



Evaluación de diferentes alternativas de control de acceso y filtrado Web.

Autor: Juan Eduardo Funes

Director: Ing. Luis Marrone

**Trabajo final integrador para obtener el postgrado de
Especialización en Redes y Seguridad.**

**Facultad de Informática – Universidad Nacional de
La Plata agosto de 2023**

RESUMEN

El presente trabajo tiene su campo de acción en Internet, en especial en la navegación de los usuarios y la posibilidad de ejercer control sobre los mismos. Inicialmente, en el capítulo 1, se verá la Arquitectura de la Web, para conocer la historia y los diferentes componentes de la web (HTML, XML, HTTP, TLS, etc) y su evolución. Luego en el capítulo 2 se verán distintas metodologías y herramientas para el filtrado Web. En el capítulo 3 se definirá el escenario a trabajar, en el cual, se tomará una muestra de navegación de un mes de un organismo de la administración pública, con un IDS conformando un lote de prueba, sometiendo a diferentes herramientas de acceso y filtrado Web comerciales y populares de código abierto a su configuración, evaluación, categorización y filtrado del mismo.

En el capítulo 4 estarán las conclusiones del lote de prueba, teniendo como objetivo, asesorar la herramienta o metodología adecuada para lograr un eficiente filtro de acceso Web en un organismo de la administración pública.

Índice de contenido

Objetivo general:	7
Objetivos específicos:	7
INTRODUCCION	7
Capítulo 1. Arquitectura de la Web.....	10
Introducción	10
1.1 Funcionamiento.....	10
1.1.1 HTTP su evolución	13
1.1.1.1 HTTP 0.9.....	13
1.1.1.2 HTTP 1.0.....	13
1.1.1.3 HTTP 1.0+.....	13
1.1.1.4 HTTP 1.1.....	14
1.1.1.5 SPDY	15
1.1.1.6 HTTP 2.....	15
1.1.1.7 HTTP 3.....	15
1.1.1.8 Mensajes HTTP.....	17
1.2 Recursos.....	20
1.2.1 Sintaxis	20
1.3 Lenguajes para la creación de páginas Web.....	22
1.3.1 HTML.....	22
1.3.2 XML	23
1.3.3 LENGUAJES PARA LA GENERACION DE CONTENIDO INTERACTIVO EN EL CLIENTE.....	24
1.3.4 GENERACION DE PAGINAS WEB DINAMICAS CON CGI.....	25
1.3.5 GENERACION DE PAGINAS WEB DINAMICAS CON PHP	25
1.3.6 ALTERNATIVAS DE GENERACION DE PAGINAS WEB DINAMICAS ..	25
1.4 Interpretación de contenidos - MIME.....	26

Capítulo 2. Metodologías y herramientas de filtrado.....	27
2.1 METODOLOGIAS.....	27
2.1.1 Listas Negras.....	27
2.1.2 Listas Blancas.....	28
2.1.3 Filtrado Web por extensiones de archivos.....	28
2.1.4 Filtrado Web por MIME.....	29
2.1.5 Filtrado Web por palabras y frases en sitio web.....	29
2.1.6 Filtrado Web por headers.....	29
2.2 HERRAMIENTAS DE FILTRADO.....	29
2.2.1 SQUID CON SQUIDGUARD.....	30
2.2.2 OpenDNS.....	30
2.2.3 FORTIGATE – FORTINET.....	31
2.2.4 INTERSCAN Web Security - TREND MICRO.....	32
Capítulo 3. Configuración y comprobación de las diferentes propuestas.....	33
3.1 ESCENARIO DE COMPROBACION.....	33
3.2 SQUID CON SQUIDGUARD.....	36
3.3 OPENDNS.....	39
3.4 FORTIGATE – FORTINET.....	44
3.5 INTERSCAN WEB SECURITY - TREND MICRO.....	46
Capítulo 4 - CONCLUSIONES.....	50
4.1 Conclusiones.....	50
4.2 Futuras líneas de investigación.....	53
Capítulo 5 - BIBLIOGRAFIA.....	54

Índice de figuras

Figura 1-1: Diagrama de conexión a una página web	11
Figura 1-2: Diagrama de conexión HTTP 1.1.....	14
Figura 1-3: Encabezados HTTP.....	18
Figura 1-4: Código de respuesta HTTP.....	19
Figura 1-5: Evolución HTML.....	23
Figura 1-6: Tipos y subtipos MIME.....	26
Figura 3-1: Archivo sitios.txt	35
Figura 3-2: Archivo resultado.txt.....	36
Figura 3-3: Configuración de redes en OpenDNS.....	40
Figura 3-4: Comprobación de funcionamiento OpenDNS en Windows.....	40
Figura 3-5: Comprobación DNS en Linux.....	41
Figura 3-6: Comprobación de funcionamiento de navegación en Linux.....	41
Figura 3-7: Configuración de filtros web en OpenDNS.....	42
Figura 3-8: Configuración seguridad en OpenDNS.....	43
Figura 3-9: Configuración proxy explícito en Fortigate.....	44
Figura 3-10: Configuración de filtros web en Fortigate.....	45
Figura 3-11: Política de aplicación de acceso y filtrado en Fortigate.....	45
Figura 3-12: Licencia Trend Micro InterScan Web Security.....	46
Figura 3-13: Proxy explícito Trend Micro InterScan Web Security.....	47
Figura 3-14: Filtros Trend Micro InterScan Web Security.....	48
Figura 3-15: Creación de política Trend Micro InterScan Web Security.....	49
Figura 3-16: Vista de política creadas - Trend Micro InterScan Web Security.....	49
Figura 4-1: Explotación de resultados del archivo resultado.txt.....	50
Figura 4-2: Conclusiones en base a los resultados.....	51
Figura 4-3: Ejemplo resultado filtrado con OpenDNS.....	52
Figura 4-4: Ejemplo resultado filtrado con Fortigate.....	52
Figura 4-5: Ejemplo resultado filtrado con Trend Micro InterScan Web Security.....	53

Objetivo general:

Evaluar distintas metodologías y herramientas para el control de acceso y filtrado Web.

Objetivos específicos:

- ✓ Estudiar herramientas de códigos abiertos y comerciales para el control de acceso y filtrado Web.
- ✓ Evaluar sus diferencias en cuanto a metodología de filtrado.
- ✓ Proponer que metodologías y herramientas son las más robustas.

Con mi aporte se podrá evaluar distintas herramientas comerciales y de código abierto para el control de acceso y filtrado Web, determinando si los mecanismos utilizados por ambas son similares o exclusivos de cada solución, lo que nos dará un aporte significativo a la hora de emprender sobre dicha temática.

INTRODUCCION

De la experiencia obtenida trabajando para la administración y en el sector privado prestando mis servicios profesionales como consultor de Informática, he notado que cada día son más frecuentes y comunes las problemáticas relacionadas con el uso de Internet, no he trabajado en ninguna institución en donde el usuario pueda navegar a Internet libremente, siempre se solicitan distintos tipos de filtros y accesos a diferentes sitios según perfiles evaluados para cada solución.

Si bien a internet se las puede describir como un conjunto de partes sean los componentes hardware y/o software o como la infraestructura de red que proporciona servicios a aplicaciones (James F. Kurose, Keiht W. Ross, 2010), es cierto que el boom de internet fue durante los años 1990 y 2000 con la aparición de la Web llamados "sitios o páginas web" como se le dice al contenido web, estos crecieron en forma exponencial hasta que hubo millones de sitios y millones de páginas, algunos de ellos más populares y otros no tanto (Andrew S. Tanenbaum, David J. Wetherall, 2012)

La realidad es que al día de hoy Internet está compuesta por diferentes servicios y aplicaciones, pero el tráfico masivo es a través de la Web.

Si se quiere hablar de seguridad en Internet hay que tener en cuenta que la misma siempre fue considerada de naturaleza abierta, y que se ha implementado una capa de seguridad mediante la implementación capa de socket seguro (**SSL**, por sus siglas en ingles), la cual sufrió una actualización conocida como protocolo de seguridad en la capa de transporte (**TLS**, por sus siglas en inglés) (Andrew S. Tanenbaum, Marteen V. Steen, 2008). Hay muchas personas que consideran que accediendo a un sitio con estas características lo realizarán de manera segura ya que el mismo nos proporciona confidencialidad (cifrado), integridad y autenticación del servidor y autenticación del cliente (James F. Kurose, Keiht W. Ross, 2010), en la realidad si un sitio posee contenido malintencionado todos sus clientes serán víctimas del mismo.

Podemos decir que la Web y el correo electrónico, son los servicios más utilizados por los usuarios, como consecuencia los más vulnerables a poder sufrir alguna alteración en perjuicio del usuario. Si bien como hemos mencionado que internet es abierta, cada país determina su propio posicionamiento al tratamiento de la misma.

El 18 de diciembre de 2014 se sanciona la ley 27.078 (ley Argentina digital), que mediante el Título 1 (Disposiciones Generales - Capítulo I) artículo 1° - se declara de interés público el desarrollo de las Tecnologías de la Información y las Comunicaciones, las telecomunicaciones, y sus recursos asociados, estableciendo y garantizando la completa neutralidad de las redes con objeto de posibilitar el acceso de la totalidad de los habitantes de la República Argentina. y que mediante el artículo 2° marca como finalidad garantizar el derecho humano a las comunicaciones y a las telecomunicaciones y reconocer a las mismas como un factor preponderante en la independencia tecnológica y productiva de nuestra Nación (Ley argentina digital – 2014).

En dicha ley en el Título VII (Consideraciones generales sobre los Servicios de TIC) artículo 56. - Neutralidad de red se garantiza a cada usuario el derecho a acceder, utilizar, enviar, recibir u ofrecer cualquier contenido, aplicación, servicio o protocolo a través de Internet sin ningún tipo de restricción, discriminación, distinción, bloqueo, interferencia, entorpecimiento o degradación y la misma en su artículo 57 prohíbe a los prestadores de Servicios de TIC bloquear, interferir, discriminar, entorpecer, degradar o restringir la utilización, envío, recepción, ofrecimiento o acceso a cualquier contenido,

aplicación, servicio o protocolo salvo orden judicial o expresa solicitud del usuario (Ley argentina digital – 2014).

Como se puede observar la ley imposibilita a los prestadores bloquear o entorpecer el acceso a Internet de los usuarios, pero bien marca salvo expresa solicitud del usuario.

Si un ciudadano quisiera que se le bloquee cierto tipo de contenido como por ejemplo pornografía, el mismo no tiene herramientas para poder realizarlo. De la misma manera pasa con los organismos públicos del estado y empresas, en donde para evitar que sus funcionarios utilicen Internet como ocio y no como herramienta de trabajo, adquieren independientemente diferentes soluciones de empresas privadas para limitar y controlar el uso de Internet.

Dada esta continua problemática siendo Internet el medio de infección más popular y la Web como medio, me motiva poder investigar sobre las diferentes metodologías y herramientas para el filtrado de contenido en el uso de la Web.

Capítulo 1. Arquitectura de la Web.

Introducción

La Web (World Wide Web) fue iniciada en los años 1989 en el Centro Europeo de Investigación Nuclear (CERN), con el propósito de generar una herramienta para el intercambio de información, la propuesta surgió de Tim Berners-Lee, la misma se basó en la idea de (Bush, 1945), en una colección de contenido que vincula a distintos contenidos, lo que se conoce hoy como hipertexto.

La web “en 10 años paso de ser una manera de coordinar el diseño de los experimentos de física de alta energía en Suiza a la aplicación que millones de personas piensan que es “Internet”” (Andrew S. Tanenbaum, David J. Wetherall, 2012).

En el año 1994 se creó el W3C (Consortio World Wide Web), quien estaba integrado por el Centro Europeo de Investigación Nuclear (CERN) y el MIT y su principal objetivo fue la de definir estándares y fomentar la interoperabilidad entre sitios. Desde entonces miles de universidades y compañías se unieron al consorcio W3C.

1.1 Funcionamiento

El funcionamiento básico para la obtención de una página web consiste básicamente en lo siguiente.

