COMPLEMENTANDO LAS SOLICITUDES CON UN MONITOREO DE RED AUTOMATIZADO

Lic. Javier F. Diaz
jdiaz@unlp.edu.ar

AC. Marisa Andrea Malvaso
mmalvaso@info.unlp.edu.ar

Facultad de Ciencias Exactas
Universidad Nacional de La Plata

ABRIL DE 1999

RESUMEN

El objetivo de realizar monitoreo en una red es, generalmente, identificar un problema de performance o analizar aspectos de diseño y utilización de la misma. Para poder llevar a cabo esta tarea se debe comprender el entorno de los usuarios, las solicitudes de servicios que los mismos realizan y qué clase de respuestas obtienen.

Una técnica comunmente utilizada en el análisis del comportamiento de una red es el proceso de registro de logs, *logging*, que provee información exhaustiva respecto de las actividades que realizan determinada población de usuarios. Actualmente existen cuatro categorías de logging: basado en el servidor, basado en el proxy, basado en el cliente y basado en la red (*network monitoring*)[1].

Luego de un análisis exhaustivo y una comparación de los métodos anteriormente mencionados, en el presente artículo se realiza una descripción de los mismos, comparándolos y analizando las ventajas y desventajas que presentan. En términos más específicos, considerando que la World Wide Web es en la actualidad una fuente dominante del tráfico en el backbone de Internet, y muy pronto también lo será en redes de área local, se seleccionan dos herramientas que permiten a estaciones de trabajo UNIX recolectar de la red logs del tráfico de Web[2]; y se estudia, entre otros, su eficiencia y performance. *httpfilt*, la primera de las herramientas analizadas, basada en tcpdump[3], monitorea el tráfico desde y hacia ports específicos. *httpdump*[4] es la segunda herramienta analizada, que presenta mayor flexibilidad que la anterior y un aumento en la eficiencia como consecuencia de la utilización de un filtro de paquetes altamente optimizado[5].

1 - INTRODUCCIÓN

Con el explosivo crecimiento de la World Wide Web[6] surge la necesidad de caracterizar a quienes utilizan el servicio de Web y qué tipo de accesos se realizan. Por otra parte, para asistir al planeamiento del crecimiento de las redes de computadoras, quienes realizan network management[7] requieren información del tráfico actual generado por el Web y una proyección del tráfico futuro. Para responder a este tipo de preguntas, está disponible la siguiente técnica:

> Logging: un proceso de registro de logs que consiste en almacenar en un archivo, en forma automática, una secuencia de eventos de máquina

observables. Normalmente, esos eventos son *solicitudes* (por ejemplo: el retorno de un documento, la ejecución de un script, el download de un applet, ...) o *respuestas* (por ejemplo: documentos contenidos en una solicitud, salidas desde los scripts, código de applets, ...). Técnicamente, el logging puede realizarse sin conocimiento de los usuarios, a pesar de que eventualmente esto podría ser prohibido por normas éticas o legales.

1.1 - LOGGING

El *logging* puede representar un registro casi perfecto[1] de las solicitudes alcanzadas por un servidor de Web o por un proxy, siempre que hayan sido enviadas a través de la red o tipeadas en un browser. Existen distintos métodos para realizar logging, y los mismos se pueden caracterizar utilizando el lugar de la Web desde donde se recolecta la información.

En la arquitectura de Web más sencilla, cada vez que un cliente envía una solicitud, es enviado un paquete HTTP (Hypertext Transfer Protocol)[8] sobre la red, desde el cliente hacia el servidor nombrado en el campo URL (Universal Resource Locator)[9] de la solicitud. Luego el servidor retorna uno o más paquetes conteniendo o bien la respuesta, o bien un código de error. Dicha arquitectura es algo más compleja cuando se utilizan servidores proxy, ya sea que desempeñen el rol de firewall, de cache[10], o de ambos.

Existen cuatro lugares posibles desde donde registrar logs, correspondiendo a las cuatro categorías de logging: en el servidor, en el proxy, en el cliente y sobre la red.

2 - CATEGORÍAS DE LOGGING

2.1 - LOGS BASADOS EN UN SERVIDOR DE WEB

El logging realizado en un servidor es la forma más ampliamente utilizada[16]. En esta estrategia se captura información acerca de las solicitudes de los clientes a un simple sitio Web, en cualquier parte del mundo en que el mismo se encuentre. Los servidores de Web pueden recolectar un log de solicitudes de documentos registrando cada HTTP GET, POST o HEAD[8] que reciben.

