RESUMEN – TRABAJO DE GRADO

AUTORES	HERNANDÉZ SIERRA OSCAR EDWIN MARÍN BENAVIDES JONATHAN CONRADO
FACULTAD	FACULTAD DE INGENIERIAS
PLAN DE ESTUDIOS	INGENIERIA DE SISTEMAS
DIRECTOR	INGENIERO BARRIENTOS AVENDAÑO EDWIN
TÍTULO DE LA TESIS	DESARROLLO DE UN PROTIPO DE FIREWALL, QUE PERMITA DETECTAR ATAQUES DE INJECTION (SQL, JAVASCRIPT, CSS) DENTRO DE UNA RED IPV6 CONTROLADO DENTRO UN SERVIDOR PROXY MULTIPLATAFORMA

RESUMEN
(70 palabras aproximadamente)

LAS TECNOLOGÍAS DE COMPUTACIÓN SURGIERON DENTRO DE LAS COMUNIDADES ACADÉMICAS Y DE INVESTIGACIÓN PARA SATISFACER LAS NECESIDADES DE CONEXIÓN Y COLABORACIÓN EN ESTOS SECTORES, PERO POCO A POCO HAN PASADO A FORMAR PARTE DEL CONGLOMERADO DE TECNOLOGÍAS DE LAS EMPRESAS. EN LA ACTUALIDAD EXISTE UNA GRAN DEPENDENCIA TANTO EN LAS EMPRESAS COMO EN LAS COMUNIDADES DE USUARIOS DE LOS SISTEMAS. DE ESTA FORMA, HAN APARECIDO AMENAZAS PRINCIPALMENTE ORIENTADAS A EXPLOTAR VULNERABILIDADES EN LOS COMPONENTES DE LAS APLICACIONES DE LA WEB, ENTRE OTRAS. ESTAS AMENAZAS AFECTAN PRINCIPALMENTE A LA CAPA DE APLICACIÓN YA QUE ES CONSIDERADA POR LOS ATACANTES COMO UN PUNTO CLAVE Y SUSCEPTIBLE A PROBLEMAS DE SEGURIDAD.

CARACTERÍSTICAS

PÁGINAS:100	PLANOS:	ILUSTRACIONES:	CD-ROM: 1

DESARROLLO DE UN PROTIPO DE FIREWALL, QUE PERMITA DETECTAR ATAQUES DE INJECTION (SQL, JAVASCRIPT, CSS) DENTRO DE UNA RED IPV6 CONTROLADO DENTRO UN SERVIDOR PROXY MULTIPLATAFORMA

AUTORES

HERNANDÉZ SIERRA OSCAR EDWIN

MARÍN BENAVIDES JONATHAN CONRADO

DIRECTOR

INGENIERO BARRIENTOS AVENDAÑO EDWIN

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS

Ocaña, Colombia Julio de 2016.

Índice

Pág.

Listado de tablas

Pág.

Listado de figuras

Pág.

Resumen

Las tecnologías de computación surgieron dentro de las comunidades académicas y de investigación para satisfacer las necesidades de conexión y colaboración en estos sectores, pero poco a poco han pasado a formar parte del conglomerado de tecnologías de las empresas. En la actualidad existe una gran dependencia tanto en las empresas como en las comunidades de usuarios de los sistemas. De esta forma, han aparecido amenazas principalmente orientadas a explotar vulnerabilidades en los componentes de las aplicaciones de la WEB, entre otras. Estas amenazas afectan principalmente a la capa de aplicación ya que es considerada por los atacantes como un punto clave y susceptible a problemas de seguridad. Las amenazas que han ganado relevancia en los últimos tiempos por la frecuencia de los ataques y su impacto negativo son los ataques de inyección SQL y los ataques de denegación de servicio basados en XML, en entornos de servicios Web. Ambos ataques se caracterizan por su amplia variedad de técnicas de ataques, además de poner en riesgo tanto la confidencialidad e integridad de los datos y las aplicaciones, pero sobre todo la disponibilidad. Las medidas de seguridad existentes se centran primordialmente en garantizar la confiabilidad e integridad de los datos, sin prestar atención a la disponibilidad.

Es por ello que se hace necesario proponer nuevas soluciones de seguridad que hagan frente a este tipo de amenazas. En este trabajo se presenta un modelo de prototipo, diseñado para detectar intrusiones en REDES IPV6. La detección de ataques de inyección SQL, XML y

JAVASCRIPT requiere un enfoque que pueda adaptarse a los constantes cambios en las técnicas de ataque y este tipo de agente resulta adecuado para ser aplicado en entornos altamente dinámicos. Para ello en el primer capítulo se elaborara un Diagnóstico actual de los firewalls orientados a detectar ataques de injection (SQL, JavaScript, CSS) bajos entornos IPV6, en el segundo capítulo se definirán los requerimientos funcionales y no funcionales del firewall de acuerdo a la estructura del protocolo IPV6, en el capítulo 3 se diseñaran las interfaces de acuerdo a los requerimientos funcionales y no funcionales planteados para el prototipo de firewall y finalmente re realizaran las diferentes codificaciones y pruebas del firewall bajo un ambiente controlado IPV6.

Introducción

La globalización de las comunicaciones y la necesidad de controlar de manera automática la mayor cantidad de procesos industriales de las organizaciones de una manera centralizada, han conducido a las empresas a converger en la red TCP/IP una gran variedad de servicios de voz, video y datos sobre la misma infraestructura.

Así mismo las tecnologías y software desarrollados en el mundo es un producto de la inteligencia y conocimiento humano, y como producto de este no están exentas de errores. Estos errores de las tecnologías y software conocidos comúnmente como vulnerabilidades. Las vulnerabilidades informáticas pueden ser aprovechadas por intrusos con la intención de obtener informaciones de un sistema o adueñarse de él violando normas de seguridad. No obstante el objetivo de los administradores de sistemas y usuarios es reconocer las vulnerabilidades que van apareciendo con el fin de señalar los habituales ataques que contienen estas informaciones. Esto podría estandarizar esta información y brindar la posible solución a los problemas de seguridad que estas representan. Un tipo de ataques a vulnerabilidades son las inyecciones SQL, las cuales consisten en la inserción o "inyección" de una consulta SQL a través de los datos de entrada que posee las aplicaciones o mediante la URL. En efecto el empleo de técnicas de minería de datos podría mitigar muchos ataques de inyecciones SQL entre otros. Por lo anterior, se desarrolla este proyecto de grado que pretende desarrollar un prototipo de firewall, que permita detectar ataques de injection (SQL, JavaScript, CSS) dentro de una red ipv6.

Capítulo 1. Titulo

Desarrollo de un prototipo de firewall, que permita detectar ataques de injection (SQL, JavaScript, CSS) dentro de una red IPv6 controlado dentro un servidor proxy multiplataforma.

1.1 Planteamiento del problema

La seguridad es en la actualidad, un tema de especial relevancia para los servicios de una red de comunicaciones que permiten el desarrollo de las actividades, sean estas personales o de carácter empresarial.

Los ataques a los sistemas de información, cada vez son más frecuentes, con el fin de comprometer, suprimir o modificar información de gran valor para los usuarios de dichos sistemas; esta es la razón por la que los sistemas de defensa están en constante evolución para enfrentar nuevos retos en términos de seguridad.

En un equipo de cómputo, es de vital importancia contar con software antivirus para la protección de la información frente a programas dañinos que desean acceder a través de la red a la que está conectado.

El camino de IPv4 a IPv6 no es una cuestión de migración, sino de transición, evolución, de integración, pero se trata de evolución disruptora, rompedora y al mismo tiempo

necesaria. Actualmente IPv6 lo componen evoluciones en desarrollo de firewall que pretenden mejorar sus especificaciones y sobre todo en los ataques de injection (SQL, JavaScript, CSS) hay pocos controles.

No obstante el diseño del prototipo de firewall estará enfocado a leer las incidencias de ataques que se le hacen al servidor en un entorno de red IPv6 ayudaría a asegurar el futuro, no hipotecarlo, frente al inevitable comercio electrónico, móvil, web, etc.
Los servidores generan logs de accesos muy detallados que contienen información muy importante sobre las visitas a nuestros sistemas de información, siendo muy útil analizarlos cuidadosamente, para esto realizaremos un analizador de logs, aplicación que analiza el archivo Access.log que genera el servidor. El analizador genera un informe detallado sobre los ataques de injection (SQL, JavaScript, CSS) que le hallan echo al sistema de información alojado en el servidor.

1.2 Formulación del problema

¿Con la construcción de un prototipo de firewall orientado a detectar ataques de injection (SQL, JavaScript, CSS) se podría contribuir al fortalecimiento de la seguridad en redes que implementan el protocolo IPv6?

1.3 Objetivos de la investigación

1.3.1 Objetivo General

Desarrollar un prototipo de firewall, que permita detectar ataques de injection (SQL, JavaScript, CSS) dentro de una red IPv6 controlado bajo un servidor proxy multiplataforma.

1.3.2 Objetivos Específicos

- Desarrollar un estado del arte actual de firewall, que permitan detectar ataques de injection (SQL, JavaScript, CSS) bajos entornos IPv6.
- Diseñar la interfaz de acuerdo a los requerimientos funcionales y no funcionales planteados para el prototipo de firewall.
- Generar conexiones seguras mediante un servidor proxy multiplataforma que permita el control y el anonimato de peticiones realizadas por el usuario al intentar conectarse a la web.
- Analizas las incidencias realizadas al servidor mediante la lectura del archivo ACCESS.log.

1.4 Justificación

El estudio de las vulnerabilidades informáticas, permite definir políticas que gestionan la operatividad del acceso restringido a los recursos de la red para los usuarios que interactúan con cada uno de los servicios que provee la infraestructura de comunicación que tienen a su disposición.

De acuerdo a este análisis, se pueden desarrollar prototipos que controlen y administren las intromisiones que intenten perjudicar la integridad, la confiabilidad y disponibilidad de la información que es generada en la red de una determinada empresa.

La construcción de un prototipo firewall enfocado a detectar cuando un atacante trata de realizar una injection (SQL, JavaScript, CSS) específicamente sobre redes IPv6, persigue una finalidad práctica, puesto que su desarrollo se beneficiaría notablemente el control de la seguridad de los sistemas de información dentro del entorno IPv6 permitiendo reducir el riesgo de un ingreso malintencionado de terceros al a red con el propósito de comprometer las características primarias de la información.

Se realizarán pruebas de ataque al sistema manejador de bases de datos y a las páginas fuentes bajo un ambiente controlado IPv6 con el fin de identificar el correcto funcionamiento del firewall.

Teniendo en cuenta que el firewall en su prototipado inicial, solo se enfocará a realizar reporte de incidencias ocurridas en el servidor, dando como análisis las diferentes intrusiones que se han realizado sobre el mismo.

Para administrar de manera eficaz un servidor, es necesario tener registros de la actividad y el rendimiento del servidor así como el registro sobre todas las peticiones que procesa. Por supuesto, almacenar información en el registro de acceso es solamente el principio en la gestión de los registros. El siguiente paso es analizar información que contiene para producir estadísticas que resulten de utilidad. (Apache.org, 2013)

1.5 Delimitaciones

1.5.1 Delimitación Operativa

Esta propuesta se limita a proponer un prototipo que sirva como refuerzo estructural contra los ataques realizados a un servidor proxy manejando así una interfaz sencilla de acuerdo a los requerimientos planteados para el desarrollo de este. Es así que podrá ser aplicado en plataforma operativa de la distribución de Windows 7. Del mismo modo se utilizara también POSTGRESQL como servidor para el almacenamiento de datos, el cual también proveerá la información para los reportes requeridos dando como resultado la eficacia y veracidad de la información, enfocando al firewall inicialmente solo al reporte de incidencias de intrusiones tipo injection (SQL, JAVASCRPT,CSS) que se hayan realizado sobre el servidor logrando así, que en futuros proyectos se dé continuidad al proceso de filtrado de tráfico para permitir o denegar el flujo de información hacia el servidor.

1.5.2 Delimitación Conceptual

Se desarrollara dentro del ámbito de infraestructura de TI, por lo que se deben tener presente los diferentes conceptos como sistema operativo, firewall, IPv4, IPv6, injection SQL, injection JavaScript, injection CSS, servicios, servidores, protocolos.

1.5.3 Delimitación Geográfica

El desarrollo de este proyecto de grado tendrá lugar en la universidad Francisco de Paula Santander Ocaña, en el municipio de Ocaña, departamento Norte de Santander, país Colombia.

1.5.4 Delimitación Temporal

El proyecto se desarrollara con una duración de máximo 6 meses a partir del momento en que sea aprobado el anteproyecto de grado.

Capítulo 2. Marco referencial

2.1 Marco Histórico

Las herramientas analíticas nacieron a mediados de los años 90 y se empezaron a utilizar de forma masiva a partir de 1995 por parte de los departamentos de informática (ahora llamados IT) originalmente la finalidad era conocer la carga de trabajo de los servidores ya que toda su actividad se registraba en un archivo llamado log.

Las primeras herramientas analíticas era software que interrogaba estos archivos y generaba una serie de informes en entornos gráficos. El log estaba formado por 18 campos de datos, desde la dirección IP hasta los detalles de accesos a los archivos. (Google central de conversiones, 2009)

El internet es uno de los avances más significativos de la historia, por la facilidad que este permite de estar siempre en comunicación sin importar en lugar donde se encuentre, tan solo con tener un punto de internet puedes estar informado e informar de lo que pasa en el mundo.

Los inicios de Internet nos remontan a los años 60. En plena guerra fría, Estados Unidos crea una red exclusivamente militar, con el objetivo de que, en el hipotético caso de un ataque ruso, se pudiera tener acceso a la información militar desde cualquier punto del país.

Esta red se creó en 1969 y se llamó *ARPANET*. En principio, la red contaba con 4

ordenadores distribuidos entre distintas universidades del país. Dos años después, ya contaba con unos 40 ordenadores conectados. Tanto fue el crecimiento de la red que su sistema de comunicación se quedó obsoleto. Entonces dos investigadores crearon el Protocolo *TCP/IP*, que se convirtió en el estándar de comunicaciones dentro de las redes informáticas *(actualmente seguimos utilizando dicho protocolo).*

ARPANET siguió creciendo y abriéndose al mundo, y cualquier persona con fines académicos o de investigación podía tener acceso a la red.
Las funciones militares se desligaron de ARPANET y fueron a parar a MILNET, una nueva red creada por los Estados Unidos.
La NSF *(National Science Fundation)* crea su propia red informática llamada *NSFNET*, que más tarde absorbe a *ARPANET*, creando así una gran red con propósitos científicos y académicos.

El desarrollo de las redes fue abismal, y se crean nuevas redes de libre acceso que más tarde se unen a *NSFNET*, formando el embrión de lo que hoy conocemos como *INTERNET* (Computación Aplicada al desarrollo SA de CV, s.f.)

A principios de los 80 se comenzaron a desarrollar los ordenadores de forma exponencial. El crecimiento era tan veloz que se temía que las redes se bloquearan debido al gran número de usuarios y de información transmitida, hecho causado por el fenómeno e-mail. La red siguió creciendo exponencialmente como muestra el gráfico (Anonimo, 2016).

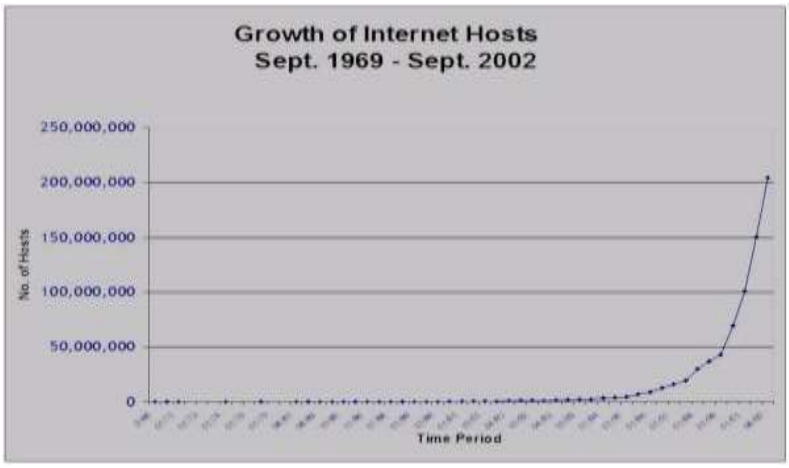


Figura: 1 Crecimiento de desarrollo de ordenadores

Fuente: Faculta de informática Barcelona

Para el año 1974, se publicó el diseño básico del protocolo de Internet (IP) de Cerf - Kahn, convirtiéndose, junto con un protocolo de capa superior llamado TCP, en el punto de partida para la explotación y uso de las redes de interconexión. No fue sino hasta inicios de la década de los años 90, que la Internet y su protocolo base TCP/IP, inicio su proliferación y uso masivo, cuando, por primera vez, iniciaron operaciones dos compañías llamadas UUnet y Psinet como ISP (Internet Service Provider) en los Estados Unidos, consiguiendo la tecnología, crecimientos exponenciales nunca previstos en tan poco tiempo. Desde entonces, el Protocolo IP, en su versión actual, 4 (IPv4), ha sido muy exitoso, por su diseño flexible y poderoso. Ha permitido que la Internet maneje redes heterogéneas, cambios bruscos en las tecnologías de hardware y aumentos enormes de escala (Pinillos, 2008)

Con el transcurrir del tiempo, la tecnología de redes ha madurado considerablemente, han surgido nuevas aplicaciones y nuevos protocolos. De manera, que el protocolo IP (IPv4) se quedó corto en el alcance de soportar estas nuevas tendencias tecnológicas. Tanto así, que

ya se están presentando algunas limitaciones al funcionamiento de las redes actuales, como lo es, fundamentalmente, la inminente saturación del espacio de direcciones IP, por el abrupto crecimiento de la Internet, limitando el crecimiento de la misma; soporte inadecuado de las nuevas aplicaciones, ya que son más demandantes de factores como: tiempos de respuestas y disponibilidad de ancho de banda; y por último, la inminente necesidad de manejar altos grados de seguridad, pero de manera nativa, ya que IPv4 para manejar seguridad, se basa en protocolos (Patches) como IPSec (IP Security Protocol), SSL (Secure Sockets Layer) , SHTTP (Secure HyperText Transfer Protocol) , los cuales ninguno es un estándar (Mediana et al, 2013, p. 72).

