Técnicas de minería de datos aplicadas a la informática forense

Julián Alberto Monsalve Pulido

Ingeniero de Sistemas. Msc. en software Libre, Universidad Autónoma de Bucaramanga - Universitat Oberta de Catalunya España. Investigador Grupo GIBRANT, Facultad de Ingeniería de Sistemas, Universidad Santo Tomás Tunja.
jmonsalve@ustatunja.edu.co

Recibido: 15 de noviembre de 2013 Aprobado: 10 de diciembre de 2013

Artículo de investigación, como producto del desarrollo parcial del grupo GIBRANT.

Resumen

El presente artículo muestra los avances de investigación en el análisis de patrones en servidores web apache, utilizando técnicas de minería de datos. Se presenta en esta investigación, un análisis y un software que pretende mejorar los procesos de minería Web para la toma de decisiones en los procesos de informática forense. En esta investigación se aplicaron reglas de asociación, con un algoritmo APRIORI para identificar los hechos comunes dentro de la información del servidor apache.

Para el desarrollo de la aplicación se tuvo en cuenta cinco pasos fundamentales para garantizar la usabilidad y funcionalidad del mismo, el primer paso se inicia con la extracción de los logs del servidor apache, el segundo es filtrar la información que se encuentra para el análisis, el tercer paso es transformar el archivo Log en un archivo de sesiones, el cuarto paso es aplicar la minería de datos utilizando reglas de asociación, donde es utilizado para descubrir hechos que ocurren en un determinado conjunto de datos y el quinto paso es la visualización de los informes con los resultados estadísticos del análisis de la información para la toma de decisiones en el proceso forense.

Palabras Claves: *Minería de datos, usabilidad, servidor web, apache, computer forensics.*

Abstract

This paper shows the research advances in the analysis in the analysis of patterns in apache web server using data mining techniques. An analysis and a software is presented with the aim to improve Web mining processes for making decisions in the process of computer forensics. In this research we applied association rules, a priori algorithm to identify common events in the apache server information.

To develop the application five steps were taken into account to ensure the usability and functionability of it: the first step begins with the extraction of logs from apache server, the second is to filter the information found in the analysis, the third step is to transform the log file in a file session, the fourth step is to apply data mining using association rules, which is used to discover facts that occur in a given data set and the fifth step is the visualization of reports the results of the analysis of statistical information for decision-making in the forensic process.

Key words: *Data Mining, usage, web server, apache*

I. INTRODUCCIÓN

La informática forense es una ciencia que ayuda a reconstruir evidencias informáticas para apoyar procesos jurídicos en cualquier legislación internacional. Bajo la ley 1273 de 2009 (Ley de protección de la información, 2009) en Colombia se estableció la protección de la información y de los datos donde se preservan integralmente los sistemas que utilicen las tecnologías de la información y comunicaciones, entre otras disposiciones. Con este artículo se quiere informar a los lectores cómo es el funcionamiento de un servidor web y cómo extraer los diferentes log para la aplicación de las técnicas forenses con la ayuda de la minería de datos. Este artículo se estructura en cinco capítulos, el primero muestra la conceptualización básica, en la segunda parte se describe la herramienta desarrollada, en la tercera parte se explica las reglas de asociación aplicadas al proyecto, en la cuarta parte se explica el módulo forense de apache y por último se muestra las conclusiones de la investigación y referencias que se utilizaron en el proceso.

II. CONCEPTUALIZACIÓN BÁSICA.

Un servidor web es un recurso de software que actúa como compilador al lado del servidor que realiza peticiones bidireccionales con cliente en cualquier lenguaje de programación compatible y con diferentes protocolos. En la actualidad existen varios servidores web en el mercado libres y propietarios, en el caso de investigación se tomó como prueba el servidor web apache (Apache Software, Fundation, 2011), ya que cuenta con 152 millones de sitios web en todo el mundo, catalogado como el servidor web más utilizado en el mundo y la facilidad uso bajo su licencia apache license (Apache Software, Fundation, 2011) donde ofrece las ventajas del software libre pero con una licencia permisiva.

