UNIVERSIDAD SAN FRANCISCO DE QUITO

Colegio de Ciencias e Ingeniería

Implementación de Seguridad en el Servidor de Hosting XpertSites.com

David Esteban Puente Guerrero

Fausto Pasmay, MSc. Director de Tesis

Tesis de grado presentada como requisito para la obtención del título de

Ingeniero en Sistemas

Quito, 21 de enero 2013

UNIVERSIDAD SAN FRANCISCO DE QUITO

Colegio de Ciencias e Ingeniería

HOJA DE APROBACIÓN DE TESIS

Implementación de Seguridad en el Servidor de Hosting XpertSites.com

David Esteban Puente Guerrero

Fausto Pasmay, MSc .....

Director de Tesis

José Luis Medina, Ing. .....

Miembro del Comité de Tesis

Fausto Pasmay, MSc .....

Director de Ingeniería de Sistemas

Ximena Córdova, PhD .....

Decana de la Escuela de Ingeniería

Colegio de Ciencias e Ingeniería

Quito, 21 de enero 2013

Firma:

Nombre: David Esteban Puente Guerrero

C. I.: 171262011-9

Fecha: 21 de enero 2013

Dedicatoria

Dedico la presente tesis a mis padres por el inmenso apoyo que me han dado para seguir esta carrera de ingeniería y culminarla. También le dedico a mi Dios por esta hermosa oportunidad de vida que tengo.

RESUMEN

Para la actual Tesis se ha propuesto el siguiente caso de estudio práctico. Tenemos un Servidor de Hosting con la siguiente dirección: XpertSites.com, en el cual se realiza desarrollo de aplicaciones y hospedaje de sitios web con sus bases de datos. El servidor a través de su sistema Snort, actualmente reporta recibir constantes escaneos de vulnerabilidades e intentos de intrusiones desde el Internet, razón por la cual el servidor requiere mejoras en seguridad para reducir así la probabilidad de una intrusión fructuosa al sistema. Se plantea como hipótesis que la solución a este problema, es la combinación entre la realización de correctivos de seguridad y la posibilidad de implementar un sistema automatizado de detección y respuesta a intrusiones basados en los ataques mas frecuentes que ha tenido el servidor. Para mejorar la seguridad del servidor, se evaluará el estado actual en seguridad, detectando así sus necesidades en seguridad a cubrir y elaborando las políticas de seguridad, a través de la creación de un “Plan de Seguridad”.

ABSTRACT

For the actual Thesis, the following case study has been proposed. We have a Web Hosting Server on the following address: XpertSites.com, which has been used for web application development and website hosting with its databases. The server through its intrusion detection system called Snort, actually reports to receive vulnerability scans and intrusion attempts from the Internet, reason for which the server requires security improvements to reduce the probability of a successful intrusion to the system. As a hypothesis, has been proposed the idea that the solution to this problem, is the combination between security corrective measures and the possibility of implementing an automated intrusion detection and response system based on the frequent attacks that the server has received. For improving security on the server, an evaluation of the actual state of security will be made, detecting its security needs and developing security policies through the creation of a "Security Plan".

ÍNDICE GENERAL

INDICE DE TABLAS

INDICE DE FIGURAS

CAPITULO 1: Introducción

1.1 Síntesis

La siguiente Tesis buscará reducir el riesgo de intrusión al Servidor XpertSites.com, buscará detectar los componentes críticos del servidor y protegerlos a través de correctivos de seguridad y con la posible implementación de un Sistema Automatizado de Detección y Respuesta a Intrusiones.

Para esto se evaluará el estado actual de la seguridad del servidor, se encontrarán sus necesidades críticas en seguridad, se procederá a documentar estos hallazgos y a convertirlos en políticas de Seguridad a seguir, creándose el Plan de Seguridad.

1.2 XpertSites.com

El servidor XpertSites.com hace ya algunos años atrás en un inicio fue creado con la finalidad de crear sitios web para empresas y posicionarlas en Internet. XpertSites comenzó y se ha mantenido como un negocio familiar, donde se juntaron talentos en el área de informática, marketing y ventas. Es decir XpertSites no es una empresa grande con cientos de empleados, al contrario, es un negocio familiar donde el mismo especialista en IT hace un trabajo personalizado a sus clientes de confianza. En ese proceso se ha asesorado a compañías pesqueras, compañías mineras, hasta artistas plásticos y organizaciones sin fines de lucro para su posicionamiento en Internet.

Con el pasar del tiempo, XpertSites se concentró mas en el desarrollo de

tecnología web para sitios del mercado de Bienes Raíces en Ecuador. Entrando específicamente a la investigación para el desarrollo de estas aplicaciones en el framework Django basado en Python. De esta manera con el tiempo fue delegando algunas cuentas de sus clientes a otros proveedores, para concentrarse principalmente en el desarrollo de aplicaciones de Bienes Raíces en Django.

Los clientes de los sitios web alojados principalmente suben sus archivos, actualizan su información y administran su cuenta, únicamente a través del panel de control de los manejadores de contenido Wordpress y Django Mezzanine, es decir sus interacciones están relacionadas únicamente a lo que se puede realizar dentro de un manejador de contenidos, esto comprende que los clientes, no acceden a la consola de Linux, ni realizan tareas de administración en Linux como creación de bases de datos, instalación de nuevas aplicaciones en el servidor, o modificación de archivos importantes de configuración del servidor, muy rara vez solicitan un acceso FTP y cuando lo hacen se lo realiza temporalmente, de la misma manera en el caso de un acceso SSH, todo este trabajo técnico que tenga que ver con configuraciones en el servidor lo hace el administrador del mismo. En base a esta realidad del presente caso de estudio se espera evaluar la seguridad del servidor e implementar mejorías en su seguridad.

En cuanto al sistema operativo, el servidor está funcionando con una distribución de Linux denominada Debian Lenny versión 5.0. Su ubicación física es en el exterior, actualmente se encuentra en el país de Panamá y ha sido rentado a una empresa que se dedica al alquiler de servidores web, por lo que el acceso al servidor se lo llevará acabo remotamente.

1.2.1 Componentes Críticos del Servidor XpertSites.com

Les llamaremos componentes críticos u objetivos de seguridad principales a aquellos servicios del Servidor que son básicos para que mantenga su funcionamiento actual como Servidor de Hosting. Básicamente se evaluará la seguridad de los siguientes componentes críticos:

- Apache Servicio Web
- BIND Servicio DNS
- Dovecot IMAP / POP3 Servicio de Correo
- Servicio FTP
- Servicio SSH
- MySQL Servicio de Base de datos

Un análisis mas detallado de estos componentes y su real importancia será analizada en la elaboración del Plan de Seguridad.

1.2.2 Topología del ambiente de Red del Servidor

El servidor se encuentra conectado directamente a un Data Center. Dentro del mismo servidor, corre el servicio de DNS, Correos, FTP, SSH. Por lo tanto no existen servidores aparte conectados para ofrecer este servicio. El análisis de tráfico será únicamente el que tenga que ver como destino al Servidor de Hosting.

1.3 Sistema Automatizado de Detección de Intrusiones Snort

Dentro del servidor se tiene instalado un Sistema de Detección de Intrusiones llamado Snort en su versión 2.7.0. con la finalidad de ir observando los intentos de intrusiones que ha venido teniendo . El sistema ha estado grabando las últimas alarmas de intrusiones que se han activado los últimos cuatro meses anteriores al inicio de este estudio, información que servirá para analizar los ataques más frecuentes que se han suscitado. Actualmente Snort se encuentra funcionando únicamente como un sistema para la detección de intrusiones (IDS) mas no como un sistema de respuesta a intrusiones (IRS).

1.4 Sistema Automatizado de Respuesta a Intrusiones

En este caso particular, en la actualidad el Servidor XpertSites.com no cuenta con un sistema (IRS), por lo tanto se ha pensado en la posibilidad de llegar a implementar un Sistema Automatizado de Respuesta a Intrusiones que trabaje en conjunto con Snort. Los sistemas de respuesta pueden responder a los ataques de diversas formas, una práctica comúnmente utilizada es bloquear la dirección ip del atacante. Para esto algunos sistemas trabajan en coordinación con un Sistema de Detección de Intrusiones, el cual le avisa a través de una alarma activada, al Sistema de Respuesta, para que realice una acción de bloqueo del ataque. Esta implementación tiene el riesgo de que mal podría el Sistema de Respuesta responder a una cantidad de falsas alarmas activadas, bloqueando direcciones ip o paquetes de datos inofensivos.

1.5 Metodología de Evaluación de la Seguridad Actual en el Servidor

Para la evaluación de como se encuentra la Seguridad actual del Servidor, se usará como guía, la metodología que propone el OSSTMM3 (Open Source Security Testing Methology Manual). La razón por la cual se ha escogido esta metodología es porque su enfoque es reduccionista, donde busca identificar claramente las Superficies de Ataque existentes, para reducirlas y luego concentrarse en las vulnerabilidades que éstas puedan tener debido a las operaciones que las mismas realizan. Su visión es buscar la seguridad desde el punto de vista de la Seguridad Operacional.

1.6 Evaluación de las Mejoras Implementadas en Seguridad

Se analizará el avance logrado en seguridad propuesto por este proyecto de Tesis, comparando las seguridades que existían antes y la seguridad que existe después de implementar el Plan de Seguridad.

1.7 Justificación e importancia del proyecto

La actual tesis nos permite descubrir cómo se puede llegar a mejorar la seguridad en un Servidor Linux de Hosting de un caso real, llegando a evaluar su seguridad para luego mejorarla implementando el Plan de Seguridad descubierto.

Es importante este proyecto porque logrará evaluar la seguridad de un caso real y permitirá llegar a comprender lo que se necesita para mantener en equilibrio la seguridad para el presente caso de estudio.

La seguridad en los servidores web, para todos sus involucrados es de eminente importancia. Servidores que se encuentran en el gobierno, en

universidades, escuelas, colegios, institutos, empresas, o que son parte de redes de servidores que ofrecen el servicio de hospedaje de contenido en Internet, todos ellos deben considerar cada vez sistemas de seguridad más efectivos, más aún cuando se almacena en estos servidores información sensible.

Esta Tesis nos permitirá comprender que mejorar la seguridad en un servidor de hosting, no es necesariamente implementar un Sistema Automatizado de Detección y Respuesta y que más bien para realizarlo hay que primeramente hacer una evaluación más precisa de la necesidades en seguridad, encontrar posibles soluciones, ver si es realmente factible este tipo de implementaciones y proceder a realizar las mejoras en seguridad.

1.8 Objetivos específicos del proyecto

- Evaluar el estado de la seguridad actual en el que se encuentra el servidor.
- Detectar los componentes críticos del servidor.
- Análisis del estado actual de las versiones de software de los principales componentes del servidor.
- Analizar los ataques mas frecuentes que ha venido teniendo el servidor.
- Analizar la posibilidad de implementar un Sistema de Respuesta de Intrusiones.
- Elaborar un “Plan de Seguridad” donde se trace el procedimiento de seguridad a implementarse.
- Fortalecer la seguridad del Servidor realizando correctivos en los componentes que lo necesitan.
- Documentar las implementaciones de seguridad realizadas.

CAPÍTULO 2: Fundamentos Teóricos

En la siguiente tesis, utilizaremos ciertas abreviaciones y nombres en inglés para su fácil identificación con la presente bibliografía de referencia la cual tiene su mayoría de textos en este idioma. Así mismo facilitará las búsquedas de contenido en Internet relacionados al tema usando la misma terminología en el idioma inglés, la cual es comúnmente usada en el medio informático.

Se discutirán en este capítulo algunos de los conceptos que dan sustento al tema principal de nuestra investigación.

2.1 Detección de Intrusiones

A continuación se explica las metodologías y tecnologías en detección de intrusiones que se han investigado para la presente Tesis.

2.1.1 Metodologías de Detección

En cuanto a las Metodologías de Detección de Intrusiones se conocen principalmente tres categorías: Detección basada en Firmas, Detección basada en Anomalías y Detección basada en el Análisis de Estado de un Protocolo. (Scarfone y Mell 2.3).

En la presente Tesis se utiliza la metodología de Detección basada en Firmas, que es lo que realiza el sistema Snort.

Detección basada en Firmas: en este contexto, una firma es un patrón que corresponde al procedimiento de una amenaza conocida. La Detección basada en firmas es el proceso de comparar firmas existentes con los eventos observados para identificar posibles incidentes (Scarfone y Mell 2.3.1).