Los esfuerzos para desarrollar un sucesor de IPv4 se inició en la década de 1990, dentro de la Internet Engineering Task Force (IETF) "El IETF es una comunidad abierta internacional encargado de la evolución de las arquitecturas de Internet y de sus normas. Un estándar de Internet que comienza como un proyecto de Internet, que generalmente se desarrolla durante la publicación de versiones sucesivas. A continuación, podrán ser publicados en documento de solicitud de comentarios (RFC). Algunos definen las normas IETF RFC, mientras que otros son documentos informativos o describir los protocolos experimentales". El objetivo era resolver las limitaciones de espacio de direcciones, así como proporcionar una funcionalidad adicional. El IETF comenzó los trabajos en el año 1993, con un Protocolo de Internet de Nueva Generación (IPng) para investigar diferentes propuestas y hacer recomendaciones para futuras acciones.

El IETF recomienda IPv6 en el año 1994. (El nombre de IPv5 había sido asignado a un protocolo de secuencia experimental). Su recomendación se especifica en el RFC 1752: *La*

Recomendación para Protocolo IP de Nueva Generación. Después de esta, siguieron nuevas propuestas donde *The Internet Engineering Steering Group* aprobó la recomendación de IPv6 y redactó un Proyecto de Norma el 17 de noviembre de 1994. El RFC 1883: *Protocolo de Internet versión 6 (IPv6),* el cual se publicó en 1995. El núcleo básico de protocolos IPv6 "Dos de los actuales grupos de trabajo IETF que se concentran en las operaciones y protocolos IPv6 son: El grupo de trabajo de operaciones de IPv6 (v6ops) y el grupo de trabajo de mantenimiento de IPv6 (6man)" se convirtió en un proyecto de norma IETF el 10 de agosto de 1998. Esto incluye RFC 2460, que sustituyó a RFC 1883. En pocas palabras, se puede decir que IPv6 es un protocolo diseñado para manejar la tasa de crecimiento de Internet y para hacer frente a los exigentes requisitos de calidad de servicios, la movilidad y la seguridad de extremo a extremo.

Este nuevo Protocolo de Internet IPv6, se convierte en una evolución natural del protocolo anterior IPv4, más no es un cambio abrupto del mismo, ya que funciones que servían en IPv4 se mantuvieron y mejoraron en IPv6, y funciones que no servían se eliminaron, produciendo una serie de características determinantes en la mejora del protocolo anterior (Frankel, Gaveman, Pearce, & IPv6, 2010).

IPV6

El IPv6 fue diseñado por Steve Deering y Craig Mudge, adoptado por Internet Engineering Task Force (IETF) en 1994. IPv6 también se conoce por “IP Next Generation” o “IPng”. IPv6 es la segunda versión del Protocolo de Internet que se ha adoptado para uso general. También hubo un IPv5, pero no fue un sucesor de IPv4; mejor dicho, fue un protocolo

experimental orientado al flujo de streaming que intentaba soportar voz, video y audio, la IPv6 está destinada a sustituir al estándar IPv4, la misma cuenta con un límite de direcciones de red, lo cual impide el crecimiento de la red (Jacome, 2009).

A continuación se indica cómo se ve un datagrama IPv6:

<---------- 32 bits ---------->

Versión (Versión)	**Traffic class (Clase de tráfico)**	**Flow label (Etiqueta de flujo)**
Payload length (Longitud de carga útil)	Next header (Siguiente encabezado)	Hop limit (Límite de saltos)
Source IP address (Dirección IP de origen)		
Destination IP address (Dirección IP de destino)		
Data (Datos)		

Figura 2 encabezado de datagrama de ipv6

Fuente: es.ccm.net

A continuación se indican los significados de los diferentes campos (Vantiaute, 2016):

- **El campo Versión (Versión)** siempre es equivalente a 4 bits para IPv6. Durante el período de transición de IPv4 a IPv6, los Routers deberán fijarse en este campo para saber qué tipo de datagrama están enrutando.

- **El campo Traffic Class (Clase de tráfico)** (codificado con 8 bits) se utiliza para distinguir las fuentes que deben beneficiarse del control de flujo de otras. Se asignan prioridades de 0 a 7 a fuentes que pueden disminuir su velocidad en caso de congestión. Se asignan valores de 8 a 15 al tráfico en tiempo real (datos de audio y video incluidos) en donde la velocidad es constante.

- Esta distinción en los flujos permite que los routers reaccionen mejor en caso de congestión. En cada grupo de prioridad, el nivel de prioridad más bajo se relaciona con los datagramas de menor importancia.

- **El campo Flow label (Etiqueta de flujo)** contiene un número único escogido por la fuente que intenta facilitar el trabajo de los routers y permitir la implementación de funciones de calidad de servicio como RSVP (*Resource reSerVation setup Protocol [Protocolo de reserva de recursos]*). Este indicador puede considerarse como un marcador de un contexto

en el router. El router puede entonces llevar a cabo procesamientos particulares: escoger una ruta, procesar información en "tiempo real", etc.

El campo de etiqueta de flujo puede llenarse con un valor aleatorio, que se utilizará como referencia del contexto. La fuente mantendrá este valor para todos los paquetes que envíe para esta aplicación y este destino. El procesamiento se optimiza debido a que el router ahora sólo tiene que consultar cinco campos para determinar el origen de un paquete. Además, si se utiliza una extensión de confidencialidad, la información relacionada con los números de puerto está enmascarada para los routers intermediarios.

- **El campo Payload limit (Longitud de carga útil)** de dos bytes contiene sólo el tamaño de la carga útil, sin tener en cuenta la longitud del encabezado. Para paquetes en los que el tamaño de datos es superior a 65.536, este campo vale 0 y se utiliza la opción de jumbo grama de la extensión "salto a salto".
- **El campo Next header (Siguiente encabezado)** tiene una función similar a la del campo*protocol (protocolo)* en el paquete IPv4: simplemente identifica el encabezado siguiente (en el mismo datagrama IPv6). Puede ser un protocolo (de una capa superior ICMP, UDS, TCP, etc.) o una extensión.
- **El campo Hop limit (Límite de saltos)** reemplaza el campo *"TTL"* (*Time-to-Live [Tiempo de vida]*) en IPv4. Su valor (de 8 bits) disminuye con cada nodo que reenvía el paquete. Si este valor llega a 0 cuando el paquete IPv6 pasa por un router, se rechazará y se enviará un mensaje de error ICMPv6. Esto se utiliza para evitar que los datagramas circulen indefinidamente. Tiene la misma función que el campo *Time to live (Tiempo de vida)* en IPv4, es decir, contiene un valor que representa la cantidad de saltos y que

disminuye con cada paso por un router. En teoría, en IPv4, hay una noción del tiempo en segundos, pero ningún router la utiliza. Por lo tanto, se ha cambiado el nombre para que refleje su verdadero uso.

- Los siguientes campos son **Source address (Dirección de origen)** y **Destination address (Dirección de destino)**.

  Después de diferentes debates, se acordó que lo mejor era que las direcciones tuvieran una longitud fija equivalente a 16 bytes.

  Los primeros bits de la dirección —el prefijo— definen el tipo de dirección. Las direcciones que comienzan con 8 ceros se reservan, en particular para las direcciones IPv4. Por lo tanto, todas las direcciones que comienzan con 8 ceros se reservan para las direcciones IPv4. Se admiten dos variantes, que se distinguen según los 16 bits siguientes (o sea 16 bits a 0 ó 1)

Diferencias entre IPv4 E IPv6

Diferencias entre IPv4 e IPv6

	Protocolo Internet versión 4 (Ipv4)	Protocolo Internet versión 6 (IPv6)
Lanzada en	1981	1999
Tamaño de las direc-ciones	número de 32 bits	número de 128 bits
Formato de las direc-ciones	Notación decimal con puntos 199.43.0.202	Notación hexadecimal: 2001:500:4::/48
Cantidad de direccio-nes	2^{32} = ~4 mil millones de direcciones	2^{128} = ~16 trillones de direcciones

Tabla 1.

Fuente: **Seguridad en IPv6**

IPv6 incluye explícitamente la posibilidad de utilizar el modelo de seguridad IPSec (Internet Protocol Security) que proporciona autenticidad, integridad y confidencialidad a las comunicaciones de extremo a extremo. IPsec es un conjunto de protocolos abiertos que tienen como fin proporcionar seguridad en las comunicaciones de la capa de red del modelo OSI (a la que pertenece el protocolo IPv6) y, de ese modo, a todos los protocolos de capas superiores.

En IPv4, la implementación de IPsec se define en una especificación diferente a la del propio protocolo IPv4, por lo que la inclusión del protocolo se hace con mecanismos definidos fuera del mismo, mientras que, en IPv6, la propia arquitectura "extensible" del protocolo permite implementar IPsec de forma natural. Es importante reseñar que IPv6 habilita la posibilidad de usar IPsec, y no los mecanismos de cifrado y autenticación propios de IPsec[3].

FIREWALL

Han pasado 20 años desde que Check Point Software Technologies embarcó su primer *firewall* de red empresarial, marcando el inicio de un mercado masivo para *firewalls* que ha protegido millones de redes en todo el mundo.

El FireWall-1 de Check Point, develado en el NetWorld Interop en 1994, no fue el primer *firewall*, por supuesto. El *firewall* había comenzado a tomar forma con el

surgimiento de Internet. Las empresas y universidades entre los 80 y los 90 vieron la necesidad de bloquear el tráfico IP no deseado, creando una barrera de ingreso perimétrica. En esa época, a veces ellos mismos "implementaban" sus propios *routers* u otros equipos, hasta que los fabricantes finalmente llegaron al rescate con productos de *firewall* que les facilitaran esta labor.

Marcus Ranum, ahora director de seguridad en Tenable Network Security, es considerado el más prominente de los primeros innovadores de *firewalls* comerciales, porque diseñó el firewall DEC SEAL en 1990, y trabajó con el firewall Gauntlet y el juego de herramientas TIS en Trusted Information Systems. TIS, fundada en 1983 por el ex empleado de la NSA Steve Walker, se enfocó en clientes gubernamentales de alta seguridad; la compañía fue vendida a Network Associates (que después se convertiría en McAfee) en 1998. Otros esfuerzos tempranos, como el *firewall* Raptor, también existían. Pero fue el lanzamiento del FireWall-1 de Check Point lo que terminó creando una especie de mercado masivo al que pronto se unieron no solo los grandes proveedores de redes, como Cisco y Juniper, sino un grupo de otros jugadores, como WatchGuard.

A lo que hoy se le llama *firewall*, típicamente hace más que un simple filtrado y control de puertos. También puede incluir detección de intrusos y un sistema de protección (IPS), un antivirus, filtrado de URL, actuar como dispositivo de prevención de pérdida de datos y mucho más, incluyendo detección de amenazas de día cero al estilo *sandbox*. Los analistas de seguridad en consultoras tecnológicas han dejado su marca criticando todas las cosas

que los fabricantes de seguridad han estado haciendo por años, y urgiéndolos a hacer más, como tener mayores velocidades de salida o mejor administración.

En todo este tiempo, el mercado de *firewalls* se ha multiplicado y Gartner piensa que superará los nueve mil millones de dólares este año. Los *firewalls* han sido usados por largo tiempo no solo en el perímetro, sino también dentro de las redes empresariales para acordonar segmentos. Pero independientemente de todo esto, la ironía es que el rol del *firewall* de red está en duda más que nunca antes, debido al crecimiento del uso de servicios basados en la nube y de dispositivos móviles (CIO PERU, 2014).

ANALIZADOR DE LOGS

Un log es un registro oficial de eventos durante un rango de tiempo en particular. Para los profesionales en seguridad informática es usado para registrar datos o información sobre quién, que, cuando, donde y por qué un evento ocurre para un dispositivo en particular o aplicación.

Analizador de logs: es un software libre de análisis web de registro de log, programa que se ejecuta bajo un sistema operativo, y proporciona la capacidad de hacer búsquedas de DNS inversa en los archivos log, para indicar sitios web donde fueron originados. (marlisq.wordpress.com, 2012)

2.2 Marco contextual

El aumento de la cantidad de asignaciones de recursos de IPv4 tuvo un incremento del 59% entregado a empresas y organizaciones de la región latinoamericana, pasando de 28 millones en 2012 a 28,5 millones en 2013 indico Luisa Villa, gerente de Lacnic (Registro de direcciones de internet para América Latina y el Caribe), llevando esto que para finales del 2014 la región de Latinoamérica se quede sin direcciones de internet IPv4, un protocolo anterior al IPv6 Habilitado en 2012 para ampliar la cantidad de direcciones disponibles en el mundo.

Esta organización, con sede en Montevideo, informó que la región entró en el proceso de agotamiento de sus direcciones de internet Protocol versión 4 (IPv4) tras superar las 178 millones de asignaciones. "con evaluaciones más estrictas en todo el mundo". Ahora el desafío es asegurar el crecimiento continuo de la red a través de una correcta transición a la versión 6 (IPv6) del protocolo de Internet, explicó el director ejecutivo de Lacnic, Raúl Echeverría a la agencia AFP.

Los países de la región que han recibido más direcciones IPv4 durante los 11 años de existencia de Lacnic son Brasil con 73 millones, seguido por México con 27,6 millones. Argentina está tercero con 18,3 millones, seguido de Colombia con 16,3 millones y Chile con 9,4 millones.

"Este cambio en el comportamiento de las solicitudes puede atribuirse al crecimiento acelerado de la penetración de internet en la región, así como al cercano agotamiento de

IPv4 previsto para este año", aseguró Villa. Lacnic advirtió que en este marco "la necesidad del despliegue de IPv6 es hoy más que nunca una realidad inevitable e inaplazable si los proveedores de conectividad desean satisfacer la demanda de sus clientes y de nuevos usuarios" (Revista el Tiempo, 2014).

Figura 3 Reporte de espacio libre para IPv4
Fuente: http://www.mintic.gov.co/

Desde los años 80 y 90 cuando las empresas y las universidades vieron la necesidad de empezar a controlar el tráfico IP no deseado, empezaron a surgir los primeros firewall no de manera comercial si no de forma privada cada empresa o universidad implementaba su propio firewall para uso interno de la misma.

En 1994 sale al mercado el primer firewall comercial desarrollado por la empresa Check Point Software Technologies llamado Firewall-1, con el auge del internet fueron saliendo nuevas aplicaciones y Check Point con su capacidad de responder realmente despegó porque no hacía ningún análisis de capa 7 y era fácil escribir una regla para permitir el tráfico (Messmer, 2014).

Los avances empresariales y tecnológicos han ido mejorando de manera continua la Protección que proporcionaba el firewall tradicional. Los usuarios ya esperan poder Trabajar desde cualquier lugar que deseen, ya sea la oficina, su casa, una habitación de hotel o una cafetería, lo que deja obsoleto el concepto tradicional de perímetro.

Además, las aplicaciones esquivan con facilidad los tradicionales firewalls basados en puertos mediante el salto de puertos, el uso de SSL y SSH, el acceso a través del puerto 80 o el uso de puertos no estándar (Alto, 2014).

Hay muchas personas que suelen creer que un Antivirus es igual a un Firewall o hasta en ocasiones decir que nos es menester el uso de un firewall, esto depende del entorno de trabajo porque si estamos hablando de una empresa es necesario el uso de estos.

- **El antivirus** es un programa que detecta la **presencia de virus informáticos** (malware que altera el funcionamiento normal del ordenador sin que el usuario lo sepa o consienta) y los elimina o repara.

- **Firewall, o cortafuegos**, es una parte de la red o el sistema que se realiza para bloquear accesos no autorizados y permitiendo los que sí lo están. Se pueden hacer por medio de software o hardware, y permiten una mayor protección a las redes, especialmente importante en empresas que cuentan con datos que han de ser bien protegidos (Netweb, 2013).

2.3 Marco conceptual

Nuestro estudio se ubica dentro del contexto geográfico de la seguridad de una RED IPV6 y en tanto como responder ante un evento adverso.

En este contexto hacemos nuestra una metodología adecuada al objetivo de estudio que perseguimos y usamos un conjunto de conceptos básicos que revisaremos a continuación.

SOFTWARE

IAN SOMMERVILLE define el software como "como un programa de ordenador, la documentación asociada y la configuración de datos que se necesitan para que estos programa operen de manera correcta" (Sommerville, 2005)

Otros autores como Roger Pressman lo definen como "el producto que construyen los programadores profesionales y al que después le dan mantenimientos durante un largo tiempo. Incluye programas que se ejecutan en una computadora de cualquier tamaño y arquitectura, contenido que se presenta a medida de que se ejecutan los programas de cómputo e información descriptiva tanto en una copia dura como en formatos virtuales que engloban virtualmente a cualesquiera medios electrónicos" (Pressman, 2010).