Existen límites en la información que proveen los logs de un servidor de Web:

- Un gran número de solicitudes puede provenir desde servidores proxy. De esta manera, el nombre del host del cliente tal vez sea anónimo, debido a que el log del servidor contiene el nombre del servidor proxy.
- El administrador de Web tal vez deshabilite la posibilidad de recolectar los nombres de los hosts clientes, debido a que dicha recolección puede no ser ética o legal[11].
- El log del servidor solamente contiene aquellas solicitudes que actualmente alcanzan el servidor, y excluye aquellos casos en los que el usuario examina un documento que está cacheado por un servidor proxy o por el Web browser del cliente. De esta manera, debido a que la utilización de caches, particularmente de proxy caches, crece día a día en la Web, declina la precisión de los logs de servidores como una medida de solicitudes de páginas Web.

---

[1] Se dice "casi perfecto" porque el dispositivo que recolecta los logs puede fallar, el dispositivo de recolección (por ejemplo el software) puede contener bugs, o los logs recolectados pueden ser maliciosamente modificados o fabricados.

Finalmente, existen distintas consideraciones que, en ciertos casos, hacen más dificultosa la implementación de esta categoría de logging.

- Si existe más de un servidor que provee el mismo servicio, surge la necesidad de relacionar los logs entre los mismos y de planear una estrategia para que toda la información registrada conserve el mismo formato y no sea redundante.
- Si en un servidor se provee más de un servicio, se debe tener especial cuidado para que los procesos de registro de información no interfieran unos con otros.

Se podría pensar que la solución ideal para estas circunstancias es el agregado de un host entre los clientes y servidores, que sea quien realice el proceso de logging, funcionando como un *filtro* para el acceso a los servicios. Las principales desventajas están relacionadas con la inserción de un único punto de falla[2] y la posible degradación en la performance de un servicio como consecuencia de un alto número de solicitudes que no están dirigidas al mismo pero que son filtradas por el mencionado host.

2.2 - Logs basados en un Servidor Proxy

El logging realizado en un servidor proxy captura la información que llega a dicho servidor. De esta manera caracteriza el conjunto de solicitudes realizadas por una población de clientes, que realizan solicitudes a servidores Web en cualquier parte del mundo, y cuyos browsers están configurados para utilizar el servidor proxy.

La realización de logging en servidores proxies también tiene limitaciones:

- La mayoría de los Web browsers crean caches en la memoria en forma automática y siempre crean caches en los discos de las máquinas en donde corren dichos browsers. De esta manera, aquellas solicitudes de los usuarios que son satisfechas por el cache de memoria o de disco en el cliente no son enviadas al servidor proxy, y así el log de dicho servidor no las puede capturar.
- A diferencia de los caches de los browsers que son creados en forma automática, en la actualidad los Web browsers solamente utilizan un servidor proxy cuando el usuario explícitamente lo configura para tal finalidad. Como concecuencia de ello, los logs de un servidor proxy podrían representar sólo una muestra parcial de clientes, aquellos que poseen usuarios lo suficientemente sofisticados como para conocer la existencia de un servidor proxy y la manera de configurar el browser para su utilización.

De esta manera, los costos ocultos de esta categoría de logging involucran, entre otros, el proceso de configuración de los clientes y servidores para la utilización del servidor proxy.

2.3 - Logs basados en un Cliente

En este caso el logging es realizado en una máquina cliente, ya sea por el mismo Web browser o por un proceso separado de monitoreo que ejecuta en dicha máquina. Los logs pueden ser una caracterización precisa de cada documento que examina el usuario[12].

---

[2] Si este host falla, además de impedir que se continúe con el logging, se esta quebrando la conexión con los servidores, con lo cual disminuye la disponibilidad del servicio.

En la actualidad, normalmente los clientes no tienen facilidades adicionales para generar logs. De esta manera, no existe un formato o un conjunto de información a listar "estándar" para los logs de clientes, de donde se deduce la dificultad que provoca unificar la información obtenida en cada uno de ellos. De todos modos, técnicamente es posible registrar este tipo de logs. Por ejemplo, los Web browsers NCSA Mosaic[13] fueron modificados en las universidades para recolectar logs de clientes[3] (por ejemplo en la Universidad de Boston[12]). Otro ejemplo del potencial para el logging basado en el cliente es HindSite[14], un plug-in para Netscape Navigator[4].