Un ejemplo de lo mencionado anteriormente sería, la inyección de ciertos comandos de sql a través de un navegador web para irrumpir con la base de datos del servidor. La detección basada en firmas lo que haría es buscar en los paquetes de datos que llegan al servidor esos comandos sql y al encontrarlos activar la respectiva alarma en Snort.

Detección basada en Anomalías: es el proceso de comparar definiciones de lo que una actividad es considerada normal contra los eventos observados para identificar desviaciones significantes. Un SPI que usa este tipo de detección tiene registrado perfiles que representan el comportamiento normal de usuarios, servidores, conexiones de red o aplicaciones (Scarfone y Mell 2.3.2).

Un ejemplo de esto sería la comparación entre el consumo típico de ancho de banda de un servidor dentro de un rango de horas en el día contra el desfase de consumo que se detecte en ese mismo rango, pudiendo ser esto debido a algún ataque, levantando así el sistema una alarma.

Detección basada en el Análisis de Estado del Protocolo: es el proceso de comparar perfiles de definiciones generalmente aceptadas que corresponden a actividades de protocolo benignas contra eventos observados para identificar las desviaciones (Scarfone y Mell 2.3.3).

2.1.2 Tecnologías en Detección de Intrusiones

En cuanto a las tecnologías de Detección de Intrusiones, éstas se clasifican basándose en su funcionalidad, principalmente se dividen en las siguientes

tres categorías. NIDS (Network IDS), HIDS (Host IDS) y DIDS (Distributed IDS) (Esler 5).

NIDS (Network IDS): deriva de su nombre debido a que monitorea la red entera de su entorno, para esto su tarjeta de red funciona en modo promiscuo (Esler 5). Su objetivo es analizar el tráfico de todo el segmento de red objetivo. Un ejemplo de esto puede ser analizar todo el tráfico de red que entre y salga a un laboratorio específico de computación.

HIDS (Host IDS): difieren de los NIDS en dos formas. Los HIDS protegen solo el servidor en donde residen y su tarjeta de red opera en modo no promiscuo. En este modo de operación, solo los paquetes destinados para la tarjeta de red del servidor son analizados. Una ventaja de esto es que para el análisis de paquetes este modo no utiliza tanto CPU como el modo NIDS, esto por su análisis localizado. Otra ventaja es que se pueden escoger solo las reglas de seguridad que tienen que ver con el servidor en cuestión, alivianando el trabajo en el CPU y mejorando la organización del set de reglas (Esler 6).

También tiene que ver con los cambios que se den dentro del servidor en cuanto a los logs del sistema, acceso a archivos, procesos activos y cambios de configuración en las aplicaciones y en el sistema (Scarfone 9).

DIDS (Distributed IDS): se compone de sensores NIDS ubicados de forma remota, conectados a una consola de administración, donde se procesa la información recibida para ser analizada y a su vez para generar las alertas (Esler 7).

El actual proyecto debido a que se centra en las intrusiones que se dan en el servidor de hosting, el sistema de detección estaría bajo la categoría de un HIDS, monitoreando la actividad que entra y sale por la tarjeta de red del servidor.

2.1.3 Sistema de Detección de Intrusiones Snort

Snort es un network IDS capaz de realizar en tiempo real un análisis de tráfico y grabación de paquetes para las redes del protocolo IP (Internet Protocol). Snort puede realizar análisis del protocolo y búsqueda / emparejamiento de contenido, para detectar una variedad de ataques, como “buffer overflows”, amenazas de escaneo de puertos, ataques CGI (Common Gateway Interface), en sí mucho mas (Esler 10).

Snort funciona en tres modos principales: “sniffer” (escucha paquetes), registro de paquetes, detección de intrusiones en la red. El modo sniffer simplemente lee los paquetes de la red y los despliega en vivo en una pantalla. El modo registro de paquetes graba los paquetes en el disco. El modo detección de intrusiones es el más complejo y configurable, permitiendo a Snort analizar el tráfico para encontrar patrones de datos que empaten con las reglas de alertas definidas por el usuario (Esler 10).

En este proyecto de Tesis, se tiene funcionando a Snort en el modo detección de intrusiones.

2.2 Respuesta Automatizada a Intrusiones (IRS)

En cuanto a los sistemas de Respuesta Automatizada, su tecnología co-múnmente se divide en las siguientes categorías:

- Respuesta Activa de Red (Network Active Response): tiene la habilidad de interactuar con el tráfico de la red de forma indirecta a través de la modificación de políticas de un cortafuegos o de las listas de acceso de un rutea-dor (ACLs). También tiene la habilidad de desactivar los puertos de un Swi-tch, así como engañar con códigos de error en los paquetes como en TCP, RST, o paquetes inalcanzables de ICMP. Esto es muy útil para desconec-tar sesiones individuales o para tratar de convencer al host atacante que el objetivo es inalcanzable debido a errores ICMP (Rash y Orebaugh 4).

- Intrusión Preventiva de Red (Network Intrusion Prevention System IPS): la principal diferencia entre la respuesta activa de red y la respuesta preventi-va de red es que ésta última es un dispositivo inline, que se encuentra ubi-cado directamente en el camino por donde pasan los paquetes a la red. Ejemplo de estos dispositivos son routers y firewalls. Cuando un IPS es “in-line” entre el atacante y el sistema objetivo, no solo los ataques pueden ser detectados por una variedad de mecanismos, sino que también los paque-tes pueden ser detenidos de manera individual, es decir puede detener los paquetes que considere maliciosos (Rash 7).

- Respuesta Activa del Host: este sistema usualmente es implementado con software en el servidor. Una vez que un evento sospechoso se ha detecta-do (puede ser con el mecanismo de análisis de logs, alteración de archivos específicos o servicios corriendo en puertos sospechosos), su sistema de respuesta activa puede tomar acción. Su enfoque principalmente es una vez detectado el ataque, tratar de mitigarlo, puede ser por ejemplo disminu-

yendo los permisos de archivos, remoción de viruses, o agregación de nuevas reglas al cortafuegos del servidor (Rash y Orebaugh 9).

- Respuesta Preventiva del Host: al igual que la Respuesta Preventiva de Red este mecanismo puede detener en el momento inicial el ataque que se está transmitiendo al servidor objetivo. Un ejemplo de esto es la modificación del kernel del sistema operativo para restringir las llamadas maliciosas al sistema operativo, a través del parche LIDS (Linux Intrusion Detection System). Otro ejemplo de prevención son los “shims” de las aplicaciones, estos son fragmentos de código que están unidos a una aplicación para realizar una inspección en el ingreso de los datos y su validación._Cuando el “shim” detecta un código malicioso, impide su acceso a un nivel más alto de la aplicación. Un buen ejemplo de esto es el módulo “mod_security” del Servicio Web Apache. La prevención de desbordamiento de buffer (buffer overflow), es una de las más importantes tecnologías de prevención para un servidor debido a la prevalencia de este tipo de vulnerabilidades. Los métodos para prevenir este tipo de ataques o por lo menos sus efectos cae en dos categorías principales: tiempo de compilación y tiempo de ejecución (Rash y Orebaugh 11).

2.2.1 Sistemas de Respuesta Automatizada a Intrusiones

La versión de Snort que tenemos instalada en el servidor XpertSites.com es la versión 2.7, la cual no incluye dentro de su propio mecanismo un sistema de Respuesta Automatizada a Intrusiones (IRS). Esto quiere decir que si deseamos

configurar un IRS en esta versión debemos valernos de algunos de los sistemas IRS que a continuación se detalla.

La nueva versión 2.9 de Snort ya viene incluido un sistema IRS el cual puede ser activado en caso de necesitarlo**,** esta versión está disponible en Internet gratuitamente para descargarla.

SnortSam

Es un plugin de respuesta activa para Snort, que realiza la operación de gateway con varios routers y dispositivos cortafuegos. SnortSam actúa como la capa de red instruyendo al gateway que altere o bloquee el tráfico por un tiempo específico para la ip especificada.

Consiste de dos partes: un agente que ejectua el dispositivo gateway y acepta comandos, y aparte un plugin de salida para Snort que envía comandos basados en reglas que se han disparado. La comunicación entre el plugin de salida y el agente es asegurada en una sesión de TCP encriptada (Rash y Orebaugh 309).

Entre los gateways que soportan están:

- Linux IPchains
- Linux IPtables
- Cisco Routers
- Checkpoint Firewall-1

El agente SnortSam provee varias características incluyendo:

- La habilidad de especificar una lista blanca de direcciones IP que nunca

deben ser bloqueadas.

- La habilidad de proveer un bloqueo por regla y tiempo de intervalo.
- La habilidad de prevenir un bloqueo repetitivo de la misma dirección IP.
- Sesiones encriptadas entre Snort y SnortSam
- La habilidad de hacer multithreading para un procesamiento rápido y bloqueos simultáneos en múltiples dispositivos.
- La habilidad de registrar eventos y enviar notificaciones por email.
- La habilidad de escalar a redes distribuidas mas grandes usando una arquitectura cliente/servidor.

Fwsnort

Funciona como una capa de transporte en línea, debido a que se la implementa directamente en la tabla IP del firewall. Funciona traduciendo las reglas de Snort en reglas equivalentes de la tabla IP, por lo tanto solo detendrá ataques para los cuales estén especificadas reglas de alarmas en Snort. No todas las reglas de Snort son fácilmente traducidas, de todas maneras Fwsnort hace un buen trabajo traduciendo al menos un 70% de ellas (Rash y Orebaugh 323).

Snort Inline

Es un verdadero IPS, implementado entre segmentos de red con la capacidad de alterar o descartar paquetes en tiempo real mientras fluyen a través del sistema. Corre en el sistema Linux y utiliza en filado de paquetes en IPtables

para recoger y tomar decisiones sobre los paquetes mientras estos atraviesan las interfaces del sistema. Puede ser usado en modo sigilo como un puente entre segmentos de red, para no ser detectado como un punto en la red. Una de las características más interesantes es su habilidad de mitigar ataques alterando datos de la capa de aplicación mientras los paquetes atraviesan el sistema. Su instalación se puede volver compleja al requerir la configuración de varias utilidades incluyendo un parche en el kernel del sistema operativo (Rash y Orebaugh 336).

Modsecurity

Es un módulo de Apache Servicio Web, que actúa como un motor detector y de prevención a intrusiones para aplicaciones Web. Es un shim de aplicación que está unida al servicio Apache y por lo tanto corre dentro del mismo espacio del proceso de ejecución que Apache. Las reglas de Modsecurity pueden analizar los encabezados, cookies, variables de ambiente, variables del servidor, variables de página, carga del POST y salida de datos de un script.

LIDS

Linux Intrusion Detection System (LIDS), es un sistema de detección y prevención de intrusiones que reside en el kernel de Linux. Consiste en un parche que se le aplica al kernel, tiene también herramientas administrativas. LIDS implementa control de acceso mandatorio, protección de archivos y restricción de procesos en el sistema Linux restringiendo acceso a archivos, operaciones de red, acceso directo a dispositivos, acceso o uso de memoria, y acceso de entrada

y salida, también tiene un detector de escaneo de puertos (Rash y Orebaugh 348).

Portsentry

Portsentry fue desarrollado para detectar y responder a los escaneos de puerto en un host. Monitorea los puertos TCP y UDP en un sistema y responde cuando un escaneo es identificado (Rash y Orebaugh 352).

Portsentry provee tres opciones de respuesta activa:

- Inserta una ruta nula en la tabla de ruteo del host. Esto redirigiría el escaneo del atacante a una dirección ip no existente, la desventaja es que si el atacante usa “ip spoofing” podría generar en el host un DoS, Denegación de Servicio.

- Inserta una regla en el cortafuegos para bloquear el tráfico de la dirección ip desde donde se realizan los escaneos. PortSentry soporta ipfw, ipfilter, ipfwadm, ipchains e IPtables. De nuevo esto podría terminar en un DoS para el host, si el atacante camufla su dirección ip.

- Añade una regla que envuelve el TCP para la dirección ip atacante, registrándola en el archivo /etc/hosts.deny Esto impediría al atacante conectarse a los servicios del host objetivo. Si bien no es un mecanismo de protección muy fuerte, aliviaría el potencial de DoS de las anteriores dos opciones (Rash y Orebaugh 353).

PSAD

Port Scan Attack Detector (PSAD) corre en Linux y analiza los registros del firewall IPtables para detectar escaneos de puerto y tráfico sospechoso. Está diseñado para ser usado como un IDS de red que utiliza los registros del firewall IPtables par abloquear y registrar paquetes (Rash y Orebaugh 353).