FIREWALL

Un firewall es software o hardware que comprueba la información procedente de Internet o de una red y, a continuación, bloquea o permite el paso de ésta al equipo, en función de la configuración del firewall.

Un firewall puede ayudar a impedir que hackers o software malintencionado (como gusanos) obtengan acceso al equipo a través de una red o de Internet. Un firewall también puede ayudar a impedir que el equipo envíe software malintencionado a otros equipos (windows.microsoft.com, 2015).

IPv6

Es la versión 6 de IP, diseñado para coexistir con IPv4 durante una fase de transición, hasta que de forma transparente, IPv4 deje de utilizarse y desaparezca de la red.

Cuando utilizamos Internet para cualquier actividad, ya sea correo electrónico, navegación web, descarga de archivos, o cualquier otro servicio o aplicación, la comunicación entre los diferentes elementos de la red y nuestro propio computador o teléfono celular, utiliza un protocolo que denominamos Protocolo de Internet (IP, Internet Protocol).

En los últimos años, prácticamente desde que Internet tiene un uso comercial, la versión de este protocolo es el número 4 (IPv4).

Para que los dispositivos se conecten a la red, necesitan una dirección IP. Cuando se diseñó IPv4, casi como un experimento, no se pensó que pudiera tener tanto éxito comercial, y

dado que sólo dispone de 2^32 direcciones (direcciones con una longitud de 32 bits, es decir, 4.294.967.296 direcciones), junto con el imparable crecimiento de usuarios y dispositivos, implica que en pocos meses estas direcciones se agotarán.

Por este motivo, y previendo la situación, el organismo que se encarga de la estandarización de los protocolos de Internet (IETF, Internet Engineering Task Force), ha trabajado en los últimos años en una nueva versión del Protocolo de Internet, concretamente la versión 6 (IPv6), que posee direcciones con una longitud de 128 bits, es decir 2^128 posibles direcciones (340.282.366.920.938.463.463.374.607.431.768.211.456), o dicho de otro modo, 340 sextillones.

El despliegue de IPv6 se irá realizando gradualmente, en una coexistencia ordenada con IPv4, al que se irá desplazando a medida que dispositivos electrónicos con conexión a Internet, equipos de red, aplicaciones, contenidos y servicios se vayan adaptando a la nueva versión del protocolo de Internet.
Por ello, es importante que entendamos cómo se realiza el despliegue del nuevo protocolo de Internet, tanto si somos usuarios residenciales, como corporativos, proveedores de contenidos, proveedores de servicios de Internet, así como la propia administración pública (MINTIC, 2015).

Podemos recordar algunas “famosas frases” que nos ayudarán a entender hasta qué punto, los propios ‘precursores’ de la revolución tecnológica que estamos viviendo, no llegaron a prever:

- “Pienso que el mercado mundial de ordenadores puede ser de cinco unidades”
- “640 Kbps. de memoria han de ser suficientes para cualquier usuario”
- “32 bits proporcionan un espacio de direccionamiento suficiente para Internet”

ATAQUE INFORMÁTICO

Es un intento organizado e intencionado causada por una o más personas para causar daño o problemas a un sistema informático o red.

Existen muchas herramientas creadas por personas informáticas o que tienen conocimiento de cómo es el movimiento de la información en la red. Hoy los botnets (conjunto formado por ordenadores infectados por un tipo de software malicioso, permite al atacante controlar dicha red de forma remota) son responsables de la mayor parte de los ataques DDoS (ataque distribuido denegación de servicio) en Internet. La comprensión de las características de este tipo de ataques DDoS es fundamental para desarrollar esquemas de mitigación de DDoS eficaces (Wang et al, 2014).

La evolución de las secuencias de comandos y programas creados para la intrusión informática se han realizado con el fin de dar la evaluación de la vulnerabilidad y análisis de los logs (registro oficial de eventos durante un rango de tiempo en particular), en donde la computación evolutiva ha contrarrestado esto mediante la construcción de secuencias de comandos de ataques a un ordenador indetectable. El uso de un sistema operativo simulado,

muestra las secuencias de comandos que se pueden desarrollar para cubrir sus pistas y llegar a ser difícil de detectar desde el análisis del archivo de registro (Fogla et al, 2006).

Los atacantes a menudo tratan de evadir un sistema de detección de intrusos (IDS) al lanzar sus ataques. El estudio más reciente mostró que algunos intrusos de red basados en la carga útil de anomalía de sistemas de detección puede ser eludida por un polimórfica ataque de mezcla (PBA). la idea principal de un PBA es crear cada instancia polimórfica de tal manera que las estadísticas de paquete (s) de ataque que coincida con el perfil normal de tráfico (Fogla, 2006).

El problema de la propagación de los virus informáticos puede ser significativo teniendo en cuenta que un virus puede dañar o eliminar datos del equipo, usar el programa de correo electrónico para propagarse a otros equipos o incluso borrar todo el contenido del disco duro.

Un ataque informático consiste en aprovechar alguna debilidad o falla en el software, en el hardware, e incluso, en las personas que forman parte de un ambiente informático; para obtener un beneficio, por lo general de condición económica, causando un efecto negativo en la seguridad del sistema, que luego pasa directamente en los activos de la organización (EcuRed, s.f.).

INJECTION SQL

La inyección directa de comandos SQL es una técnica donde un atacante crea o altera comandos SQL existentes para exponer datos ocultos, sobrescribir los valiosos, o peor aún, ejecutar comandos peligrosos a nivel de sistema en el equipo que hospeda la base de datos. Esto se logra a través de la práctica de tomar la entrada del usuario y combinarla con parámetros estáticos para elaborar una consulta SQL. Los siguientes ejemplos están basados en historias reales, desafortunadamente.

Una vulnerabilidad en las aplicaciones web permite a los usuarios maliciosos obtener un acceso sin restricciones a la información privada y confidencial. Las Vulnerabilidades de inyección SQL son particularmente relevantes, como servicios web que con frecuencia acceden a una base de datos relacional utilizando comandos SQL. La inyección SQL se coloca a la cabeza en los mecanismos de ataque de aplicaciones web usadas por los hackers para robar datos de las organizaciones. Los hackers 'pueden tener ventajas debido al diseño defectuoso, inadecuadas prácticas de codificación, validaciones incorrectas de entrada del usuario, errores de configuración, u otras deficiencias de la infraestructura (A.S., 2003)
En la terminología de seguridad informática, los ataques de inyección SQL (SQLIAs) son ataques que suponen una amenaza de seguridad para aplicaciones web mediante la manipulación, modificación, recuperar o destruyendo la información sensible que subyace en el servidor de base de datos a través de aplicaciones web. Este tipo de ataques podría comprometer la confidencialidad, integridad y disponibilidad de los sistemas de bases de datos de las aplicaciones en línea. Aunque muchos investigadores y desarrolladores se centran en la prevención de este tipo de ataque, y proponer técnicas para superar este

problema, los métodos, ya sea falla al abordar correctamente este tipo de ataques o tienen alguna limitación en la prevención de todo tipo de SQLIAs (Alkhashab, 2011)

La Refactorización a menudo requiere la reordenación de fragmentos de código; tal es el caso cuando se migra de una API (Interfaz de Programación de Aplicaciones) a otro. Realizar dicha reordenación de forma manual es complejo y propenso a errores. Un ejemplo específico en el ámbito de la seguridad implica la ejecución de consultas de bases de datos, en el que algunos de los parámetros provienen de fuentes no fiables. En Java, la API de comandos proporciona oportunidades para que los ataques de inyección SQL (Abadi, 2011) debido a la falta de validación en la entrada de datos y a la conexión a la base de datos con privilegios de súper usuario o de alguien con privilegios para crear usuarios, el atacante podría crear un súper usuario en la base de datos (The php group, s.f.). p.11

INJECTION JAVASCRIPT

Es una técnica que permite modificar un contenido de un sintió sin tener que salir del sitio. Esto puede ser muy útil cuando por ejemplo, tiene que suplantar el servidor mediante la edición de algunas opciones de formulario, Inyecciones de JavaScript se ejecutan desde la barra de URL de la página que está visitando. (thisis legal, s.f.)

Proteger auto ligera JavaScript, el objetivo es prevenir o modificar el comportamiento inapropiado causado por un comando malicioso inyectado por scripts o código de terceros mal diseñado. El enfoque se basa en la modificación del código con el fin de hacer que sea auto-protección: el mecanismo de protección (política de seguridad) está incrustado en el

propio código y de seguridad intercepta llamadas a la API pertinentes. Los desafíos que provienen de la naturaleza del lenguaje JavaScript: las variables en el ámbito del programa pueden ser redefinidos, y el código se pueden crear y ejecutar en la marcha. Esto crea problemas potenciales, respectivamente, para la inviolabilidad del mecanismo de protección, y para asegurar que los eventos relevantes de seguridad sin pasar por alto la protección (Phung, 2009).

Actualmente los navegadores web se utilizan a menudo como un medio popular para comprometer ordenadores conectados a Internet. Un atacante podría inyectar un malware de JavaScript en una página web. Cuando una víctima visita esta página, el malware se ejecuta e intenta aprovechar una vulnerabilidad específica del navegador o descargar un programa no deseado. Ofuscado JavaScript malicioso puede evadir fácilmente la detección basada en firmas cambiando la apariencia de código JavaScript. Para hacer frente a este problema, algunos estudios previos han utilizado análisis estático en el que algunas de las características se extraen de las dos páginas web benignas y malignas, y luego un clasificador está capacitado para distinguir entre ellos. Debido a que el código JavaScript en la actualidad benigna es a menudo ofuscado, las técnicas de análisis estático generan muchas falsas alarmas (Gorji, 2014)

INJECTION CSS

Cascading Style Sheets vulnerabilidad de inyección también se conoce como inyección CSS. CSS vulnerabilidad de inyección, donde un código CSS arbitraria inyección atacante en la aplicación web que dictó dentro del navegador del Vitim. El impacto de esta

vulnerabilidad está en función de la carga útil atacante CSS, también puede conducir a Cross site scripting también (jain, s.f.)

Aunque este ataque es un poco raro, hay ocasiones en que las necesidades de accesibilidad de los usuarios de lectores de pantalla parecen estar en contradicción con las necesidades de los usuarios visuales. Este tipo de conflicto se produce cuando los desarrolladores web ponen elementos de formulario dentro de una matriz de tabla de datos, cuando quieren utilizar las imágenes como las partidas en lugar de texto, y en otras situaciones. La adición de texto ayuda a los usuarios de lectores de pantalla, pero puede complicar el diseño visual, reduciendo así la comprensibilidad. Una solución es utilizar CSS para ocultar el texto de los usuarios con visión de una manera que todavía es accesible a los lectores de pantalla (Bohman, 2004).

Estos ataques son de origen cruzado que en donde utilizan el estilo de importación de hoja para robar información confidencial de un sitio web de la víctima, el secuestro de sesión autenticada existente de un usuario; XSS defensas existentes son ineficaces (Huang, 2010).

LENGUAJE DE PROGRAMACION

Como cualquier lenguaje de programación, el lenguaje Java tiene su propia estructura, reglas de sintaxis y paradigma de programación. El paradigma de programación del lenguaje Java se basa en el concepto de programación orientada a objetos (OOP), que las funciones del lenguaje soportan.

El lenguaje Java es un derivado del lenguaje C, por lo que sus reglas de sintaxis se parecen mucho a C: por ejemplo, los bloques de códigos se modularían en métodos y se delimitan con llaves ({ y }) y las variables se declaran antes de que se usen.

Estructuralmente, el lenguaje Java comienza con *paquetes*. Un paquete es el mecanismo de espacio de nombres del lenguaje Java. Dentro de los paquetes se encuentran las clases y dentro de las clases se encuentran métodos, variables, constantes, entre otros. En este tutorial, aprenderá acerca de las partes del lenguaje Java (Perry, 2012).

PHP (Acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML, lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no sabrá el código subyacente que era.

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales.

Aunque el desarrollo de PHP está centrado en la programación de scripts del lado del servidor, se puede utilizar para muchas otras cosas. (PHP.net, 2001).

METODOLOGIAS AGILES

Las metodologías ágiles son sin duda uno de los temas recientes en ingeniería de software que están acaparando gran interés. Prueba de ello es que se están haciendo un espacio destacado en la mayoría de conferencias y workshops celebrados en los últimos años. Es tal su impacto que actualmente existen 4 conferencias internacionales de alto nivel y específicas sobre el tema. Además ya es un área con cabida en prestigiosas revistas internacionales. En la comunidad de la ingeniería del software, se está viviendo con intensidad un debate abierto entre los partidarios de las metodologías tradicionales (referidas peyorativamente como "metodologías pesadas") y aquellos que apoyan las ideas emanadas del "Manifiesto Ágil". La curiosidad que siente la mayor parte de ingenieros de software, profesores, e incluso alumnos, sobre las metodologías ágiles hace prever una fuerte proyección industrial. Por un lado, para muchos equipos de desarrollo el uso de metodologías tradicionales les resulta muy lejano a su forma de trabajo actual considerando las dificultades de su introducción e inversión asociada en formación y herramientas. Por otro, las características de los proyectos para los cuales las metodologías ágiles han sido especialmente pensadas se ajustan a un amplio rango de proyectos industriales de desarrollo de software; aquellos en los cuales los equipos de desarrollo son pequeños, con plazos reducidos, requisitos volátiles, y/o basados en nuevas tecnologías.

Aunque los creadores e impulsores de las metodologías ágiles más populares han suscrito el manifiesto ágil y coinciden con los principios enunciados anteriormente, cada metodología tiene características propias y hace hincapié en algunos aspectos más específicos. A

continuación se resumen dichas metodologías ágiles, dejando el análisis más detallado de XP para la siguiente sección (Cockbun, 2001)

Diferencias entre metodologías ágiles y no ágiles

Tabla 2.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Fuente: http://noqualityinside.com/nqi/nqifiles/XP_Agil.pdf

SCRUM

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint*es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto. Éstas son las verdaderas protagonistas,

especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración (R.C, 2001).

2.4 Marco teórico

INTERNET:

Es impresionante el grado y la rapidez con la que la modalidad interconectada de comunicación electrónica –conocida indistintamente como la red, Internet, la web o el ciberespacio –ha entrado en casi todos los aspectos de la vida cotidiana. Pero, a pesar de su popularidad y de su amplia propagación, es todavía muy nueva, demasiado nueva como para permitir una reflexión retrospectiva sobre su naturaleza y su impacto. Aun así, es posible negar su importancia y, por eso, la tentación de enjuiciar qué es y qué puede significar para la cultura, la ley y la política es muy grande (Graham, 2009)

EL ANTIVIRUS:

El programa llamado **Elk Corner** tiene el honor de ser sindicado como el primer virus esparcido entre computadoras fuera de un laboratorio. Fue creado en 1982 por el norteamericano **Richard Skrenta** y atacaba a los equipos que utilizaban el sistema operativo **Apple II**. Se transmitía, mediante la copia y la reproducción de discos removibles o disquetes.

Un **Virus** es un programa que puede reproducirse en forma automática haciendo copias de si mismo. Su objetivo consiste en infectar la mayor cantidad posible de computadoras,

distribuyéndose por redes o siendo transportado por los usuarios en medios de almacenamiento removibles o en E-mails.

HACKER:

El hacker es un experto informático cuyo accionar aparece asociado al vandalismo. Muchos les temen, aunque los hackers no necesariamente son malvados. La etimología de la palabra, según el gurú informático Richard Stallman, se refiere a divertirse con el ingenio.

DE QUIEN NOS PROTEGE EL FIREWALL?

Un firewall nos protege en tanto controlar el tráfico entrante y saliente de la red y evita que éste pueda poner en riesgo nuestra información. Sin embargo, y a través de la misma técnica, el firewall impide además que usuarios malintencionados accedan a nuestro equipo y lo controlen de manera remota o husmeen la información que contiene. Sin un firewall instalado en el equipo, sería muy fácil acceder a una computadora aun a pesar de que existan políticas de red y recursos compartidos que lo impiden (Burgos, 2010).

ANALIZADORES DE LOGS.