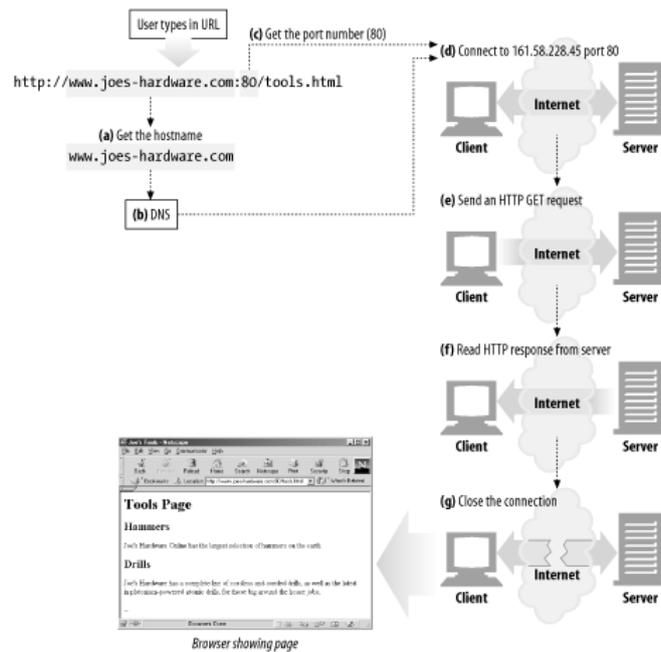


Figura 1-1: Diagrama de conexión a una página web.
Fuente: (HTTP: The Definitive Guide, 2002)

1. El usuario escribe en el navegador una dirección web.
2. El navegador pide al DNS¹ la dirección IP² del servidor.
3. El DNS responde con la dirección IP correspondiente.
4. El navegador realiza una conexión TCP a la dirección IP del sitio en el puerto 80, el puerto conocido para el protocolo HTTP.
5. Después envía una solicitud HTTP para pedir la página solicitada.
6. El servidor envía la página como una respuesta HTTP, por ejemplo, enviando el archivo /index.html.

¹ Sistema de nombre de dominio, es el que se encarga de traducir nombres de páginas a direcciones IP.

² Protocolo de Internet, Una dirección IP es el identificador de un recurso en la red.

7. Si la página incluye los localizadores URL³ necesarios para desplegar en pantalla, el navegador obtiene los otros URL mediante el mismo proceso.
8. Se liberan las conexiones TCP⁴ si no hay más solicitudes para los mismos servidores durante un periodo corto.

Lo importante de toda esta infraestructura es que, aunque difieran cualquiera de los componentes anteriormente mencionados incluso evolucionen, la web está preparada para poder soportar esos cambios. Todo ello gracias al protocolo que lo hace posible que es el protocolo HTTP (Protocolo de Transferencia de HiperTexto), el cual veremos en detalle su funcionamiento y evolución

³ Localizador uniforme de recursos.

⁴ Protocolo de control de transmisión.

1.1.1 HTTP su evolución

1.1.1.1 HTTP 0.9

Fue la versión prototipo del protocolo HTTP realizada en el año 1991, solo aceptaba el método GET⁵ y no soportaba contenido multimedia, fue reemplazado por el HTTP 1.0 ya que solamente fue construido para buscar objetos simples en formato HTML⁶.

1.1.1.2 HTTP 1.0

Fue la primera versión estándar del protocolo a través del RFC⁷ 1945 del año 1996, soporta contenido multimedia y los métodos GET, POST⁸ y HEAD⁹, su funcionamiento consistía en la de realizar una solicitud, establecer la conexión TCP y ser liberada tras su respuesta (HTTP no persistente), útil para un sitio completo HTML sin referencias externas.

1.1.1.3 HTTP 1.0+

Versión informal de HTTP 1.0 que contenía algunas mejoras algunas de ellas luego se incorporaron en HTTP 1.1 como soporte de proxy¹⁰, alojamiento virtual y conexiones duraderas conocidas como “*Keep-Alive*”.

⁵ Método HTTP para la lectura de una página Web.

⁶ Lenguaje de marcado de hipertexto, se utiliza para la programación de una página Web.

⁷ Documento en donde se describen y definen protocolos.

⁸ Método HTTP que permite el envío de datos a un servidor HTTP para su posterior procesamiento.

⁹ Método HTTP que permite leer el encabezado de una página Web.

¹⁰ Aplicación que permite compartir una conexión de Internet a diferentes usuarios.

1.1.1.4 HTTP 1.1

Posee correcciones del protocolo HTTP 1.0 mediante los RFC 2068 y 2616, soporte a lenguajes de programación avanzados de los finales de los años 1990 y utilización de conexiones HTTP-persistentes en su modo por defecto por lo que múltiples objetos pueden ser enviados a través de una única conexión TCP.

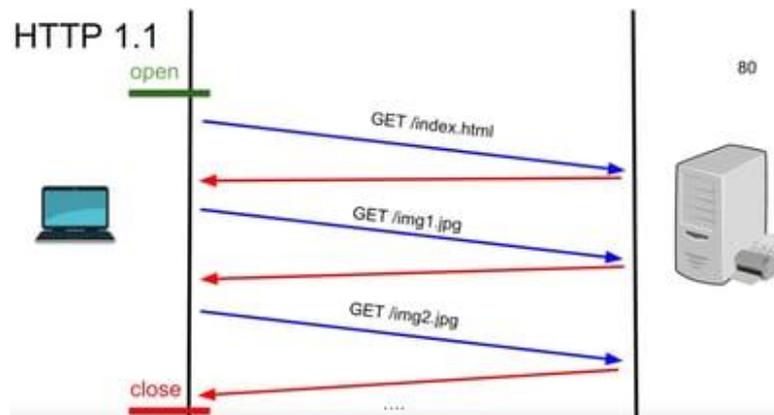


Figura 1-2: Diagrama de conexión HTTP 1.1.
Fuente: (Cisco.com, 2022)

HTTP 1.1 implementa los métodos adicionales de PUT¹¹, DELETE¹², TRACE¹³ CONNECT¹⁴ Y OPTIONS¹⁵.

¹¹ Método HTTP que permite almacenar una página web.

¹² Método HTTP que permite eliminar una página web.

¹³ Método HTTP que permite realizar repeticiones de solicitud entrante en una página web.

¹⁴ Método HTTP que permite conectarse a través de un proxy.

¹⁵ Método HTTP que permite comprobar las opciones que soporta una página web.

1.1.1.5 SPDY

SPDY de sus siglas en ingles Speedy fue desarrollado por la empresa Google para mejorar las conexiones entre un cliente y el servidor hasta un 64 % optimizando las conexiones. Este fue sustituido en el año 2015 con la aparición de HTTP 2

1.1.1.6 HTTP 2

Basado en algunas de las ideas de SPDY para su desarrollo, es un protocolo binario que conserva la compatibilidad con su antecesor HTTP 1.1, incorpora las siguientes características:

- a. Multiplexación: La multiplexación permite enviar y recibir al mismo tiempo optimizando la comunicación, con ella se consigue reducir el número de conexiones.
- b. Server push: Esta técnica permite enviarle información al navegador del cliente, la cual lo guardara en la cache¹⁶ por si posteriormente el cliente lo solicite, el contenido sea entregado de inmediato.
- c. Compresión de HEADERS: En HTTP 1.X por cada solicitud hay información de HEADERS que indican algún comportamiento en el navegador o el servidor. Con HTTP 2.0 todos los HEADERS se empaquetan en un solo bloque comprimido para ser enviados como una unidad, estas se envían y una vez finalizada la transmisión, se decodifican.

1.1.1.7 HTTP 3

El protocolo HTTP/3 define el uso del protocolo QUIC¹⁷ como transporte para HTTP/2. QUIC es un complemento del protocolo UDP que admite la multiplexación de varias conexiones y proporciona métodos de cifrados equivalentes a TLS/SSL.

¹⁶ Memoria utilizada para reducir tiempos de respuestas.

¹⁷ Conexiones rápidas a Internet.

Creado en 2013 por Google como una alternativa a TCP + TLS, resolviendo el problema de la configuración de conexiones prolongadas como el tiempo de negociación en TCP, permitiendo eliminar las demoras como consecuencia de pérdida de paquetes durante la transferencia de datos, HTTP 3 incorpora las siguientes características.

- a. Seguridad similar a TLS brinda la capacidad de usar TLS sobre UDP.
- b. Control de transmisión para evitar la pérdida de paquetes asegurando la integridad.
- c. Capacidad de garantizar demoras mínimas entre el envío de una solicitud y la recepción de una respuesta.
- d. Uso de número de secuencia diferente para retransmitir un paquete, evitando la ambigüedad al determinar los paquetes recibidos y deshacerse de los tiempos de espera.
- e. La pérdida de un paquete afecta solo el flujo asociado y no detiene la entrega de datos en flujos transmitidos en paralelo a través de la conexión actual.
- f. No existen problemas con el bloqueo de cola TCP
- g. Soporte de identificación de conexión para reducir el tiempo de reconexión para clientes móviles
- h. Posibilidad de conectar mecanismos avanzados para control de sobrecargas.
- i. Técnicas de predicción de ancho de banda para garantizar tasas óptimas de envío de paquetes, evitando que se produzcan condiciones de congestión.

1.1.1.8 Mensajes HTTP

Los mensajes HTTP se dividen en dos tipos: mensajes de solicitud y mensajes de respuesta para solicitar una acción a un servidor web. Los mensajes de respuesta transmiten los resultados de una solicitud a un cliente. Tanto los mensajes de solicitud como de respuesta tienen la misma estructura básica de mensajes a continuación se detallarán sus diferencias.

a. Estructura de los mensajes de solicitud

<Método><recurso><versión>

<Encabezados>

<Cuerpo del mensaje>

Un **<Método>** HTTP, normalmente pueden ser un GET o POST, aunque en ocasiones se puede hacer otros tipos de peticiones.

El **<recurso>** pedido, la URL el recurso.

La **<versión>** del protocolo HTTP.

<Encabezados> HTTP opcionales, que pueden aportar información adicional a los servidores, existen diferentes tipos de encabezados

Encabezado	Tipo	Contenidos
User-Agent	Solicitud	Información sobre el navegador y su plataforma.
Accept	Solicitud	El tipo de páginas que puede manejar el cliente.
Accept-Charset	Solicitud	Los conjuntos de caracteres que son aceptables para el cliente.
Accept-Encoding	Solicitud	Las codificaciones de página que puede manejar el cliente.
Accept-Language	Solicitud	Los idiomas naturales que puede manejar el cliente.
If-Modified-Since	Solicitud	Hora y fecha para verificar la actualidad de un mensaje.
If-None-Match	Solicitud	Etiquetas enviadas previamente para verificar la actualidad de un mensaje.
Host	Solicitud	El nombre DNS del servidor.
Authorization	Solicitud	Una lista de las credenciales del cliente.
Referer	Solicitud	El URL anterior desde el cual provino la solicitud.
Cookie	Solicitud	La cookie establecida previamente que se regresa al servidor.
Set-Cookie	Respuesta	La cookie que debe guardar el cliente.
Server	Respuesta	Información sobre el servidor.
Content-Encoding	Respuesta	Cómo se codifica el contenido (por ejemplo, gzip).
Content-Language	Respuesta	El lenguaje natural utilizado en la página.
Content-Length	Respuesta	La longitud de la página en bytes.
Content-Type	Respuesta	El tipo MIME de la página.
Content-Range	Respuesta	Identifica una parte del contenido de la página.
Last-Modified	Respuesta	Hora y fecha de la última modificación de la página.
Expires	Respuesta	Hora y fecha en que la página dejará de ser válida.
Location	Respuesta	Indica al cliente a dónde enviar su solicitud.
Accept-Ranges	Respuesta	Indica que el servidor aceptará solicitudes de rango de bytes.
Date	Ambas	Fecha y hora en que se envió el mensaje.
Range	Ambas	Identifica una parte de una página.
Cache-Control	Ambas	Directivas para manejar las cachés.
ETag	Ambas	Etiqueta para el contenido de la página.
Upgrade	Ambas	El protocolo al que el emisor desea conmutar.

Figura 1-3: Encabezados HTTP.
Fuente: (Tanenbaum, Wetherall, 2012)

<Cuerpo del mensaje> el cual puede ser opcional, con el método POST, se puede enviar información al servidor.

Ejemplo:

```

GET /index.html HTTP/1.1 } Solicitud
Host: google.com         } Encabezados
Accept: text/html        }
    
```

b. Estructura de los mensajes de respuesta

<Versión><código de respuesta><mensaje>

<Encabezados>

<Cuerpo del mensaje>

En esta estructura solamente varían dos campos código de respuesta y mensaje los cuales están asociados, el mensaje detalla de alguna forma la respuesta del código, los cuales se detallan a continuación

<Código de respuesta > estos indican el éxito o no de la solicitud, está compuesto por tres dígitos y están divididos en cinco categorías.

