a excepción de la columna time) que componen el cuadro de mando de los eventos permite obtener información agrupada en diferentes criterios. Al lado izquierdo parte baja de la interfaz se encuentra un panel de control que ofrece las diferentes opciones de filtro de tiempo (hora, día, mes y personalizada), con la característica adicional de poder guardar los filtros para su posterior uso.
Estadísticas	La sección de estadísticas es distribuida en 5 fichas: Categorizaciones (categorizations): Despliega 3 tipos de gráficos circulares clasificando las alertas de acuerdo a severidad y relevancia. En la ficha fuentes (sources) los gráficos despliegan el top de los recursos o fuentes de análisis y el país origen. En la ficha objetivos (targets) se representan el Top de la direcciones y usuarios objetivo. La ficha analizadores (analyzers) viene acompañada de 5 tipos de gráficos con el Top 10 para ilustrar: Clases de analizadores, los propios analizadores, modelos de analizadores, nodos, y ubicaciones. La ficha línea del tiempo (timeline) muestra un gráfico de líneas representando la severidad de los eventos en el tiempo.
Gráficos	Basada su interfaz de gráficos en la librería 2D Cairo, ha sido desarrollado el archivo Chart.py para facilitar el trabajo de representación de datos.
Configuración HIDS-OSSEC	X

Adicionales	Configuración de preferencias: selección de idioma, cambio de password. Administración de permisos a través de la creación de usuarios. Control de sesión inactiva: El sistema cierra automáticamente la sesión si no se esta realizando ninguna tarea durante un tiempo determinado. Esto tiene como excepción si la actualización automática esta activada y el tiempo para actualizar es inferior al período de sesión inactiva. La sección agentes (Agents) permite gestionar y monitorear los analizadores. Esta sección contiene 2 fichas. La ficha agentes (agents) muestra toda la información de los agentes, su estado, lista de alertas enviadas por cada analizador. La ficha Látidos de corazón (Heartbeats) despliega un listado con notificaciones de cada agente.

TABLA V
ANÁLISIS FUNCIONAL DE LA ALTERNATIVA KIBANA

Funcionalidades y/o Características	Observaciones
Tableros de información o cuadros de mando	La estructura dinámica de los dashboard o cuadros de mando de Kibana permiten desde la cración de tableros de información, o a los ya creados agregar filas y estas a su vez agregarle los diferentes componentes que dispone para visualizar datos, entre ellos se tienen: histogramas, tablas , mapas, campos, entre otros. El despliegue del cuadro informativo en la parte baja de la vista, permite una graficación en la parte central de la información desplegada por Kibana. El detalle a un solo clic de cada evento mostrado en la tabla es otra de las particularidades en sus cuadros de mando o tablero de información.
Funciones de búsqueda o Filtros	En la parte superior de la vista de Kibana se encuentra la sección Query, que brinda la posibilidad de realizar consultas a través de varios términos, los cuales pueden agregarse para ejecutar una búsqueda más especifica. Uno de los componentes fundamentales en los tableros de información son las tablas de datos, estas traen como funcionalidad en sus títulos o rótulos la posibilidad de ordenamiento ascendente o descendente. Otra de las características de Kibana radica en la periodicidad que ofrece para la consulta de los eventos de OSSEC , desde 30 días hasta tiempo real incluyendo parametrizable.
Estadísticas	Se puede afirmar que es uno de los puntos débiles de la herramienta, ya que solo ofrece un conteo de eventos y representación porcentual de acuerdo al filtro que se aplique.
Gráficos	El componentes de gráficos es soportado por la Librería D3 de javascript. A través de este componente permite la agregación de visualizaciones de datos.
Configuración HIDS-OSSEC	X
Adicionales	Como característica relevante no se puede pasar por alto la flexibilidad de configuración de los dashboard en cuanto a ubicación de sus paneles y componentes, almacenamiento y carga de dashboard y la característica de compartir el cuadro o tablero de mando que se este visualizando.

TABLA VI
ESPECIFICACIONES FUNCIONALES A CONSIDERAR PARA LA BIFURCACIÓN DEL PROYECTO OSSEC WEB UI.

Tendencia	Especificación funcional
Tableros de información o cuadros de mando. (Dashboard)	-La información relacionada con el eventos de OSSEC debe ser desplegada a través de tablero o cuadro de mando de forma ordenada. - Debe permitir parámetrizar la consulta de los eventos en períodos de tiempo: Minutos, horas, días o tiempo real). -El tablero o cuadro de mando debe tener como fuente de información una base de datos para permitir acelerar las consultas.
Búsquedas o filtros	-La búsqueda debe permitir a través de una sola entrada o barra de búsqueda teclear el término o términos por los cuales se desee filtrar los eventos. -La barra de búsqueda debe permitir el uso comillas, operadores booleanos y comodines. -Exportar los resultados de las consultas a formatos como CSV, PDF, XML, JSON
Estadísticas	-Exportar los resultados de las estadísticas a formatos como CSV, PDF, XML, JSON
Gráficos	-El sistema deberá generar gráficos que permitan resumir la información desplegada en el cuadro de mando de eventos. -El sistema deberá generar gráficos para representar la característica Top de eventos por severidad y relevancia.
Configuración HIDS-OSSEC	-Debe controlar el acceso al sistema. -El sistema debe permitir adicionar o crear agentes, empleando la creación automática de llaves). -Debe listar los agentes. -Debe Remover agentes. -Informar el estado actual del agente.

REFERENCIAS

[1] Free Software Foundation (2007). General Public License version 3 disponible en http://www.gnu.org/licenses/gpl.html

[2] Jacobson, I., Booch, G., Rumbaugh, J. (2000). (El proceso unificado de desarrollo de software) Addison Wesley Pág. 6-7