El uso de este servidor se hace en la gran mayoría en servidores Linux, ya que cuenta con mayor adaptabilidad, escalabilidad, robustez y seguridad con respecto a otros sistemas operativos. El servicio o proceso de apache en de Linux es el httpd, donde se ejecuta como demonio forma continua en background escuchando todas las peticiones que hace el cliente, es necesario administrar el servicio como usuario root donde al iniciar configura procesos básicos de arranque y crea procesos hijos que tienen como tarea escuchar y responder las peticiones del cliente. El puerto que usa apache por defecto es el 80 ó se puede configurar cualquier otro por debajo del 1024 dependiendo las políticas de administración o seguridad de la empresa.

En la siguiente gráfica se muestra el proceso de transacción del protocolo http, donde un proceso servidor escucha en un puerto de comunicaciones TCP y espera las solicitudes de conexión de los clientes web.

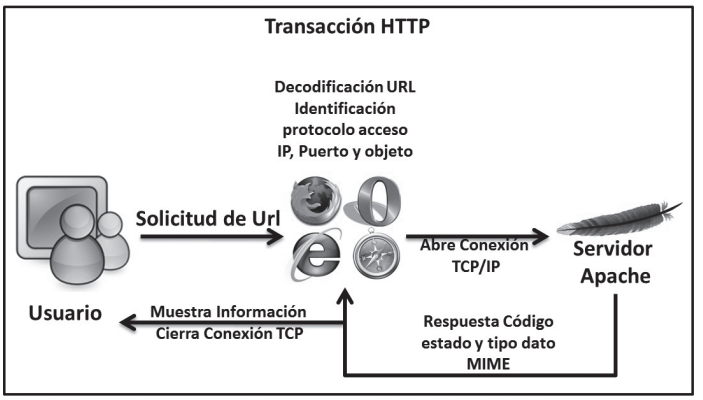

FIGURA 1. ETAPAS DE UNA TRANSACCIÓN HTTP.
Fuente: autor.

Para cada transacción con el servidor apache devuelve un código numérico donde informa sobre los resultados de la operación, puede ser mensajes de error o de confirmación del mensaje, con el siguiente formato: versión de protocolo HTTP utilizado, código numérico de estados (tres dígitos) y descripción del código numérico.

Para las cinco categorías de los mensajes de estado, apache tiene la siguiente codificación:
1xx Mensajes Informativos.
2xx Mensajes asociados con operaciones realizadas correctamente.
3xx Mensajes de redirección, que informan de operaciones complementarias que se deben realizar para finalizar la operación.
4xx errores del cliente; el requerimiento contiene algún error, o no puede ser realizado.
5xx errores del servidor, que no ha podido llevar a cabo una solicitud.

Los registros log son registros que el servidor hace en algunos archivos para hacer un seguimiento de todas las peticiones de los usuarios, los más comunes son el log de acceso donde registra las peticiones correctas que ha respondido el servidor y el log de errores donde registra los errores de las peticiones.

En el Log de acceso o trazos de demanda de las páginas muestran los patrones de visualización

del usuario proporcionando información exacta, activa y objetiva sobre los usos de la Web de los usuarios. Los datos almacenados en los logs siguen un formato estándar diseñado por CERN y NCSA [Luotonen, 1995]. Una entrada en el Log siguiendo este formato contiene entre otras cosas, lo siguiente: dirección IP del cliente, identificación del usuario, fecha y hora de acceso, requerimiento, URL de la página accedida, el protocolo utilizado para la transmisión de los datos, un código de error, agente que realizó el requerimiento, y el número de bytes transmitidos. Esto es almacenado en un archivo de texto separando cada campo por comas (“,”) y cada acceso es un renglón distinto.