Rango Glogal	Rango definido	Categoría
100-199	100-101	Información
200-299	200-206	Éxito
300-399	300-305	Redirección
400-499	400-415	Error en el cliente
500-599	500-505	Error en el servidor

Figura 1-4: Código de respuesta HTTP.

El código de respuesta más conocidos es el **404** el cual indica que el recurso no fue encontrado.

<Mensaje> está asociada con el código de respuesta.

Ejemplo:

```

HTTP/1.1 200 OK } Respuesta
Server: Apache
Content-Lenght: 6821 } Encabezados
Content-Type: text/html

<html>
data data } Cuerpo del mensaje
</html>

```

1.2 Recursos

Las URL son en un subconjunto de recursos de una clase más general a esa se la denomina URI (identificador de recurso uniforme). Las URI se subdividen en dos subconjuntos principales las URL que son las que nos interesan a nosotros y las URN (nombres uniformes de recursos) estas sirven para identificar, pero no para localizar.

1.2.1 Sintaxis

La RFC 9110 (2022) establece las compatibilidades entre las diferentes versiones y establece su sintaxis común a emplearse entre las diferentes versiones del protocolo HTTP mediante semántica de HTTP que incluye las intenciones definidas por cada método de solicitud, las extensiones, los campos del encabezado, los códigos de estado u otros datos de control y metadatos de recursos que pueden proporcionarse en los campos de respuesta. La semántica también incluye metadatos de representación que describen cómo el destinatario debe interpretar el contenido, solicitar campos de encabezado que podrían influir en la selección de contenido y los diversos algoritmos de selección.

Una URL está compuesta de tres partes:

- a. La primera parte el esquema que indica cual es el protocolo para acceder a ese recurso.
- b. La segunda parte es la ubicación del servidor.
- c. La tercera parte la ruta específica del recurso.

Estas tres partes se pueden representar bajo la siguiente sintaxis

<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>

<scheme> Indica que protocolo se usara para acceder al recurso estos podrán ser http/https/ftp¹⁷.

<user>:<password> Algunas aplicaciones necesitan de autenticación para poder acceder al recurso, esto se logra mediante el ingreso de un usuario y contraseña, un ejemplo de esto es con el protocolo ftp.

ftp://user:password@funesjuan.com.ar

¹⁸ Protocolo de transferencias de archivos.

<host>:<port> Nombre o dirección ip del servidor que posee el recurso más el número de puerto de la aplicación, si se utilizan los puertos por defectos estos pueden ser omitidos.

<http://www.funesjuan.com.ar:80>

<path> Nombre del recurso que se está solicitando

<http://www.funesjuan.com.ar/sitio/index.html>

Para la ubicación de un recurso que se encuentra en subcarpetas se utiliza el carácter “/”.

<params> Parámetros de entrada al recurso

<query> Parámetros de entrada al recurso en especial este es utilizado cuando una aplicación realiza consultas a una base de datos.

<http://www.funesjuan.com.ar/sitio/index.php?id=7>

Para la utilización de las querys se menciona un nombre de un campo con su igualdad “=” a un valor dado, para agregar otro campo solo hay que agregar el carácter “&”.

<frag> Referencia a una porción del recurso local

<http://www.funesjuan.com.ar/sitio/index.html#menu>

Aquí el carácter para definir a esa porción del recurso dentro del mismo servidor se utiliza el carácter “#”.

Ya mencionado la sintaxis de cómo se compone una URL veamos algunos ejemplos de URL para distintos protocolos.

mailto: jefunesc@gmail.com

ftp://funesjuan.com.ar/pub

rstp://funesjuan.com.ar:554

https://www.funesjuan.com.ar

Es importante remarcar que dentro de una URL quizás se componga de algunos caracteres especiales como espacio “ ”, el “~”, o “%”, para poder representar los

mismos ah de colocarse antes el carácter especial “%” el cual permite escapar el carácter ingresado y luego su valor en hexadecimal, vemos algunos ejemplos.

0x7E= ~ <http://www.funesjuan.com.ar/%7Ecarpeta>

0x20= ESPACIO “ “ <http://www.funesjuan.com.ar/carpeta%20publica/>

0x25=”%” <http://www.funesjuan.com.ar/carpeta/100%25satisfecho/>

1.3 Lenguajes para la creación de páginas Web

Dentro de la evolución de la Web existen diferentes formatos de representación el lenguaje inicial fue el HTML, veamos algunos de ellos.

1.3.1 HTML

Como se ha mencionado anteriormente HTML fue el primer lenguaje para generar páginas Web, se dice que cuando se utiliza este tipo de lenguaje la página Web es del tipo estática ya que solo permite mostrar contenido sin modificación de estados o un base a un criterio dado lo que si lo permite las páginas Web del tipo dinámico.

Igualmente, HTML tuvo su evolución en donde fue incorporando distintas utilidades, una de las más significativas fue la incorporación de **CSS** (hojas de estilos en cascada) que permitió modificar la apariencia de la Web. Veamos en la siguiente imagen la evolución de HTML.

Elemento	HTML 1.0	HTML 2.0	HTML 3.0	HTML 4.0	HTML 5.0
Hipervínculos	x	x	x	x	x
Imágenes	x	x	x	x	x
Listas	x	x	x	x	x
Mapas e imágenes activas		x	x	x	x
Formularios		x	x	x	x
Ecuaciones			x	x	x
Barras de herramientas			x	x	x
Tablas			x	x	x
Características de accesibilidad				x	x
Incrustación de objetos				x	x
Hojas de estilo				x	x
Secuencias de comandos				x	x
Vídeo y audio					x
Gráficos vectoriales en línea					x
Representación de XML					x
Hilos en segundo plano					x
Almacenamiento del navegador					x
Lienzo de dibujo					x

Figura 1-5: Evolución HTML
Fuente: (Tanenbaum, Wetherall, 2012)

1.3.2 XML

XML (Lenguaje de marcado extensible), se creó para especificar contenido de manera estructurada, fue desarrollado por el W3C y es mucho más fácil de analizar. XML es por lo general como lenguaje para la comunicación entre programas de manera que se pueda realizar una transacción entre un cliente y servidor con lenguajes totalmente diferentes de programación.

La sintaxis de XML requiere los elementos estén siempre anidados, se cierran las etiquetas, los atributos deben estar siempre entre comillas y que es sensible a mayúsculas y minúsculas. Veamos un ejemplo

```
<?xml version= "1.0" encoding="ISO-8859-1"?>
<nota date="28/09/17">
<para>Juan</para>
<de>Paola</de>
</nota>
```

1.3.3 LENGUAJES PARA LA GENERACION DE CONTENIDO INTERACTIVO EN EL CLIENTE

JAVASCRIPT fue el primer lenguaje de programación de secuencia de comandos que se ejecutó en el cliente, muy útil para crear páginas web interactivas. Su sintaxis es similar al lenguaje de programación C.

Una alternativa de generación de página web del lado del cliente en la plataforma Microsoft Windows es con **VBSCRIPT** que se basa en el lenguaje de programación Visual Basic.

Otro método de generación es mediante la utilización de **applets**, estos son pequeños programas de **JAVA** compilados en instrucciones máquina para una computadora virtual conocida como **JVM** (máquina virtual JAVA), estos **applets** son incrustados en el HTML.

Del surgimiento de **JVM** Microsoft lanzo los controles ActiveX, que son programas compilados para lenguaje maquina x86 y ejecutados sobre el hardware.

De la utilización de javascript surgieron **AJAX** (javascript asíncrono) que no es un lenguaje de programación si no un grupo de tecnologías (HTML, javascript, XML Y DOM¹⁸) que juntas hacen más poderosas las aplicaciones.

Así mismo también surgió **JSON** (notación de objetos javascript) como lenguaje ligero de intercambio de datos en reemplazo o alternativa de XML.

Otra de las evoluciones fue **YAML** diseñado para ser fácilmente legible y escribible por humanos. Utiliza una sintaxis clara y simple que se basa en la indentación y el uso de espacios en blanco. YAML admite una amplia variedad de tipos de datos, incluidos números, cadenas, listas, diccionarios y tipos de datos compuestos más complejos.

1.3.4 GENERACION DE PAGINAS WEB DINAMICAS CON CGI

CGI (interface de puerta de enlace común) provee a los servidores web una interface que permite que se comuniquen programas de soportes y secuencia de comandos que por ejemplo como resultado de la búsqueda en un formulario se genere una página web con contenido HTML dinámico, se ejecuta la entrada del lado del servidor ejecutando el CGI, este interpreta y genera el contenido. El lenguaje de programación para ejecutar CGI puede ser **C** que producen archivos ejecutables o **Python, Ruby, Perl** que produce archivos interpretados.

1.3.5 GENERACION DE PAGINAS WEB DINAMICAS CON PHP

PHP (preprocesador de hipertexto), es otra alternativa para generar páginas web dinámicas del lado del servidor y se caracteriza por poseer código incrustado dentro de HTML, diseñado para interactuar con base de datos, su sintaxis es similar al lenguaje de programación C.

1.3.6 ALTERNATIVAS DE GENERACION DE PAGINAS WEB DINAMICAS

Por el momento hemos mencionado dos formas de generación de páginas web dinámicas, una es a través de CGI y la otra a través de PHP incrustado. Existen otras alternativas como el **JSP** (Java Server Pages), que es similar al PHP, pero utiliza el lenguaje de programación JAVA para la generación de las mismas. Otra alternativa es **ASP.NET** (Active Server Page.NET) que se considera como la versión de PHP y JSP para Microsoft, utiliza programas en el marco de trabajo (framework) de aplicaciones en red .NET para la generación de las páginas web dinámicas.

1.4 Interpretación de contenidos - MIME

MIME (Extensiones multipropósito para de correo de Internet) fue diseñado para resolver problemas él envió de contenidos de los correos electrónicos, tal fue su éxito que fue incorporado en el protocolo HTTP para la web.

Los servidores web para poder desplegar una página web deben conocer su formato, para que todos los navegadores web puedan interpretar la misma. Existen diferentes tipos de objetos en la web como, documentos, imágenes, audio, video etc. Todos ellos deben poder ser interpretados, **MIME**¹⁸ es un estándar bajo las RFC (2045 a 2049 y 2077), y propone los siguientes tipos y subtipos de contenidos para su interpretación.

Tipo	Subtipos de ejemplo	Descripción
text	plain, html, xml, css	Texto en diversos formatos.
image	gif, jpeg, tiff	Imágenes.
audio	basic, mpeg, mp4	Sonidos.
video	mpeg, mp4, quicktime	Películas.
model	vrmf	Modelo 3D.
application	octet-stream, pdf, javascript, zip	Datos producidos por aplicaciones.
message	http, rfc822	Mensaje encapsulado.
multipart	mixed, alternative, parallel, digest	Combinación de múltiples tipos.

Figura 1-6: Tipos y subtipos MIME.
Fuente: (Tanenbaum, Wetherall, 2012)

En el siguiente link <http://www.iana.org/assignments/media-types/media-types.xhtml> se puede ver todos los tipos MIME registrados en IANA¹⁹

¹⁸ Modelo de objeto de documentos, se utiliza para modificar parte de la página mientras se despliega en pantalla.

¹⁹ Entidad que supervisa la asignación global de direcciones IP y otros recursos relacionados a los protocolos de Internet

Capítulo 2. Metodologías y herramientas de filtrado.

2.1 METODOLOGIAS.

Las metodologías y herramientas de filtrado que se investigaran son aquellas que tengan relación con el destino final y siempre a través de servidores PROXY¹, no se investigara ninguna metodología de control de acceso relacionada con el cliente en cuanto a su dirección IP o DIRECCION MAC²: tampoco la restricción de acceso entre rangos horarios.

2.1.1 Listas Negras.

Método de bloqueo del acceso a determinados sitios o dominios. Existen diferentes listas algunas comerciales y otras libres sobre un conjunto de sitios, las cuales son agrupadas por diferentes tipos de categorías. Un administrador puede autorizar o denegar el acceso a esos sitios según la categoría. Las categorías más comunes son las siguientes.

- Evasión de proxies
- Hacking
- Pornografía
- Radio y tv
- Telefonía IP
- Phishing

¹ Servidor que intercepta conexiones de red hechas desde un cliente a un servidor de destino, cuando un equipo de la red desea acceder a una página web, es realmente el proxy quien realiza la comunicación y a continuación traslada el resultado al cliente que la solicitó.

² Identificador de 48 bits que se corresponde de forma única con una interfaz de red.

