Revista Cubana de Ciencias Informáticas
Vol. 6. No. 4, Octubre –Diciembre, 2012
ISSN: 2227-1899 | RNPS: 2301
Pág. 62-69
http://rcci.uci.cu

Tipo de artículo: Artículo original
Temática: Desarrollo de aplicaciones informáticas
Recibido: 23/09/2012 | Aceptado: 17/11/2012

Detección de *software* malicioso para el filtro de contenido Smart Keeper

Detection of malicious software for the content filtering Smart Keeper

Yurisleidy Hernández Moya[1]*, Daileny Hernández Barreiro[1], Luis Enrique Sánchez Arce[1], Juan Carlos Lobaina Guzmán[2], Dovier Antonio Ripoll Méndez[3]

[1] Departamento de Soluciones Informáticas para Internet. Centro de Ideoinformática. Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2 ½, Torrens, Boyeros, La Habana, Cuba, CP.: 19370
[2] Empresa Nacional de Software. División Camagüey, DESOFT, Calle 2da, e/ E y Pineda, Reparto La Guernica, Camagüey, Cuba, CP.: 71200
[3] Departamento de Técnicas de Programación. Facultad 1. Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2 ½, Torrens, Boyeros, La Habana, Cuba, CP.: 19370

*Autor para la correspondencia: ymoya@uci.cu

Resumen

La detección de *software* malicioso en el filtro de contenido *Smart Keeper* presenta varias limitantes, tales como: no se consideran las preferencias del cliente respecto al antivirus que este desea emplear, se ignoran los ficheros de gran tamaño en la búsqueda de *software* malicioso y las peticiones de escaneo se procesan únicamente de manera secuencial (lo que puede provocar un incremento en el tiempo de respuesta para el usuario). En la presente investigación se propone el desarrollo de un subsistema que contribuirá a erradicar las deficiencias anteriores. La propuesta se basa en dos elementos fundamentales: el componente para el envío de mensajes (solicitudes de trabajo) y la capa de abstracción que propicia la comunicación con los diferentes antivirus.

Palabras clave: antivirus, filtro de contenido, smart keeper, software malicioso.

Revista Cubana de Ciencias Informáticas
Vol. 6. No. 4, Octubre –Diciembre, 2012
ISSN: 2227-1899 | RNPS: 2301
Pág. 62-69
http://rcci.uci.cu

Abstract

The detection of malicious software in the content filtering Smart Keeper has several limitations, such as: the customer's preferences aren't considered regarding the antivirus that it wishes to employ, the big files in search of malware are ignored and the petitions for scan are only processed in sequence (which may cause an increase in the response time to user). In this research it is proposed the development of a subsystem that will help eradicate previous shortcomings. The proposal is based on two key elements: the component for sending messages (work requests) and the abstraction layer that facilitates communication with different antivirus.

Keywords: *antivirus, content filtering, malicious software, smart keeper.*

Introducción

Un software malicioso o *malware* es un programa informático que, sin el conocimiento del usuario, puede provocar anomalías en el comportamiento de los sistemas informáticos (Kramer y Bradfield, 2010). Tales software suelen aparecer en archivos de diferentes tipos (textos, imágenes, música, aplicaciones, entre otros). Las facilidades que ofrece Internet para obtener y publicar materiales hacen de la Red de Redes un medio no exento de la presencia de software maliciosos. Debido a la fácil adquisición de tal tipo de software, resulta imperativo tener precaución de los archivos que son adquiridos desde otros sistemas. En el caso específico de controlar el acceso a materiales disponibles en Internet, los filtros de contenido constituyen una de las soluciones técnicas más empleadas.

Los filtros de contenido son sistemas informáticos que limitan el acceso a determinados recursos que son accesibles desde un ordenador. Generalmente están diseñados para ser usados en ordenadores personales o servidores; en esta última variante permiten ofrecer servicios de manera centralizada a varias computadoras en la red. Entre las funcionalidades típicas de tales sistemas se encuentran: el filtrado de páginas web, el filtrado de mensajería instantánea, la detección de navegación anónima y la detección de software malicioso.