74.6.22.162 - - [27/Apr/2008:04:20:42 -0500] “GET /robots.txt HTTP/1.0” 404 291 “-” “Mozilla/5.0 (compatible; Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)”

III. DESCRIPCIÓN DE LA HERRAMIENTA DESARROLLADA.

Teniendo la información en un archivo log de forma desorganizada y no entendible, se vio la necesidad del desarrollo de una herramienta que nos ayude a organizar la información para la aplicación de la minería de datos y obtener resultados tabulados para su respectivo análisis. La herramienta que se desarrollo fue creada bajo la licencia creative commons (Creative Commons Colombia, 2011). A continuación se muestra el diagrama de casos de uso tenido en cuenta para el desarrollo de la aplicación, donde el usuario de la herramienta, procesa la información, identifica la metodología o técnica de la minería de datos y ejecuta los informes finales para la toma de decisiones en el portal. La toma de decisión depende del proceso usable que tienen las páginas, si hay necesidad de cambiar la ubicación de las mismas o hacer modificaciones de estilo, estructura o de contenido.

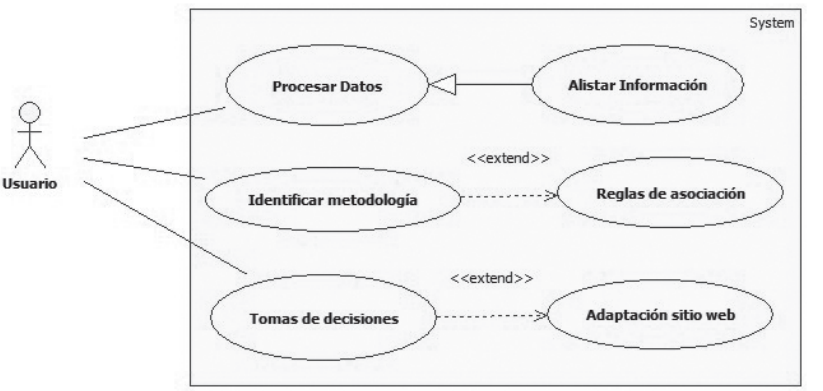

FIGURA 2. DIAGRAMA DE CASOS DE USO DE LA APLICACIÓN.
Fuente: autor.

A. Cargar Log.
Proceso por el cual seleccionamos el Log de conexión previamente extraído en el servidor.

B. Procesar Datos.
Proceso de filtrado y de almacenamiento de datos a las bases de datos.

C. Sessionización.
Proceso de transformación del Log en sesiones de usuario.

D. Minería Web.
Aplicación de técnicas para descifrar los patrones de comportamiento de usabilidad.

E. Informes.
Se presentan los resultados de la aplicación de minería de datos a la información recolectada.

FIGURA 3. PANTALLA DE LA APLICACIÓN DESARROLLADA.
Fuente Autor.

IV. REGLAS DE ASOCIACIÓN.

Para una visión general del flujo de la información se presenta la siguiente gráfica, donde muestra el proceso global que debe hacer el usuario para el uso de la herramienta. Todos los usuarios del sitio web ingresan desde su navegador por medio de la url al sitio web, los registros de ingresos son almacenados en un log que el servidor apache configura automáticamente, se extrae el log del servidor y se carga a la aplicación desarrollada, la aplicación analiza la información del log y realiza una limpieza de la información que no se necesita en el análisis, cuando se hace la limpieza se procede a realizar la minería de datos, basados en reglas de asociación que más adelante en el artículo se explicará su definición y por último el sistema arroja informes estadísticos según las peticiones del usuario.

FIGURA 4. PROCESO DE ANÁLISIS DE LOG DE CONEXIÓN SERVIDOR APACHE.
Fuente Autor.

Para la identificación de patrones en los log de conexión, se aplicó reglas de asociación que son usadas para descubrir hechos que ocurren en común dentro de un determinado conjunto de datos. Las reglas de asociación son aplicadas en diversos campos, uno de ellos el mercadeo, donde se identifica patrones de comportamientos de compras de los clientes y se trazan estrategias para subir ventas con solo cambiar la ubicación a los productos en el almacén o en un portal de comercio electrónico.