- Juegos
- Alcohol
- Armas
- Apuestas
- Material adulto
- Noticias

Existen otras categorías, aquí solamente se han mencionado las más destacadas. En el siguiente link se pueden ver las categorías que utiliza la empresa Cisco https://www.cisco.com/c/en/us/products/collateral/security/web-security-appliance/datasheet_C78-718442.html

2.1.2 Listas Blancas

Método utilizado para permitir el acceso a determinados sitios o dominios. Todas las referencias de la lista blancas deben ser permitidas sobre cualquier tipo de restricción. Comúnmente es utilizada para solamente permitir el acceso a internet a esos sitios y nada más que a ellos, no teniendo acceso a otras páginas web que no esté en esa lista.

2.1.3 Filtrado Web por extensiones de archivos

Método utilizado para bloquear el acceso a descarga de determinados tipos de archivos, los archivos que generalmente son bloqueados son los ejecutables como, por ejemplo .exe, .bat, .com, bin etc. Existen algunos productos que también utilizan antivirus para la protección de descarga de archivos, esta utilidad sobrecarga el producto y el tiempo de descarga aumenta.

2.1.4 Filtrado Web por MIME

Método similar al bloqueo por tipos de archivos, solamente que este lo realiza según el tipo y subtipo de documentos según el estándar MIME, en 1.4 Figura 1.6 se pueden ver algunos los mismos ya mencionados anteriormente.

2.1.5 Filtrado Web por palabras y frases en sitio web

Método que mediante la utilización de expresiones regulares permite bloquear sitios en base a palabras, frases, o secuencias de caracteres presentes en alguna parte de la solicitud http realizada por el cliente, es muy común utilizar esto con algunas frases o palabras que pueden ser consideradas como obscenas.

2.1.6 Filtrado Web por headers

Este método permite el bloqueo de encabezados HTTP tanto de solicitudes como de respuestas. En 1.1.1.8 Figura 1.3: se mencionaron los diferentes tipos de encabezados existentes en HTTP, los cuales se pueden aceptar o bloquear según se especifique.

2.2 HERRAMIENTAS DE FILTRADO.

Las herramientas de filtrado a investigar serán dos de las alternativas de código abierto más populares y robustas en base al proxy cache SQUID³ y alternativas de servidores firewall con proxy comerciales que no solo se utilizan para ello sino como un conjunto de herramientas de protección para diferentes fines y otra que es muy utilizada en servidores proxy y router para lograr el bloqueo de sitios.

³ Servidor proxy Web con cache.



2.2.1 SQUID CON SQUIDGUARD

Es un producto que trabaja en combinación con SQUID. Squidguard es un redirector URL y utiliza el método de listas negras para filtrar el acceso a los sitios web.

Squidguard fue desarrollado en el año 2009 y su versión estable se dio a conocer en el año 2010. El sitio <http://www.squidguard.org/> tuvo soporte hasta unos pocos años y en la actualidad en las distribuciones Linux se encuentra disponible el paquete para su instalación es muy utilizado en algunas distribuciones Linux o FreeBSD con funciones de Firewall y Router como por ejemplo PfSense.

Las listas negras deben ser descargadas para su utilización las mismas fueron cambiando con el tiempo, en la actualidad se pueden descargar del sitio <https://dsi.ut-capitole.fr/blacklists/download/>

Squid es un software que funciona como servidor proxy, de código abierto y de alto rendimiento utilizado para mejorar el rendimiento de los sitios web al cachear las páginas web frecuentemente solicitadas por los usuarios. Funciona interceptando todas las solicitudes web y reenviándolas de manera eficiente, permitiendo un acceso más rápido a los recursos y reduciendo la carga en el servidor. Squid también proporciona funciones de control de acceso, autenticación de usuarios y registro de actividades.

2.2.2 OpenDNS



OpenDNS fue creado para resolver nombres de dominios en el año 2005, inicialmente a individuos y empresas de manera gratuita como alternativa más rápida que los DNS de los ISP. La función principal de OpenDNS es actuar como un servicio de resolución de nombres de dominio (DNS).

OpenDNS fue incorporando distintas utilidades como correctores ortográficos para la resolución de dominios, protección contra phishing⁴, bloqueo de sitios clasificados como maliciosos y diferentes estadísticas y reportes de uso todo pudiendo ser configurado mediante un panel de control por usuario.

OpenDNS no solo es utilizado en los usuarios si no en distintos dispositivos como Firewall o Routers, por lo que en 2012 llego a manejar el 3% del total de todos los usuarios de Internet.

En el año 2015 OpenDNS fue adquirido por CISCO, actualmente continúa siendo gratuito para la resolución de nombres de dominios y bloqueo básico para personas y los hogares, teniendo costos y productos adicionales para protegerse contra propagación de diferentes tipos de amenazas y malware⁵.

2.2.3 FORTIGATE – FORTINET



Fortigate es un Firewall, producto comercial de la empresa Fortinet el cual puede funcionar como proxy transparente o como proxy explicito dentro de una red o como simplemente un Gateway. Inicialmente Fortigate se enfocó a la gestión unificada de amenazas con funciones de firewall, prevención de intrusiones, filtrado Web y protección frente a malware.

Hoy en día los productos fortigate tienen muchas otras prestaciones y se los denominan NGFW (nueva generación de firewall), estas por ejemplo soportan VPN⁶ y otras utilidades más sofisticadas.

Fortigate es considerado líder en el sector de NGFW según la consultora de tecnología más importante denominada Gartner, la cual dio su último reporte en noviembre de 2021. Fortigate tiene distintos modelos los cuales varía dependiendo la cantidad de usuarios que se conectaran.

Fortigate utiliza las metodologías ya vistas para el filtrado web, pero tiene su propia base de datos para el método de listas negras denominada Fortiguard, la cual es consultada constantemente antes de tomar una decisión. Estas solamente están disponibles durante el periodo que el equipo este con licencia, la cual debe renovarse como mínimo anualmente.

⁴ Método de engaño para obtener información.

⁵ Software con código malicioso.

⁶ Red privada virtual, a través de ella se puede acceder remotamente desde una computadora de manera segura a la red de una organización.

2.2.4 INTERSCAN Web Security - TREND MICRO



InterSan Web Security es un proxy, producto comercial de la empresa Trend Micro, InterSan Web Security además del filtrado web soporta la protección de escaneo tráfico FTP, brindando protección a esos protocolos. A diferencia de productos similares como InterScan VirusWall que trabaja de manera similar, pero a su vez brinda protección a correo electrónico y otros protocolos de red. InterScan Web Security esta solo dedicado a la protección de tráfico Web y el mismo puede ser utilizado como como proxy transparente o como proxy explicito dentro de una red o como simplemente un Gateway.

Trend Micro es considerado líder en el sector de End Point según la consultora de tecnología más importante denominada Gartner, la cual dio su último reporte en Enero de 2017.

InterSan Web Security también utiliza las metodologías ya vistas para el filtrado web, y tiene su propia base de datos para el método de listas negras las cuales son descargadas, estas solamente estarán disponibles durante el periodo que el equipo este con licencia, la cual debe renovarse como mínimo anualmente.

Capítulo 3. Configuración y comprobación de las diferentes propuestas.

3.1 ESCENARIO DE COMPROBACION

Para la comprobación de funcionamiento de las diferentes herramientas se reunió información de navegación de los usuarios de la red de un organismo de la administración pública (Ejército Argentino) durante un mes, generando un lote de prueba de 16499 sitios web. Para la comprobación se instalarán en ambientes virtuales y físicos a fin de poder evaluar su funcionamiento y rendimiento de filtrado de cada herramienta. Por cada sitio a comprobar solo se comprobará los sitios con acceso mediante HTTP y no con la versión segura HTTPS a fin de evitar problemas de verificación del certificado e inspección de contenido mediante inspección SSL dando conocimiento a los usuarios sobre dicha inspección. Las metodologías a utilizar solo permitirían filtrar la URL o la dirección de dominio específica para HTTPS ya que no se podrá verificar el contenido sin inspección SSL como también el acceso a sitios web mediante el protocolo DoH¹ definido en el RFC 8484.

A diferencia del DNS tradicional, DoH es una extensión del protocolo DNS que utiliza HTTPS para encriptar y proteger las consultas y respuestas de DNS durante su transmisión a través de Internet. Al utilizar HTTPS, DoH proporciona una capa adicional de seguridad al cifrar las consultas y respuestas de DNS, lo que dificulta su interceptación y asegura la integridad de los datos transmitidos, razón por la cual no la utilizaremos para la comprobación.

Así mismo nos ocurre con la inspección SSL que se requiere que un dispositivo, como firewall, proxy o un sistema de seguridad, pueda interceptar el tráfico cifrado y descifrarlo para su análisis. Se deben utilizar certificados SSL personalizados en el dispositivo de inspección SSL para establecer una conexión SSL/TLS entre el cliente y el dispositivo de inspección, y otra conexión SSL/TLS entre el dispositivo de inspección y el servidor final. Esto implica generar certificados SSL que sean reconocidos por los clientes y confiables para que no generen advertencias de seguridad en los navegadores.

¹ Consultas DNS sobre el protocolo HTTPS.

Una vez que el tráfico SSL/TLS ha sido interceptado y descifrado, se puede realizar un análisis en profundidad del contenido del tráfico. Esto puede incluir inspección de contenido y filtrado de URL entre otros.

La inspección SSL afecta a la privacidad ya que implica la capacidad de ver el contenido cifrado de las comunicaciones. También puede haber riesgos de seguridad si los certificados SSL no se manejan adecuadamente, ya que podrían ser aprovechados por atacantes para realizar ataques.

La inspección SSL se podría realizar con los siguientes escenarios:

- Squid con Squidguard ya que es un proxy que intercepta el tráfico.
- FORTIGATE – FORTINET ya que es un firewall de nueva generación que intercepta y analiza el tráfico.
- INTERSCAN Web Security - TREND MICRO ya que es un proxy que intercepta el tráfico.

La inspección SSL no se puede realizar con OpenDNS este ofrece servicios de seguridad y filtrado de contenido a nivel de DNS. Sin embargo, OpenDNS no proporciona funcionalidades directas de inspección SSL como las que pueden ofrecer los servidores proxy como Squid o firewall como Fortigate.

Como se mencionó anteriormente a fin de agilizar las comprobaciones solo se comprobará los sitios con acceso mediante HTTP y no con la versión segura HTTPS a fin de evitar problemas de verificación del certificado.

Por cada una de las comprobaciones se mostrarán las imágenes y políticas utilizadas para el filtrado Web. En todos los casos solamente se buscar filtrar las categorías que abarquen los siguientes tipos de páginas web.

- Pornografía.
- Phishing
- Sitio Web Maliciosos
- SPAM

Cabe destacar que en el lote de prueba existen páginas de diferentes categorías las cuales no se evaluara en el presente trabajo de investigación.

Para comprobar el funcionamiento de filtrado se creara un script en bash denominado “*comprobar.sh*”, donde será modificado para cada herramienta de filtrado a fin de adaptarse al funcionamiento de la misma.

El archivo con los sitios a comprobar se denomina sitios.txt con un contenido similar a la siguiente figura:

```
pics.xxxmaturevideos.com
pictures.hentai-foundry.com
pictures.share-image.com
picture-us.hismarttv.com
pic.vartuc.com
pic.wonporn.net
pic.wwwxxx.mobi
pic.xchica.com
piedrascalientes.com
piet2eix3l.com
pigdragon.cookappsgames.com
pileton.com.ar
p.im9.eu
pimpandhost.com
pinaynudeself.xyz
pinetworth.com
ping3.teamviewer.com
ping.chartbeat.net
ping.confirmid.name
pingping.fantasti.cc
pinkertube.com
pinkfineart.com
pinlust.com
pinmule.com
pinmybabe.com
pinmyhentai.com
pinofsex.com
pintarycoloreardibujos.net
pintolandiagay.com.br
pinuderest.com
pipcie.pl
pipeschannels.com
pipe.skype.com
pipoffers.apnpartners.com
piporace.com
piratepass.pw
pirluttravel.com
piroposparaunaamiga.com
piscis.ru
pisodesign.com.br
pisosrusticossaltillo.com.mx
pittylanumerologa.com.ar
pivigames.blog
piwik.mobirum.com
pix.btrll.com
pixel-a.basis.net
pixel.adacado.com
pixel.adcrowd.com
pixel.adsafeprotected.com
pixel.advertising.com
pixel-a.sitescout.com
pixel.cdnwidget.com
```

In 7319 Col 2 368.608 caracteres.