La ventaja del logging basado en el cliente es que éste se sobrepone a los problemas anteriormente mencionados que ocurren en las otras dos estrategias, debido a que también registra información de aquellas solicitudes que fueron satisfechas por el cache local del cliente (ya sea en memoria volátil o en disco). Esto se debe a que el logging basado en el cliente puede almacenar todos los accesos generados por un usuario, antes de que los mismos sean dirigidos al servidor de Web o al proxy cache.

La limitación obvia del logging basado en el cliente es que la información puede ser recolectada solamente desde aquellos clientes que estén dispuestos a ejecutar un Web browser que registre logs.

2.4 - LOGS BASADOS EN LA RED

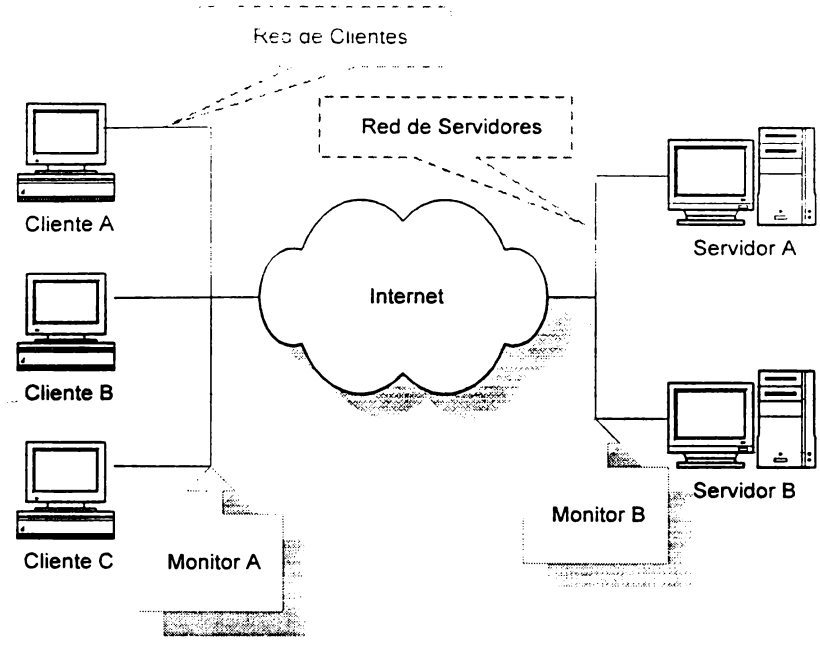
En esta alternativa, una máquina es conectada a la red para que realice el monitoreo de la misma[5]. El monitor escucha pasivamente todo el tráfico que viaja en la red, identifica paquetes que contienen partes de mensajes HTTP y construye un log de las URLs solicitadas en dichos paquetes. La población de clientes que el logging basado en la red puede caracterizar depende del lugar de la red en donde se conecta el monitor. Por ejemplo, un monitor ligado a una red de hosts clientes que ejecutan Web browsers, como es el caso del "monitor A" en el siguiente gráfico, registra información respecto de dicho conjunto de clientes.

---

[3] Donde se registran los siguientes items: nombre de la máquina, hora, URL, tiempo de retorno y si la referencia URL fue retornada directamente o desde una referencia en otra URL.

[4] Este plug-in almacena en una base de datos la URL, título, fecha y hora de acceso, fecha y hora de la última modificación del documento y tamaño del documento para cada URL solicitada por el usuario.

[5] Una máquina que realiza monitoreo de red, *network monitor*, también suele llamarse *network sniffer*.



Alternativamente, un monitor ligado a una red que conecta uno o más servidores de Web, como es el caso del "monitor B" en el gráfico anterior, registra información respecto de solicitudes destinadas a cualquiera de dichos servidores.

Finalmente, los monitores de red pueden ser útiles para mejorar el logging basado en un servidor proxy. Por ejemplo: el monitor puede registrar solicitudes de clientes que no están configurados para utilizar un servidor proxy y de clientes que no están conectados a redes con servidores proxies.

El dispositivo que monitorea la red puede ser tanto un instrumento de monitoreo (como por ejemplo un LANalyzer para Ethernet) como un computador de propósito general. Un ejemplo de utilización de un computador de propósito general para monitoreo es la herramienta *tcpdump*[6].

Esta estrategia de monitoreo de red, que no es nueva[7], posee varias ventajas:

- Es transparente: no se requieren cambios en el cliente, proxy o servidor de Web, y no tiene impacto sobre la performance de los clientes, proxies, servidores de Web o sobre la red.
- Es segura: nadie puede acceder a los datos monitoreados, excepto quien controle el dispositivo de monitoreo.
- Puede utilizarse para verificar información registrada por otros mecanismos de logging.