SPLUNK: Es un software para entender casi cualquier formato de información de log, desde la seguridad hasta la inteligencia analítica empresarial (comúnmente conocida como business anlytics), pasando por el monitoreo de la infraestructura. Las herramientas de búsqueda y gráficos de Splunk son tan ricas en funcionalidades que probablemente no existe ningún conjunto de datos al que no puedas acceder a través de su interfaz de usuario o de su API (splunk.com, 2015)

LOGGLY: Es una robusta herramienta para el análisis de los logs, enfocada en la simplicidad y la facilidad de uso para un público DevOps, ayudando a encontrar y resolver problemas operacionales. Esto la hace muy amigable para los desarrolladores. (loggly.com, 2016)

ETAPAS TIPICAS DE UN ATAQUE:

A continuación se describe el típico y genérico proceso de vulnerar la infraestructura informática de una entidad. Cabe resaltar que existen infinitas maneras y combinaciones de hacerlo, donde si bien todas recorren este camino, en general se pueden describir las siguientes etapas:

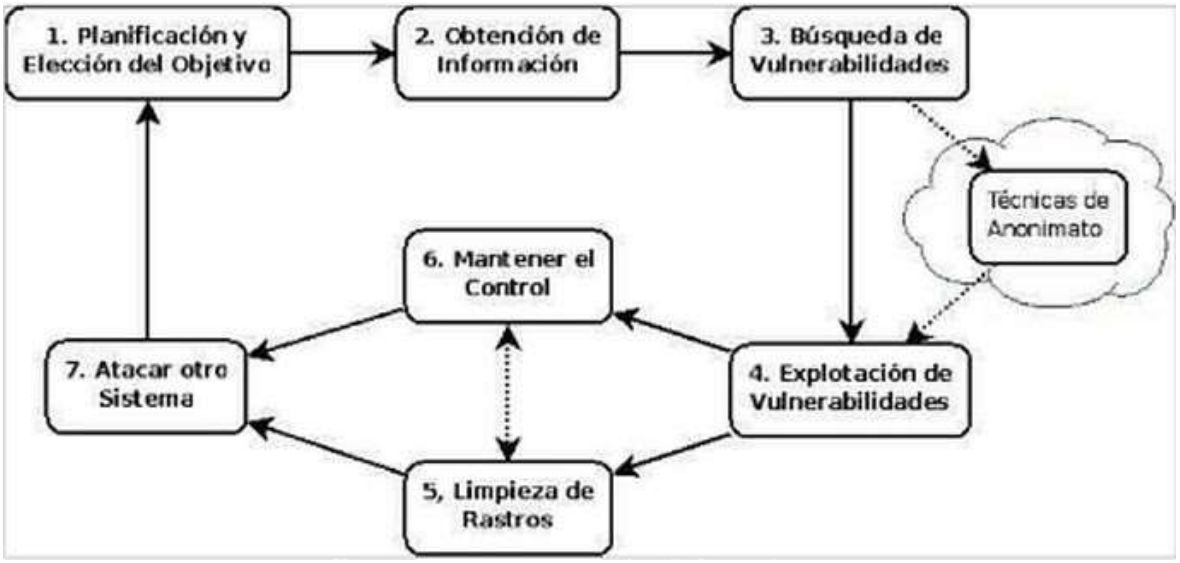


Figura 4 Etapas de un ataque

INYECCION SQL:

Un tipo de ataques a vulnerabilidades son las inyecciones SQL, las cuales consisten en la inserción o "inyección" de una consulta SQL a través de los datos de entrada que posee las aplicaciones o mediante la URL.

ATAQUES CON JAVASCRIPT:

Los ataques de tipo Cross-Site Scripting (a menudo abreviados en la literatura como ataques XSS) son ataques contra aplicaciones Web en los que un atacante toma control sobre el navegador de un usuario con el objetivo de ejecutar código o scripts malicioso (generalmente scripts escritos en lenguaje HTML o Javascript) dentro del entorno de confianza del sitio Web asociado a la aplicación final. Si dicho código es ejecutado satisfactoriamente, el atacante puede obtener acceso, de forma activa o pasiva, a recursos del navegador Web asociados con la aplicación (tales como cookies e identificadores de sesión).

Los ataques XSS o CSS se basan en inyectar una URL o código malicioso dentro de una URL valida de acceso a un sitio web embebiéndola en el campo de datos. Así, se aprovecha el sitio web original y se manipula solo una parte del mismo, normalmente la parte donde el usuario introducirá sus datos (login y password o número de tarjeta de crédito) para que sean enviados al sitio web fraudulento y no al oficial de la entidad (Nuñez, 2007)

2.5 Marco legal

LEY N^{O}.1341 ~ **2009**

"POR LA CUAL SE DEFINEN PRINCIPIOS Y CONCEPTOS SOBRE LA SOCIEDAD DE LA INFORMACIÓN Y LA ORGANIZACIÓN DE LAS TECNOLOGIAS DE LA INFORMACIÓN Y LAS COMUNICACIONES - TIC-, SE CREA LA AGENCIA NACIONAL DE ESPECTRO Y SE DICTAN OTRAS DISPOSICIONES"

EL CONGRESO DE COLOMBIA

DECRETA:

TITULO I DISPOSICIONES GENERALES

CAPITULO I - PRINCIPIOS GENERALES

ARTICULO 1.- OBJETO. La presente Ley determina el marco general para la formulación de las políticas públicas que regirán el sector de las Tecnologías de la Información y las Comunicaciones, su ordenamiento general, el régimen de competencia, la protección al usuario, así como lo concerniente a la cobertura, la calidad del servicio, la promoción de la inversión en el sector y el desarrollo de estas tecnologías, el uso eficiente de las redes y del espectro radioeléctrico, así como las potestades del Estado en relación con la planeación, la gestión, la administración adecuada y eficiente de los recursos, regulación, control y vigilancia del mismo y facilitando el libre acceso y sin discriminación

de los habitantes del territorio nacional a la Sociedad de la Información. Parágrafo. El servicio de televisión y el servicio postal continuarán rigiéndose por las normas especiales pertinentes, con las excepciones específicas que contenga la presente ley. Sin perjuicio de la aplicación de los principios generales del derecho.

ARTÍCULO 2.- PRINCIPIOS ORIENTADORES.

La investigación, el fomento, la promoción y el desarrollo de las Tecnologías de la Información y las Comunicaciones son una política de Estado que involucra a todos los sectores y niveles de la administración pública y de la sociedad, para contribuir al desarrollo educativo, cultural, económico, social y político e incrementar la productividad, la competitividad, el respeto a los derechos humanos inherentes y la inclusión social.
Las Tecnologías de la Información y las Comunicaciones deben servir al interés general y es deber del Estado promover su acceso eficiente y en igualdad de oportunidades, a todos los habitantes del territorio nacional.
Son principios orientadores de la presente Ley:

Prioridad al acceso y uso de las Tecnologías de la Información y las Comunicaciones. El Estado y en general todos los agentes del sector de las Tecnologías de la Información y las Comunicaciones deberán colaborar, dentro del marco de sus obligaciones, para priorizar el acceso y uso a las Tecnologías de la Información y las Comunicaciones en la producción de bienes y servicios, en condiciones no discriminatorias en la conectividad, la educación los contenidos y la competitividad.

Libre competencia. El Estado propiciará escenarios de libre y leal competencia que incentiven la inversión actual y futura en el sector de las TIC y que permitan la concurrencia al mercado, con observancia del régimen de competencia, bajo precios de mercado y en condiciones de igualdad. Sin perjuicio de lo anterior, el Estado no podrá fijar condiciones distintas ni privilegios a favor de unos competidores en situaciones similares a las de otros y propiciará la sana competencia.

Uso eficiente de la infraestructura y de los recursos escasos. El Estado fomentará el despliegue y uso eficiente de la infraestructura para la provisión de redes de telecomunicaciones y los servicios que sobre ellas se puedan prestar, y promoverá el óptimo aprovechamiento de los recursos escasos con el ánimo de generar competencia, calidad y eficiencia, en beneficio de los usuarios, siempre y cuando se remunere dicha infraestructura a costos de oportunidad, sea técnicamente factible, no degrade la calidad de servicio que el propietario de la red viene prestando a sus usuarios y a los terceros, no afecte la prestación de sus propios servicios y se cuente con suficiente infraestructura, teniendo en cuenta la factibilidad técnica y la remuneración a costos eficientes del acceso a dicha infraestructura. Para tal efecto, dentro del ámbito de sus competencias, las entidades de orden nacional y territorial están obligadas a adoptar todas las medidas que sean necesarias para facilitar y garantizar el desarrollo de la infraestructura requerida, estableciendo las garantías y medidas necesarias que contribuyan en la prevención, cuidado y conservación para que no se deteriore el patrimonio público y el interés general.

Protección de los derechos de los usuarios. El Estado velará por la adecuada protección de los derechos de los usuarios de las Tecnologías de la Información y de las Comunicaciones, así como por el cumplimiento de los derechos y deberes derivados del Habeas Data, asociados a la prestación del servicio. Para tal efecto, los proveedores y/u operadores directos deberán· prestar sus servicios a precios de mercado y utilidad razonable, en los niveles de calidad establecidos en los títulos habilitantes o, en su defecto, dentro de los rangos que certifiquen las entidades competentes e idóneas en la materia y con información clara, transparente, necesaria, veraz y anterior, simultánea y de todas maneras oportuna para que los usuarios tomen sus decisiones.

Promoción de la Inversión. Todos los proveedores de redes y servicios de telecomunicaciones tendrán igualdad de oportunidades para acceder al uso del espectro y contribuirán al Fondo de Tecnologías de la Información y las Comunicaciones.

Neutralidad Tecnológica. El Estado garantizará la libre adopción de tecnologías, teniendo en cuenta recomendaciones, conceptos y normativas de los organismos internacionales competentes e idóneos en la materia, que permitan fomentar la eficiente prestación de servicios, contenidos y aplicaciones que usen Tecnologías de la Información y las Comunicaciones y garantizar la libre y leal competencia, y que su adopción sea armónica con el desarrollo ambiental sostenible.

El Derecho a la comunicación, la información y la educación y los servicios básicos de las TIC: En desarrollo de los artículos 20 y 67 de la Constitución Nacional el Estado propiciará

a todo colombiano el derecho al acceso a las tecnologías de la información y las comunicaciones básicas, que permitan el ejercicio pleno de los siguientes derechos: La libertad de expresión y de difundir su pensamiento y opiniones, la de informar y recibir información veraz e imparcial, la educación y el acceso al conocimiento, a la ciencia, a la técnica, y a los demás bienes y valores de la cultura. Adicionalmente el Estado establecerá programas para que la población de los estratos desarrollará programas para que la población de los estratos menos favorecidos y la población rural tengan acceso y uso a las plataformas de comunicación, en especial de Internet y contenidos informáticos y de educación integral.

Masificación del gobierno en línea. Con el fin de lograr la prestación de servicios eficientes a los ciudadanos, las entidades públicas deberán adoptar todas las medidas necesarias para garantizar el máximo aprovechamiento de las Tecnologías de la Información y las Comunicaciones en el desarrollo de sus funciones. El Gobierno Nacional fijará los mecanismos y condiciones para garantizar el desarrollo de este principio. Y en la reglamentación correspondiente establecerá los plazos, términos y prescripciones, no solamente para la instalación de las infraestructuras indicadas y necesarias, sino también para mantener actualizadas y con la información completa los medios y los instrumentos tecnológicos.

Artículo 71 de la constitución política de Colombia

La búsqueda del conocimiento y la expresión artística son libres. Los planes de desarrollo económico y social incluirán el fomento a las ciencias y, en general, a la cultura. El Estado

creará incentivos para personas e instituciones que desarrollen y fomenten la ciencia y la tecnología y las demás manifestaciones culturales y ofrecerá estímulos especiales a personas e instituciones que ejerzan estas actividades (Congreso de Colombia, 1991)

Capítulo 3. Diseño metodológico

3.1 Metodología de desarrollo de software

"Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo. (Avinson & Fitzgerald, 2006).

Para el desarrollo de este proyecto se va a tener en cuenta la metodología de desarrollo ágil SCRUM, la cual comienza con la visión general del producto, especificando y dando detalle a las funcionalidades o partes que tienen mayor prioridad de desarrollo en bloques temporales o cortos (iteraciones que pueden llevarse a cabo en un periodo de tiempo breve de un mes o hasta de dos semanas, si así se necesita) cada iteración debe proporcionar un resultado completo.

3.2 Tipo de investigación

El presente Proyecto de Investigación es de tipo analítica descriptivo pues pretende exponer los diferentes elementos, procedimientos y características que se deberán tener en cuenta para la implementación de un prototipo de firewall orientado a las redes IPV6. La investigación no se detiene a comprobar hipótesis ni a hacer predicciones al respecto.

De esta manera, en este documento se detalla el procedimiento para implementar el firewall, se propone una arquitectura en un ambiente controlado y se realizan pruebas de funcionalidad con el fin de describir las diferentes funcionalidades que se pueden implementar.

En el caso concreto de este trabajo, el proceso se dividió en 5 actividades importantes. Las actividades son las siguientes:

Identificación y planteamiento del problema: Se requiere identificar el problema para limitar su alcance, plantear una hipótesis y establecer los objetivos. En este trabajo, la descripción del problema comprende la detección de intrusiones en los entornos de las aplicaciones distribuidas, más concretamente los ataques de inyección SQL, CSS y JAVASCRIPT, apuntados a poner en riesgo la disponibilidad de los datos, las aplicaciones y los servicios.

Creación del estado del arte: Dentro de esta actividad se realiza una revisión de las tecnologías disponibles y las propuestas existentes, para analizar el problema, conocer lo que está hecho y proponer una nueva solución. La revisión del estado del arte realizada en este trabajo comprende las tecnologías creadas a solucionar intentos de fraudes, técnicas de aprendizaje automático, y principalmente las propuestas existentes como solución al problema de la detección de los ataques estudiados.

Definición de la propuesta: En esta actividad se plantea la nueva solución partiendo de una investigación preliminar. La nueva solución debe corresponder con los objetivos propuestos. En este caso, se propone el DESARROLLO DE UN PROTOTIPO DE FIREWALL, QUE PERMITA DETECTAR ATAQUES DE INJECTION (SQL, JAVASCRIPT, CSS) DENTRO DE UNA RED IPV6.

Implementación y evaluación de la solución en entornos reales: Se realiza una implementación del prototipo de la propuesta, en un entorno real, para llevar a cabo una evaluación. La arquitectura propuesta se ha implantado en dos escenarios reales para abordar los ataques de inyección SQL, CSS y JAVASCRIPT.A partir de la implementación se ha podido obtener los resultados que han permitido valorar la efectividad de la arquitectura en la detección de las intrusiones estudiadas. Además de facilitar una comparación teórica, por sus características, con las propuestas existentes.

Publicación de los resultados: Con los resultados obtenidos y su evaluación, se inicia el proceso de divulgación de los resultados para someterlos a la valoración de la comunidad científica.

Capítulo 4. Presentación de resultados

4.1 Estado del arte

Como sabemos el protocolo IPv6 no es nuevo, en 1992 IETF (Internet Engineering Task Force) anuncio una convocatoria para la creación de los grupos de trabajo de "IP de próxima generación" ("IP Next Generation") o (IPNG) lo cual conocemos hoy como IPv6.

En la feria Networld Interop del 2001 en Paris la empresa Consulintel S.L presenta el firewall IPv6 6WINDGate y el primer equipo de acceso IPv6 completo, este firewall IPv6 completa la gama de funciones ofrecidas por el 6WINDGate, en entornos IPv4 e IPv6, ofreciendo calidad de servicio, movilidad y enrutamiento. 6WINDGate como el único equipo completo de acceso IPv4/IPv6 confirma su papel en la migración hacia IPv6 y el suministro de valor añadido, como declaro Pactrick Cocquet, consejero Delegado de 6WIND y vicepresidente del foro IPv6 "Debido a que la principal necesidad de los administradores de redes es crear una nueva gama de servicios IP de valor añadido y prepararse para la migración e IPv6, con el 6WINDGate ofrecemos la respuesta adecuada en el momento justo" (Tin Can Comunicacion, 2001)

En 2013 la empresa FORTINET proveedor global en seguridad de redes de alto rendimiento anuncio un nuevo dispositivo el FortiGate 3700D para brindar paridad de desempeño de firewall de IPv4 a IPv6, con la capacidad de alcanzar un rendimiento de hasta 160 GBps, el FortiGate entrega rendimiento y traducción IPv6 e IPv4 comparable,

eliminando el cuello de botella en el desempeño que causan otros dispositivos de seguridad. (Fortinet, 2001)

En la universidad del valle hicieron el diseño e implementación de una red IPv6 para transición eficiente desde IPv4; como lo mencionan sus autores, Luis E. Bolívar, Fabio G. Guerrero, Oscar Polanco; el proyecto se implementa en un modelo que tiene una red IPv6 nativa en la oficina central de una red corporativa y que utiliza mecanismos de transición para la conexión a internet IPv4, internet IPv6 y oficinas remota (Bolivar, 2012).

La UFPSO por su parte también ha estado realizando avances sobre el protocolo IPv6 de la mano del estudiante lobo contreras, Josué. Que alcanzo su reconocimiento como ingeniero con la tesis IMPLEMENTACION DE LA SEGURIDAD DEL PROTOCOLO DE INTERNET IPSEC EN REDES DE AREA LOCAL CON IPv6, esta tesis se centra en mostrar como la seguridad del protocolo de internet IPSEC puede brindar un nivel de seguridad a la información en las redes de datos que soportan el protocolo a la información de redes de siguiente generación IPv6. (Contreras, 2011)

En España para tesis de grado desarrollaron un analizador de logs para detectar posibles accesos no deseados, por el estudiante David Monforte Ruiz con la colaboración de la empresa Arsys Internet S.L con el objeto de ayudar a detectar posibles ataques a servidores alojados en esta empresa. (Ruiz, 2015).

AWStats (Advenced Wed Statics) es un analizador de logs muy rápido, configurable y sencillo de usar, escrito en Perl y con soporte para muchos idiomas, AWStats analiza el acccess.log y genera una página HTML con estadísticas y graficas sobre documentos más pedidos, procedencia de los visitantes, horas a las que entran, etc. (awstats.org, s.f.)

4.2 Diseño de interfaz.

EL proyecto "DESARROLLO DE UN PROTOTIPO DE FIREWALL QUE PERMITA DETECTAR ATAQUES DE INJECTION SQL, JAVASCRIPT Y CSS DENTRO DE UNA RED IPV6 CONTROLADO DENTRO DE UN SERVIDOR PROXY MULTIPLATAFORMA" desarrollado dentro de las instalaciones de la Universidad Francisco de Paula Santander Seccional Ocaña fue diseñado acorde a los requerimientos funcionales y no funcionales.

En este capítulo, se mostrara el diseño en base al cual se construyó el prototipo de firewall dando el inicio de su primera etapa de análisis de incidencias acorde a los requerimientos funcionales y no funcionales planteados, en donde a través del empleo del lenguaje de modelamiento Unificado UML, se presentarán los distintos diagramas que permitirán comprender la funcionalidad del prototipo mencionado.