Figura 3-1: Archivo sitios.txt

El resultado a explotar será similar a la siguiente imagen en donde según el código de error se comprobará el resultado del filtro realizado.

```

www.psnwzksttygfs.com, 503
www.ptqagsagtb.bid, 503
www.publisex.cl, 403
www.puerto80it.com, 301
www.puertojardin.com.ar, 200
www.puntanasvip.com, 200
www-punterlink-co-uk.dualstackcdn.com, 404
www.puppypark.club, 200
www.pureadexchange.com, 403
www.purejapanese.com, 403
www.pure-ts.com, 403
www.puricelliabogados.com.ar, 200
www.puritanas.com, 403
www.pussybook.xyz, 403

```

Figura 3-2: Archivo resultado.txt.

3.2 SQUID CON SQUIDGUARD

Para la comprobación del Squid con SquidGuard se utilizará un servidor Linux Ubuntu Server con Squid3 y los paquetes necesarios para lograr el filtrado web en apoyo con SquidGuard, para ello se deberán bajar listas negras actualizadas.

```
wget https://ds1.ut-capitole.fr/blacklists/download/blacklists.tar.gz
```

Luego se deberá configurar el Squid para que pueda redirigir las peticiones a SquidGuard, editando el archivo squid.conf

```
url_rewrite_program /usr/bin/squidGuard
```

Ahora vamos configurar SquidGuard utilizando las categorías de sitios ya establecidos como se muestra a continuación editando el archivo SquidGuard.conf.

```

src localnet {
    ip 10.0.0.0/8
}
dest hacking {
    domainlist hacking/domains
    urlist hacking/urls
}
dest porn {
    domainlist porn/domains
    urlist porn/urls
}
acl {
    localnet {
        pass hacking !porn !in-addr all
        redirect http://10.10.8.109/prohibido.html
    }
    default {
        pass none
    }
}

```

Para comprobar su funcionamiento del servidor proxy, se ha configurado en el equipo cliente la variable **export http_proxy:http://10.10.21.99:3127** a fin de obligar a que dicho dispositivo utilice el servidor proxy para realizar las peticiones a internet, luego sobre el mismo se ha utilizado el siguiente código en el script comprobar.sh

```
#!/bin/bash

while read sitio
do

    estado=$(curl -I -s $sitio -m 10 | grep "HTTP/1.1" | awk ' (print $2} ')

    echo $sitio, $estado >> resultado.txt

    echo $sitio, $estado

    sleep 2

done < sitios.txt
```

Si quisiéramos utilizar el squid en modo transparente a fin de no estar indicando en los navegadores el puerto de conexión al squid el mismo debería realizarse modificándose el archivo squid.conf la línea **http_port 3127 transparent** y utilizar iptables para enrutar todo el tráfico de la red con destino al puerto 80 al puerto 3137, como se menciona a continuación.

```
iptables -A PREROUTING -s 10.0.0.0/8 -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 3127
```

Otra de las alternativas es utilizar squid con el protocolo WCCP, la integración de Squid con WCCP se debe configurar el router local para interceptar el tráfico web saliente de la red y redirigirlo hacia el servidor Squid. Esto se logra mediante la configuración de políticas de red y la activación del servicio WCCP en el router.

```
access-list 102 remark ACL for WCCP proxy access
access-list 102 remark Squid proxies bypass WCCP
access-list 102 remark LAN clients proxy port 80 only
access-list 102 permit tcp 10.0.0.0 255.0.0.0 any eq 80
access-list 102 remark all others bypass WCCP
access-list 102 deny ip any any
!
! Assign ACL to WCCP
ip wccp web-cache redirect-list 102
```

**! Now set WCCP version 2:
ip wccp version 2**

El router y el servidor Squid establecen una conexión utilizando el protocolo WCCP. Esta comunicación permite al router enviar al servidor Squid el tráfico web que necesita caché, así como también recibir información sobre el estado del servidor Squid, el el proxy debe ir la siguiente configuración.

```
http_port 3127 transparent  
wccp2_router 10.0.0.254  
wccp2_forwarding_method 1  
wccp2_return_method 1  
wccp2_assignment_method hash  
wccp2_service standard 0
```

Para una explotación de los resultados, se analizará los resultados independientemente al igual que el resto de las herramientas como se ha mencionado en el ejemplo de la “Figura 3.2: Archivo resultado”

3.3 OPENDNS

Para la comprobación de OpenDns se ha solicitado una licencia denominada "Home Free" para tal fin, procediendo a su registro y configuración. Para el correcto funcionamiento de OpenDns, se debe configurar en los equipos que se conecten a Internet, como servidor DNS, los que indique la herramienta OpenDNS.

Usar OpenDNS como reemplazo de un servidor de resolución de nombres de dominio (DNS) local tiene varias ventajas.

- Velocidad y disponibilidad: OpenDNS tiene servidores distribuidos globalmente que pueden mejorar la velocidad de resolución de nombres de dominio en comparación con un servidor DNS local.
- Filtrado de contenido y seguridad: OpenDNS ofrece funcionalidades de filtrado de contenido y seguridad adicionales que pueden no estar disponibles en un servidor DNS local estándar.
- Facilidad de configuración y administración: Configurar y administrar OpenDNS generalmente es más fácil que mantener un servidor DNS local, especialmente si no se tiene la experiencia técnica
- Registro y análisis: OpenDNS ofrece herramientas de registro y análisis que pueden proporcionar información detallada sobre el tráfico de red y el uso de Internet, lo que permite monitorear la actividad de los usuarios y detectar posibles amenazas o problemas de seguridad.

A continuación, algunas pantallas necesarias para su configuración y funcionamiento.

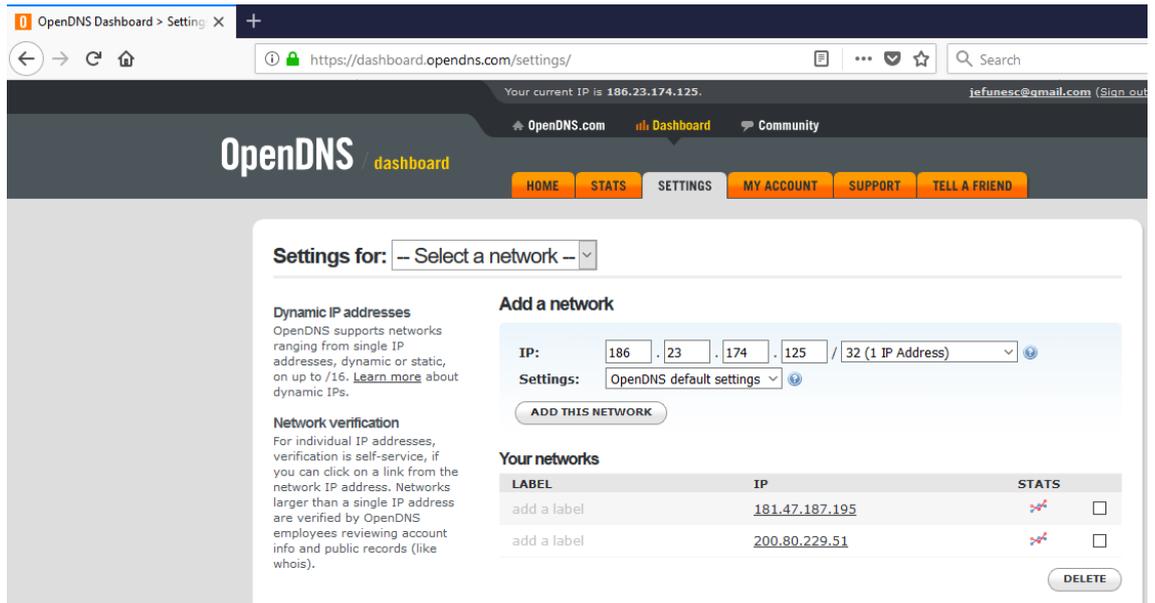


Figura 3-3: Configuración de redes en OpenDNS.

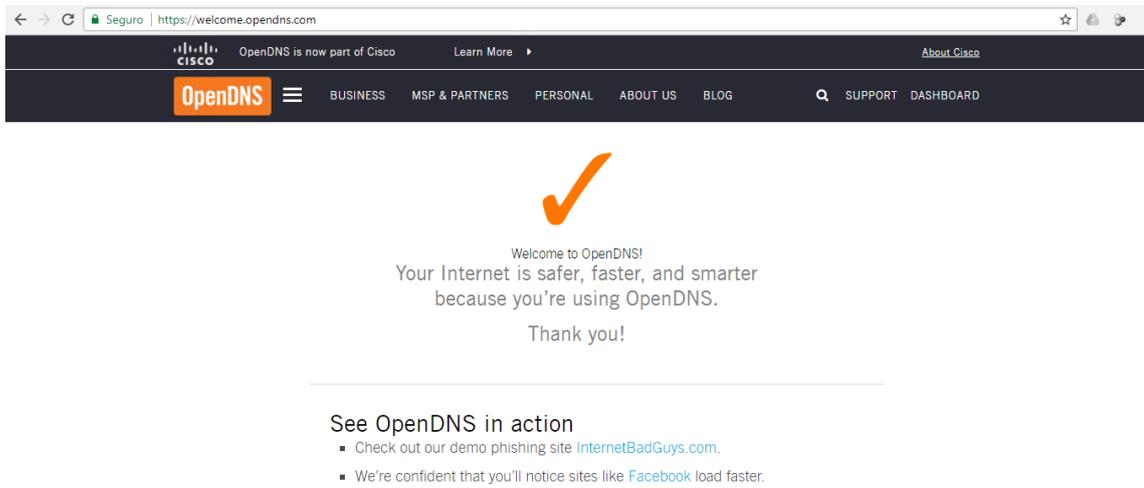
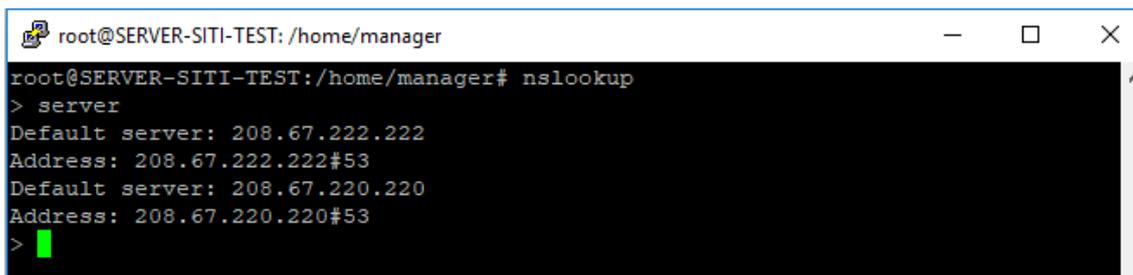
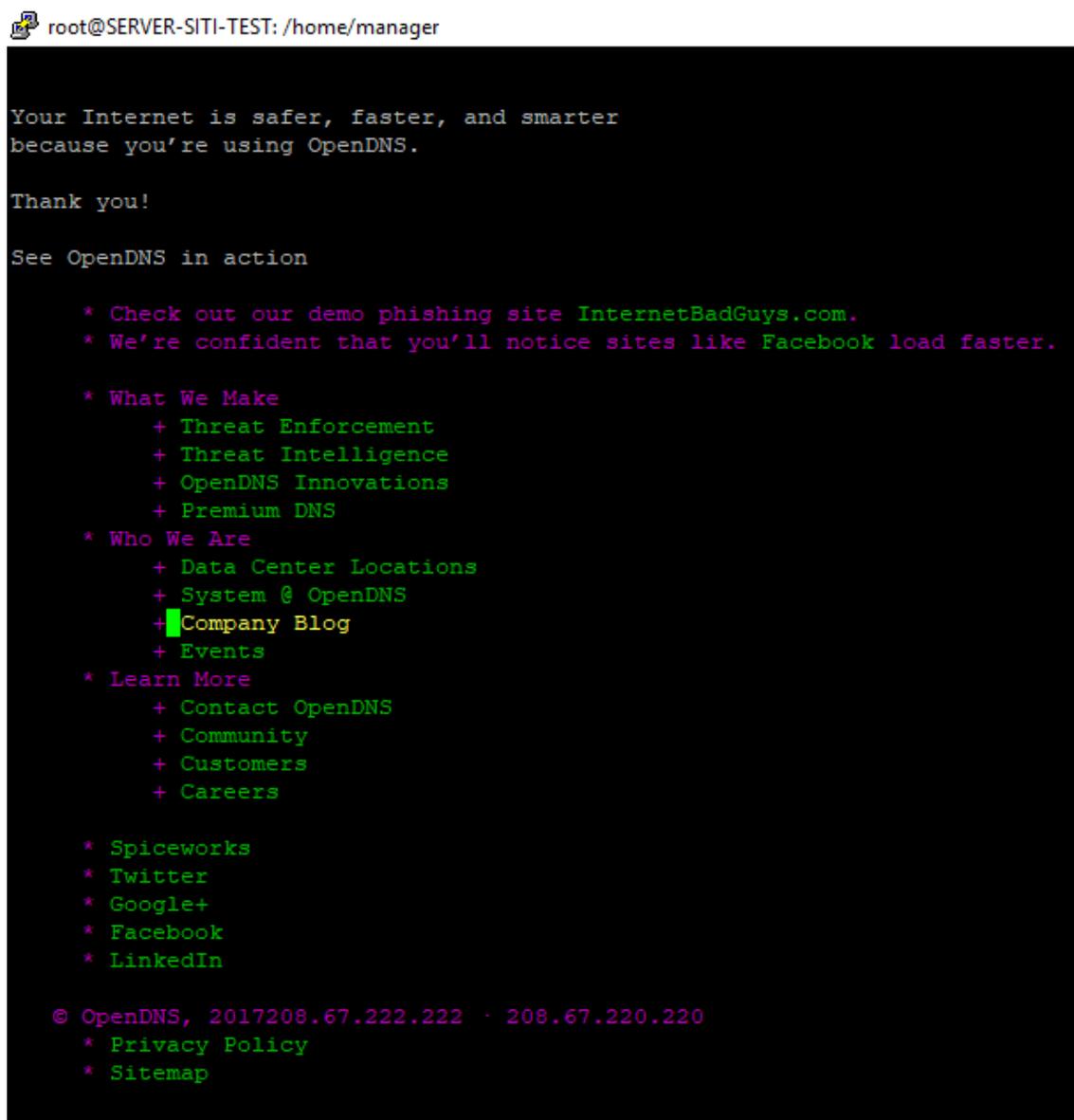