PSAD y FwSnort hacen una buena combinación para una respuesta activa a intrusiones.

PSAD maneja las siguientes limitaciones que tiene PortSentry (Rash y Orebaugh 353).

- Mejor integración al firewall; PortSentry escucha en puertos para detectar escaneos, lo que sugiere mayor administración en el firewall.
- Mecanismos de puntaje para priorizar las acciones y las respuestas.
- Registro de huellas pasivo.
- Detección de exploración a ICMP
- Detección de exploración a Backdoor y DDOS
- Revisión de Whois integrada
- Alertas de e-mails integradas.

2.3 Escaneo de Seguridad

Zenmap

Este software es la versión gráfica del popular programa Nmap.

Nmap es un programa de código abierto que sirve para efectuar rastreo de puertos. Se usa para evaluar la seguridad de sistemas informáticos, así como para descubrir servicios o servidores en una red informática. Ha llegado a ser una de las herramientas imprescindibles para todo administrador de sistema, y es usado para pruebas de penetración y tareas de seguridad informática en general. Como muchas herramientas usadas en el campo de la seguridad informática, es también una herramienta muy utilizada para hacking. Los administradores de sistema pueden utilizarlo para verificar la presencia de posibles aplicaciones no autorizadas ejecutándose en el servidor, así como los crackers pueden usarlo para descubrir objetivos potenciales (Nmap).

2.4. Metodología de Seguridad OSSTMM3

Sus siglas en inglés representan las palabras “Open Source Security Testing Methology Manual”. Si bien es cierto que existen varias metodologías para buscar la seguridad. Para este caso de estudio me basaré principalmente en la metodología de la guía OSSTMM3. La razón que se ha escogido esta metodología es porque su enfoque es reduccionista, donde busca identificar claramente las Superficies de Ataque existentes, para reducirlas y luego concentrarse en las vulnerabilidades que éstas puedan tener debido a las operaciones que las mismas realizan. Su visión es buscar la seguridad desde el punto de vista de la Seguridad Operacional.

Herzog nos comenta en su manual lo siguiente:

> El propósito principal de este manual es proveer una metodología científica para una caracterización mas precisa de la Seguridad Operacional (OpSec). Este manual es adaptable a casi cualquier tipo de auditoría, incluyendo pruebas de penetración, hacking ético, evaluación de la seguridad, red-teaming, blue-teaming, consecutivamente. Está escrito como un documento de investigación y está diseñado para la verificación de la seguridad basada en hechos y presentación de métricas en un nivel profesional. Un propósito secundario sería proveer delineamientos los cuales seguidos correctamente, permitirán al analista desempeñar una auditoría certificada de OSSTMM (Herzog 13).

A continuación pongo un resumen de lo que describe Pete Herzog sobre la metodología OSSTMM3.

Este manual es sobre Seguridad Operacional (OpSec). Busca medir cuan bien la seguridad funciona. La OpSec es una combinación de separación y controles. Bajo la Opsec, para que una amenaza sea efectiva, ésta debe interactuar directa o indirectamente con un asset. Separar la amenaza de un asset es evitar una posible interacción. De ahi sería posible obtener una Seguridad Equilibrada si la amenaza y el asset están completamente separados entre ellos. Caso contrario lo que se tendría ahi es una protección del asset, que sería implementar Controles para disminuir el impacto de la amenaza.

De ahi que en el contexto de la Seguridad Operacional, se le llama Seguridad a la separación de un asset y una amenaza, y Protección al Control de una amenaza y sus efectos.

Para obtener una verdadera Protección de los assets, diferentes tipos de Controles se requieren. Pero, los Controles así mismo pueden aumentar el número de interacciones dentro del alcance lo que significa que mas controles no es necesariamente mejor.

Para comprender como la OpSec puede trabajar en un ambiente operacional, ésta debe ser reducida a sus elementos. Estos elementos permiten cuantificar la Superficie de Ataque, la cual está determinada por la cantidad de separaciones y controles que existan para ese Vector, el cual es la dirección de la interacción. El enfoque reduccionista nos lleva a la necesidad de que veamos la Seguridad y Protección de una nueva manera, una que les permite existir independientes del riesgo y con la total capacidad de alcanzar la Seguridad en Equilibrio o "Perfect Security" que se menciona en la OpSec, que es el balance exacto de seguridad y controles con las operaciones y limitaciones. Para ver la seguridad de esta nueva forma, se requiere comprender una nueva terminología también.

2.4.1 Concepto de Seguridad OSSTMM3

Seguridad es una función de la separación. Existen 3 formas lógicas de crear esta separación.

1. Mover el asset para crear una separación física o barrera lógica entre éste y la amenaza.
2. Transformar la amenaza a un estado de inofensivo.
3. Destruir la amenaza.

En este concepto cuando deseamos analizar el estado de Seguridad de algo, buscamos donde existe la posibilidad de una interacción y donde no. Pode-

mos conocer algunas, todas o ninguna de las interacciones que se puedan estar requiriendo para determinadas operaciones de algún servicio que se esté ofreciendo. Pero quien evalúa la seguridad posiblemente no conozca al momento la justificación para todos estos puntos interactivos, por lo cual nos referimos a ellos como la Porosidad. La Porosidad reduce la separación entre una amenaza y un acceso. Luego se la categoriza en tres elementos, Visibilidad, Acceso y Confianza, describiendo la función que tienen dentro de las operaciones, lo cual permite buscar los controles adecuados para la fase de remediación de mejorar la protección.

Tabla 2.1 - Terminología OSSTMM3

Término	Definición
Alcance (Scope)	Está determinado por el Canal y el Vector donde se realiza la interacción de pruebas o ataques.
Canal	Es el medio en donde se da una interacción. Pueden ser: Humano, Físico, Inalámbrico, Telecomunicaciones, Redes de Datos.
Superficie de Ataque	Es determinada por la falta de separaciones específicas y de controles funcionales que existen para ese vector.

Vector	Es la dirección de la interacción. Los vectores vendrían a ser en este caso cada una de las oportunidades de interacción con el Blanco Objetivo. Estos vectores nacen al momento de ejecutar un ataque. Si un atacante decide vulnerar el Servicio DNS entonces establecería un Vector de Ataque dirigido hacia éste Blanco Objetivo, utilizando alguna herramienta de software para atacarlo.
Vector de Ataque	Es la dirección de la interacción entre la amenaza y el Blanco Objetivo, con la finalidad de irrumpir en la Seguridad.
Controles	Permiten la reducción de pérdidas y de impacto. Son la garantía de que los assets físicos y de información así como los canales en sí mismos estén protegidos de varios tipos de interacciones indebidas.
Limitaciones	Se refiere a las limitaciones que una medida de seguridad o control pueda tener. Este es el estado actual de los límites conocidos y percibidos para los canales, operaciones y controles que se han podido verificar en una auditoría. Los tipos de limitaciones son clasificados por como interactuan con la seguridad y las protecciones en un nivel operacional.
Operaciones	Son las acciones que permiten interactuar de forma abierta y disponible.

Seguridad en Equilibrio ("Perfect Security")	El balance exacto de Seguridad con sus Controles en las operaciones y las Limitaciones de sus mismos Controles.
Porosidad	Todos los puntos interactivos, operaciones, que son categorizadas como Visibilidad, Acceso o Confianza.
Protección	Una forma de protección donde una amenaza o sus efectos son controlados. Para esto los controles deben ser ubicados de tal manera que aseguren que la amenaza en si misma o sus efectos sean minimizados a un nivel aceptable por el administrador o propietario del asset.
Seguridad	Una forma de protección donde se crea una separación entre los assets y la amenaza. Esto incluye pero no es limitado a la eliminación de ya sea el asset o su amenaza. Para que exista seguridad el asset es removido de la amenaza o la amenaza es removida del asset. El manual OSSTM cubre la seguridad desde una perspectiva operacional, verificando las medidas de seguridad en un ambiente vivo y operativo.

Rav	El rav es una escala de medida de una Superficie de Ataque. Representa cuánto de las operaciones está expuesto a una ataque. Su cálculo representa en forma de porcentaje el balance cuantitativo entre Porosidad, Controles de Pérdida y Limitaciones de los Controles. En esta escala 100% de Rav en la Seguridad Total representa el balance entre interacciones y controles.
Blanco Objetivo (Target)	Se encuentra dentro del Alcance que se requiere atacar, está compuesto por uno o mas assets y por las protecciones que pueda tener el asset.
Vulnerabilidad	Es una clasificación de Limitación donde una persona o proceso pueden acceder, denegar acceso a otros, o esconder los assets o así mismo.
Visibilidad	La ciencia policial ha identificado como “oportunidad” como uno de los elementos que motivan un robo, acompañado de “beneficio” y “riesgo disminuido”. Visibilidad es una manera de calcular la oportunidad. Puede ser el asset de un Blanco Objetivo que se lo ha descubierto dentro del Alcance.

Acceso	Como se ha comprendido que la Seguridad significa la separación entre una amenaza y un asset, entonces la habilidad de interactuar directamente con el asset significa tener Acceso a él. El Acceso es calculado por el número de lugares diferentes donde una interacción puede ocurrir donde se requiere una autenticación para proceder a interactuar.
Confianza	Medimos Confianza como parte de OpSec, a cada relación que existe donde un asset acepta interacción libremente de otro asset sin necesidad de autenticación dentro del Alcance.

2.5 Servicios Activos en el Servidor XpertSites.com

A continuación describo los servicios que han estado en funcionamiento en el Servidor XpertSites.com

2.5.1 Apache Servicio Web

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web. Apache es

el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python (Servidor HTTP Apache).

2.5.2 Mysql Base de Datos

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código (Mysql).

2.5.3 Bind Servicio DNS

BIND (Berkeley Internet Name Domain, anteriormente : Berkeley Internet Name Daemon) es el servidor de DNS más comúnmente usado en Internet, especialmente en sistemas Unix, en los cuales es un Estándar de facto. Es patrocinado por la Internet Systems Consortium. BIND 9 incluye entre otras características importantes: TSIG, notificación DNS, nsupdate, IPv6, rndc flush, vistas, procesamiento en paralelo, y una arquitectura mejorada en cuanto a portabilidad. Es comúnmente usado en sistemas GNU/Linux" (Bind).

2.5.4 Servicio SSH

OpenSSH (Open Secure Shell) es un conjunto de aplicaciones que permiten

realizar comunicaciones cifradas a través de una red, usando el protocolo SSH. Fue creado como una alternativa libre y abierta al programa Secure Shell, que es software propietario *(*Open SSH*).*

2.5.5 ProFTP Servicio FTP

ProFTPd es un servidor FTP. Puede ser fácilmente configurado como múltiples servidores FTP virtuales, y tiene capacidades para ser enjaulado dependiendo del sistema de archivos que haya por debajo. Puede ejecutarse con un demonio propio o como un servicio más de inetd. Es capaz de trabajar sobre Ipv6 (ProFTPd).

2.5.6 Dovecot Servicio IMAP/POP3

Dovecot es un servidor de IMAP y POP3 de código abierto para sistemas GNU/Linux / UNIX.. Puede trabajar con el estándar mbox, Maildir y sus propios formatos nativos dbox de alto rendimiento. Es completamente compatible con implementaciones de servidores UW IMAP y Courier IMAP, así como con clientes que accedan directamente a los buzones de correo. También incluye un Agente de Entrega de Correo llamado Local Delivery Agent (agente de entrega local*)* (Dovecot)*.*

2.5.7 Postfix Mail Server

Postfix es un servidor de correo de software libre / código abierto, un programa informático para el enrutamiento y envío de correo electrónico, creado con la intención de que sea una alternativa más rápida, fácil

de administrar y segura al ampliamente utilizado Sendmail. Anteriormente conocido como VMailer e IBM Secure Mailer, fue originalmente escrito por Wietse Venema durante su estancia en el Thomas J. Watson Research Center de IBM, y continúa siendo desarrollado activamente (Postfix).

CAPÍTULO 3: Evaluación de Seguridad del Servidor

3.1 Situación Actual

El servidor XpertSites.com hace ya algunos años atrás en un inicio fue creado con la finalidad de crear sitios web para empresas y posicionarlas en Internet. XpertSites comenzó y se ha mantenido como un negocio familiar, donde se juntaron talentos en el área de informática, marketing y ventas. Es decir XpertSites no es una empresa grande con cientos de empleados, al contrario, es un negocio familiar donde el mismo especialista en IT hace un trabajo personalizado a sus clientes de confianza. En ese proceso se ha asesorado a compañías pesqueras, compañías mineras, hasta artistas plásticos y organizaciones sin fines de lucro para su posicionamiento en Internet.