---

[6] Para utilizar *tcpdump*, se debe configurar el adaptador de red de un sistema operativo UNIX en modo "promiscuo", para que éste acepte todos los paquetes que viajan por la red, más que aquellos que están destinados a la dirección IP de la estación de trabajo en donde la misma está instalada. *tcpdump* decodifica los paquetes para producir una lista de los campos que se encuentran en los headers de los protocolos (por ejemplo TCP e IP).

[7] Los monitores han sido utilizados por mucho tiempo como herramientas para análisis de problemas en redes de comunicaciones.

- El monitor de red puede estar programado para registrar información de otros protocolos además del HTTP[8].
- No existe problema de muestreos o auto-selección: el monitor recolecta todas las solicitudes de documentos, y si el mismo es lo suficientemente lento como para no poder registrar todos los paquetes de la red, realiza una muestra aleatoria.
- Un monitor de red puede utilizar el header de los mensajes HTTP capturados para realizar cálculos basados en múltiples paquetes, y de esta manera producir más información respecto de la utilización del Web de la que se puede obtener con el formato de logs estándar de los servidores proxy o de Web[9].

La principal desventaja del monitoreo de red es la necesidad de o bien una red que permita broadcast (tal como una Ethernet, token ring o FDDI ring) o bien configurar el monitor como un gateway sobre un enlace de red punto-a-punto.

Por otro lado, se debe tener especial cuidado con el número de clientes y servidores que posee la red, debido a que existe un límite en el porcentaje de paquetes que puede registrar el monitor. Una solución a esta restricción puede ser conectar más de un monitor a la red, configurando cada uno de ellos para que registre un conjunto de paquetes provenientes de diferentes direcciones de clientes o servidores. Para tener una aproximación de cuánto tráfico puede manejar un monitor de red, se puede considerar un computador de propósito general como dispositivo de monitoreo. Dicho monitor debería estar capacitado para manejar al menos el tráfico que podría manejar el servidor de Web más rápido, corriendo sobre la misma plataforma de hardware. Esto se debe a que el monitor de red lee paquetes de red pero nunca los escribe[10], mientras que un servidor de Web debe leer paquetes, encontrar el documento solicitado en el disco y escribir paquetes.

3 - Aspectos de Privacidad

Cualquier estrategia de logging (ya sea basada en el servidor, en el cliente, en el proxy o en la red) promueve aspectos de privacidad. Estos aspectos suelen discutirse debido a que en algunos países no existen guías de ética o reglas legales que describan cómo pueden utilizarce los registros de logs sin atentar contra la privacidad de los usuarios[15]. Por ejemplo: actualmente en la Web, los servidores y proxies recolectan en forma rutinaria logs; que al almacenar los nombres de los hosts clientes; identifican usuarios en forma indirecta[11].

En particular, el logging basado en la red acentúa este aspecto, debido a que se realiza en forma transparente, sin modificaciones en los clientes o servidores.

---

[8] Además de capturar información de otros protocolos populares en la Web (FTP, Real audio,...) podría capturar eventos tales como conexiones no concluídas o erróneas, así como también tráfico excesivo de algún protocolo en el que no suele repararse (ICMP, POP,...) y puede degradar la performance de la red cuando se utiliza en forma indiscriminada.

[9] Realizar este tipo de cálculos en un servidor que tiene una funcionalidad tal como cachear, filtrar tráfico (firewall) o proveer un servicio (en este caso Web), puede ser crítico para su óptimo funcionamiento.

[10] Debe existir otro proceso dedicado a tal finalidad para no degradar la performance del servidor.

[11] Además, algunos de estos servidores utilizan herramientas adicionales para recolectar información de los usuarios conectados a los clientes y combinan esa información con bases de datos que reúnen características personales de los mismos.