Figura 3-4: Comprobación de funcionamiento OpenDNS en Windows



```
root@SERVER-SITI-TEST: /home/manager
root@SERVER-SITI-TEST:/home/manager# nslookup
> server
Default server: 208.67.222.222
Address: 208.67.222.222#53
Default server: 208.67.220.220
Address: 208.67.220.220#53
>
```

Figura 3-5: Comprobación DNS en Linux.



```
root@SERVER-SITI-TEST: /home/manager

Your Internet is safer, faster, and smarter
because you're using OpenDNS.

Thank you!

See OpenDNS in action

* Check out our demo phishing site InternetBadGuys.com.
* We're confident that you'll notice sites like Facebook load faster.

* What We Make
  + Threat Enforcement
  + Threat Intelligence
  + OpenDNS Innovations
  + Premium DNS

* Who We Are
  + Data Center Locations
  + System @ OpenDNS
  + Company Blog
  + Events

* Learn More
  + Contact OpenDNS
  + Community
  + Customers
  + Careers

* Spiceworks
* Twitter
* Google+
* Facebook
* LinkedIn

© OpenDNS, 2017208.67.222.222 - 208.67.220.220
* Privacy Policy
* Sitemap
```

Figura 3-6: Comprobación de funcionamiento de navegación en Linux.

EL filtrado Web se realizará sobre las categorías ya establecidas como se muestra en las siguientes imágenes.

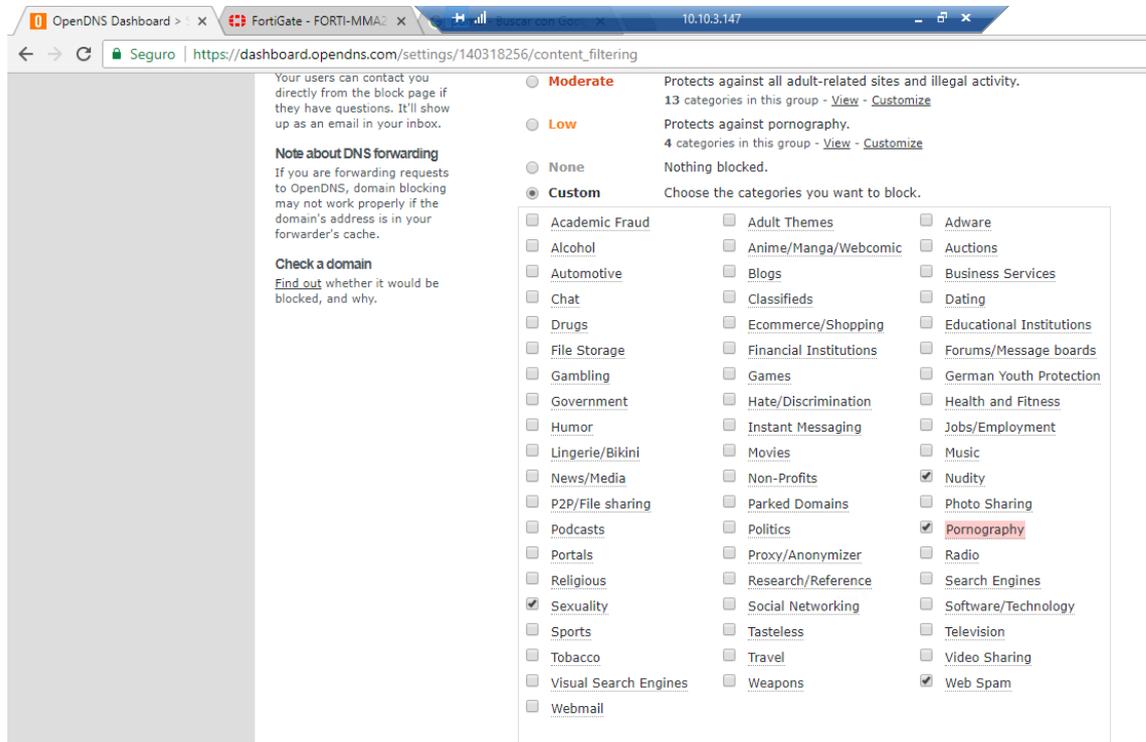


Figura 3-7: Configuración de filtros web en OpenDNS.

OpenDNS a diferencia de otras herramientas tiene un módulo adicional para brindar protección sobre redes Botnet y Phishing, a continuación, imagen de su configuración.

Settings for: Add/manage networks

Security

- [Web Content Filtering](#)
- Security**
- [Customization](#)
- [Stats and Logs](#)
- [Advanced Settings](#)

Malware/Botnet Protection **Enable basic malware/botnet protection**
 When certain Internet-scale botnets are discovered or particularly malicious malware hits, we offer protection to all our users so that as many people as possible can be protected from the threat. At this time, this feature blocks the Conficker virus and the Internet Explorer Zero Day Exploit, and is continually expanded to include other types of malicious sites.

Phishing Protection **Enable phishing protection**
 By enabling phishing protection, you'll protect everyone on your network from known phishing sites using the best data available.

Suspicious Responses **Block internal IP addresses**
 When enabled, DNS responses containing IP addresses listed in [RFC1918](#) will be filtered out. This helps to prevent [DNS Rebinding attacks](#). For example, if badstuff.attacker.com points to 192.168.1.1, this option would filter out that response.

The three blocks of IP addresses filtered in responses are:

```
10.0.0.0 - 10.255.255.255 (10/8)
172.16.0.0 - 172.31.255.255 (172.16/12)
192.168.0.0 - 192.168.255.255 (192.168/16)
```

Apply to all my networks

Figura 3-8: Configuración seguridad en OpenDNS.

Para comprobar su funcionamiento se ha utilizado el siguiente código en el script comprobar.sh.

```
#!/bin/bash
while read sitio
do
    estado=$(curl -s -L $sitio -m 10 | grep block.opendns
    | awk 'BEGIN { FS="/" } (print $4)')
    echo $sitio, $estado >> resultado.txt
    echo $sitio, $estado
    sleep 2
done < sitios.txt
```

3.4 FORTIGATE – FORTINET

Para la comprobación del Fortigate se utilizará el modelo FortiGate-1500D con las licencias necesarias para el filtrado web y bajo la configuración de proxy explícito como se muestra a continuación especialmente indicando que la política por defecto es de manera restrictiva denegando todo el tráfico y habilitar solo lo que se necesita.

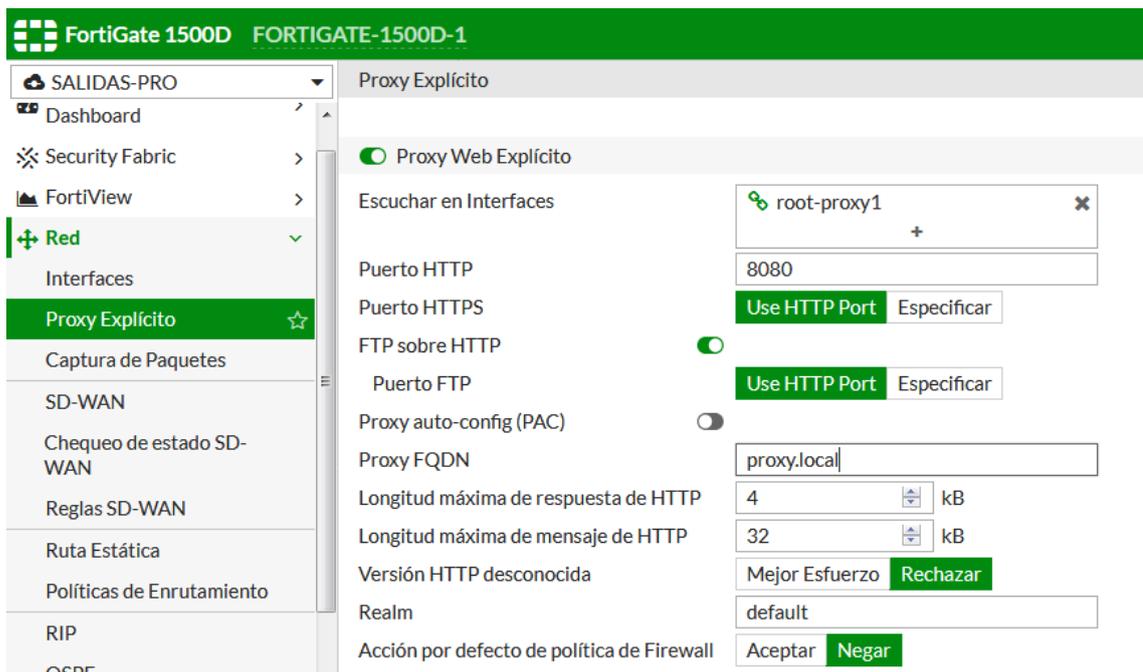


Figura 3-9: Configuración proxy explícito en Fortigate.

Sobre el mismo solamente se comprobará las categorías de sitios ya establecidos y como se muestra en la siguiente imagen su configuración.

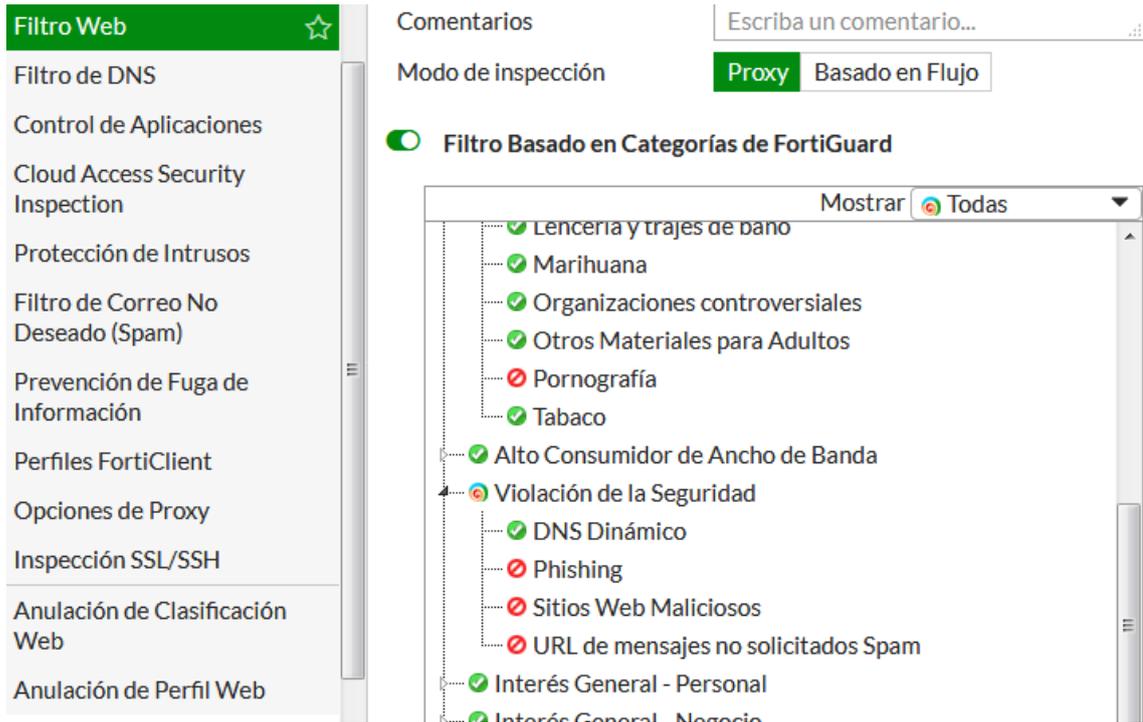


Figura 3-10: Configuración de filtros web en Fortigate.