Para el caso de usabilidad tenemos a **P={ índex, académica, administrativo…}** conjunto de páginas del portal web, usados como ítems.

Se muestra 5 transacciones registradas en el log de apache, donde se usó algunos de los 3 elementos de prueba.

ID	Índex	Academia		Administrativo		Bienestar	Investigación
1	1	0	1	1	0		
2	1	1	0	0	0		
3	1	1	1	1	0		
4	0	1	0	1	1		
5	0	0	1	0	1		

ID	Índex	Academia	Administrativo	Bienestar	Investigación
1	1	0	1	1	0
2	1	1	0	0	0
3	1	1	1	1	0
4	0	1	0	1	1
5	0	0	1	0	1

FIGURA 5. EJEMPLO DE TRANSACCIONES, EN EL USO DE UN PORTAL WEB.
Fuente Autor.

Se identifica el conjunto de transacciones con sus identificadores, para fijar la regla se debe hallar en un conjunto de elementos que son usados frecuentemente. En nuestro ejemplo se analiza que el conjunto (index y académica) tiene un soporte de S 2/5= 0,4 con un 40% que de cada 5 transacciones se cumplan en conjunto.

Para la confianza analizamos (index, académica) => (administrativa) = 0,2/0,4 = 0,5 el 50% que ingresan a (index, académica) ingresan también a la página administrativa.

Para la confianza analizamos (index, académica) => (Bienestar) = 0,3/0,4 = 0,75 el 75% que ingresan a (index, académica) ingresan también a la página Bienestar, regla bastante aceptable para tenerla en cuenta.

Para la confianza analizamos (index, académica) => (Investigación) = 0,1/0,4 = 0,25 el 25% que ingresan a (index, académica) ingresan también a la página investigación, regla muy débil no se tiene en cuenta como regla de asociación.

El algoritmo apriori se usa en minería de datos para encontrar Reglas de asociación en un conjunto de datos. Este algoritmo se basa en el conocimiento previo o “a priori” de los conjuntos frecuentes, esto sirve para reducir el espacio de búsqueda y aumentar la eficiencia (Hernández, 2004).

El algoritmo apriori, genera todos los ítems sets con un elemento. Usa estos para generar los de dos elementos, y así sucesivamente. Se toman todos los posibles pares que cumplen con las medidas mínimas de soporte. Esto permite ir eliminando posibles combinaciones ya que no todas se tienen que considerar. Por último genera las reglas revisando que cumplan con el criterio mínimo de confianza.

Algoritmo 1 Algoritmo apriori(D:datos, MinC:Cobertura Minima)

```
i=0
Rellena_Item(Ci)//incluye en C0 todos los ítems de Tamaño 1
Mientras Ci <> Null
        Para Cada x = Elemento de Ci
                Si cobertura(x) >= MinC Entonces Li = Li U X
        Fin Para
        Ci = Selecciona_candidatos (Li)
        i=i+1
fin mientras
retornar C
```

Para el desarrollo de la aplicación, se tomó como base las transacciones realizadas por los usuarios, identificado por un id de sesión analizado por medio de un algoritmo de perfilamiento que identifica las sesiones anónimas y se agrupa por ip y por fecha y hora de acceso, con un umbral de 10 a 20 minutos. Con el resultado del perfilamiento se crea una tabla con todas las páginas que componen el portal web como podemos observar en la figura No. 6, donde se registra 1 (uno) si en la sesión visita la página y 0 (cero) en caso contrario. Para la creación de las reglas de asociación se usó un método elitista donde buscamos las páginas con más visitas en las sesiones ya que su probabilidad de soporte y confianza es más alta.