Con el pasar del tiempo, XpertSites se concentró mas en el desarrollo de tecnología web para sitios del mercado de Bienes Raíces en Ecuador. Entrando específicamente a la investigación para el desarrollo de estas aplicaciones en el framework Django basado en Python. De esta manera con el tiempo fue delegando algunas cuentas de sus clientes a otros proveedores, para concentrarse principalmente en el desarrollo de aplicaciones de Bienes Raíces en Django.

Los clientes de los sitios web alojados principalmente suben sus archivos, actualizan su información y administran su cuenta, únicamente a través del panel de control de los manejadores de contenido Wordpress y Django Mezzanine, es decir sus interacciones están relacionadas únicamente a lo que se puede realizar dentro de un manejador de contenidos, esto comprende que los clientes, no acceden a la consola de Linux, ni realizan tareas de administración en Linux como

creación de bases de datos, instalación de nuevas aplicaciones en el servidor, o modificación de archivos importantes de configuración del servidor, muy rara vez solicitan un acceso FTP y cuando lo hacen se lo realiza temporalmente, de la misma manera en el caso de un acceso SSH, todo este trabajo técnico que tenga que ver con configuraciones en el servidor lo hace el administrador del mismo. En base a esta realidad del presente caso de estudio se espera evaluar la seguridad del servidor e implementar mejorías en su seguridad.

Dentro del servidor se tiene instalado un Sistema de Detección de Intrusiones llamado Snort en su versión 2.7.0. con la finalidad de ir observando los intentos de intrusiones que ha venido teniendo. El sistema ha estado grabando las últimas alarmas de intrusiones que se han activado los últimos cuatro meses anteriores al inicio de este estudio.

Snort ha sido instalado con una configuración predeterminada tal como viene en su versión para Debian Lenny y con una base de datos de alarmas de licencia libre, igualmente tal como vienen para su respectiva versión, es decir con alrededor de 21200 alarmas, de las cuales vienen activadas para su uso por default alrededor de unas 16400 alarmas (Over 21,000 Snort Rules).

Actualmente Snort se encuentra funcionando únicamente como un sistema para la detección de intrusiones (IDS) mas no como un sistema de respuesta a intrusiones (IRS). La versión 2.7 de Snort que se tiene instalada no funciona como un IRS, para lograr que funcione de esa manera en caso de querer implementarlo habría que instalar la actual versión que incluye ya un sistema IRS en su versión de Snort 2.9 que es la versión que está actualmente disponible en Internet en el momento del estudio de la presente Tesis.

En cuanto al sistema operativo, el servidor está funcionando con una distribución de Linux denominada Debian Lenny versión 5.0.

Se usarán los dominios XpertSites.com y XpertSites.net de forma indistinta durante esta Tesis, ya que apuntan al mismo servidor y a la misma dirección ip.

3.1.1 Hardware del Servidor

Tabla 3.1 - Actuales componentes físicos Servidor XpertSites.com	
Componentes físicos	**Requerimientos**
Memoria Ram	1gb RAM
Disco Duro	40 gigas
Sistema Operativo	Linux Debian 5.0 (Lenny)
Procesador	AMD Athlon(tm) 64 X2 Dual Core Processor 4400+

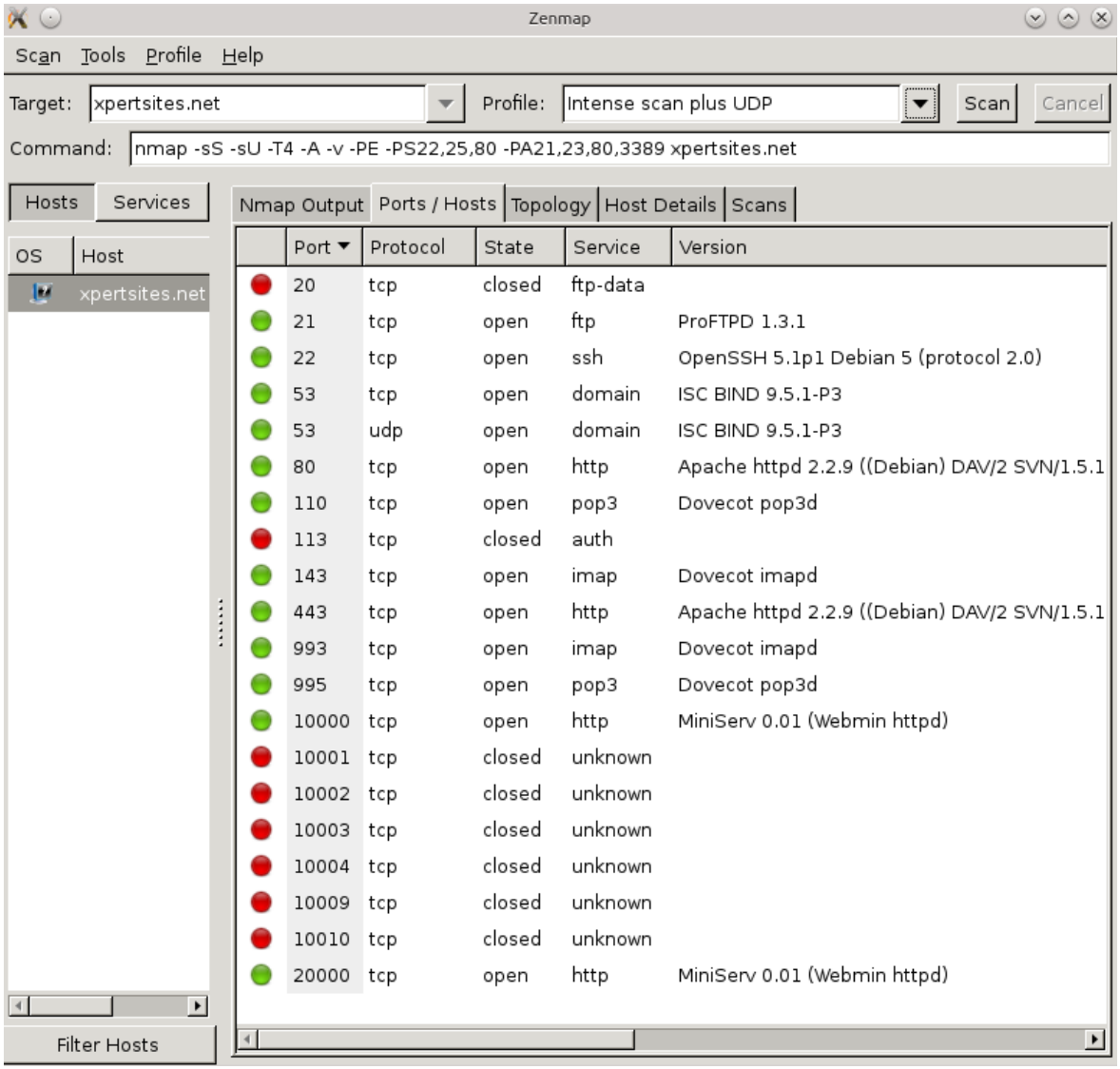
3.1.2 Software del Servidor

Tabla 3.2 - Versión Actual del Software del Servidor	
Software	**Versión Actual**
Apache WebServer	2.2.9
BIND DNS Server	9.5.1
Dovecot IMAP/POP3 Server	1.0.15
MySQL Database Server	5.0
Postfix Mail Server	2.5
ProFTPD Server	1.31
SSH Server	5.1
PHP	5.2
Python	2.5
Webmin Server	1.0
Phpmyadmin	2.11.8
Django Web Framework	1.2

Wordpress	2.8
Joomla	1.5

3.1.3 Escaneo de Vulnerabilidades con Zenmap

Figura 3.1: Escaneo de Vulnerabilidades con Zenmap



A continuación los puertos que tenemos abiertos en el servidor:

Tabla 3.3 - Puertos abiertos en el Servidor	
21 tcp	ProFTPD 1.3.1
22 tcp	OpenSSH 5.1p1
53 tcp	ISC Bind 9.5.1
53 udp	ISC Bind 9.5.1
80 tcp	Apache httpd 2.2.9
110 tcp	Dovecot pop3d
143 tcp	Dovecot imapd
443 tcp	Apache httpd 2.2.9
993 tcp	Dovecot imapd
995 tcp	Dovecot pop3d
10000 tcp	MiniServ Webmin
20000 tcp	MiniServ Webmin

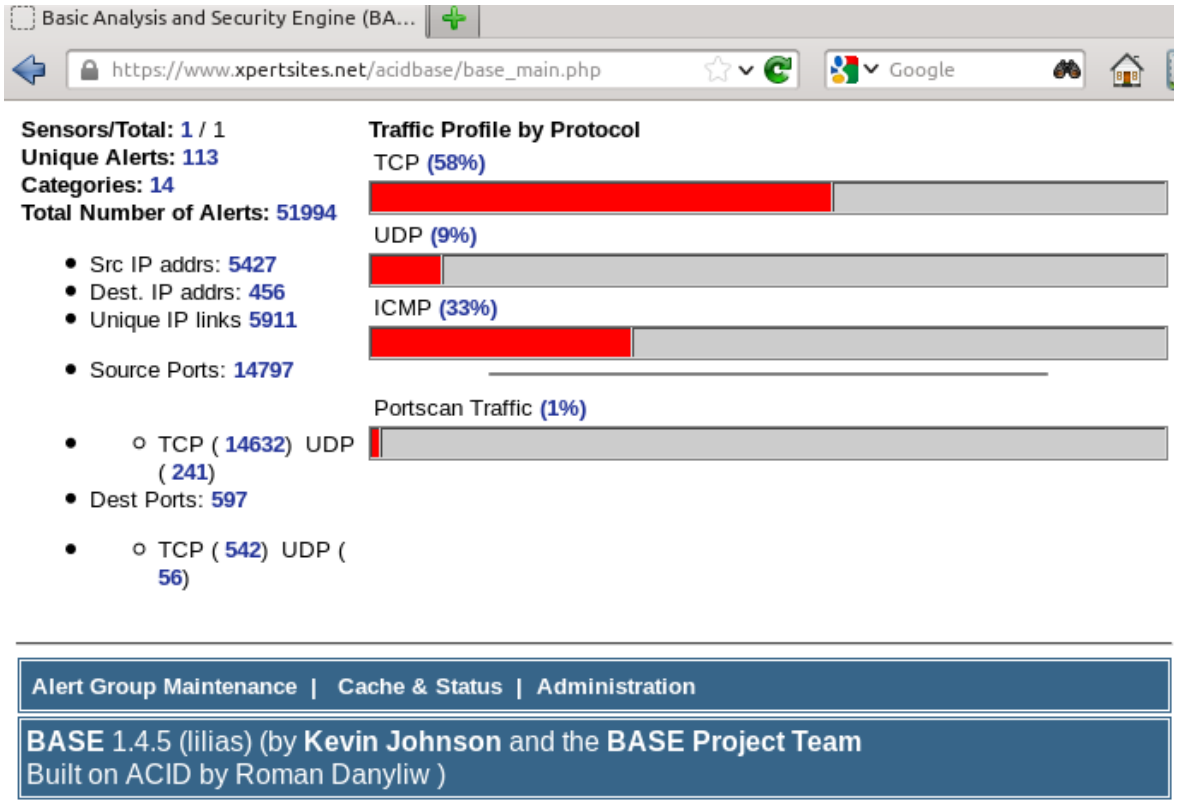
3.2 Evaluación de los ataques registrados en el servidor a través de Snort.

Se procedió a realizar el análisis de las estadísticas de los ataques recibidos, que habían sido en gran parte previamente registrados por XpertSites.com en la base de datos del servidor a través del sistema de Detección de Intrusos Snort. Se ha decidido evaluar los 20 tipos de ataques mas frecuentes recibidos en los 4 meses anteriores.

Snort no ofrece una interfaz gráfica para visualizar los ataques almacenados en la base de datos, para esto se utilizó el software AcidBase, el cual permite clasificar la información almacenada por Snort para su mejor interpretación.

En la siguiente figura se muestra el resumen de ataques que ha venido teniendo el servidor.