Luego se habilitarán las políticas necesarias para poder brindar internet a los usuarios aplicando en la misma la política de filtrado web que se ha establecido.



Figura 3-11: Política de aplicación de acceso y filtrado en Fortigate.

Para comprobar su funcionamiento se ha utilizado el script comprobar.sh, que se utilizó también en la herramienta SquidGuard.

3.5 INTERSCAN WEB SECURITY - TREND MICRO

Para la comprobación de InterScan Web Security se descargó una ISO denominada IWSVA-6.5-SP2-1548-x86_64.iso del sitio http://downloadcenter.trendmicro.com/index.php?regs=NABU&clk=latest&clkval=4599&lang_loc=1 y se ha solicitado una licencia de prueba para tal fin.

Product License 

To receive your Activation Code, enter your Registration Key at the [Trend Micro Product Registration Server](#).

 **[IWSVA with DLP] Your trial period will end in 20 days.** [View license upgrade instructions](#)

License Information		View detailed license online
Product:	InterScan Web Security Virtual Appliance	
Version:	Trial with DLP	
Activation code:	IH-ZXLZ-8JYRL-5K8YP-ALJHY-J3KSP-2UB2L	Enter a new code
Seats:	001000	
Status:	Activated	

Figura 3-12: Licencia Trend Micro InterScan Web Security.

InterScan Web Security posee una política por predeterminada permitiendo y negando algunas categorías de sitios, esta debe ser ajustada a las necesidades particulares de cada implementación. Sobre el mismo solamente se comprobará las categorías de sitios ya establecidos como proxy explícito.

Proxy Settings



Please specify the relevant proxy settings.

HTTP Listening Port	
Port number:	<input type="text" value="8080"/>
Forward Proxy Mode	
<input type="checkbox"/>	Enable upstream proxy (dependent mode)
Proxy server:	<input type="text"/>
Port number:	<input type="text" value="8080"/>
Guest User Login	
<input type="checkbox"/>	Enable guest user login
Port number:	<input type="text" value="8081"/>
Anonymous FTP over HTTP	
Email address:	<input type="text" value="anonymous@iwss.trendmicro.com"/>

Figura 3-13: Proxy explicito Trend Micro InterScan Web Security.

EL filtrado Web se realizará sobre las categorías ya establecidas como se muestra en la siguiente imagen su configuración.

Disease Vector	<input type="checkbox"/>	✓	Allow
Hacking	<input type="checkbox"/>	✗	Block
Joke Program	<input type="checkbox"/>	✓	Allow
Made for AdSense	<input type="checkbox"/>	✓	Allow
Malware Accomplice	<input type="checkbox"/>	✓	Allow
New Domain	<input type="checkbox"/>	✓	Allow
Password Cracking	<input type="checkbox"/>	✗	Block
Potentially Malicious Software	<input type="checkbox"/>	✓	Allow
Proxy Avoidance	<input type="checkbox"/>	✓	Allow
Remote Access Program	<input type="checkbox"/>	✓	Allow
Spam	<input type="checkbox"/>	✗	Block
Spyware	<input type="checkbox"/>	✓	Allow
Web Advertisement	<input type="checkbox"/>	✓	Allow
Scam	<input type="checkbox"/>	✓	Allow
Ransomware	<input type="checkbox"/>	✗	Block
+ Communications and Search	<input type="checkbox"/>	Allow	Action
- Adult	<input type="checkbox"/>	Allow	Action
Abortion	<input type="checkbox"/>	✓	Allow
Adult/Mature Content	<input type="checkbox"/>	✓	Allow
Alcohol/Tobacco	<input type="checkbox"/>	✓	Allow
Gambling	<input type="checkbox"/>	✓	Allow
Illegal Drugs	<input type="checkbox"/>	✓	Allow
Illegal/Questionable	<input type="checkbox"/>	✓	Allow
Intimate Apparel/Swimsuit	<input type="checkbox"/>	✓	Allow
Marijuana	<input type="checkbox"/>	✓	Allow
Nudity	<input type="checkbox"/>	✓	Allow
Pornography	<input type="checkbox"/>	✗	Block

Figura 3-14: Filtros Trend Micro InterScan Web Security.

Luego se habilitarán las políticas necesarias para poder brindar internet a los usuarios creando una política de filtrado web.

URL Filtering Policy: Add Policy ?

[Policy List](#) > (New Policy) Enable policy

1. Select Accounts
2. Specify Rules
3. Specify Safe Search Engines
4. Specify Exception Lists

Create new policy:*

Copy from existing policy:* URL Filtering Global Policy ▼

IP range:

From:

To:

IP address:

IP Subset:

Address:

Prefix Length:

Type	Identification	
IP RANGE	10.10.1.0 ~ 10.10.1.254	

Figura 3-15: Creación de política Trend Micro InterScan Web Security.

URL Filtering Policies Enable URL filtering Enable Dynamic URL Categorization ?

✔ Your changes have been saved, but not deployed. By default, deployment will occur at the time specified on the Administration > IWSVA Configuration > Policy Deployment screen. Before deployment, you can make further changes to the policy. ✖

	Account	Policy Name	Priority	Status
<input type="checkbox"/>	10.10.1.0 ~ 10.10.1.254	proxy-exlicito	1	<input checked="" type="checkbox"/>
<input type="checkbox"/>	(All accounts)	URL Filtering Global Policy	2	<input type="checkbox"/>

Figura 3-16: Vista de política creadas - Trend Micro InterScan Web Security.

Para comprobar su funcionamiento se ha utilizado el script comprobar.sh, que se utilizó también en la herramienta SquidGuard y Fortinet.

Capítulo 4 - CONCLUSIONES

4.1 Conclusiones

Del trabajo realizado se ha obtenido diferentes resultados en un archivo denominado resultado.txt, se ha consolidado el mismo por cada herramienta, todo en una hoja de cálculo en la cual se han contabilizado los diferentes códigos HTTP devueltos en el archivo resultado.txt, generado por el script comprobar.sh

conclusiones-filtrado							
	A	B	C	D	E	F	G
1	SQUIDGUARD	PERMITIDOS	4262		FORTINET	PERMITIDOS	2840
2		PROHIBIDOS	1096			PROHIBIDOS	6132
3		REMOVIDOS	3922			REMOVIDOS	1384
4		REMOVIDOS	1346			REMOVIDOS	1217
5		INALCANZABLE	3067			INALCANZABLE	2840
6		CODIGOS HTTP IDENTIFICADOS	13693			CODIGOS HTTP IDENTIFICADOS	14413
7		OTROS CODIGOS HTTP	2806			OTROS CODIGOS HTTP	2086
8		TOTAL	16499			TOTAL	16499
9							
10							
11	TREND	PERMITIDOS	3909		OPENDNS	PERMITIDOS	4209
12		PROHIBIDOS	4883			PROHIBIDOS	4103
13		REMOVIDOS	1413			REMOVIDOS	1877
14		REMOVIDOS	1365			REMOVIDOS	870
15		INALCANZABLE	3267			INALCANZABLE	3004
16		CODIGOS HTTP IDENTIFICADOS	14837			CODIGOS HTTP IDENTIFICADOS	14063
17		OTROS CODIGOS HTTP	1662			OTROS CODIGOS HTTP	2436
18		TOTAL	16499			TOTAL	16499
19							
20		SQUIDGUARD	TREND	FORTINET	OPENDNS		
21	TOTAL SITIOS	16499	16499	16499	16499		
22	PROHIBIDOS	1096	4883	6132	4103		

Figura 4-1: Explotación de resultados del archivo resultado.txt.

Como se puede ver se separó los códigos 200 como permitido, los códigos 403 como prohibidos, salvo para el caso del filtrado de Opendns en el cual el error es identificado con un valor en el encabezado HTTP **“block.opendns.com”**, los códigos de removidos son los 301 y 302, el código 404 como sitio inalcanzable, como otros los demás código HTTP.

Con la explotación de los datos solamente se tomó como totales los códigos 403 y mensajes de OpenDNS **“block.opendns.com”**, sobre el total de la muestra de sitios que fueron 16499. Sobre esa explotación se realizó la siguiente grafica que demuestra las conclusiones del presente trabajo de investigación.

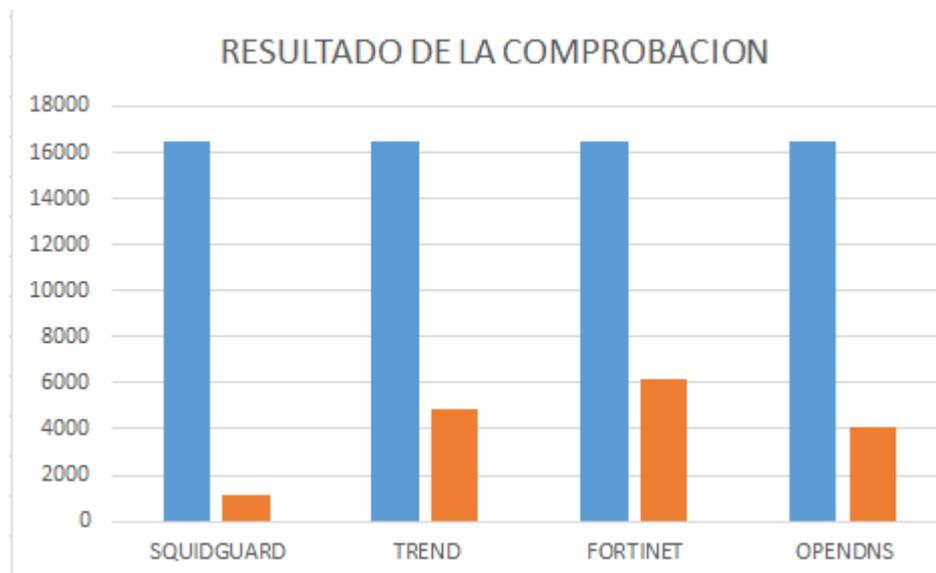


Figura 4-2: Conclusiones en base a los resultados.

Con el presente trabajo se ha querido manifestar que existen diferentes herramientas para el filtrado web, todas ellas trabajan de manera similar, teniendo una lista de los diferentes dominios y/o url a ser denegadas. Es muy notorio que existe un gran esfuerzo por cada herramienta para tener actualizada estas listas, existe una gran diferencia entre las listas públicas que utiliza SquidGuard, con la de Trend y SquidGuard que son pagas, no es así para el caso de OpenDNS versión gratuita, con el resto de las herramientas pagas.

Así mismo resulta dificultoso poder encontrar una herramienta que se ajuste a las necesidades de una organización, ya que internet es un mundo y cada día crece más y más.

De las herramientas estudiadas en el presente trabajo de investigación la que considero más eficiente en su accionar son las herramientas pagas, pero para el caso de que una organización no cuente con recursos económicos para tal fin, OpenDNS Free recomiendo que es la más adecuada para brindar la solución de filtrado Web.

A continuación, se muestran imágenes con el resultado de filtrado de las diferentes herramientas utilizadas.