Id	Pág1	Pág2	Pág3	Pág4	Pág5	Pág6
1	1	0	0	1	1	0
2	0	0	0	1	1	1
3	1	0	1	0	0	1
4	0	0	0	1	0	0
5	1	1	1	1	0	0
6	1	1	1	1	0	0
7	0	0	0	1	0	1
8	1	1	1	0	0	1
9	1	0	1	1	1	1
10	0	1	0	0	1	1
S=	6	4	5	7	4	6

FIGURA 6. PROCESO DE IDENTIFICACIÓN DE REGLAS DE ASOCIACIÓN.
Fuente Autor.

Como resultado de la aplicación del proceso de minería de datos, el prototipo nos da como resultado lo siguiente:

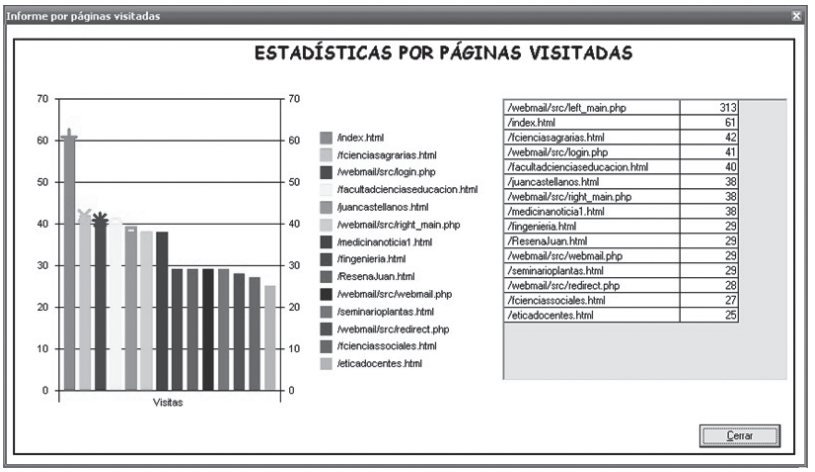

FIGURA 7. ESTADÍSTICAS DE ANÁLISIS DE PÁGINAS VISITADAS.
Fuente Autor.

La página de ingreso al correo electrónico ha sido la más visitada hasta el momento con 313 visitas, y con 61 visitas la página principal del portal. Cada uno de los resultados debe ser tenido en cuenta para el mantenimiento de cada una de las páginas que más frecuentan los usuarios.

Permite evaluar qué páginas son más consultadas. A la vez se le podrá dar más atención a las páginas que se visitan con menos frecuencia.

En cada portal de Internet se identifican las páginas y archivos más visitados para ser actualizados con mayor frecuencia, para que el usuario sea concurrente en el portal.

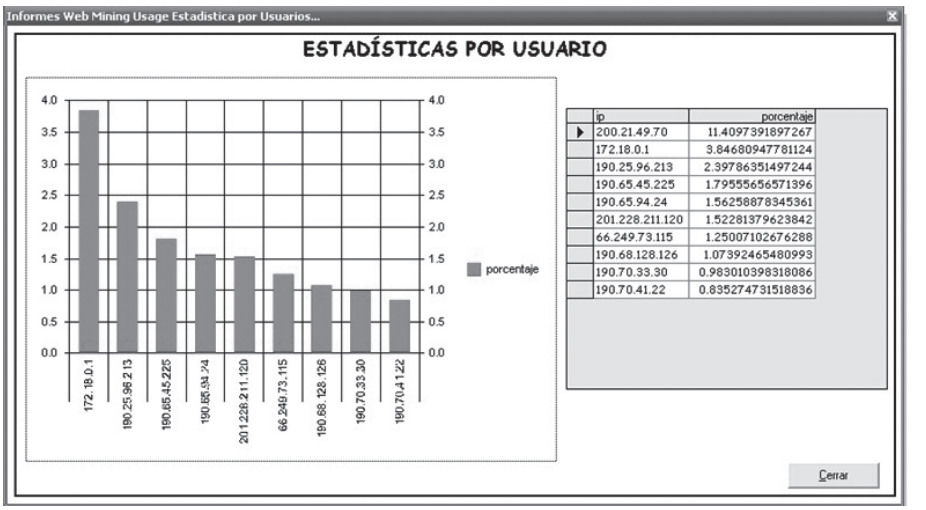

FIGURA 8. INFORME DE ESTADÍSTICA POR USUARIO.
Fuente Autor.