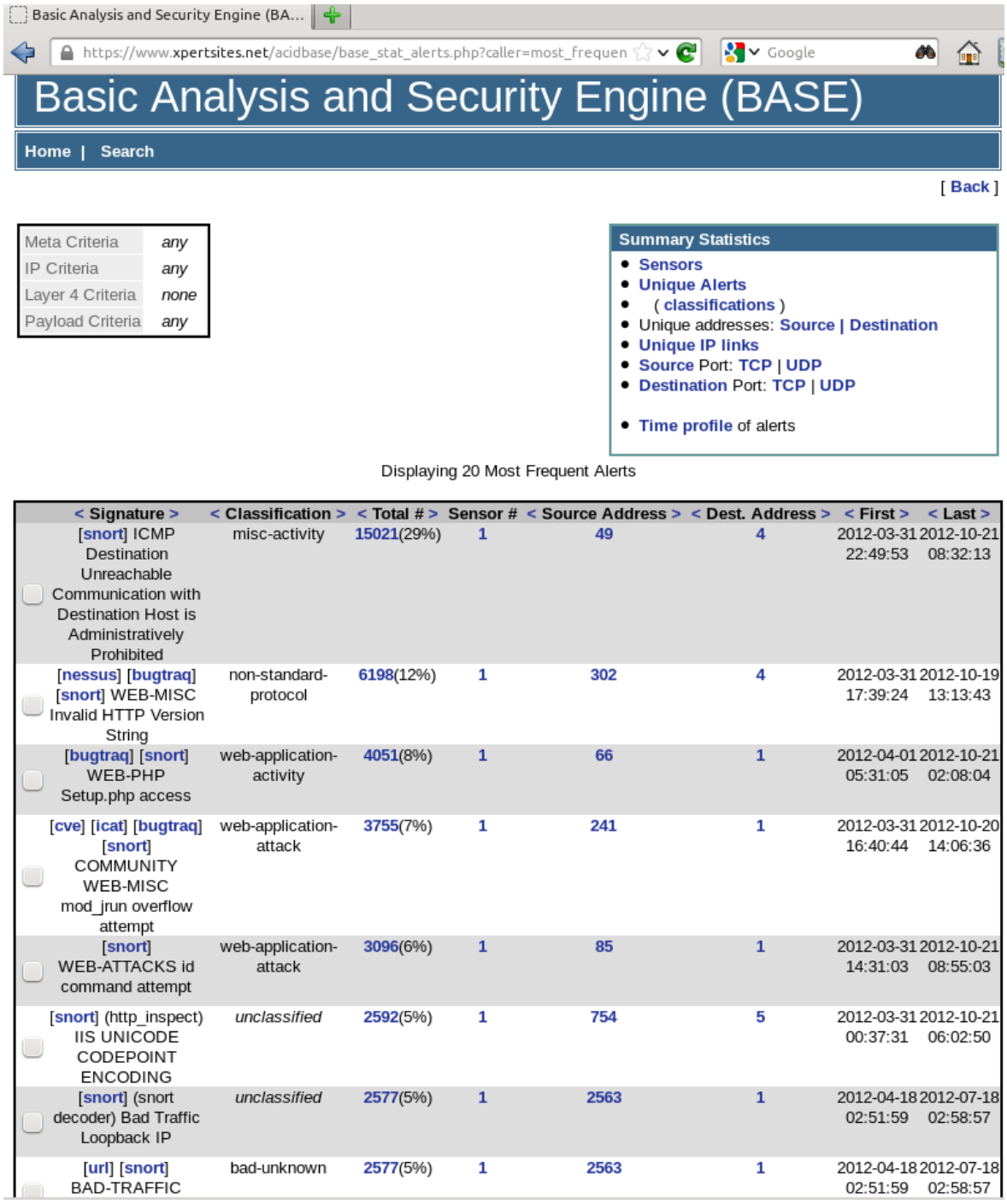
Figura 3.2: Ataques registrados por Snort en el servidor



En la figura anterior se puede observar que se activaron 51994 alertas sobre presuntos ataques de los cuales si no contamos los ataques repetidos quedarían en 113 ataques únicos. También nos podemos fijar que el origen de los ataques se supone que vienen de 5427 direcciones ip distintas, hago esta presunción porque existe la probabilidad de que muchos de esos ataques se originaron desde una misma dirección ip, pero que el atacante pudo haber camuflado su dirección ip con otras direcciones alterando la información de los paquetes de datos que ha envíado, esto suele ser una práctica bastante común en el mundo de la intrusión a sistemas informáticos en el Internet y posiblemente explique el porqué tenemos un número tan alto de direcciones ip origen.

A continuación los 20 tipos de ataques mas frecuentes hacia el servidor:

Figura 3.3: 20 tipos de ataques mas frecuentes registrados



< Signature >	< Classification >	< Total # >	Sensor #	< Source Address >	< Dest. Address >	< First >	< Last >
[snort] ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited	misc-activity	15021(29%)	1	49	4	2012-03-31 22:49:53	2012-10-21 08:32:13
[nessus] [bugtraq] [snort] WEB-MISC Invalid HTTP Version String	non-standard-protocol	6198(12%)	1	302	4	2012-03-31 17:39:24	2012-10-19 13:13:43
[bugtraq] [snort] WEB-PHP Setup.php access	web-application-activity	4051(8%)	1	66	1	2012-04-01 05:31:05	2012-10-21 02:08:04
[cve] [icat] [bugtraq] [snort] COMMUNITY WEB-MISC mod_jrun overflow attempt	web-application-attack	3755(7%)	1	241	1	2012-03-31 16:40:44	2012-10-20 14:06:36
[snort] WEB-ATTACKS id command attempt	web-application-attack	3096(6%)	1	85	1	2012-03-31 14:31:03	2012-10-21 08:55:03
[snort] (http_inspect) IIS UNICODE CODEPOINT ENCODING	*unclassified*	2592(5%)	1	754	5	2012-03-31 00:37:31	2012-10-21 06:02:50
[snort] (snort decoder) Bad Traffic Loopback IP	*unclassified*	2577(5%)	1	2563	1	2012-04-18 02:51:59	2012-07-18 02:58:57
[url] [snort] BAD-TRAFFIC	bad-unknown	2577(5%)	1	2563	1	2012-04-18 02:51:59	2012-07-18 02:58:57

	[bugtraq] [snort] POP3 PASS format string attempt	attempted-admin	2314(4%)	1	30	1	2012-04-01 22:14:01	2012-10-05 15:21:52
	arachNIDS[snort] WEB-MISC http directory traversal	attempted-recon	2163(4%)	1	11	1	2012-05-12 15:27:29	2012-10-18 21:13:22
	arachNIDS[snort] ICMP PING NMAP	attempted-recon	1613(3%)	1	441	3	2012-03-30 19:32:50	2012-10-21 01:38:00
	[nessus] [snort] WEB-MISC robots.txt access	web-application-activity	709(1%)	1	294	1	2012-03-30 18:40:52	2012-04-02 23:43:34
	[snort] (http_inspect) DOUBLE DECODING ATTACK	*unclassified*	642(1%)	1	231	5	2012-03-30 22:58:25	2012-10-19 21:41:46
	[snort] ATTACK-RESPONSES 403 Forbidden	attempted-recon	543(1%)	1	1	309	2012-03-30 19:39:05	2012-10-19 05:23:36
	[snort] (http_inspect) OVERSIZE REQUEST-URI DIRECTORY	*unclassified*	464(1%)	1	136	4	2012-04-01 12:09:25	2012-10-19 23:32:37
	[cve] [icat] [bugtraq] [snort] FTP command overflow attempt	protocol-command-decode	308(1%)	1	4	1	2012-10-10 11:55:36	2012-10-17 10:44:01
	[snort] (portscan) UDP Portsweep	*unclassified*	291(1%)	1	1	66	2012-04-01 06:11:14	2012-10-21 00:30:39
	[snort] (http_inspect) OVERSIZE CHUNK ENCODING	*unclassified*	204(0%)	1	11	1	2012-04-05 09:59:27	2012-10-11 20:21:06
	[bugtraq] [snort] IMAP login format string attempt	attempted-admin	194(0%)	1	11	1	2012-04-01 22:36:21	2012-10-11 03:27:00
	[snort] (snort_decoder): Experimental Tcp Options found	*unclassified*	188(0%)	1	23	2	2012-04-17 17:07:51	2012-10-18 13:18:23

A continuación se enumera las alarmas en orden de frecuencia de mayor a menor, con su respectivo id, con el cual se puede acceder a la información de la respectiva alarma en el sitio web www.snortid.com (Somerville).

Vamos a evaluar las siguientes alarmas como:

- Ataque Falso Positivo: cuando el evento es un ataque que no es aplicable a la configuración del servidor debido a que ese elemento que el ataque buscaba no existe. O también cuando el evento no es un ataque ya que es

un comportamiento natural de la red o ha existido algún error en los paquetes de datos.

- Ataque Potencial: cuando la alarma en efecto representa un intento de ataque y a su vez el ataque busca vulnerar un elemento que si se encuentra en el servidor.

#1 - Frecuencia: 15021 - Snort Id: 1:486

Título: "ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited"

Análisis

La alarma se activa cuando un paquete de datos no logra atravesar una red. Esto puede ser un indicador de un problema de ruteo o de la red. (Somerville) Se puede observar que esta alarma no es necesariamente un ataque, sino que mas bien tienen que ver con un problema de la red. Por lo tanto es un falso positivo.

Evaluación obtenida: Ataque Falso Positivo.

#2 - Frecuencia: 6198 - Snort Id: 1:2570

Título: "WEB-MISC Invalid HTTP Version String"

Análisis

Se genera la alarma cuando se realiza una petición HTTP al servidor de una manera no standard. Puede darse en los casos de querer vulnerar una aplicación en el servidor (Somerville).

Se sospecha que este tipo de paquetes puede usarse en un ataque, pero

asi mismo se menciona también que puede ser un falso positivo: "La versión actual de la alarma incorrectamente asume que el texto donde se menciona la versión HTTP del paquete deba estar en letras mayúsculas" (Case-sensitivity in 2570.6). Hay que recalcar que en ésta cita su autor comete un error tipográfico al describir esta situación como un falso negativo, cuando realmente se refiere a un falso positivo.

Por la alta frecuencia de veces en que esta alarma se ha activado, y por el alto número de fuentes ip que activaron la alarma que son 342, es mas probable que ésta alarma sea un falso positivo.

Evaluación obtenida: Ataque Falso Positivo.

#3 - Frecuencia: 4051 - Snort Id: 1:2281

Título: "WEB-PHP Setup.php access"

Análisis

El evento se genera cuando se hace un intento por explotar una vulnerabilidad en la aplicación PHP MediaWiki que pudiera estar corriendo en un servidor(Somerville).

En este caso se puede observar que pudiera realmente ser un intento de ataque, pero la aplicación MediaWiki no la tenemos instalada en el servidor, así que se procede a acceder al registro de la base de datos de Snort donde se almacena el comando utilizado por los atacantes para encontrar una vulnerabilidad y encuentro lo siguiente:

```
"GET /phpmyadmin/scripts/setup.php HTTP/1.1"
```

Con esto se puede observar que el ataque fue efectuado no a la aplicación

MediaWiki sino mas bien a la página setup.php de la aplicación PhpMyAdmin. En este caso, si tenemos instalado en el servidor la aplicación PhpMyAdmin, pero el ataque fue infructuoso porque se tiene protegida la carpeta de la aplicación con una protección de clave a nivel de directorio.

Evaluación obtenida: Ataque Potencial.

#4 - Frecuencia: 3755 - Snort Id: 1:100000122

Título: “COMMUNITY WEB-MISC mod_jrun overflow attempt”

Análisis

Este es un ataque efectuado a los conectores mod_jrun and mod_jrun20 de Apache para el servicio web JRun 3.0 a 4.0 (Somerville).

Se puede ver que es un ataque real, pero no se aplica al servidor ya que no disponemos de esa aplicación instalada en el servidor.

Evaluación obtenida: Ataque Falso Positivo.

#5 - Frecuencia: 3096 - Snort Id: 1:1333

Título: “WEB-ATTACKS id command attempt”

Análisis

Ésta alarma busca que en la dirección url escrito por el atacante, contenga el texto "id", suponiendo que el atacante usará esto como un comando para consultar al servidor información sobre los usuarios del mismo (Somerville).

El problema de ésta alarma es que no discierne de las páginas web especialmente de los manejadores de contenido como Joomla, que hacen uso común

del texto "id" para identificar y desplegar páginas web. Por lo tanto la mayoría de sus alertas se convierten en una falsa alarma.

A continuación podemos ver ésta situación donde tenemos el texto registrado en la base de datos de Snort que disparó la alarma:

```
GET /index.php?option=com_content&amp;task=view&amp;id=8&amp;Itemid=12
```

Evaluación obtenida: Ataque Falso Positivo.