4 - MONITORES DE RED PARA EL WEB

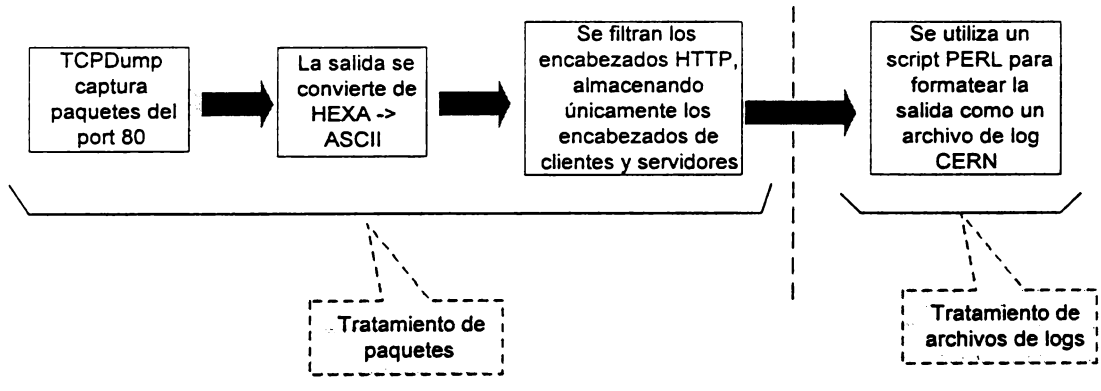
Dos de las principales herramientas que permiten a una estación de trabajo UNIX recolectar logs de una red son *httpfilt* y *httpdump*. Ambas herramientas permiten registrar logs de todas las solicitudes HTTP que ocurren en la red a la cual está ligada el monitor. *httpfilt* permite registrar solicitudes HTTP con un alto rendimiento, mientras que *httpdump* provee información más extensa de cada solicitud y puede ser extendida para registrar tráfico proveniente de otros protocolos diferentes de HTTP.

Para maximizar su utilidad, la salida que se obtiene de *httpfilt* y *httpdump* se encuentra en un formato que permite que otras herramientas de análisis de WWW puedan beneficiarse utilizándolas, y además crean una versión extendida de dicho formato de registro de información. El formato utilizado se conoce como el archivo de log de *formato común*[17].

En el momento de analizar las herramientas, se deben tener en cuenta distintos aspectos que permitan arribar a conclusiones respecto de su utilidad, rendimiento y flexibilidad, entre otras características de importancia. Por ejemplo, sería de utilidad evaluar el tipo y cantidad de tráfico que permiten recolectar, el rendimiento en cuanto a performance, formato de la información registrada y en qué momento permiten visualizar la mencionada información[12].

4.1 - HTTPFILT

httpfilt, la primera de las dos herramientas, usualmente identificada como una variante de *httpdump*, filtra la salida del popular paquete *tcpdump*[3] para producir uno de los dos archivos de salida. *tcpdump* es ejecutado con parámetros que permiten recolectar únicamente paquetes que están dirigidos al port 80 (el port estándar de httpd) y su salida es conducida a través de tres scripts PERL que se ejecutan en forma separada como procesos concurrentes.



Cada uno de los filtros realiza las siguientes operaciones:

- Primer filtro: invoca a *tcpdump* para capturar todos los paquetes de la red destinados al port 80 sobre cualquier dirección IP.
- Segundo filtro: convierte el contenido del paquete de hexadecimal a ASCII, y lo pasa al siguiente filtro.

[12] Es decir, ¿permite la herramienta visualizar los logs registrados en tiempo real?, o ¿primero debe realizarse el registro de datos y posteriormente su visualización?.

- Tercer filtro: examina los caracteres iniciales del campo de datos de cada paquete, buscando alguna de las siguientes cadenas de caracteres: GET, HEAD, HTTP y POST, para identificar paquetes que contienen un encabezado HTTP. Los paquetes obtenidos como resultado son enviados al último filtro.
- Cuarto filtro: categoriza cada paquete y genera el archivo de logs en el formato especificado.

4.2 – HTTPDUMP

Una limitación de *httpfilt* es que solamente permite monitorear tráfico desde y hacia ports especificados. De esta manera, se debe conocer con anterioridad sobre qué números de ports, en cualquier red, se ejecutan servidores de Web, y si alguno se desconoce, el tráfico que lo involucra no será registrado. La segunda herramienta, *httpdump*, se sobrepone a esa limitación, pero su actual implementación es específica para una DECstation ejecutando un sistema operativo Ultrix.

httpdump está basada en una herramienta de captura de paquetes altamente optimizada en lugar de utilizar *tcpdump*, y de esta manera, permite monitorear todos los números de ports para filtrar luego aquellos paquetes que contengan un encabezado de HTTP. Por otra parte, se incrementa el nivel de performance implementando todos los filtros y pasos de conversión en código "C".

httpdump consiste actualmente de dos programas "C" que se ejecutan en forma concurrente para producir logging de red de las solicitudes de Web. El primero de los mencionados programas, *tcppf* (TCP packet filter), crea un filtro que monitorea los paquetes de red para identificar paquetes TCP y enviar sus datos al segundo programa, *httpc* para un análisis posterior, y registrar, finalmente, entradas en el archivo de logs de formato especificado.