Dentro de los requerimientos funcionales se definió el comportamiento interno del prototipo de firewall de la manipulación de datos y otras funcionalidades específicas que se muestran en los casos de uso que fueron llevados a la práctica. Estos requerimientos se especificaron como el comportamiento de entrada y salida del sistema y surgen de la razón

fundamental de la existencia del prototipo. Así mismo estos requerimientos funcionales se plantearon de la siguiente manera:

- **Información de Ataques.**

Figura 5: información de ataques

Fuente: Autores del proyecto.

A partir de este requerimiento se diseña una interfaz como lo muestra la figura anterior con los siguientes campos:

Ip_origen: dentro de este campo de podrá visualizar la **Ip** del atacante.

Fecha-Hora: Muestra la Fecha y hora exacta en que el atacante realiza

Su proceso de intrusión.

Recurso: Consiste en mostrar la afectación que tuvo el código fuente dentro del servidor al momento de ser atacado.

Código de Respuesta:

User-Agent: Muestra desde que Sistema Operativo fue atacado mi servidor, así mismo el navegador que se utilizó para realizarlo.

Referencia: Identifica el tipo de ataque que realiza dicho atacante.

Nro-Linea: Numero de Línea Afectado por el la inyección de ataques.

Detalle: Muestra en Detalle qué tipo de Ataque fue realizado al Servidor.

Cargue de Archivos.

Dentro de este requerimiento se diseña el botón CARGAR que tiene como funcionalidad cargar el Archivo Access de extensión .log. Este archivo es llamado Access porque describe y almacena todo el comportamiento del servidor durante peticiones realizadas.

Figura 6: Botón Cargar

Fuente: Autores del proyecto.

Seleccionamos la opción cargar, luego debemos seleccionar el archivo access.log que se encuentra en donde anteriormente instalamos nuestro servidor.

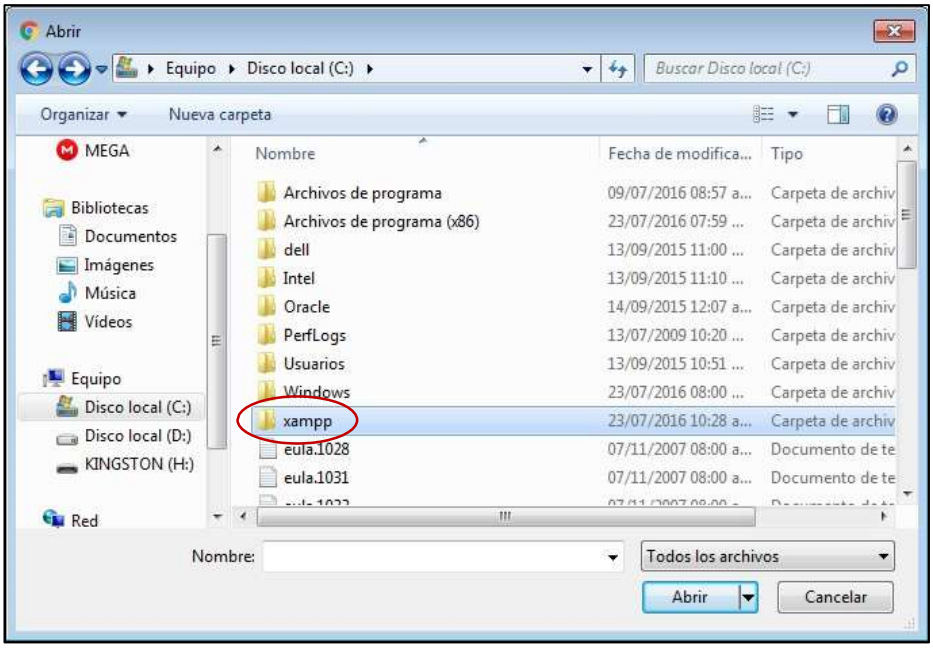


Figura 7: Carpeta de selección del servidor

Fuente: Autores del proyecto.

Seleccionamos la carpeta del servidor como se muestra en la figura anterior. Después de haber seleccionado la carpeta del servidor, seleccionamos la carpeta apache.

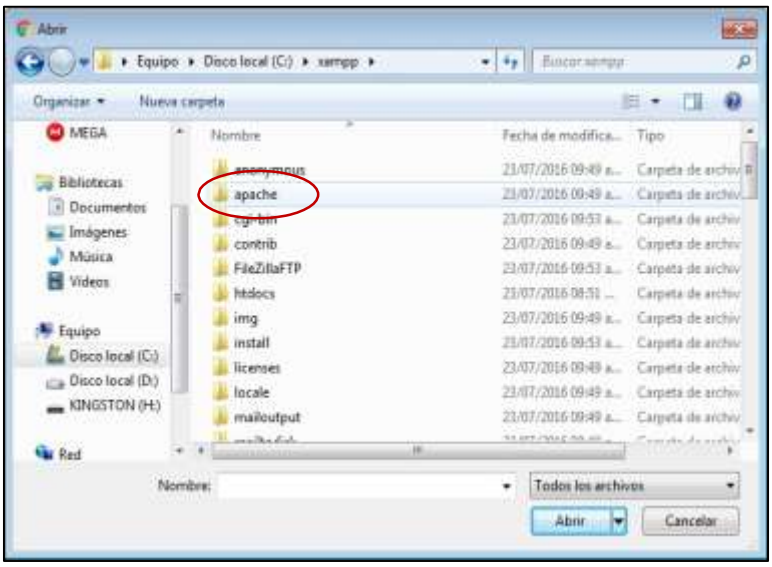


Figura 8: Carpeta Apache

Fuente: Autores del proyecto.

Selecionamos la carpeta logs.

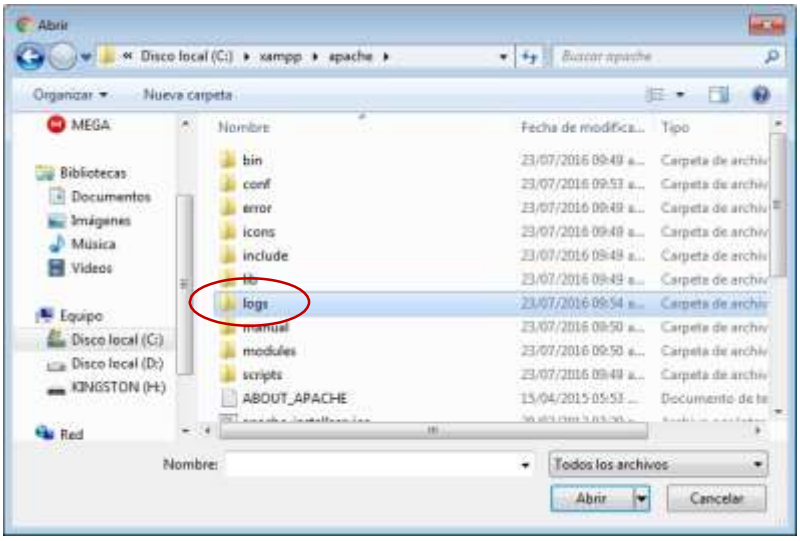


Figura 9: Carpeta Logs

Fuente: Autores del proyecto.

Seleccionamos el archivo access y damos clic en la opcion Abrir.

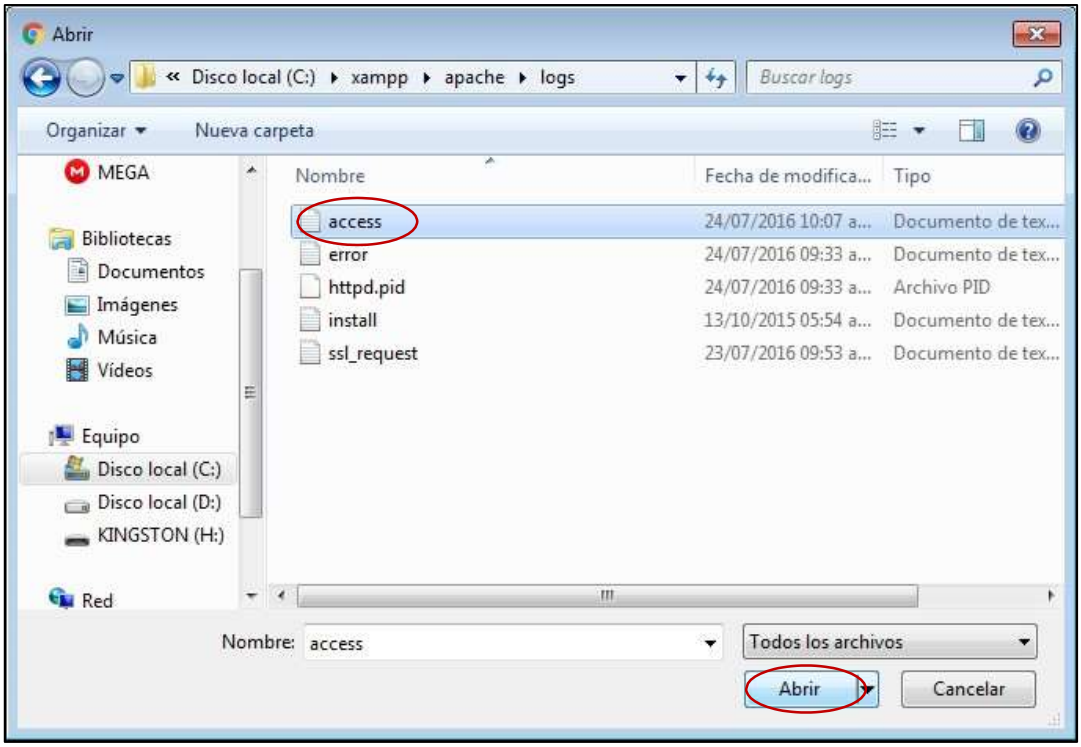


Figura 10: Archivo Access.log

Fuente: Autores del proyecto.

Envío de Archivo.

Dentro de este requerimiento se diseña de igual manera el botón ENVIAR que tiene como funcionalidad enviar el archivo cargado previamente en el prototipo para que este lea detalladamente toda esa información contenida dentro del archivo Access del servidor y así visualizar la información expuesta en el requerimiento

Información de Ataques.

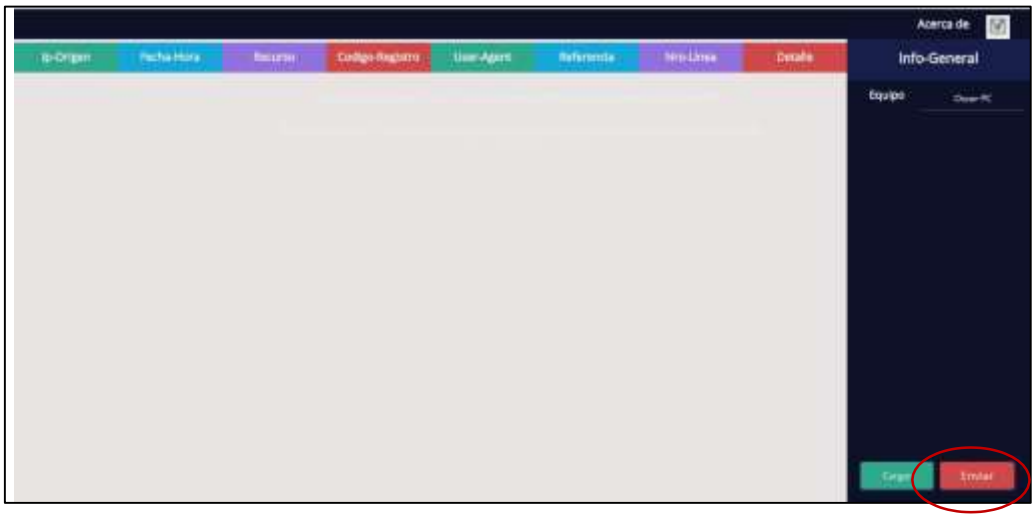


Figura 11: Interfaz Prototipo.

Fuente: Autores del proyecto.

Los requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de este como la funcionalidad, usabilidad, confiabilidad, compatibilidad con hardware y software, especificaciones del producto, etc. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en la interfaz del sistema. Así mismo estos requerimientos no funcionales se plantearon de la siguiente manera:

Confiabilidad de la Información: se define como requerimiento no funcional del prototipo, puesto que, será de máxima confianza, pues su utilización y funcionamiento brindara confiabilidad y la veracidad de la información.

Portabilidad: Se diseñó de tal manera de tal manera de que fuera funcional en cualquier dispositivo móvil o sistema operativo.

Usabilidad: El uso es de manera sencilla, puesto que, se diseñó con una interfaz sencilla de tal manera de que el usuario final pueda utilizarlo con la mayor rapidez posible.

Accesibilidad: El diseño del prototipo se diseñó de forma multiplataforma de que cualquier sistema operativo o dispositivo móvil pueda iniciarlo correctamente y hacer cualquiera de sus funcionalidades.

Rendimiento.

4.2.1 DiagramaS UML

A través del lenguaje de Modelamiento Unificado, se podrá visualizar, especificar y documentar el prototipo de firewall que se desarrolló en su primera de etapa como el análisis de incidencias realizadas al servidor dando como objetivo la identificación de cada una de ellas mencionadas dentro del transcurso del documento.

4.2.1.1 Diagrama de clases

En el siguiente diagrama, se mostrara las clases utilizadas en el servidor utilizado para la construcción del buscador que es el cual en donde el usuario atacante podrá realizar sus distintas peticiones y el prototipo de firewall podrá detectar.

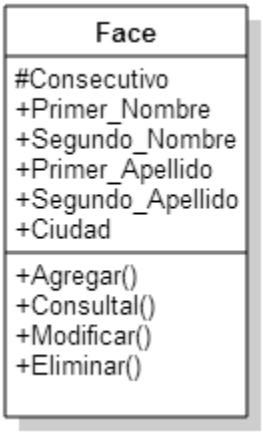


Figura 12: Diagrama UML de Clases Pagina Buscador

Fuente: Autores del proyecto

4.2.1.2 Diagrama de casos de uso

El siguiente diagrama permitirá conocer las distintas acciones que realizará el administrador, al interactuar con el prototipo de software, entre estas están:

Cargar archivo: En esta etapa, el administrador podrá cargar el archivo de incidencias del servidor.

Enviar archivo: Ya luego de haber cargado el archivo anteriormente, el administrador podrá enviar el archivo para que el prototipo de firewall analice, detecte y separe las incidencias realizadas al servidor.

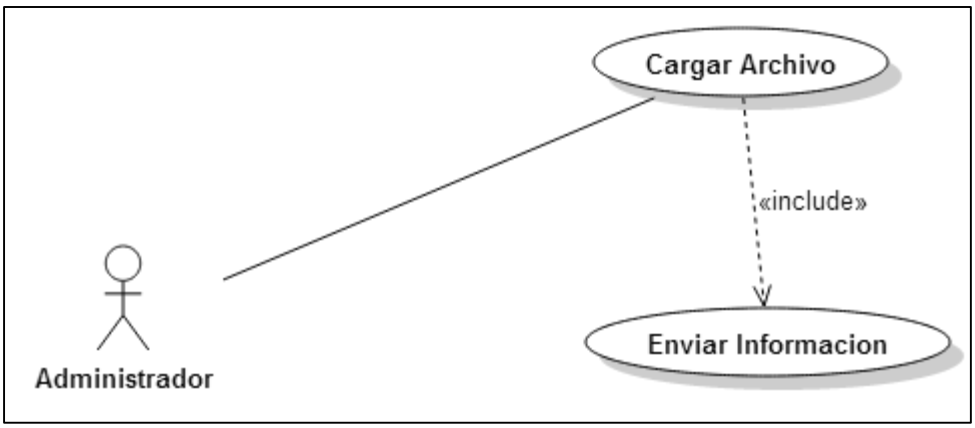


Figura 13: Diagrama UML de Casos de Uso Prototipo de Firewall

Fuente: Autores del proyecto

Donde no hay restricción ni acciones específicas para un usuario determinado, porque el mismo usuario puede navegar y hacer uso de todo el prototipo.

4.2.1.3 Especificación de casos de uso

A continuación se describe detalladamente cada caso de uso por medio de plantillas donde se especifican cómo interactúa el actor con el sistema.

4.2.1.3.1 Caso de uso logueo en el sistema. Breve descripción

Este caso de uso permite al administrador CARGAR el ARCHIVO en donde se encuentra la información oficial de eventos transcurridos en un rango de tiempo llamados comúnmente como archivos .logs.

Actores:

Administrador.

- **Precondiciones**

El administrador debe haber iniciado la aplicación dentro de un servidor local dentro de la maquina a utilizar.

- **Flujo de eventos**

Este caso de uso comienza cuando el administrador desea CARGAR la información en el sistema.

- **Flujo Básico**

1. El sistema muestra una interfaz sencilla y en su parte inferior derecha una sección de dos cajas de texto; Cargar y Enviar.
2. El administrador hace clic en el botón de cargar.
3. El sistema muestra una ventana especificando el directorio en donde el archivo access.log se pueda hallar.
4. El administrador selecciona el archivo.
5. El sistema muestra una interfaz sencilla y en su parte inferior derecha una sección de dos cajas de texto; Cargar y Enviar.
6. El administrador selecciona el botón ENVIAR.
7. El sistema valida la información y muestra de forma detallada la información encontrada dentro del archivo.

- **Flujos Alternativos**

a. Sí, en el **Flujo Básico,** el administrador no logra encuentra el archivo access.log, debe remitirse a la carpeta de instalación del servidor de base de datos instalada en la máquina.