#6 - Frecuencia: 2592 - Snort Id: 119:7

Título: "UNICODE CODEPOINT ENCODING"

Análisis

La alarma se dispara cuando el atacante utiliza una vulnerabilidad que funciona únicamente con el servidor Microsoft Information Server (Somerville).
El servidor que estamos evaluando utiliza el sistema operativo Linux Debian Lenny, lo que hace que esta alarma sea un falso positivo.

Evaluación obtenida: Ataque Falso Positivo.

#7 - Frecuencia: 2577 - Snort Id: 1:528

Título: "Traffic Loopback IP"

Análisis

El evento es generado cuando Snort detecta un tráfico anómalo en la red (Somerville). En vista de la no muy clara descripción de que tipo de anomalía en la red puede ser, se procede a revisar en la base de datos de Snort la informa-

ción que pudiera contener el paquete de datos que disparó la alarma. Al revisar los distintos paquetes que dispararon la alarma, éstos no contenían ningún texto adentro, es decir no contenían ninguna instrucción de ataque, estaban vacíos. Con esto determinamos que esta alarma no representa un ataque, es un falso positivo.

Evaluación obtenida: Ataque Falso Positivo.

#8 - Frecuencia: 2577 - Snort Id: 116:150

Título: “BAD-TRAFFIC loopback traffic”

Análisis

El evento es generado cuando Snort detecta un tráfico anómalo en la red (Somerville). Curiosamente se detecta que esta alarma es disparada por el mismo paquete de datos que activó la alarma anterior “Snort id: 1:528”, incluso tiene la misma fecha y hora registrada. Al tratarse del mismo paquete de datos, realizamos la misma conclusión que en el anterior evento.

Evaluación obtenida: Ataque Falso Positivo.

#9 - Frecuencia: 2314 - Snort Id: 1:2666

Título: "POP3 PASS format string attempt"

Análisis

Algunas versiones de "Courier IMAP/POP" son vulnerables al formato de la cadena de datos enviado durante una sessión de autenticación con el servidor

POP. El ataque es aplicable desde las versiones 1.6 a 3.0.2 de Courier IMAP/POP (Somerville).

El servidor no tiene instalado el programa Courier IMAP/POP, en vez de éste tiene instalado el programa Dovecot IMAP/POP3, por lo tanto éste ataque no es aplicable al servidor.

Evaluación obtenida: Ataque Falso Positivo.

#10 - Frecuencia: 2163 - Snort Id: 1:1112

Título: “WEB-MISC http directory traversal"

Análisis

Este es un ataque que intenta buscar acceder a otros directorios del servidor (Somerville). Se procedió a revisar la cadena de datos grabada por Snort y se encontró lo siguiente:

ssage\"\05420\054\"Se a\\u00f1adi\\u00f3 con \\u00e9xito el entrada \\\"Cr\\u00e9dito Biess vivienda nueva o usada: borrador\\\".\"]\054[\"__**json_message**\"\05420\054\"Se a\\u00f1adi\\u00f3 con \\u00e9xito el text \\\"Afiliados Cr\\u00e9dito Biess .....\\\". Puede editarlo de nuevo abajo.\"]\054[\"__**json_message**\"\05420\054\"Se modific\\u00f3 con \\u00e9xito el entrada \\\"Cr\\u00e9dito Biess vivienda nueva o usada: publicado\\\".\"]]"

Se puede observar que en el ataque intentaron usar comandos usando la notación **json** de javascript. El servidor si utiliza tecnología javascript y ajax, pero no se sabe realmente si este tipo de ataque vulneraría el sistema, en todo caso existe la probabilidad al tener esta tecnología, por lo que le denominaremos como un Ataque Potencial.

Evaluación obtenida: Ataque Potencial

#11 - Frecuencia: 1613 - Snort Id: 1:469

Título: “ICMP PING NMAP”

Análisis

Este evento se dispara cuanto hay un intento de obtener información del servidor a través del programa Nmap, para verificar que servidores están funcionando en una red. Al ser posible este tipo de escaneos en nuestro servidor, se lo calificará como un Ataque Potencial.

Evaluación obtenida: Ataque Potencial

#12 - Frecuencia: 709 - Snort Id: 1:1852

Título: “WEB-MISC robots.txt access”

Análisis

El evento avisa que un posible atacante está accediendo al archivo robots.txt de un sitio web en el servidor para luego acceder a algún archivo que se encuentre mencionado en el mismo.

En este caso se procedió a examinar la cadena de datos utilizada por el atacante para verificar si realmente era un ataque quien quería acceder al archivo o era el comportamiento normal de algún motor de búsqueda en querer obtener información sobre las páginas webs del servidor.

Se encontró lo siguiente en la base de datos de Snort respecto a este ataque registrado:

From: msnbot(at)microsoft.com

From: googlebot(at)googlebot.com

From: bingbot(at)microsoft.com

Esto demuestra que éste evento realmente no representa un ataque, sino que se trata de los motores de búsqueda de Google y Microsoft intentando indexar las páginas web almacenadas en el servidor.

Evaluación obtenida: Ataque Falso Positivo.

#13 - Frecuencia: 642- Snort Id: 119:2

Título: "(http_inspect) DOUBLE DECODING ATTACK"

Análisis

La alarma se dispara cuando el atacante utiliza una vulnerabilidad que funciona únicamente con el servidor Microsoft Information Server (Somerville). El servidor que estamos evaluando utiliza el sistema operativo Linux Debian Lenny, razón por la cual este ataque no es aplicable.

Evaluación obtenida: Ataque Falso Positivo

#14 - Frecuencia: 543 - Snort Id: 1:1201

Título: "ATTACK-RESPONSES 403 Forbidden"

Análisis

Este evento ocurre cuando un cliente trata de acceder a un recurso del sistema que no está permitido.

Eventualmente la descripción es muy clara, se ha querido acceder a un re-

curso del sistema que no está permitido. Podemos evaluarlo como un Ataque Potencial.

Evaluación obtenida: Ataque Potencial.

#15 - Frecuencia: 464 - Snort Id: 119:15

Título: "OVERSIZE REQUEST-URI DIRECTORY"

Análisis

El evento ocurre cuando el atacante utiliza una cadena de datos bien grande en la dirección URL, pudiendo esto convertirse en un ataque (Somerville).

A continuación un extracto obtenido del ataque registrado en la base de datos de Snort:

.php%22%3Bi%3A30%3Bs%3A34%3A%22%2Fthemes%2FFlexxDark%2Fsidebar-left.php
%22%3Bi%3A31%3Bs%3A28%3A%22%2Fthemes%2FFlexxDark%2Fsingle.php%22%3Bi
%3A32%3Bs%3A35%3A%22%2Fthemes%2FFlexxDark%2Fsidebar-right.php%22%3Bi
%3A33%3Bs%3A32%3A%22%2Fthemes%2FFlexxDark%2Fmenu-pages.php%22%3Bi
%3A34%3Bs%3A28%3A%22%2Fthemes%2FFlexxDark%2Fsearch.php

Evaluación obtenida: Ataque Potencial.

#16 - Frecuencia: 308 - Snort Id: 1:1748

Título: "FTP command overflow attempt"

Análisis

El ataque trata de enviar una cadena de datos mayor o igual a 400 bytes para el servicio FTP 3CDaemon de Windows, con la intención de confundir al servicio y ganar acceso (Somerville).

Como se puede ver este ataque no se aplica a la realidad de nuestro servidor Linux.

Evaluación obtenida: Ataque Falso Positivo.

#17 - Frecuencia: 291 - Snort Id: 122:19

Título: "(portscan) UDP Portsweep "

Análisis

El siguiente evento se genera cuando se detecta un escaneo de puertos hacia el servidor (Somerville).

Se procede a revisar la cadena de datos capturada por Snort sobre el ataque:

```
Priority Count: 5
Connection Count: 30
IP Count: 25
Scanned IP Range: 50.57.11.90:216.255.129.246
Port/Proto Count: 1
Port/Proto Range: 53:53
```

Se observa que en efecto es un escaneo de puertos donde se ha especificado el rango de direcciones ip a escanear.

Evaluación obtenida: Ataque Potencial

#18 - Frecuencia: 204 - Snort Id: 119:16

Título: "(http_inspect) OVERSIZE CHUNK ENCODING"

Análisis

El evento ocurre cuando el atacante utiliza una cadena de datos bien grande en la dirección URL, pudiendo ser signos de un ataque (Somerville). Para verificar que realmente fue un intento de ataque se procede a inspeccionar la cadena de datos registrada por Snort:

```
-----------------------------9ce1b9ba0c21
[2 non-ASCII characters]
Content-Disposition: form-data; name="submit"
[4 non-ASCII characters]
Login
[2 non-ASCII characters]
-----------------------------9ce1b9ba0c21
[2 non-ASCII characters]
Content-Disposition: form-data; name="lang"
[4 non-ASCII characters]
en-GB
[2 non-ASCII characters]
-----------------------------9ce1b9ba0c21
[2 non-ASCII characters]
Content-Disposition: form-data; name="username"
[4 non-ASCII characters]
admin
[2 non-ASCII characters]
-----------------------------9ce1b9ba0c21
[2 non-ASCII characters]
Content-Disposition: form-data; name="passwd"
[4 non-ASCII characters]
198400
[2 non-ASCII characters]
```

```
----------------------------9ce1b9ba0c21--
```

Se observa que efectivamente es un ataque, se puede ver que el atacante intentaba adivinar la clave de admin a través de un formulario web.

Evaluación obtenida: Ataque Potencial

#19 - Frecuencia: 194 - Snort Id: 1:2664

Título: "IMAP login format string attempt"

Algunas versiones de Courier IMAP/POP son vulnerables al formato de la cadena de datos enviado durante una sesión de autenticación con el servidor POP. El ataque es aplicable desde las versiones 1.6 a 3.0.2 de Courier IMAP/POP (Somerville).

El servidor no tiene instalado el programa Courier IMAP/POP, en vez de éste, tiene instalado el programa Dovecot IMAP/POP3, por lo tanto éste ataque no es aplicable al servidor.

Evaluación obtenida: Ataque Falso Positivo

#20 - Frecuencia: 188 - Snort Id: 116:58

Título: "(snort_decoder): Experimental Tcp Options found"

Análisis

Esto es la indicación de un comportamiento anómalo entre elementos de la

red (Somerville). Se procede a revisar la cadena de datos de los paquetes involucrados con este evento registrado en la base de datos de Snort y se encuentra que los paquetes no tienen cadenas de datos con instrucciones de algún ataque.

Evaluación obtenida: Ataque Falso Positivo

Tabla 3.4 - Resultado Evaluación de Ataques Frecuentes		
Evaluación obtenida	Conteo	Porcentaje
Ataques Falsos Positivos	13	65%
Ataques Potenciales	7	35%
Total	20	100%

De la muestra de los 20 tipos de ataques más frecuentes, se pudo evaluar que el 65% fueron ataques falsos positivos, es decir que en ese porcentaje están ataques que no tenían nada que ver con la configuración del servidor y otros que simplemente no eran ataques sino comportamiento inusual de la red.

El 35% restante representa los ataques que apuntan a elementos que el Servidor si tiene en su configuración, pero que realmente no son prueba de que sean efectivos contra estos elementos. Simplemente indica que estos elementos existen en el servidor, mas no que éstos realmente puedan ser vulnerados. Su determinación de vulnerabilidad efectiva, se muestra incierta, ya que son comandos que han utilizado en los paquetes de datos, pero no necesariamente estos comandos han probado tener algún sentido para el servidor. Al menos en el servidor no se ha registrado ninguna intrusión real o daño debido a estos ataques potenciales.

Observando esta situación, notamos que no tiene sentido seguir observando los presuntos ataques al servidor registrados por Snort para implementar mayor seguridad, donde hasta el momento del conteo de los 20 tipos de ataques mas frecuentes el 65% es un falso positivo y sin duda este porcentaje aumentará según aumentemos la muestra, ésto debido a que tenemos registrado un total de 51994 ataques hacia el servidor, de los cuales en XpertSites.com jamás se ha tenido noticias de que han tenido éxito.

En vista de esto, se observa que no tiene sentido invertir tiempo y recursos en construir reglas de Snort que respondan a los ataques más frecuentes recibidos, peor aún podemos proceder a desactivar o activar reglas de Snort en base a esta información que en su mayoría son falsos positivos.