Smart Keeper, actualmente en su versión 2.0, es un filtro de contenido que se desarrolla en la Universidad de las Ciencias Informáticas (UCI). Este sistema para su funcionamientos emplea diversos componentes, entre los que destacan: Squid (software que constituye un servidor *proxy-cache*) y Greasyspoon (programa informático que implementa el protocolo ICAP -del inglés, *Internet Content Adaptation Protocol*-, el cual posibilita manejar las respuestas obtenidas de Internet). Limitación del acceso por tiempo y subredes, restricciones de fichero por tamaño y formato, creación de excepciones y detección de software malicioso son algunas de las principales funcionalidades de dicho filtro. *Smart Keeper,* para la detección de software malicioso, cuenta con parámetros de configuración que le permiten al cliente especificar el formato y tamaño de los archivos que serán analizados. La detección de *software*

malicioso en dicho filtro presenta varias limitantes, tales como:

- Solo se puede emplear el antivirus Clamav.
- En el proceso se ignoran los ficheros superiores a 10 MB (del inglés, *Mega Bytes*). La detección de *software* malicioso se realiza en "tiempo real", almacenando los datos a escanear en la RAM (del inglés, *Random Access Memory*) de la computadora. Esta situación provoca que sea necesario limitar el tamaño y la cantidad de los ficheros concurrentes que pueden ser analizados, con el objetivo de evitar que se utilice toda la RAM de la computadora.
- No es posible la incorporación de varios nodos que trabajen de forma paralela. Las solicitudes de trabajo, para escanear archivos, son atendidas de manera secuencial y en el orden de llegada. Lo anterior, en caso de existir una demanda significativa de trabajo, puede implicar un incremento en el tiempo de respuesta al usuario (pues el sistema solo pasará a realizar la siguiente tarea una vez haya terminado la actual).

Las limitantes anteriores afectan la aceptación y uso del sistema, así como su adaptabilidad en los lugares donde se despliegue; por tanto, se evidencia la necesidad de obtener un mecanismo para *Smart Keeper* que permita mitigarlas.

Materiales y métodos

La detección de *software* malicioso es realizada por varios filtros de contenido, tales como:

- **Surf-Secure**. Sistema desarrollado por la empresa PineApp1. Realiza el filtrado de páginas web empleando ACR (del inglés, *Advanced Content Recognition*), técnica novedosa que incluye elementos de Inteligencia Artificial. Inspecciona el tráfico HTTP y FTP, mediante F-Secure (un motor antivirus), para la protección contra virus, troyanos y gusanos. Bloquea el tráfico iniciado por el spyware; además, identifica y deniega diversas aplicaciones como las siguientes: *peer-to-peer* (P2P), mensajería instantánea, VoIP y juegos. Esto último permite optimizar el consumo del ancho de banda y reduce el ingreso de contenido malicioso a la red (PineApp, 2007).
- **Blue Coat WebFilter**. Producto que posee una base de datos con millones de URL que son organizadas en categorías temáticas, varias de las cuales están definidas para incluir páginas que hospedan o distribuyen *malware* (Blue Coat, 2009). Este sistema está sustentado por la tecnología de *Blue Coat*: Clasificación Dinámica en Tiempo Real (DRTR, por sus siglas en inglés). Ofrece un tratamiento a varias formas de *malware*, así como a otros contenidos ofensivos y/o poco productivos de la web (Westcon Security, 2010). Esto último es realizado mediante el análisis de: contenido del sitio web, formularios, enlaces, direcciones URL de origen y otros elementos.