- **Requerimientos Especiales**

  Ninguno.

- **Post-condiciones**

  Una vez que el administrador haya cargado el archivo Access.log podrá verla información detallada de los eventos ocurridos en el servidor dentro de una rango de tiempo.

- **Puntos de extensión**

  Ninguno.

4.3 Lenguaje utilzado y clases php creadas

4.3.1 Buscador

El buscador se encuentra desarrollado en lenguaje PHP y cada clase es utilizada por cada una de las interfaces.

Las clases creadas son las siguientes:

Busqueda.php:

Se establece un método en donde se recibe la cadena que queremos buscar, luego una clase ID para recuperar el ID y pasarlo a otra página. Así mismo se encuentran las líneas en donde se pueden colocar las fotos recuperadas de la base de datos, el nombre y la dirección. (Obsérvese en la Figura)

```
<?php
include_once('conexion.php');
if($_POST){

$q=$_POST['palabra'];
$consultar = new conexion();
$consultar->consulta("SELECT * FROM principal.face WHERE nombre LIKE '%$q%'");
while($data = $consultar->extraerRegistro()){
$id=$data['id'];
$nombre=$data['nombre'];
$direc=$data['direccion'];
$foto=$data['url'];

?>
<a href="usuario_completo.php?id=<?php echo $id; ?>" style="text-decoration:none;" >
<div class="display_box" align="left">
<div style="float:left; margin-right:6px;"><img src="<?php echo $foto?>" width="60" height="60" /></div>
<div style="margin-center:6px;"><b><?php echo $nombre; ?></b></div>
<div style="margin-right:6px; font-size:14px;" class="desc"><?php echo $direc; ?></div></div>
</a>

<?php
}

}
else
{

}

?>
```

Conexion.php:

Se establece las variables del usuario, host, el nombre de la base de datos, el puerto a utilizar y la contraseña que se designó para poder establecer la conexión en el servidor. (Obsérvese en la Figura)

```php
<?php

date_default_timezone_set('America/Bogota');


class conexion
{
        private $conexion;
        private $resultado;

        function __construct()
        {
                $this->conectar_base_datos();
        }

         private function conectar_base_datos()
    {
                $user = "dolphin";
                $host = "127.0.0.1";
                $db = "face";
                $puerto = "5432";
                $pass= "dolphin2016*";

             $this->conexion = pg_connect("host=$host dbname=$db user=$user password=$pass port=$puerto")
                                                                        or die
                                                                        ("Ocurrio un error");
         }

          function consulta($consulta)
          {

                  $this->resultado=pg_query($this->conexion,$consulta) or die ("Ocurre algo en el servidor");
          }

          function extraerRegistro()
          {
                  if($fila = pg_fetch_array($this->resultado))
                  {
                          return $fila;
                  }
                  else
                  {
                          return false;
                  }
          }
```

Index.php:

Se establecen las variables a recibir, se crea el diseño con su respectiva caja de texto y se le da un diseño simple. (Obsérvese en la Figura)

```
<style type="text/css">

#caja_busqueda
{
width:400px;
height:25px;
border:solid 2px #979DAE;
font-size:16px;
}
#display
{
width:300px,
display:none;
overflow:hidden;
z-index:10;
border: solid 1px #666;
}
.display_box
{
padding:2px;
padding-left:6px;
font-size:18px;
height:63px;
text-decoration:none;
color:#3b5999;
}

.display_box:hover
{
background: #415AB5;
color: #FFF;
}
.desc
{
color:#666;
font-size:16;
}
.desc:hover
{
color:#FFF;
}
```

```
</style>

<script language="JavaScript" src="jquery-1.5.1.min.js"></script>
<!--<script language="JavaScript" src="jquery.watermarkinput.js"></script> -->

<script type="text/javascript">
$(document).ready(function(){

$(".busca").keyup(function(){
var texto = $(this).val();
var dataString = 'palabra='+ texto;

if(texto==''){

}else{

$.ajax({
type: "POST",
url: "busqueda.php",
data: dataString,
cache: false,
success: function(html){
$("#display").html(html).show();
}
});

}
return false;
});
});

</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Facebook</title>
</head>

<body>
<form>
  <div style=" width:240px; " >
     <input type="text" class="busca" id="caja_busqueda" name="clave" placeholder="Buscar amigos, personas y lugares"/><br />
  </div>
<div id="display"></div>
</form>
<p>
</body>
</html>
```

Usuario completo.php:

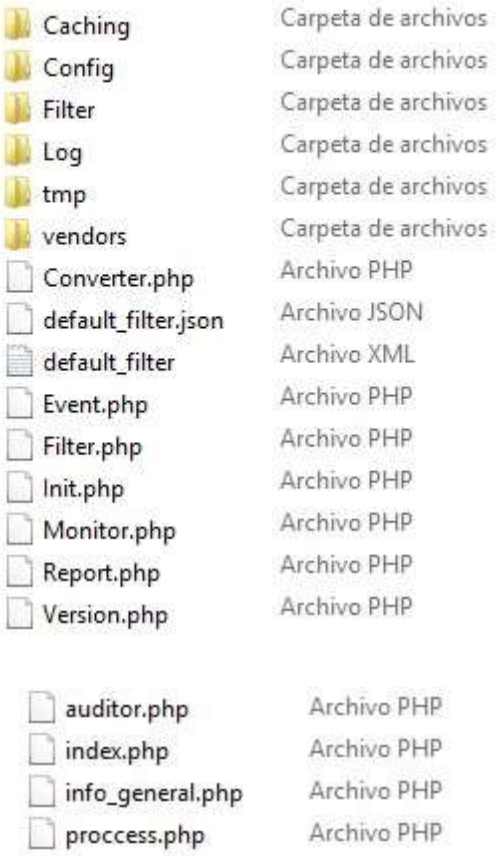
Se establece este método para recibir el usuario en el buscador. (Obsérvese en la figura)

```php
<?php
$id=$_REQUEST["id"];
echo $id;
?>
```

4.3.2 Prototipo de firewall

El prototipo de firewall se encuentra desarrollado en lenguaje PHP y cada clase es utilizada por cada una de las interfaces.

Las clases creadas son las siguientes:



Auditor.php:

Esta clase se encarga de estar revisando la información dentro del archivo de incidencias del servidor. (Obsérvese en la figura)

```
<?php

require_once 'IDS/Init.php';

try {
    $init = IDS_Init::init('IDS/Config/Config.ini.php');
    $init->config['General']['base_path'] = 'IDS/';
    $init->config['General']['use_base_path'] = true;
    $init->config['Caching']['caching'] = 'none';
} catch (Exception $e) {
    printf('An error occured: %s',$e->getMessage());
    exit();
}

$fdesc = fopen($_FILES['log']['tmp_name'], 'r');
if(!$fdesc) {
    echo "<p>Error al abrir el fichero de log</p>";
    exit();
}

$linea = fgets($fdesc);
$nlinea = 0;
while( $linea ) {

    $nlinea++;

    $patron = "/^([^ ]+) ([^ ]+ [^ ]+ \[([^\]]+)\] "([^ ]+ [^ ]+ [^ ]+)" ([^ ]+) ([^ ]+) "([^"]+)" "([^"]+)"$/";
    $matches = array();
    preg_match($patron, $linea, $matches);

    if(count($matches) != 8) {
        echo "<p>Error al parsear la linea $linea</p>\n";
        $linea = fgets($fdesc);
        continue;
    }

    $ipaddr = $matches[1];
    $datetime = $matches[2];
    $httpreq = stripslashes($matches[3]);
    $status = $matches[4];
    $reqlen = $matches[5];
    $referer = stripslashes($matches[6]);
    $useragent = stripslashes($matches[7]);
```

```
if(count($matches) != 4) {
    echo "<p>Error al parsear la solicitud HTTP en la linea $nlinea</p>\n";
    $linea = fgets($fdesc);
    continue;
}

$method = $matches[1];
$resource = $matches[2];
$version = $matches[3];

$partes = explode('?', $resource, 2);
if(count($partes) == 2) {
    $uri = $partes[0];
    $params = explode('&', $partes[1]);
    $get = array();

    foreach($params as $param) {
        $aux = explode('=', $param, 2);
        if( count($aux) == 2) {
            $varname = urldecode($aux[0]);
            $varvalue = urldecode($aux[1]);
            $get[$varname] = $varvalue;
        }
    }

    $ids = new IDS_Monitor($get, $init);
    $result = $ids->run();


    if (!$result->isEmpty()) {
        echo "<div style=\"font-size:0.8em;font-family:sans-serif;border:solid 1px #AAA;padding:10px;margin:10px;\">\n";
        echo "<h3 style=\"color:red;\">Ataque Detectado</h3>\n";
        echo "<p>\n";
        echo "<b>IP de origen:</b> $ipaddr<br/>\n";
        echo "<b>Fecha/Hora:</b> $datetime<br/>\n";
        echo "<b>Recurso:</b> " . htmlspecialchars($resource) . "<br/>\n";
        echo "<b>C&oacute;digo de respuesta:</b> $status<br/>\n";
        echo "<b>User-Agent:</b> " . htmlspecialchars($useragent) . "<br/>\n";
        echo "<b>Referer:</b> " . htmlspecialchars($referer) . "<br/>\n";
        echo "<b>Nro. de L&iacute;nea:</b> $nlinea<br/>\n";
        echo "</p><p><b>Detalle:</b><br/>\n";
        echo $result;
        echo "</p></div>\n";
    }
```

Index.php:

Dentro de esta clase se crea el diseño del prototipo de firewall y se establecen las variables a mostrar. (Obsérvese en la figura)

```
<!DOCTYPE html>
<html lang="es">
    <head>
        <meta name="viewport" content="width=device-width,initial-scale=1,maximun-scale=1">
        <meta charset="utf-8">
        <link type="text/css" href="css/estilos.css" rel="stylesheet">
        <link href='https://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css
        <script type="text/javascript" src="js/funciones.js"></script>
        <title></title>
    </head>
    <body>
        <div class="menu">
            <a href="#"><img src="img/btn-menu1.png"></a>
            <h4><a href="#">Acerca de</a></h4>
        </div>
        <form method="post" name="" action="" target="">
            <div class="datos">
            <div class="titulo1"><h4>Info-General</h4></div>
            <div class="ti-contenido">
              <!-- cargamos la info de info_general.php -->
              <div id="info_general"></div>
                <div class="contenedor1">
                    <!--div class="titulo1 espacio1"><h4>Top Procees</h4></div-->
                <!--div class="info-procees">
                    <div id="proccess_system" class="valor-proceso "><p></p></div>
                </div-->
                </div>
            </div>
        </div>

        <div class="tabla">
            <div class="contenido">
                <div class="atributo" id="color1"><h5>Ip-Origen</h5></div>
                <div class="atributo" id="color2"><h5>Fecha-Hora</h5></div>

            </div>
            <div class="contenido">
                <div class="atributo" id="color3"><h5>Recurso</h5></div>
                <div class="atributo" id="color4"><h5>Codigo-Registro</h5></div>


            </div>
            <div class="contenido">
                <div class="atributo" id="color1"><h5>User-Agent</h5></div>
                <div class="atributo" id="color2"><h5>Referencia</h5></div>
```

```
$linea = fgets($fdacc);
$nlinea = 0;
while( $linea ) {

    $nlinea++;

    $patron = '/^([^ ]+) ([^ ]+ [^ ]+ \[([^\]]+)\] "([^ ]*) ([^ ]*) [^ ]*" ([^ ]+) ([^ ]+) "([^"]*)" "([^"]*)"$/';
    $matches = array();
    preg_match($patron, $linea, $matches);

    if(count($matches) != 8) {
        echo "<p>Error al parsear la linea $nlinea</p>\n";
        $linea = fgets($fdacc);
        continue;
    }

    $ipaddr = $matches[1];
    $datetime = $matches[2];
    $httpreq = stripslashes($matches[3]);
    $status = $matches[6];
    $resplen = $matches[5];
    $referer = stripslashes($matches[8]);
    $useragent = stripslashes($matches[7]);


    $patron = '/^([^ ]+) ([^ ]+) ([^ ]+)$/';
    $matches = array();
    preg_match($patron, $httpreq, $matches);

    if(count($matches) != 4) {
        echo "<p>Error al parsear la solicitud HTTP en la linea $nlinea</p>\n";
        $linea = fgets($fdacc);
        continue;
    }

    $method = $matches[1];
    $recurso = $matches[2];
    $version = $matches[3];

    $partes = explode('?', $recurso, 2);
    if(count($partes) == 2) {
        $uri = $partes[0];
        $params = explode('&', $partes[1]);
        $get = array();
```

Info_general.php:

En esta clase se establecen las variables asociadas a los procesos del servidor que está funcionando en segundo plano. (Obsérvese en la figura)

```
<?php
        function get_server_memory_usage(){
        $free = shell_exec('free');
        $free = (string)trim($free);
        $free_arr = explode("\n", $free);
        $mem = explode(" ", $free_arr[1]);
        $mem = array_filter($mem);
        $mem = array_merge($mem);
        $memory_usage = $mem[2]/$mem[1]*100;

        return $memory_usage;
}

function get_server_cpu_usage(){
        $load = sys_getloadavg();
        return $load[0];

}

function get_uptime(){
$get_uptime = file_get_contents('/proc/uptime');
$uptime = explode(' ', $get_uptime);
$uptime_days = floor($uptime[0] / $84000);
$uptime_hours = floor(($uptime[0] / 3600) % 24);
$uptime_minutes = floor(($uptime[0] / 60) % 60);
$uptime_seconds = ($uptime[0] % 60);

echo $uptime_hours.' H: '.$uptime_minutes.' M: '.$uptime_seconds.' S';
}


?>
<div class="info"><div class="titulo-info1"><h5>Equipo</h5></div><div class="titulo-info2"><p><?=gethostname(); ?></p></div></div>
<!--div class="info"><div class="titulo-info1"><h5>Memoria</h5></div><div class="titulo-info2"><p><!-?=get_server_memory_usage() ?>%</p></div></div>
<div class="info"><div class="titulo-info1"><h5>Uso-CPU</h5></div><div class="titulo-info2"><p><!?=get_server_cpu_usage()?>%</p></div></div>
<div class="info"><div class="titulo-info1"><h5>Uptime:</h5></div><div class="titulo-info2"><p><!?=get_uptime()?></p></div></div!-->
```

Process.php:

En esta clase se analiza los procesos ejecutados por el servidor. (Obsérvese en la figura)

```php
<?php

function check_process(){
        exec("top -d 0.5 -b -n 1 | tail -n 17 | awk '{print $12}'", $top, $error );
        echo nl2br(implode("\n",$top));
        if ($error){
        exec('/usr/bin/top n 1 b 2>&1', $error );
        echo "Error: ";
        exit($error[0]);
                }
        }

        function check_process_max_consum(){
                exec("ps aux --width 30 --sort -rss | head | tail -n 15 | awk '{print $11}'",$top2 , $error);
                echo nl2br(implode("\n",$top2));
                if ($error){
                exec('/usr/bin/top n 1 b 2>&1', $error );
                echo "Error: ";
                exit($error[0]);
                        }
                }


?>
<div id="proccess_system" class="valor-proceso "><p><?= check_process(); ?></p></div>
```

4.4 Configuración del servidor y el cliente.

Un servidor proxy es aquel que sirve de intermediario entre el internet y las computadoras de que conforman una red, constituye una herramienta de seguridad que permite vulnerabilidades que se presenten en una red de comunicaciones. Un proxy permite a otros equipos conectarse a una red de forma indirecta a través de él.

Cuando un equipo de la red desea acceder a una información o recurso, es realmente el proxy quien realiza la comunicación y a continuación traslada el resultado al equipo final. En unos casos esto se hace así porque no es posible la comunicación directa y en otros casos porque el proxy añade una funcionalidad adicional, como puede ser la de mantener resultados obtenidos en una cache que permita acelerar sucesivas consultas coincidentes.

Para la utilización de nuestro prototipo se usó del siguiente servidor proxy multiplataforma y los siguientes requerimientos del sistema:

Linux

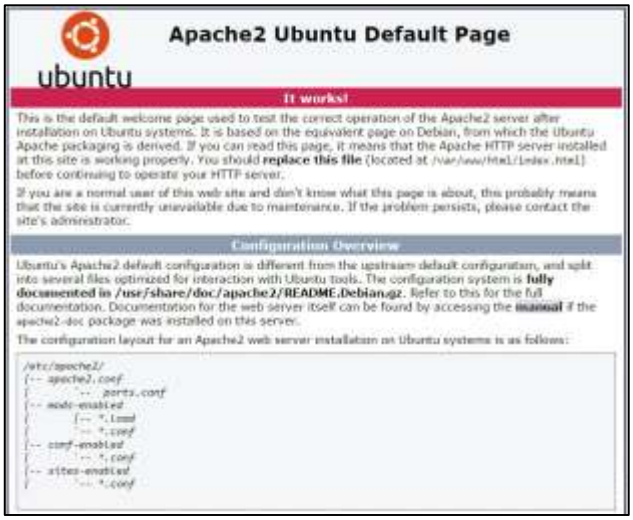


Figura 14: Servidor proxy linux

Fuente: Autores del proyecto.

Requerimientos del sistema.

Mínimo 64 MB de RAM.

Mínimo 50 MB de disco duro, para la instalación, tener en cuenta que las páginas web que usted vaya a cargar posteriormente no están calculadas en este mínimo de memoria.