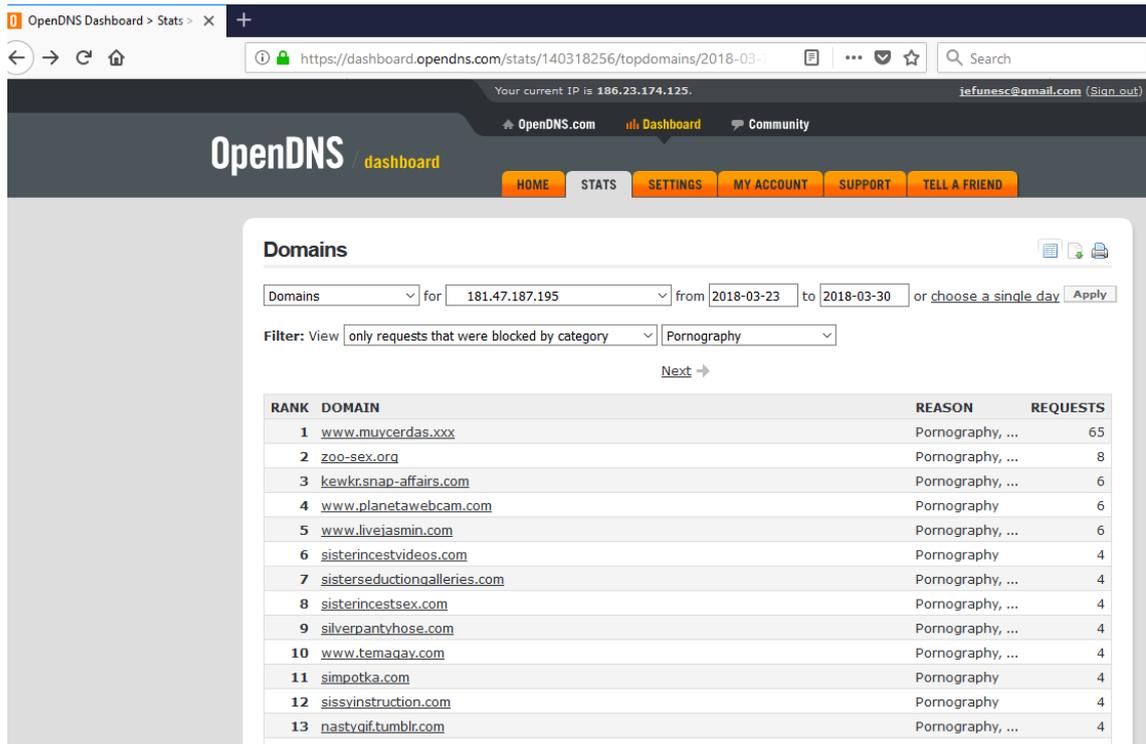


Figura 4-3: Ejemplo resultado filtrado con OpenDNS.

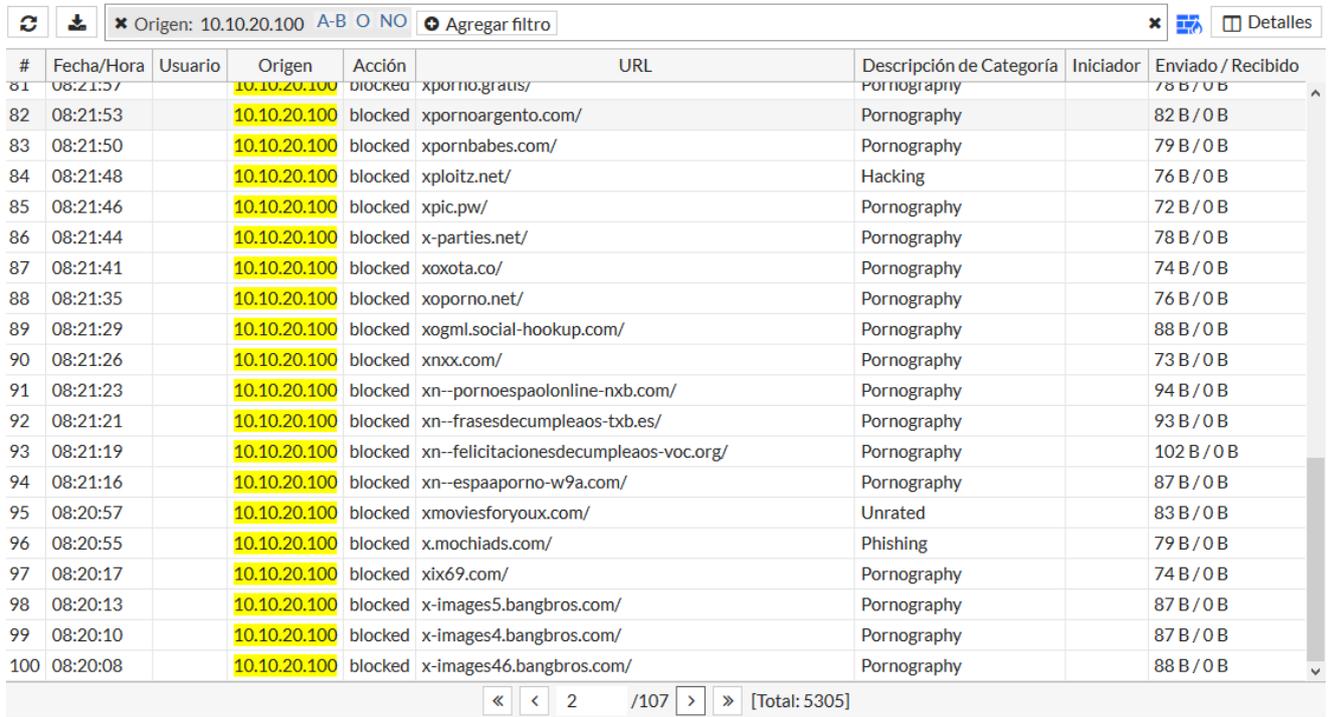


Figura 4-4: Ejemplo resultado filtrado con Fortigate.

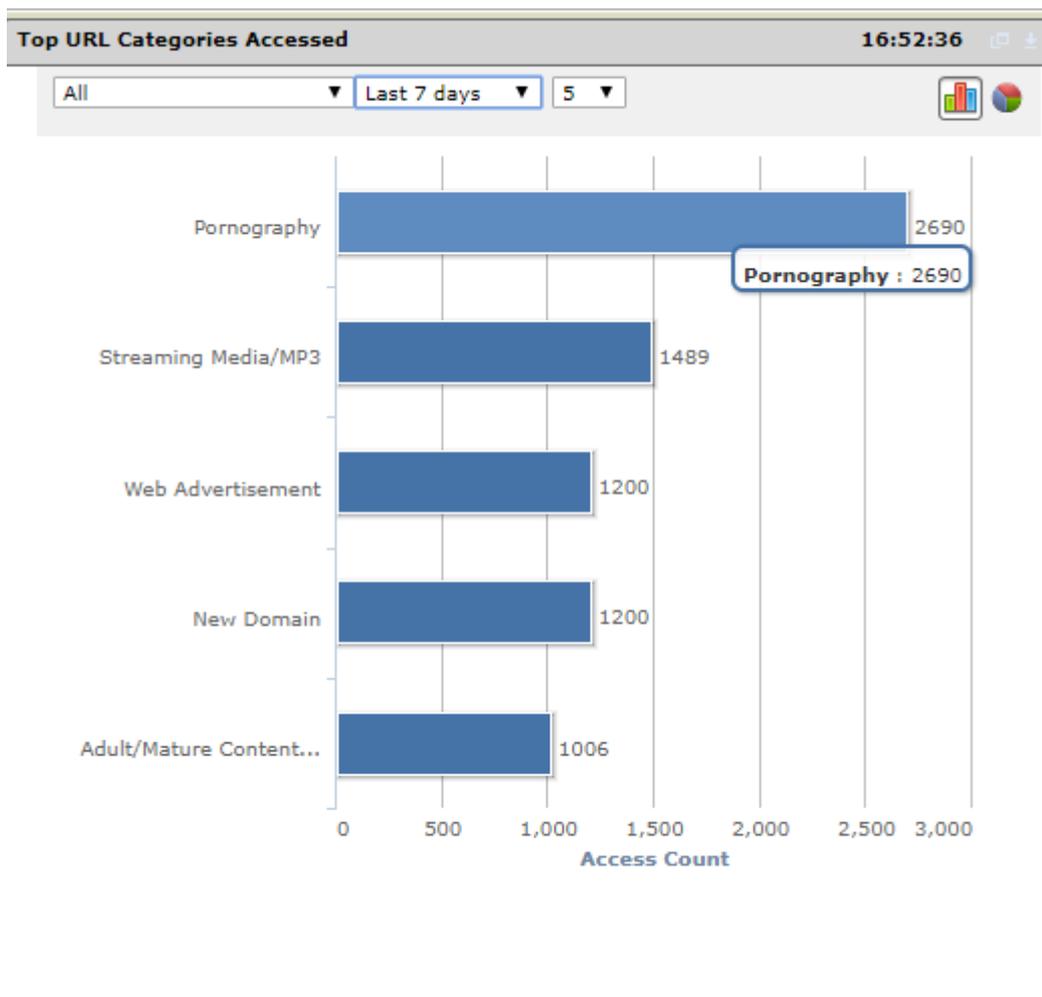


Figura 4-5: Ejemplo resultado filtrado con Trend Micro InterScan Web Security.

4.2 Futuras líneas de investigación

Herramienta de filtrado web sobre un dns, el cual pueda ser retroalimentado de diferentes fuentes, para la administración pública.

Capítulo 5 - BIBLIOGRAFIA

HTTP Semantics RFC 9110 - IETF Datatracker (2022). Recuperado de <https://datatracker.ietf.org/doc/rfc9110/>

HTTP Semantics RFC 9114 - IETF Datatracker (2022). Recuperado de <https://datatracker.ietf.org/doc/html/rfc9114>

DNS Queries over HTTPS (DoH)- (2018). Recuperado de <https://www.rfc-editor.org/rfc/rfc8484.html>

Hypertext Transfer Protocol Version 2 (HTTP/2) - IETF Datatracker (2015). Recuperado de <https://datatracker.ietf.org/doc/rfc7540/>

Andrew S. Tanenbaum, Marteen V. Steen, (2008). Sistemas distribuidos principios y paradigmas. (2da ed). Person - Prentice Hall.

James F. Kurose, Keiht W. Ross, (2010). Redes de computadoras un enfoque descendente. (5ta ed). Pearson

Andrew S. Tanenbaum, David J. Wetherall, (2012). Redes de computadoras. (5ta ed). Pearson

Silvano Da Ros, (2006). Content Networking Fundamentals. (1 ed). Cisco Press

InfoLeg. (2014) .InfoLeg - Ministerio de Economía y Finanzas Públicas. Recuperado de <http://www.infoleg.gov.ar/infolegInternet/anexos/235000-239999/239771/norma.htm>

Buenos Aires ciudad. (2016). Seguridad en Internet | Buenos Aires ciudad - Gobierno de la Ciudad Autónoma de Buenos Aires. Recuperado de <http://www.buenosaires.gob.ar/educacion/escuelas/seguridad-en-internet>

W3C (2004). Architecture of the World Wide Web, Volume One. Recuperado de <https://www.w3.org/TR/2004/REC-webarch-20041215/>

Squid. (2017). Página oficial <http://www.squid-cache.org/>

The Chromium Projects. (2011). SPDY: An experimental protocol for a faster web. Recuperado de <http://www.chromium.org/spdy/spdy-whitepaper>

SomosTechies. (2016). Qué es HTTP/2 y qué ventajas tiene sobre HTTP 1.1. Recuperado de <https://sometechies.com/que-es-http2/#.WcVcsTW1vIU>

Mozilla. (2017). Generalidades del protocolo HTTP. Recuperado de <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>

W3school. (2017). THE WORLD'S LARGEST WEB DEVELOPER SITE. Recuperado de <https://www.w3schools.com/>

Zentyal (2017). Servicio de Proxy HTTP. Recuperado de https://wiki.zentyal.org/wiki/Es/5.0/Servicio_de_Proxy_HTTP

Ubuntu documentación (2017). Dansguardian Internet Content Filtering. Recuperado de <https://help.ubuntu.com/community/DansGuardian>

Fortinet (2017). The Fortinet cookbook - Transparent Web Proxy. Recuperado de <http://cookbook.fortinet.com/transparent-web-proxy-56/>

Online Help Center – Trend Micro (2017). InterScan Web Security. Recuperado de <http://docs.trendmicro.com/en-us/enterprise/interscan-viruswall.aspx>

David Gourley, Brian Totty, Marjorie Sayer, Anshu Aggarwal, Sailu Reddy, (2002). HTTP: The Definitive Guide. O'Reilly Media