Los sistemas de filtrado que fueron estudiados ofrecen poca documentación acerca de cómo implementan

[1] Empresa especializada en sistemas de seguridad para correo electrónico y redes.

exactamente las diferentes funcionalidades. Por lo general, el código fuente no está disponible libremente. Por otra parte, no se identificó ningún componente que realice la detección de *software* malicioso y que, a su vez, pueda ser incorporado de manera completa al filtro *Smart Keeper.*

Entre las características deseadas para el proceso de detección de software malicioso en *Smart Keeper* se encuentran:(1) la posibilidad de incorporar nuevos nodos de procesamiento para la ejecución en paralelo del análisis de archivos y (2) la adaptabilidad a varios antivirus. Los nodos de procesamiento para el escaneo de los archivos podrían estar ubicados en computadoras separadas conectadas en red. Para el envío de las solicitudes de trabajo hacia un nodo en concreto, es requerido establecer la comunicación entre el componente emisor de las solicitudes y los diferentes nodos (receptores). Para el intercambio de mensajes está definido el protocolo AMQP (del inglés, *Advanced Message Queuing Protocol*).

AMQP permite la interoperabilidad entre aplicaciones informáticas denominadas clientes y servidoras de mensajería, con el establecimiento de un conjunto de componentes y normas para la comunicación entre dichos componentes. Entre las características del protocolo se encuentran: multi-canal, negociado, asíncrono, seguro y eficiente; además cuenta con tres tipos principales de componentes (Vinculantes, Cola de mensajes e Intercambiadores) que intervienen en el intercambio de información. Permite una amplia variedad de arquitecturas para la mensajería (Group, 2006). Actualmente existen varios *software* que implementan dicho protocolo, como son:

- **RabbitMQ**. Sistema multiplataforma de mensajería (intermediario para la mensajería). Permite realizar de forma sencilla la persistencia de los datos y posibilita la confirmación de la entrega de los mensajes. La aplicación servidora de *RabbitMQ* está desarrollada con el lenguaje de programación *Erlang* y utiliza el *framework OPT* (del inglés, *Open Telecom Platform*). Los mensajes son enviados a través de Intercambiadores antes de arribar a una Cola. Existen aplicaciones clientes para varios lenguajes de programación. El código fuente está disponible bajo la Licencia Pública de Mozilla y cuenta con una comunidad de personas que contribuye a su desarrollo (Videla y Williams, 2012).
- **ZeroMQ**. Programa informático desarrollado por la corporación iMatrix junto a una comunidad de contribuidores. Fue concebida para ser utilizada en aplicaciones concurrentes o distribuidas. Cuenta con una cola de mensajes y permite el envío de información de manera asincrónica. Define cuatro patrones de mensajería (Solicitud-respuesta, Publicación-suscripción, pipeline y Par exclusivo) y requiere que al menos uno de los patrones sea utilizado. Con el empleo de los patrones permite desarrollar aplicaciones con diferentes configuraciones. El sistema está desarrollada en C++ y existen programas de enlaces para varios lenguajes de programación. El código fuente posee licencia GNU-LGPL (del inglés, *Lesser General Public License*) (Hintjens, 2012).

Con el estudio de ambos *software* se identificó que RabbitMQ se adapta mejor a los requerimientos y posee

características y funcionalidades que facilitarán el desarrollo.

Una vez que la información se encuentra en el nodo de procesamiento al que le fue asignada la petición, es necesario entregar el fichero al antivirus disponible e identificar el resultado que ofrece. Existen varios sistemas que realizan la comunicación con antivirus, entre los cuales se destaca el siguiente:

- **Amavis**. Software que detecta virus en correos electrónicos. Dicho sistema permite la comunicación entre los Agentes de Transferencia de Correo (MTA, por sus siglas en inglés) y los analizadores de contenido (antivirus, filtros de correos spam, entre otros). Para la implementación del software se empleó el lenguaje de programación Perl. En la actualidad este producto puede interactuar con aproximadamente 40 detectores de virus (y resulta fácil su extensión a otros). Contiene un fichero de configuración donde se especifica los antivirus con los cuales puede trabajar, así como los parámetros para la comunicación con cada uno de ellos. Es *software* libre, licenciado bajo la GNU-GPL (del inglés, *General Public License*) versión 2 (Martinec, 2012).

En el estudio realizado no se identificó ningún componente de Amavis que pudiese ser empleado en la propuesta; sin embargo, su estudio permitió indagar sobre las opciones para la comunicación con varios antivirus.