CONFIGURACIÓN DE LA CONEXIÓN PUNTO A PUNTO ENTRE LINUX Y WINDOWS

CONFIGURACIÓN EN LINUX

Para realizar la configuración en el sistema operativo de la distribución Linux, se realizaron los siguientes pasos:

Entramos a las conexiones de red y seleccionamos solo enlace local en ajustes de IPv6.

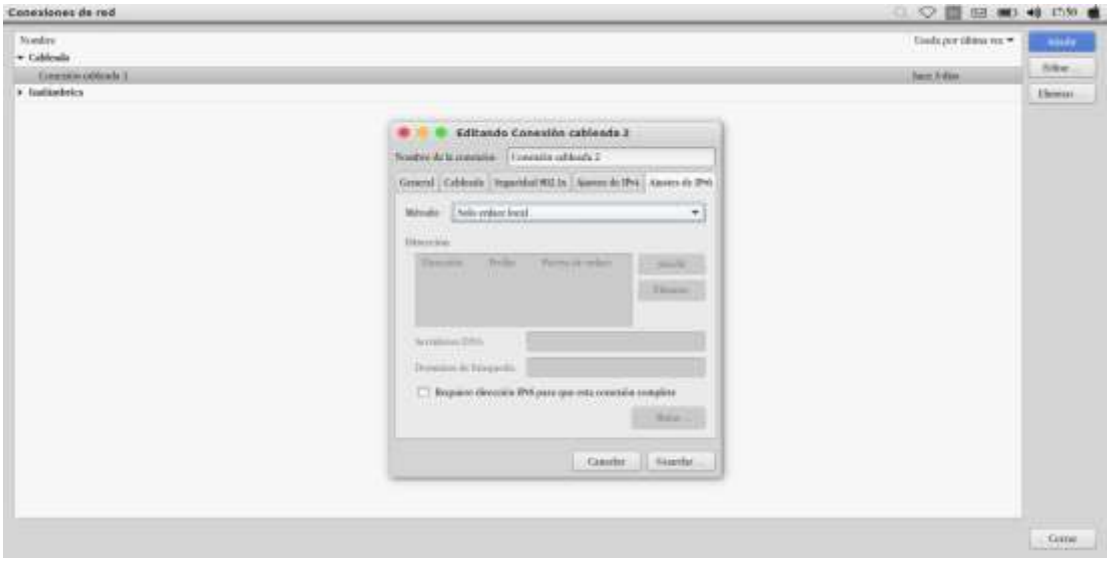


Figura 15: Configuracion de Red en Linux

Fuente: Autores del proyecto.

Elegimos el tipo de conexión que para nuestro caso sera cableada.

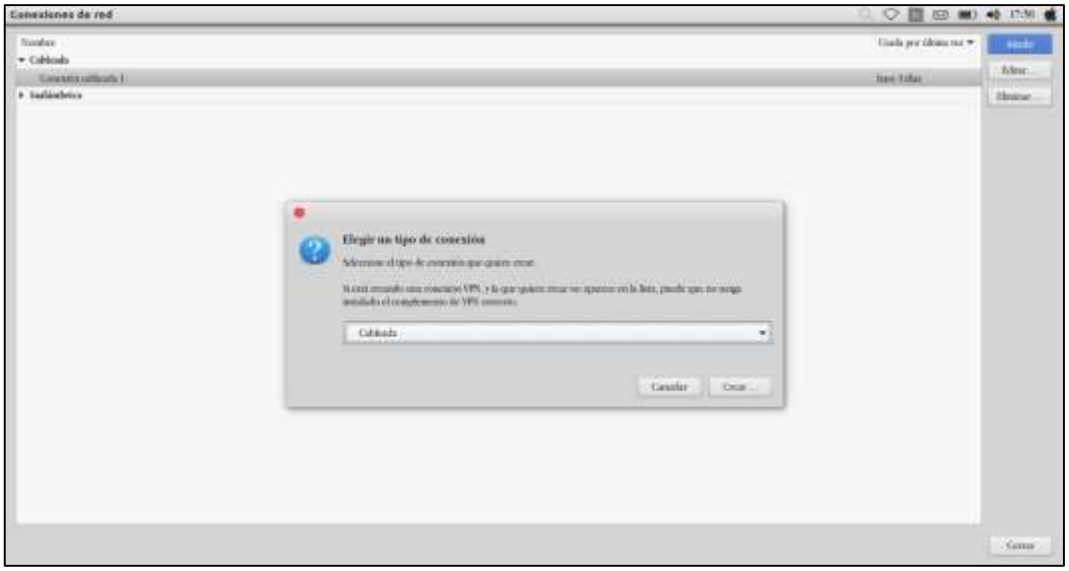


Figura 16: Configuracion de tipo de onexion de red en linux

Fuente: Autores del proyecto.

Ingresamos nuestra IP 1080:0:0:0:8:200c:4171, el prefijo 64 y opcion guardar.

Figura 17: Configuracion de IPv6 en linux.

Fuente: Autores del proyecto

CONFIGURACIÓN DE WINDOWS

Para realizar la configuración en el sistema operativo de la distribución Windows, se realizaron los siguientes pasos:

Nos dirigimos a conexiones de red, seleccionamos Conexión de área local, y propiedades.

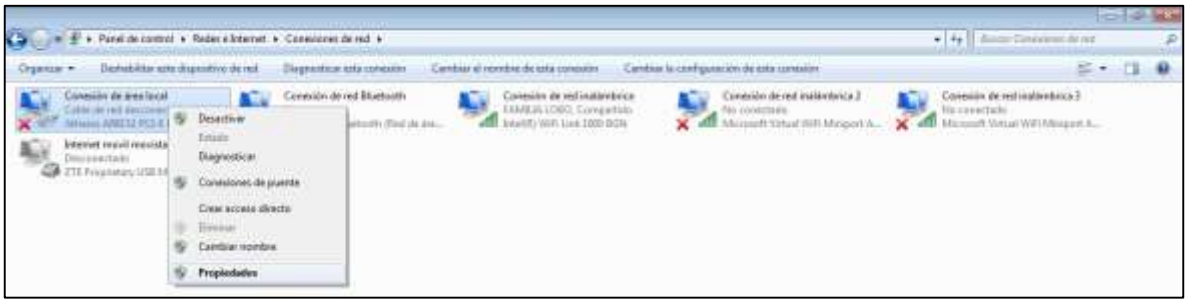


Figura 18: Configuracion de red en windows.

Fuente: Autores del proyecto.

Activamos el protocolo versión 6 (TCP/IPv6) y seleccionamos propiedades.

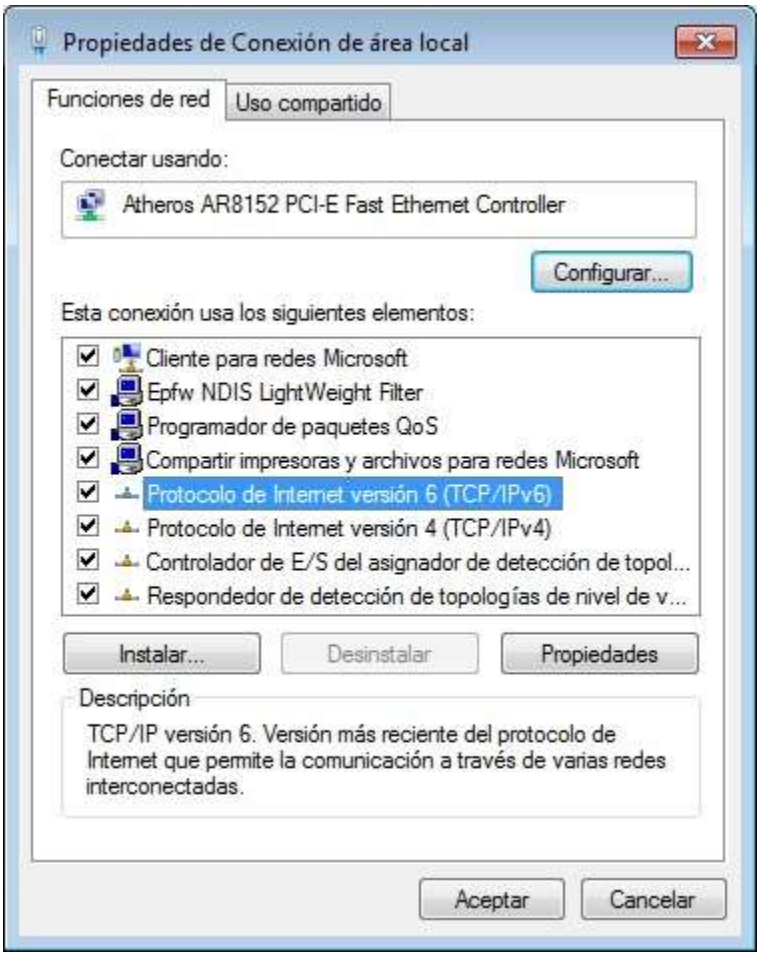


Figura 19: Activacion del protocolo IPv6 en windows.

Fuente: Autores del proyecto.

Configuramos nuestra IP 1080:0:0:0:9:200c:4171

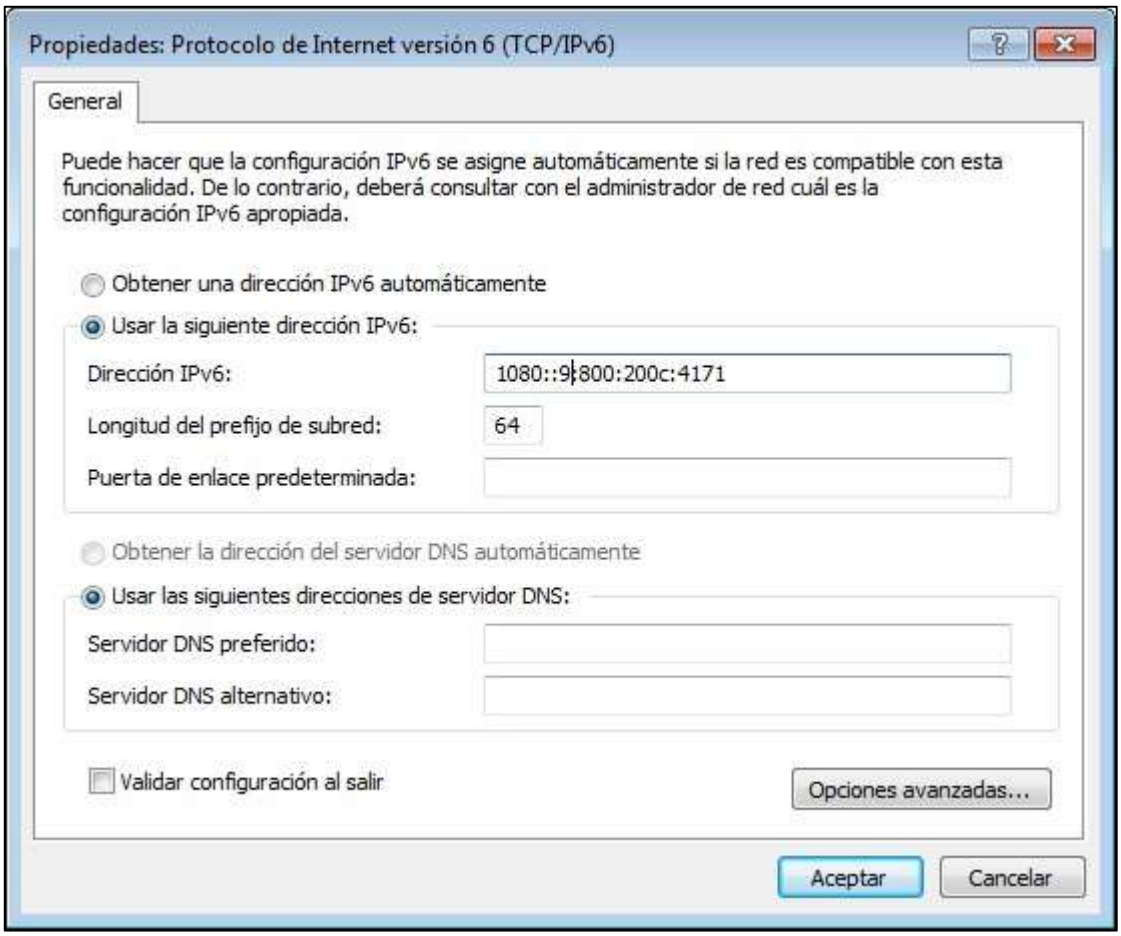


Figura 20: Configuracion de IPv6 en windows.

Fuente: Autores del proyecto

Luego de realizados los pasos siguientes como en el servidor que en nuestro caso será el Sistema Operativo de distribucion Linux y nuestro cliente o computador atacante que en nuestro caso será el computador de sistema Operativo de distribucion Windows comprobamos la conexión entre ellos con los siguientes pasos:

Para comprobar que hay conexion hacemos uso del programa CMD de cada uno de los computadores y realizamos ping entre los dos para comprobar las estadisticas entre la conexion.

Ping de Linux a Windows

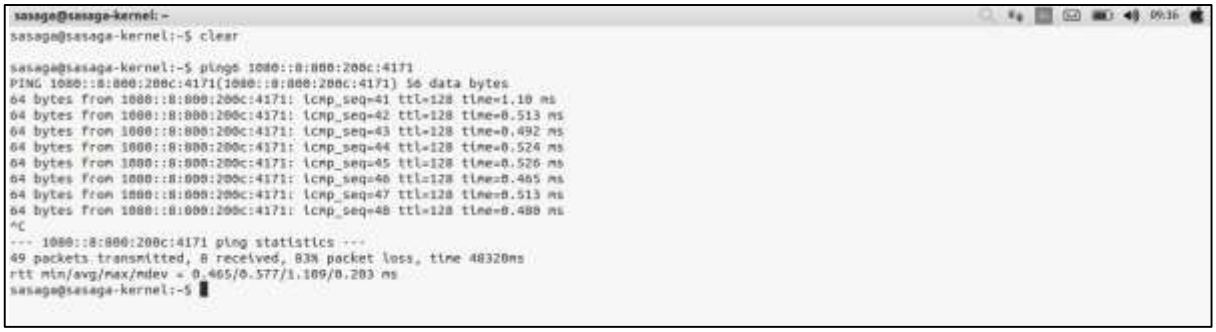


Figura 21: Ping desde Servidor linux

Fuente: Autores del proyecto.

Ping de Windows a Linux

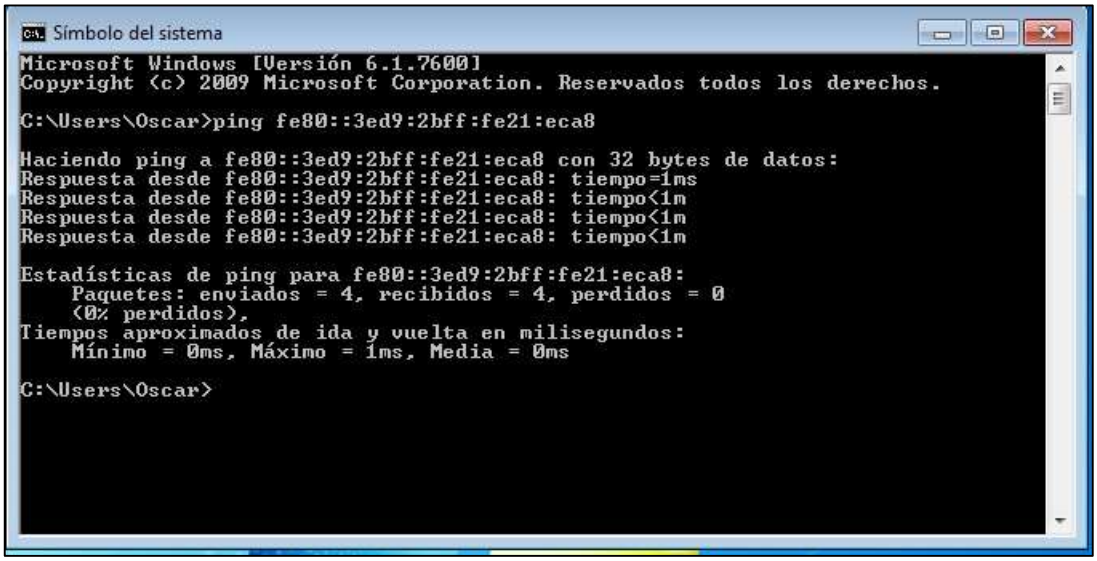


Figura 22: Ping del Cliente en windows

Fuente: Autores del proyecto.

COMO ACCEDER DESDE EL PC CON WINDOWS AL SERVIDOR QUE ESTÁ ALOJADO EN LINUX?