Después de haber hecho esta evaluación, hago la siguiente reflexión sobre Snort como un sistema de protección (IRS) ante los ataques recibidos en el servidor:

- Snort de forma inicial funciona como un sistema de detección de ataques y si lo configuramos para que Snort defienda el servidor de estos ataques, su mecanismo de protección dependerá de la pre-existencia de reglas que disparen las alarmas y bloqueen este tipo de ataques. Es decir si la regla no existe y la vulnerabilidad es nueva, existe una alta probabilidad que Snort no detecte la amenaza. Es casi la misma lógica de un antivirus, éste ofrece protección siempre y cuando el virus o su patrón de comportamiento se encuentre previamente registrado en su base de datos, es decir sea conocido previamente.

- En Snort se necesita actualizar el set de reglas continuamente si se desea defender a los sistemas de las nuevas vulnerabilidades reportadas por la comunidad en Internet y para tener reglas actualizadas donde ya se hayan realizado correcciones a errores encontrados. Si ya tenemos un set de reglas organizado y pulido, y luego ingresan nuevas reglas o cambios para las reglas anteriores, esto requeriría un significante trabajo de investigación para saber de que se tratan estas nuevas reglas y si valen la pena activarlas, algo que en XpertSites.com al menos no se dispone del presupuesto ni el personal para dedicar a este tipo de trabajo, talvez funcione en compañías con departamentos de seguridad especializados, donde existen equipos de trabajo que se dedican a estas tareas específicas.
- En cuanto a los ataques de real importancia, si el atacante desea que ciertos ataques que puedan realmente representar un peligro, éstos no sean fácilmente encontrados en el sistema de detección, él podría utilizar una herramienta automatizada para lanzar una exhaustiva variedad de ataques generando un sinnúmero de alarmas las cuales taparían o nos despistarían de las alarmas de ataques que realmente nos debemos preocupar.
- Por otro lado si queremos bloquear a un atacante para que éste no siga levantando falsas alarmas, sería a través de su dirección ip. Entonces si configuramos a Snort para que bloquee las direcciones ip que tienen relación con los ataques, nos podemos topar con el problema de que éstas direcciones ip son falsificadas y que realmente no son el origen del ataque o se puede dar el caso de que muchas de estas direcciones pueden pertenecer a sitios normalmente benévolos pero que fueron utilizados temporalmente

para un ataque. Si recordamos las cifras de la figura 3.2 se nos muestra que los ataques han sido generados desde 5427 posibles direcciones ip en tan solo unos cuantos meses, esto nos pone a pensar que en efecto un gran porcentaje de esas direcciones ip, pertenecen a puntos de red de los cuales no deberíamos bloquear su acceso, quien sabe algunas de estas direcciones ip puedan pertenecer a redes que son importantes para la conección del servidor a Internet.

- En el caso de que usáramos a Snort como IRS para elaborar reglas de alarmas que bloqueen no necesariamente los ataques mas frecuentes sino ataques que tengan relación a los elementos que tiene el servidor como el Servicio Mysql, Servicio DNS, Servicio FTP, Servicio SSH, Servicio de Correo, Servicio Web Apache, esto no tendría sentido ya que para la mayoría de éstos servicios existe una inmensa variedad de tipos de ataques, no olvidemos como se mencionó anteriormente que para la versión de Snort 2.7 se crearon alrededor de 21200 reglas de alarmas para justamente proteger los ataques que tengan que ver con estos servicios (Over 21,000 Snort Rules). Por eso en un inicio se planteó la posibilidad de proteger el servidor con Snort tomando en cuenta únicamente los tipos de ataques más frecuentes que ha venido teniendo el servidor XpertSites.com en los últimos meses.

Por lo tanto para cumplir nuestro objetivo principal de la Tesis que es mejorar la seguridad del servidor XpertSites.com, no procederemos con esta metodología de implementar Snort como un sistema de Respuesta (IRS), sino que busca-

remos una metodología que nos ayude a resolver el tema de mejorar la seguridad desde otra perspectiva.

3.3 Presentación de la metodología OSSTMM3

La OSSTMM3 definida por sus siglas en inglés como la Open Source Security Testing Methodology Manual, nos ofrece actualmente la versión 3, que es con la cual trabajaremos para evaluar nuestro caso de estudio. Esta versión tiene cambios importantes en cuanto a conceptos y terminologías en comparación con la versión 2.

La razón que se ha escogido esta metodología es porque su enfoque es reduccionista, donde busca identificar claramente las Superficies de Ataque existentes, para reducirlas y luego concentrarse en las vulnerabilidades que éstas puedan tener debido a las operaciones que las mismas realizan. Su visión es buscar la seguridad desde el punto de vista de la Seguridad Operacional.

La metodología OSSTMM fue desarrollada por la ISECOM (Institute for Security and Open Methologies) en Enero del año 2001. ISECOM es una comunidad abierta sin fines de lucro oficialmente registrada en Catalunia, España. ISECOM mantiene oficinas en Barcelona - España y Nueva York - USA (About Us).

La ISECOM también desarrolló el proyecto para colegios “Hacker Highschool" donde ofrece soporte con materiales y enseñanza sobre seguridad y privacidad en Internet para profesores de escuela primaria y secundaria (HackerHighSchool.org).

La ISECOM actualmente en este año 2012 se encuentra ofreciendo a través de su metodología OSSTMM, certificaciones, licencias y seminarios principalmente en Alemania, Italia, Canada y Chile . (Training, Isecom.org)

Entre las certificaciones que entrega la OSSTMM se encuentran:

- OPST - OSSTM Professional Security Tester
- OPSA - OSSTM Professional Security Analyst
- OWSE - OSSTM Wireless Security Expert
- CTA- OSSTMM Certified Trust Analyst
- OPSE - OSSTMM Professional Security Expert
- SAI - OSSTMM Certified Security Awareness Instructor

De entre algunos de los centros de certificación para la OSSTMM están actualmente los siguientes:

- KOENIG - INDIA/DUBAI
- ONECONSULT - AUSTRIA - ALEMANIA - SUIZA
- LAB106 - BENELUX
- DREAMLAB - FRANCIA - ALEMANIA - CHILE - SUIZA
- MEDIASERVICE.NET - ITALIA/GRECIA
- ADMERITIA - ALEMANIA
- ONECONSULT - ALEMANIA
- ISSECOM - ESPAÑA - USA

- LA SALLE UNIVERSITAT RAMON LLULL
- BARIKAT - TURKIA
- BELL - CANADA
- GCP GLOBAL - MEXICO

CAPÍTULO 4: Evaluación del Servidor a través de la OSSTMM3

En el actual capítulo se llega a evaluar los siguientes aspectos y simultáneamente se sugieren los correctivos a implementarse, formando un Plan de Seguridad.

Evaluación

- Estado Actual de las versiones de software de los principales componentes del Servidor Web.
- Análisis de Seguridad de la actual Superficie de Ataque del Servidor.
- Plan de Seguridad

Para medir la seguridad de los Blancos Objetivos, se utilizará como escala principal de medida, la Porosidad de la Superficie de Ataque, la cual es una medida propuesta por el OSSTMM3, desde el punto de vista de la Seguridad Operacional.

Definiciones del Ambiente

Canal

Recordemos que Canal es el medio en donde se da una interacción. Pueden ser: Humano, Físico, Inalámbrico, Telecomunicaciones Análogas o Digitales, Redes de Datos.

El Canal de evaluación en este caso sería, la transmisión de datos por Internet, hacia y desde el Servidor XpertSites.com.

Alcance

El Alcance está delimitado por el Canal a analizar, en este caso es la interacción que se da en la transmisión de datos que entra y sale al Servidor XpertSi-

tes.com por su tarjeta de red.

4.1 Análisis de Seguridad de las Versiones de Software de los Principales Componentes del Servidor.

Este análisis es importante ya que manteniendo actualizado todo el Software que usamos, podemos eliminar las vulnerabilidades que se han descubierto en versiones anteriores y que se han corregido en las nuevas versiones.

Linux Kernel:

A través del siguiente comando: *uname -a*

Linux Kernel 2.6.26-2-686

Distribución de Linux:

A travéz del siguiente comando: *head -n1 /etc/issue*

Debian GNU/Linux 5.0

A continuación se genera una tabla del resto de componentes de software importantes en el Servidor, información obtenida a través del Panel de Control VirtualMin, el cual es el software administrador del servidor.

Tabla 4.1 - Comparación de Versiones de Software		
Software	**Versión Instalada**	**Nueva Versión Disponible**
Apache WebServer	2.2.9	2.2.22
BIND DNS Server	9.5.1	9.8.1
Dovecot IMAP/POP3 Server	1.0	2
MySQL Database Server	5.0.	5.5

Postfix Mail Server	2.5	2.9
ProFTPD Server	1.31	3.2.8
SSH Server	5.1	5.9
PHP	5.2	5.3
Python	2.5	2.7
VirtualMin	1	1.6
Phpmyadmin	2.11.8	3.5.4
Django Web Framework	1.2	1.4.2
Wordpress	2.8	3.4.2

Situación de Riesgo Latente: Principales componentes de software del Servidor Web no se encuentran en sus últimas versiones.

Calificación Nivel de Riesgo: Alto.

Razón Calificación: un software desactualizado significa que éste ha tenido mucho mas tiempo para ser descubierto sus vulnerabilidades y éstas ya podrían utilizarse para un ataque, la última versión ofrecería correcciones a las posibles vulnerabilidades encontradas.

4.1.2 Posibles Soluciones

En esta situación se detecta dos alternativas.

Alternativa #1: actualizar desde el mismo servidor todos los paquetes a su última versión.

Esto a través del comando:

```
apt-get update
apt-get upgrade
```

```
apt-get dist-upgrade
```

Esta alternativa tendría el riesgo de si bien es cierto que se intentaría actualizar todo el software en lo posible, habrán librerías y componentes que quedan como antes, con el riesgo de no quedar compatibles con las nuevas actualizaciones. Mas que todo algunos archivos de configuración donde guardan en variables los valores para el comportamiento de los programas, podrían ya no coincidir con las variables necesitadas por los nuevas versiones actualizadas. Esto sucede principalmente en los archivos de configuración de los manejadores de contenido Wordpress, y el Web Framework Django, y las configuraciones específicas que se hayan realizado en los archivos de configuración de Apache y PHP.

Alternativa #2: consiste en contratar un nuevo servidor, donde se realizaría una “instalación en fresco”, instalando todo el software requerido en sus últimas versiones. Una vez funcionando el nuevo servidor, se realiza una migración paulatina de las aplicaciones y sitios web, del antiguo servidor al nuevo contratado. En el nuevo servidor web, se partiría con los nuevos archivos de configuración que vienen las aplicaciones y se actualizarían estos archivos con los valores de las variables de los anteriores archivos de configuración adaptándose a los nuevos requisitos. De esta manera se garantiza una migración mas fluida sin riesgo de interrupción de servicio por incompatibilidades de librerías de software como pudiera pasar en la Alternativa 1. Obteniendo así un ambiente de servidor con las últimas versiones de software y por lo tanto mayor Seguridad.

4.2 Análisis de Seguridad de la Actual Superficie de Ataque del Servidor XpertSites.com

El concepto de Seguridad Actual en la guía OSSTM3 define al Rav como la escala de medida de una Superficie de Ataque. En el Rav se pueden considerar 3 factores: Seguridad Operacional, Controles de Porosidades y Limitaciones de los Controles. (Herzog 79)

Al realizar la evaluación del Servidor XpertSites.com, nos hemos topado con un problema muy importante a resolver. Dentro del mismo servidor, operan varios Servicios los cuales son: Ambiente TCP/IP, Servicio DNS, Servicio de Correos, Servicio de Base de Datos, Servicio SSH, Servicio FTP.

Conforme a lo anterior el Servidor XpertSites.com a simple vista tiene una gran Superficie de Ataque, conformada por las superficies de ataque de los otros servicios que están corriendo dentro del mismo servidor, servicios que pueden tener sus propias vulnerabilidades cada uno.

Ante esta situación, se ha decidido resolver este problema a través del análisis de la Seguridad Operacional. Según la guía OSSTMM3, la Seguridad Operacional evalúa a una Superficie de Ataque a través de su nivel de Porosidad. Podemos representar estos conceptos con las siguientes fórmulas:

Porosidad = nivel de exposición de la Seguridad Operacional

Porosidad = Visibilidad(P_V) + Accesos(P_A) +puntos de Confianza(P_C)

Visibilidad = # de puntos descubribles

Accesos = # ingresos con autenticación por cada punto de interacción (# puertos donde interactúa un servicio)

Confianza = # accesos que conectan a otros accesos sin usar credenciales (# forwarding ports, ingreso o transmisión sin autenticación)

Entendiendo estos conceptos se procede primeramente a evaluar la Porosidad de la Superficie de Ataque del Servidor XpertSites.com analizando la Porosidad de cada una de las Superficies de Ataque de los Servicios que lo conforman.