Además se empleará mongoDB, una base de datos orientada a documentos (Chodorow y Dirolf, 2010)**,** para registrar las peticiones realizadas por los usuarios. El lenguaje de programación a utilizar será Java, por ser el usado en el desarrollo de *Greasyspoon* (herramienta a partir de la cual se realiza la detección de *software* malicioso en *Smart Keeper).*

Resultados y discusión

Se propone el desarrollo de un subsistema para la detección de *software* malicioso, el cual será incorporado al filtro de contenido *Smart Keeper.* Las funcionalidades del *Greasyspoon* continuarán empleándose para trabajar con la respuesta obtenida de Internet. La propuesta está conformada por dos elementos fundamentales: el componente para el envío de solicitudes de trabajo y la capa de abstracción para la comunicación con los antivirus. Esta separación contribuye a una mayor modularidad de la propuesta.

Para el envío de mensajes se utilizará *Rabbitmq*; este *software* ofrece varias ventajas, entre las que destacan:

- la incorporación de nuevos nodos de procesamiento puede efectuarse sin necesidad de detener el funcionamiento del sistema.
- la asignación de trabajo a un nodo específico puede realizarse en dependencia de la disponibilidad del mismo.

La capa de abstracción contará con un fichero de configuración para indicar el antivirus disponible, así como los parámetros para la comunicación con el antivirus. Este mecanismo posibilita que la incorporación de un nuevo

Revista Cubana de Ciencias Informáticas
Vol. 6. No. 4, Octubre –Diciembre, 2012
ISSN: 2227-1899 | RNPS: 2301
Pág. 62-69
http://rcci.uci.cu

antivirus no sea un proceso engorroso, pues sólo requerirá de una nueva entrada en el fichero de configuración. La comunicación se realizará mediante el empleo de las opciones que ofrecen los antivirus a través de la línea de comando. El archivo a escanear será guardado en el disco duro del ordenador, con el fin de evitar su almacenamiento completo en la RAM.

La Figura 1 muestra cómo se relacionará el subsistema con algunos de los componentes que integran el filtro. La respuesta a la petición formulada por el usuario es entregada por *Squid a Greasyspoon* (en este último se obtiene el archivo). Si el archivo en cuestión fue procesado completamente con anterioridad, la respuesta registrada será entregada de manera inmediata al usuario sin necesidad de repetir el análisis. En caso contrario, la ruta del fichero (para que este sea escaneado) será enviada mediante *Rabbitmq* a la capa de abstracción. Los diferentes estados por los que transita el fichero serán actualizados en una base de datos. Cuando se realiza una solicitud de descarga de un archivo, *Greasyspoon* remite al usuario solicitante a una página web que, mediante consultas a la base de datos, mostrará el estado del archivo hasta que finalice el proceso.

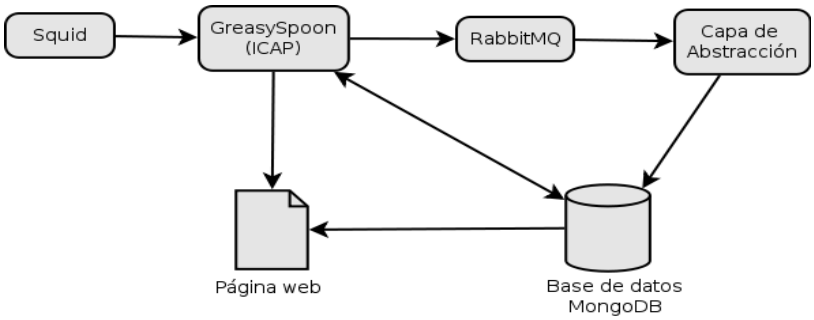

Figura 1. Subsistema que contribuye a la detección de *software* malicioso para el filtro de contenido Smart Keeper.

Como los diferentes componentes que integran la propuesta podrían estar en computadoras separadas y conectadas en red, se garantizará que los archivos a escanear estén disponibles para todos los componentes que lo necesiten dentro del subsistema. El cliente podrá configurar si el archivo será entregado o no al usuario en los siguientes casos: no hayan nodos para escanear u ocurra un error en el sistema luego de que el fichero haya sido descargado. La tarea quedará pendiente si el archivo no es entregado al usuario y existiese una ausencia de nodos para escanear. Cuando una tarea clasificada como pendiente es finalmente resuelta, el sistema informará al usuario (por correo electrónico) si existe o no *software* malicioso en el archivo.