Para hacer esto debemos colocar en nuestro buscador la IP dentro de corchetes [] seguido de un / con el nombre del proyecto al que queremos acceder. Nos quedaría de la siguiente manera. [IP]/tes

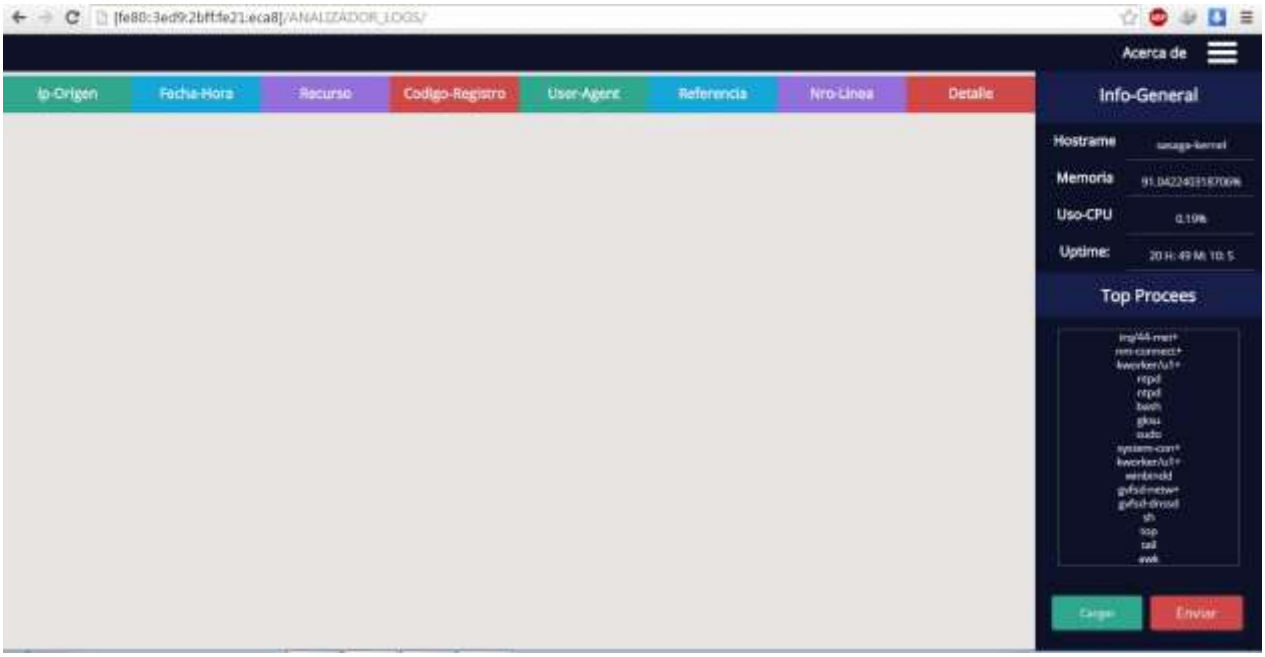


Figura 23: Interfaz del Prototipo de Firewall en el Servidor.

Fuente: Autores del proyecto.

4.5 Análisis de incidencias.

A continuación, se presentara las interfaces a través de las cuales se puede visualizar como es el funcionamiento del Prototipo de Firewall.

Para nuestro caso, creamos un simple buscador y dentro de él, incorporamos errores de programación las cuales nos servirían para realizar los ataques necesarios y comprobar que nuestro prototipo esté funcionando correctamente.

Ejecutamos el buscador de la siguiente manera.

Figura 24: Interfaz del buscador.

Fuente: Autores del proyecto.

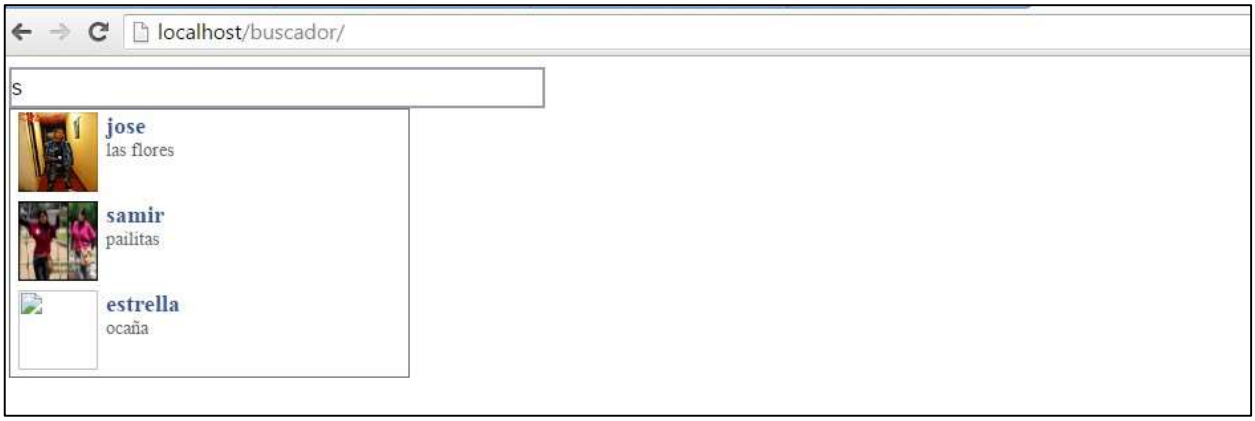


Figura 25: Interfaz del buscador de prueba.

Fuente: Autores del proyecto.

4.6 Pruebas

En esta sección, se procederá a realizar las pruebas necesarias para ver el funcionamiento del prototipo de firewall.

Se ejecuta la sentencia "**SELECT*FROM**" dentro del buscador. Ya colocando la palabra SELECT e prototipo de firewall será capaz de detectar el tipo de ataque y detallarlo dentro de su interfaz.

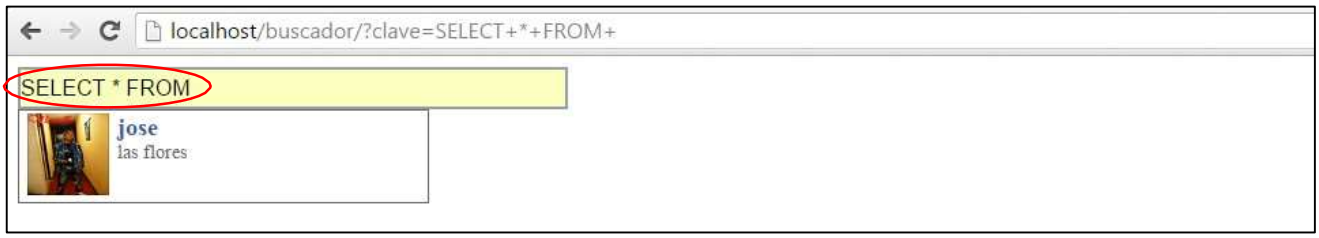


Figura 26: Primer Ataque al Buscador.

Fuente: Autores del proyecto.

Así mismo se ejecuta los siguientes ataques

Figura 27: Segundo Ataque al Buscador.

Fuente: Autores del proyecto.

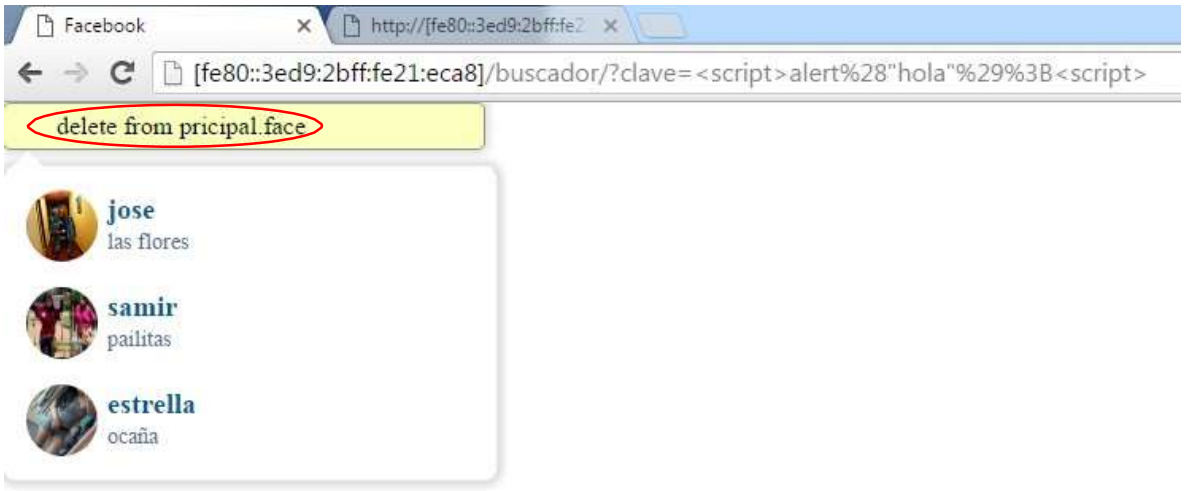


Figura 28: Tercer Ataque al Buscador

Fuente: Autores del proyecto.

Figura 29: Cuarto Ataque al Buscador.

Fuente: Autores del proyecto.

Figura 30: Quinto Ataque al Buscador.

Fuente:Autores del proyecto.

Luego de haber realizado los pasos especificados en el capítulo 4.2 Diseño de interfaz en la sección cargue de archivos para cargar el archivo Access.log del servidor, nuestro prototipo identifica los ataques y visualiza toda esa información de la siguiente manera:

Figura 31: Informacion de Ataques.

Fuente: Autores del proyecto.

Figura 32: información de ataques

Fuente: Autores del proyecto.

Capítulo 5. Recolección de información

El tipo de recolección de información se realizó de la siguiente manera; La técnica a utilizar fue las Sesiones de grupo la cual utiliza formas de estudios cualitativos y su técnica se basa en la reunión de un grupo de personas para indagar acerca de actitudes y reacciones frente a un producto, servicio, concepto, publicidad, idea o empaque. Así mismo dentro de esa gran técnica se utilizó el método de sesiones con moderador dual, en donde constó de dos participantes; encargándose uno de desarrollar la sesión de manera suave y confortable, mientras que el otro se aseguró de que se toquen todos los puntos predefinidos.

Capítulo 6. Conclusiones.

Al terminar el presente trabajo, se ha llegado a las siguientes conclusiones:

La seguridad en las redes de comunicación, se ha convertido en un factor de relevancia al momento de diseñar nuevos software, porque cada vez se busca dar robustez y protección a la información manejada por el usuario sistemas para lograr tener un verdadero control de intrusos a sus host, que hoy en día es el anhelo de cualquier organización.

En la actualidad, no existe solución técnica frente a un ataque, por lo tanto las infraestructuras de internet representan el punto más débil de la seguridad global en un sistema. Para esto se debe conocer como es el verdadero funcionamiento de un firewall, para identificar sus puntos críticos y adherir políticas de seguridad que los fortalezcan.

El uso de un firewall adecuado evita ataques y desastres que pueden ocasionar intrusos, males intencionados que intentan dañar o usufructuar la información contenida dentro de un host.

El manejo de interfaces amigables para el usuario mejoran la perspectiva y el funcionamiento de sistemas de seguridad; siendo mucho más atractivos para quienes los utilizan y menos confusos.

Capítulo 7. Recomendaciones.

Al terminar el presente trabajo, se ha llegado a las siguientes recomendaciones:

Es una buena práctica realizar la revisión continua de los reportes o logs generados para detectar situaciones que puedan ser prevenidas a tiempo.

La seguridad de la red no se concentra en el firewall, aunque es parte importante de la misma, sino en una política de seguridad coherente que se adapte a la organización y su misión, en la que se tome la red como un todo y no como sub-sistemas independientes que dependen de un firewall para protegerse.

La implementación de un firewall debe estar atada a los contantes cambios que pueden ocurrir dentro de una red.

De acuerdo al primer prototipo desarrollado del firewall y aprovechando los reportes de incidencias que este genera, se puede comenzar a desarrollar en futuros proyectos todo el proceso del filtrado de tráfico que permita denegar o permitir las diferentes peticiones que se hagan sobre el servidor, ya que gracias a este reporte, el administrador de la red podrá implementar controles bajo reglas que le permitan endurecer la seguridad del servidor.

Referencias

A.S., Y. (2003). *Analysis of different technique for detection of SQL injection in Proceedings of the internacional Conference & working on Emerging Trends in Technology.* Retrieved from http://dx.doi.org/10.1145/1980022.1980229

Abadi, A. (2011). *Code- motion for API migration fiying SQL injection vulnerabilities in Java.* Retrieved from http://dx.doi.org/10.1145/1984732.1984734

Alkhashab, E. (2011). *Preventing SQL injection attacks based on query optimization process.* Retrieved from http://dx.doi.org/10.1145/2107556.2107566

Alto, P. (2014). *Resumen del Firewall de nueva generacion.* Retrieved from http://www.xnetworks.es/contents/PaloAltoNetworks/Firewall_Feature_Overview_ESP_2011.pdf

Anonimo. (2016). *Retroinformatica.* Retrieved from Historia de internet: http://www.fib.upc.edu/retro-informatica/historia/internet.html.

Apache.org. (2013). *Apache.org*. Retrieved from https://httpd.apache.org/docs/2.0/es/logs.html

awstats.org. (n.d.). *awstats.org*. Retrieved from http://www.awstats.org/

Bohman, P. R. (2004). *An accesible method of hiding HTML .* Retrieved from http://dx.doi.org/10.1145/990657.990664

Bolivar. (2012). *Diseño e implementacion de una red IP6 transicion eficiente desde IPV4.* Retrieved from http://revistaingenieria.univalle.edu.co:8000/index.php/inycompe/article/view/320/387

Burgos, A. (2010). *Seguridad pc.* Retrieved from https://books.google.es/books?id=31lKLjo1JnQC&pg=PA177&dq=historia+del+firewall&hl=es&sa=X&ved=0CCAQ6AEwAGoVChMIh-KJ5uTWxwIVRHUeCh2MDgnP#v=onepage&q=historia%20del%20firewall&f=false

Congreso de Colombia. (1991). *Constitucion politica de Colombia.* Retrieved from https://books.google.es/books?id=GJG2EctxDjsC&pg=PA51&dq=ATAQUES+CON+CSS&hl=es&sa=X&ved=0CCEQ6AEwAGoVChMIgdaxoe_WxwIVRVYeCh3YWwJA#v=onepage&q&f=false

Contreras, J. L. (2011). *Redes de Computadores.* Retrieved from http://sib.ufpso.edu.co/det_titulo.php?no_control=19529&tipo_material=2&titulo=IMPLEMENTACION%20DE%20LA%20SEGURIDAD%20DEL%20PRTOCOLO%20DE%20INTERNET%20IPSEC%20EN%20REDES%20DE%20AREA%20LOCAL%20CON%20IPV6&tipo=algunos&ini_titulo=IPV6

Fogla, P. (2006). *Evolving computer intrusion scripts for vulnerability asessment and log analysis.* Retrieved from http://dx.doi.org/10.1145/1068009.1068331

Fortinet. (2001). *Productos.* Retrieved from www.fortinet.com/resource_center/solution_briefs/faster-firewalls-for-faster-networks.html

Google central de conversiones. (2009, septiembre 08). *Google central de conversiones*. Retrieved from http://central-de-conversiones.com.co/2009/09/breve-historia-del-analisis-web-y.html

Gorji, A. (2014). *Detecting Obfuscated Java Scrip Malware Using Sequences of Internal Function Calls.* Retrieved from http://dx.doi.org/10.1145/2638404.2737181

Graham, G. (2009). *Internet una indagacion filosofica .* Retrieved from https://books.google.es/books?id=uZr2GvcpeuwC&printsec=frontcover&dq=INTERNET&hl=es&sa=X&v ed=0CDEQ6AEwA2oVChMIjJv47eHWxwIVxh8eCh2-FAYJ#v=onepage&q=INTERNET&f=false

Huang, L. S. (2010). *Protecting browers from cross- origin CSS attacks .* Retrieved from http://dx.doi.org/10.1145/1866307.1866376

Jacome, B. A. (2009). *Propuesta investigativa de las aplicaciones y servicios para internet 2 en la universidad tecnica de Cotopaxi Ecuador.* Retrieved from http://repositorio.utc.edu.ec/bitstream/27000/1236/1/T-UTC-0861.pdf

loggly.com. (2016). *loggly*. Retrieved from https://www.loggly.com/

marlisq.wordpress.com. (2012, 08 29). *blog academico*. Retrieved from https://marlisq.wordpress.com/administracion_web/analizador-de-logs/

Mediana, C. C. (2013). *Caracterizacion del IPV6.* Retrieved from http://www.scielo.org.co/scielo.php?pid=S0123-921X2013000200010&script=sci_arttext

Messmer, E. (2014). *La evolucion del Firewall.* Retrieved from http://cioperu.pe/articulo/16839/la-evolucion-del-firewall/

Netweb, G. (2013). *Antivirus, firewall y anti- spyware; un tridente de seguridad e imprescindible.* Retrieved from http://www.gadae.com/blog/diferencia-entre-antivirus-firewall-y-anti-spyware/

Nuñez, G. (2007). *Ataques con CSS.* Retrieved from https://books.google.es/books?id=GJG2EctxDjsC&pg=PA51&dq=ATAQUES+CON+CSS&hl=es&sa=X&ved=0CCEQ6AEwAGoVChMIgdaxoe_WxwIVRVYeCh3YWwJA#v=onepage&q&f=false

PHP.net. (2001). *php.net*. Retrieved from http://php.net/manual/es/intro-whatis.php

Phung, P. (2009). *Lightweight self protecting Java Script.* Retrieved from http://doi.acm.org/10.1145/1533057.1533067

Pinillos, T. (2008). *Revista electronica de Estudios telematicos.* Retrieved from La nueva generacion IP: http://publicaciones.urbe.edu/index.php/telematique/article/view/766/1841.

Revista el Tiempo. (2014). Redaccion Tecnosfera. http://es.ccm.net/contents/268-protocolo-ipv6.

Ruiz, D. M. (2015, junio). Analizador de logs para detectar posibles accesos no deseados. Logroño, España: Universidad la Rioja, Servicio de publicaciones.

splunk.com. (2015). *SPLUNK*. Retrieved from http://www.splunk.com/es_es

Tin Can Comunicacion. (2001). *Estado del arte.* Retrieved from www.acceso.com

Vantiaute, N. (2016). *Historia del Protocolo IP.* Retrieved from http://es.ccm.net/contents/268-protocolo-ipv6

Wang, A. (2014). *POSTER: How Distributed are todays Dos Attaks.* Retrieved from http://dx.doi.org/10.1145/2660267.2662382