El usuario que más frecuenta el sitio con un 11,409% del 100% total de utilización, es el usuario con la dirección IP 200.21.49.70, el cual identificamos que es una dirección pública de la ciudad de Tunja.

El usuario 190.70.41.22 tiene 0.83% de usabilidad durante el rango de tiempo analizado en el Log de conexión y se observa en el gráfico como el Host que presenta el más bajo porcentaje de visitas al sitio.

V. MÓDULO FORENSE DE APACHE

Apache como software con arquitectura modular cuenta con una lista configurable de módulos que ayudan al buen funcionamiento del mismo, para la investigación fue necesario el análisis de varios de ellos en el servidor web, como el módulo mod_log_forensic donde registra las solicitudes del cliente a priori y posteriori registrada por medio de una identificación única (forensic-id) que debe ser la misma durante los dos procesos.

Para la configuración del módulo es necesario editar el archivo de configuración de apache / etc/httpd/config/httpd.config con usuario root, agregamos o habilitamos los siguientes módulos.

LoadModule log_forensic_module modules/mod_log_forensic.so
LoadModule unique_id_module modules/mod_unique_id.so

Después de agregar los módulos es necesario agregar la directiva ForensicLog para que apache identifique la ruta de escritura de los log forense.
ForensicLog logs/forensic_log

Se reinicia los servicios service httpd restart y comprobamos que en el directorio /var/log/httpd se registra los access_log, error_log y forensic_log.

La directiva ForensicLog en los servidores apache se emplea para registrar peticiones forenses especiales en el servidor donde tiene las siguientes características.

+Tfgj-n8AAAEAABR-CHsAAAAH|GET /favicon.ico HTTP/1.1|Host:192.168.1.33|User-Agent:Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv%3a2.0.1) Gecko/20100101 Firefox/4.0.1|Accept:text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8|Accept-Language:es-es,es;q=0.8,en-us;q=0.5,en;q=0.3|Accept-Encoding:gzip, deflate|Accept-Charset:ISO-8859-1,utf-8;q=0.7,*;q=0.7|Keep-Alive:115|Connection:keep-alive
-Tfgj-n8AAAEAABR-CHsAAAAH

El servidor apache en el LogFormat tiene como defecto la siguiente máscara para el registro de los log forenses.

LogFormat “%h %l %u %t \”%r\” %>s %b \”%{Referer}i\” \”%{User-Agent}i\”” combined

Para la realización de un análisis forense es necesario verificar el log de conexión de apache y compararlo con los registros del mod_log_forensic de apache, para esto es necesario validarlos por medio del token forensic. Para lo anterior hay varias formas de realizar este procedimiento, sin embargo para que el registro de la llave quede en los dos log se debe reconfigurar la el LogFormat agregando la siguiente instrucción %{forensic-id}n, el log forense quedaría de la siguiente forma.

LogFormat “%h %l %u %t \”%r\” %>s %b \”%{Referer}i\” \”%{User-Agent}i\” %**{forensic-id}n”** combined

Es necesario comprobar si las llaves se están registrando correctamente en el log de conexión y en el módulo forense, se va elaborar una petición al servidor para verificar los resultados.

access_log:

192.168.102.176 - - [15/Jun/2011:17:01:28 -0500] “GET /favicon.ico HTTP/1.1” 404 289 “-” “Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Gecko/20100101 Firefox/4.0.1**”TfkruH8AAAEAAAkuBjMAAAAE**

forensic_log:

+TfkruH8AAAEAAAkuBjMAAAAE|GET /favicon.ico HTTP/1.1|Host:192.168.102.57|User-Agent:Mozilla/5.0 (Windows NT 6.1; rv%3a2.0.1) Gecko/20100101 Firefox/4.0.1|Accept:text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8|Accept-Language:es-es,es;q=0.8,en-us;q=0.5,en;q=0.3|Accept-Encoding:gzip, deflate|Accept-Charset:ISO-8859-1,utf-8;q=0.7,*;q=0.7|Keep-Alive:115|Connection:keep-alive
-**TfkruH8AAAEAAAkuBjMAAAAE**

Se observa que se está registrando la llave única para cada una de las peticiones en los dos archivos log, si de alguna forma es modificado log de acceso siempre estará vinculado en paralelo al registro forense, para la verificación de la legitimidad del registro del usuario.