Conclusiones

La caracterización del proceso de detección de *software* malicioso en el filtro de contenido *Smart Keeper* permitió: identificar las principales deficiencias de este proceso y definir las propiedades que fueron incorporadas/modificadas en dicho proceso. El estudio de los filtros de contenido, las herramientas para el intercambio de mensajes y el sistema para la detección de virus en correos electrónicos contribuyó a conformar la base tecnológica para la concepción del subsistema propuesto. La propuesta corrige varias deficiencias actuales del proceso de detección de *software* malicioso en *Smart Keeper*, lo cual permitió ampliar las facilidades de dicho filtro de contenido.

Referencias

- BLUE COAT. *Blue Coat WebFilter URL Categories* [online] 2009 S.l.: s.n. [Consultado el: 20 de septiembre de 2012]. Disponible en: [http://www.clm.com.br/produtos/bluecoat/pdf/BlueCoat-WebFilter-URL-Categories.pdf].
- CHODOROW, KRISTINA y DIROLF, MICHAEL. *MongoDB: The Definitive Guide*. 2010. [online]. S.l.: s.n. ISBN 1449381561. [Consultado el: 22 de septiembre de 2012]. Disponible en: [http://www.google.com.cu/url?sa=t&rct=j&q=mongodb+the+definitive+guide+pdf+kristina+chodorow&source=web&cd=1&cad=rja&ved=0CCIQFjAA&url=http%3A%2F%2Fmoodle.openfmi.net%2Fpluginfile.php%2F22821%2Fmod_folder%2Fcontent%2F1%2FMongoDB_The_Definitive_Guide.pdf%3Fforcedownload%3D1&ei=ECVeUIW4EtGF0QH60IFo&usg=AFQjCNHEnyu5X5O2mVMtwUVE8avAnjP7lw].
- HINTJENS, Pieter, 2012. *ØMQ - The Guide*. [online]. 2012. [Consultado el: 8 de julio de 2012]. Disponible en: [http://zguide.zeromq.org/page:all].
- KRAMER, SIMON y BRADFIELD, JULIAN. *A General Definition of Malware*. En: *Journal in Computer Virology*. 2010. Vol. 6, no. 2, pp. 105–114. DOI 10.1007/s11416-009-0137-1.
- MARTINEC, MARK, *amavisd-new*. [online] 2012. [Consultado el: 4 de julio de 2012]. Disponible en: [http://amavis.org/].
- PINEAPP. *Surf-Secure Filtrado Web en tiempo real* [online]. 2007. S.l.: s.n. [Consultado el: 1 de julio de 2012]. Disponible en: [http://www.pineapp.com/de/downloadfile.php?file=U3VyZi1TZUN1cmVfQnJvY2h1cmVfR2VybWFuLnBkZg==].
- VIDELA, ALVARO y WILLIAMS, JASON J.W. *RabbitMQ in Action DISTRIBUTED MESSAGING FOR*

EVERYONE [online]. S.l.: s.n. [Consultado el: 20 de septiembre de 2012]. ISBN 9781935182979. Disponible en: [http://filepost.com/files/c6f9a1da/RabbitMQ.in.Action.pdf/].

- W. GROUP. *AMQP - A General-Purpose Middleware Standard* [online] 2006. S.l.: s.n. [Consultado el: 4 de julio de 2012]. Disponible en: [http://xml.coverpages.org/AMQPv0-10.pdf].
- WESTCON SECURITY, 2010. *Blue Coat Webfilter - Block malware and filter content according to strict policy controls* - Westcon Security UK - Westcon Security UK. [online] 2010 [Consultado el: 21 de septiembre de 2012]. Disponible en: [http://uk.security.westcon.com/content/vendors/blue-coat/blue-coat-webfilter].