5 – CONCLUSIONES Y TRABAJO FUTURO

La necesidad actual de un análisis de problemas de performance y de aspectos de diseño y utilización de las redes, desencadenan un impulso en el desarrollo de la actividad de monitoreo de tráfico. Esto genera un proceso de seguimiento exhaustivo de las actividades realizadas por los usuarios (estrategias de logging).

En general, el *logging* crea la ilusión de un perfecto conocimiento de las actividades que se realizan en la red. Sin embargo, debido a la existencia de sus distintas categorías (basadas en el servidor, en el proxy, en el cliente o en la red), no se puede afirmar que una de ellas es la panacea. A pesar de que la alternativa que promete la mayor cantidad de ventajas es el logging basado en la red (comúnmente llamado *monitoreo*) deben ser evaluadas las características del entorno al momento de seleccionar alguna de las mismas.

Si se opta por el logging basado en la red, existen diferentes herramientas que brindan la posibilidad de registrar logs de solicitudes HTTP que ocurren en la red a la cual está ligada el monitor, entre ellas *httpfilt* y *httpdump*.

Actualmente estamos desarrollando una variante de la herramienta *httpdump* que sea portable a otros sistemas UNIX.

6 – REFEFERENCIAS

[1] "Sincronización de estadísticas entre servidores y proxies", tesis de Licenciatura en Informática, de la Facultad de Ciencias Exactas, UNLP. Autor: María José Catullo. Director: Lic. Javier Diaz.

[2] "World Wide Web (WWW) Traffic Analysis Tools". Computer Science Department Virginia Polytechnic and State University.

http://www.cs.vt.edu/~chitra/WWWTrafficTools.html.

[3] En la siguiente dirección
http://science.nas.nasa.gov/Groups/LAN/ClassNotes/ant/tcpdump.html se encuentra una breve descripción de la herramienta, disponible en:
http://ftp.ost.eltele.no/pub/networking/tcpdump/

[4] Roland Wooster, Stephen Williams, Patrick Brooks, "HTTPDUMP Network HTTP Packet Snooper", Abril de 1996.

[5] packetfilter, man page.
http://apache.ethz.ch/Digital_UNIX/MAN/MAN7/0060____.HTM

[6] "Internet Growth", http://www.is-bremen.de/~mhi/inetgrow.htm
http://www.insead.fr/CALT/Encyclopedia/ComputerSciences/Internet/growth.html
"Internet Domain Survey", http://www.nw.com/zone/WWW/report.html

[7] "The SimpleWeb", http://wwwsnmp.cs.utwente.nl/
"SNMP Version 3 (SNMPv3)", http://www.ibr.cs.tu-bs.de/ietf/snmpv3/

[8] "Hypertext Transfer Protocol –HTTP/1.1". RFC 2068.
"Hypertext Transfer Protocol –HTTP/1.0". RFC 1945.

[9] "Uniform Resource Locator (URL)". RFC 1738.

[10] A.C. Marisa A. Malvaso y Lic. Miguel A. Luengo, "Alternativas de Caching para optimizar el ancho de banda", Septiembre de 1997.

[11] J. Berman, J. Goldman, D. J. Weitzner y D. K. Mulligan, "Statement of the Center for Democracy and Technology before the Federal Trade Commission Workshop on Consumer Privacy on the Global Information Infraestructure", Junio de 1996, http://www.cdt.org/publications/FTC_June96_test.test.html.

[12] En la siguiente dirección están disponibles ejemplos de logs en clientes:
ftp://cs-ftp.bu.edu/techreports/95-010-www-client-traces.tar.gz,
C. R. Cunha, A. Bestavros y M. E. Crovella, "Client Log Traces"

[13] "WWW Tools", http://www.sparc.spb.su/Cos/hotlist.html

[14] ISVS HindSite para Netscape Navigator:
http://www.isysdev.com/products/hindsite.htm

[15] Center for Democracy and Technology, "CDT Privacy Issues Page",
http://www.cdt.org/privacy/otherheadlines.html

[16] Un ejemplo de herramienta que realiza logging en el servidor para IIS puede ser ISAPI. http://www.microsoft.com/MSJ/0498/IIS/IIS.HTM

[17] "W3c httpd common log format". WWW Consortium.
http://www.w3.org/pub/WWW/Daemon/User/Config/Logging.html