VI. CONCLUSIONES

La informática forense es una ciencia que identifica delitos informáticos, pero los resultados son difíciles de identificar por la cantidad de información, una solución es la aplicación de técnicas de minería de datos por medio de una herramienta de libre distribución.

En la actualidad la gran mayoría de aplicaciones son orientadas a la web y los servidores web son los más atacados para cometer delitos informáticos, lo cual deben ser configurados ade-

cuadamente para mantener registros de acceso con una normatividad adecuada y así poder hacer un seguimiento a posibles fraudes informáticos.

El desarrollo y el potencial del Web Mining, permite detectar información invisible pero de gran importancia para consolidar y ampliar el criterio de la W3, la determinación de los patrones de conducta se establecen como redes de relaciones existentes que permiten identificar grupos homogéneos de usuarios para encausar sus intereses comunes al desarrollo de grupos participativos y líneas de investigación, con personas dedicadas a temáticas afines.

Debido a consumo computacional requerido por los grandes volúmenes de datos, es necesario una adecuada representación de los mismos, para esta investigación se representaron por medio de arrays dinámicos lo que permitió mejorar el proceso computacional salvaguardando siempre la integridad de la información.

En el desarrollo de aplicaciones como resultado de una investigación, se sugiere realizar las publicaciones con licencias libres y con código abierto para que otros investigadores que quieran continuar con el proceso, no inicien desde cero la codificación, ya que con una buena documentación y un proceso colaborativo, el software alcanzará la madurez más rápidamente.

Como trabajos futuros es necesario aplicar algunas combinaciones con técnicas de inteligencia artificial, para evitar modificaciones en los log de conexión por parte de los atacantes y prever futuros ataques en el servidor apache.

REFERENCIAS

Apache Software, Fundation. (05 de Mayo de 2011). Licencia Apache. USA.

Associated Press (1998). “Hackers: Pentagon archives vulnerables”. Mercury Center.

Botía, D.(2008) “Aplicación de las herramientas de software libre Sleuthkit y Autopsy a la informática forense”, Revista Intekhnia Volumen 3 No. 7.

Creative Commons Colombia. (2011). Licencias Creative Commons Colombia. Recuperado el 22 de Junio de 2011, de http://co.creativecommons.org/

Davara, M. (1993). Derecho Informático. Navarra: Ed. Aranzadi. Del Peso, Emilio; Piattini, Mario G. (2000). Auditoría Informática (2.ª ed.). Ed. RA-MA.

Davara, M. (julio, 1997). “El documento electrónico, informático y telemático y la firma electrónica”. Actualidad Informática Aranzadi (núm. 24). Navarra.

González, D. (2002). Sistemas de Detección de Intrusiones.

Hernández, J. (2004). Introducción a la Minería de Datos. Madrid: PEARSON EDUCACIÓN.

Jeimy, J. (2006) Introducción a la informática forense, Revista Sistemas Seguridad y computación forense ACIS.

Ley de protección de la información, 1273 (05 de Enero de 2009).

MapR Technologies, Inc.; MapR Technologies Signs Corporate Contributor License Agreement for Apache Software Foundation. (2011, June). Computers, Networks & Communications,387. Retrieved June 14, 2011, from ProQuest Computing. (Document ID: 2369059991).

Northcutt, S. (2000). Network Intrusion Detection. An analyst’s handbook. New Riders.
Spitzner, L. (2001). Honeypots: Tracking Hackers. Addison-Wesley.