4.3 Muestra Prototipo de Porosidades

En esta Tesis se buscará sacar una muestra prototipo de las Porosidades con fines demostrativos, ya que si bien es cierto realizando un análisis mucho más profundo de Vulnerabilidades se podrían encontrar un sin número de Porosidades , esto ya estaría fuera del alcance de esta Tesis que es meramente demostrar la lógica de implantar Seguridad en un Servidor Web como el de XpertSites.com utilizando muestras ejemplares que sirvan de prototipo, como las siguientes muestras de Porosidades que vamos a detectar.

A continuación el Servidor Web ofrece los siguientes Servicios, estos al momento de la inspección se encuentran funcionalmente disponibles, y se los ha identificado como Sub-Superficies de Ataque dentro de la principal Superficie de Ataque que es el servidor.

Superficie de Ataque #1: Ambiente TCP/IP

Situación de Riesgo Latente: se entiende como Ambiente TCP/IP las principales interacciones que se dan en este protocolo y que quedan expuestas al Internet, dentro de este ambiente están importantes servicios como el Servicio Web Apache y el Panel de Control VirtualMin.

Tabla 4.2 - Porosidad de la Sub-Superficie de Ataque #1: Ambiente TCP/IP		
Factores	Información obtenida	Conteo
Visibilidad	- puerto 80 abierto: Servicio Apache - puerto 1000 abierto: Panel de Control VirtualMin abierto. (Se recomienda usar otro puerto no común) - versión desactualizada de Apache - versión desactualizada de PHP - versión desactualizada de Python - versión desactualizada de Django Framework - posible vulnerabilidad de Denegación de Servicio (DoS)	7
Accesos	- login de usuario Wordpress - login de usuario Django - no existe un firewall activo	3
Confianza	- no existen porosidades	0
Total	Visibilidad + Accesos + Confianza	10

Superficie de Ataque #2: Servicio de Base de Datos

Situación de Riesgo Latente: la base de datos de los sitios y aplicaciones web se encuentra en el mismo servidor XpertSites.com.

Tabla 4.3 - Porosidad de la Sub-Superficie de Ataque #2: Servicio Base de Datos		
Factores	Información obtenida	Conteo
Visibilidad	- Puerto Mysql abierto - Versión desactualizada de Mysql	2
Accesos	- Apache se conecta internamente con el servicio de base de datos, usando un login y password.	1
Confianza	- no existen porosidades	0
Total	Visibilidad + Accesos + Confianza	3

Superficie de Ataque #3: Servicio DNS

Situación de Riesgo Latente: el Servicio DNS de los nombres de dominio de los sitios web hospedados se encuentra en el mismo Servidor XpertSites.com. Si un atacante logra ingresar al Servidor por alguna de sus superficies de ataque también podría tener acceso a la configuración DNS de los dominios. Esta situación aumenta otra razón más para y por donde ser atacado. Separando este ser-

vicio del servidor, permitiendo que el Servicio DNS de los dominios sea ofrecido por el servidor del mismo proveedor del dominio, reduciría el nivel de riesgo. Usualmente estos grandes proveedores de nombres de dominio, tienen una gran infraestructura de seguridad que se encarga de los ataques a su Servicio DNS.

Tabla 4.4 - Porosidad de la Sub-Superficie de Ataque #2: Servicio DNS		
Factores	Información obtenida	Conteo
Visibilidad	- Versión desactualizada de Bind DNS	1
Accesos	- no existen porosidades	0
Confianza	- Se intercambia libremente información DNS con servidores que lo requieren, sin autenticación de identidad, no tiene configurado DNSSEC.	1
Total	Visibilidad + Accesos + Confianza	2

Superficie de Ataque #4: Servicio de Correos

Situación de Riesgo Latente: unas pocas cuentas de correo se encuentran hospedadas dentro del mismo servidor XpertSites.com, otras se encuentran ya hospedadas en el servidor del proveedor del dominio. La interceptación de correos electrónicos para la obtención de sus contactos y acceso a información delicada como claves y datos críticos personales de los individuos naturalmente son una gran tentación para quien logre obtener esta información. Rompiendo las seguridades del Servicio de Correos, existe la probabilidad de que claves correspondientes a los accesos del mismo servidor se podrían encontrar en los mismos correos electrónicos.

Tabla 4.5 - Porosidad de la Sub-Superficie de Ataque: Servicio de Correos		
Factores	Información obtenida	Conteo
Visibilidad	-Puerto IMAP abierto -Puerto POP3 abierto -Versión desactualizada de Postfix -Versión desactualizada de Dovecot IMAP/POP3	4
Accesos	- usuarios usan login y password para acceder. No se utiliza encriptación en la comunicación. No existe control para ataque de fuerza bruta.	1
Confianza	- no tiene porosidades	0
Total	Visibilidad + Accesos + Confianza	5

Superficie de Ataque #5: Servicio FTP

Situación de Riesgo Latente: el servicio es utilizado únicamente por el administrador del servidor, de forma muy esporádica ya que principalmente se usa SSH. Este Servicio FTP no usa un canal de transmisión encriptado, en algún momento la transmisión de usuario y contraseña pueden ser capturadas. Aparte algún atacante puede intentar un ataque de fuerza bruta para adivinar el usuario y clave de ingreso de alguna de las cuentas del servidor a través de este servicio.

Tabla 4.6 - Porosidad de la Sub-Superficie de Ataque: FTP		
Factores	Información obtenida	Conteo
Visibilidad	- Versión desactualizada de ProFTP - Puerto FTP abierto	2
Accesos	- Usuarios usan login y password para acceder. No se utiliza encriptación en la comunicación. No existe control para ataque de fuerza bruta.	1
Confianza	- no existen porosidades.	0
Total	Visibilidad + Accesos + Confianza	3

Superficie de Ataque #6: Servicio SSH

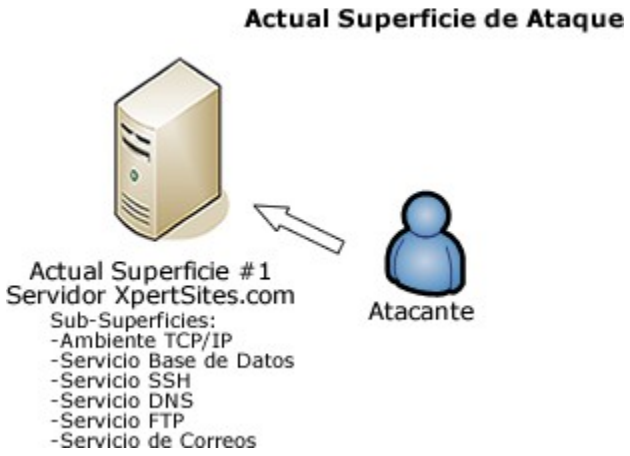
Situación de Riesgo Latente: el servicio es utilizado principalmente por el administrador del servidor, cuando existe algún usuario que lo requiera se lo activa temporalmente. El servicio requiere para su acceso un usuario y una clave, puede ser vulnerable a algún intento de ataque de fuerza bruta para adivinar las credenciales de ingreso.

Tabla 4.7 - Porosidad de la Sub-Superficie de Ataque: SSH		
Factores	Información obtenida	Conteo
Visibilidad	- Versión desactualizada de OpenSSH - Puerto SSH abierto	2
Accesos	- Acceso SSH usa login y password para ingresar. No existe control para ataque de fuerza bruta.	1
Confianza	- no existen porosidades.	0
Total	Visibilidad + Accesos + Confianza	3

4.3.1 Comprendiendo la Porosidad de las Actuales Superficies de Ataque

Se ha determinado que la Superficie de Ataque total del servidor XpertSites.com está compuesta por la sumatoria de distintas Superficies de Ataque que se encuentran en el mismo servidor, conformadas por los distintos “assets” o componentes principales que permiten ofrecer sus servicios al servidor XpertSites.com, tales como el Ambiente TCP/IP, Base de Datos, DNS, Email, FTP, SSH, tal como lo muestra la figura a continuación.

Figura 4.1: Actual Superficie de Ataque



El concepto de Seguridad Operacional (OpSec), buscaría la seguridad disminuyendo la superficie de Ataque del Servidor XpertSites.com, es decir disminuir su nivel de Porosidad. Esto se lograría disminuyendo la Superficie de Ataque Total del servidor, separándolo en lo posible de sus diferentes Superficies de Ataque que lo conforman y que hacen que tenga una mayor Superficie de Ataque Total, se entiende así que esta relación es directamente proporcional.

Representando este concepto en una ecuación sería:

Actual Superficie de Ataque Servidor XpertSites.com = ∑ Superficie de Ataque Individual de cada Servicio dentro del mismo Servidor.

Actual Superficie de Ataque Servidor XpertSites.com = Porosidad Superficie Ambiente TCP/IP + Porosidad Superficie Servicio de Base de Datos + Porosidad Superficie Servicio DNS + Porosidad Superficie Servicio de Correos + Porosidad Superficie Servicio FTP + Porosidad Superficie Servicio SSH.

Tabla 4.8 - Porosidad de la Actual Superficie de Ataque Total: Servidor XpertSites.com	
Sub-Superficies de Ataque	Porosidad
Ambiente TCP/IP	10
Servicio Base de Datos	3
Servicio DNS	2
Servicio de Correos	5
Servicio FTP	3
Servicio SSH	3
Total	26

OpSec (Seguridad Operacional)

$OpSec_{suma}$ = total de Porosidad de la Superficie de Ataque en el servidor XpertSites.com

$OpSec_{suma}$ = 26 puntos de Porosidad

Según la medida Rav de Seguridad en la guía OSSTMM3, la Seguridad en Equilibrio está representada por el balance entre la Porosidad existente, los Controles para disminuir la Porosidad y las Limitaciones de dichos Controles.

Este concepto se representa en la siguiente fórmula:

Rav de Seguridad en Equilibrio = Controles – Limitaciones de los Controles – Porosidad

“El Rav básicamente es restar de los Controles, las Limitaciones de dichos Controles y la Porosidad de la Superficie de Ataque” (Herzog 67).

Comprendiendo esto, lo anterior quiere decir que la existencia de los Controles, depende de la existencia de la Porosidad en su intento de disminuirla y compensar así la Seguridad. Tiene bastante sentido, viéndolo mas claro en el siguiente ejemplo de la vida cotidiana: si tengo varias puertas y ventanas en una casa que representan la Porosidad, eso genera la existencia de implementar Controles para las respectivas puertas y ventanas, en nuestra búsqueda de equilibrio en la Seguridad.

Si bien es cierto que para el cálculo de la fórmula de Rav Total de una Superficie de Ataque en la OSSTMM3 comprende también los factores de Controles y sus Limitaciones, estos Controles dependen de la existencia de la Porosidad. Al momento vamos a plantear en el Plan de Seguridad la posibilidad de eliminar algunos de los elementos de Porosidad para brindar una mejor Seguridad, lo cual hace que ya no tenga sentido seguir evaluando en el servidor los Controles y Limitaciones de Porosidades que ya no van a existir.

Lo sorprendente de este procedimiento es que para eliminar la Porosidad, no necesitamos indagar necesariamente o conocer a ciencia cierta una cantidad de Controles y Limitaciones que se puedan tener implantadas, o las últimas Vulnerabilidades que se han descubierto en este ámbito, esto es lo que hace que el análisis de la Seguridad desde el punto de vista de la Seguridad Operacional (OpSEC) ahorre bastante tiempo en el análisis de Seguridad, haciéndolo muy efectivo al descubrir el camino hacia una mejor Seguridad de una forma más directa y práctica, sin enredarse en antiguos Controles o Limitaciones mal o bien implantadas.

El anterior análisis se ve corroborado con lo que menciona Pete Herzog Presidente de la ISECOM y creador de la guía OSSTMM3:

> Es posible de tomar un atajo en el análisis de Seguridad y todavía tener un Rav representativo, si realmente no es de importancia el margen de error debido a que solo se desea realizar una comparación, usted puede calcular solamente la Porosidad, que significa contar los assets visibles y accesibles, luego implementar los controles respectivos (Herzog 68).

De esta manera nos centraremos en analizar el resultado que hemos obtenido de la Porosidad de la Superficie de Ataque en cuestión, mejor dicho su Seguridad Operacional.

4.4 Posible Solución de Seguridad

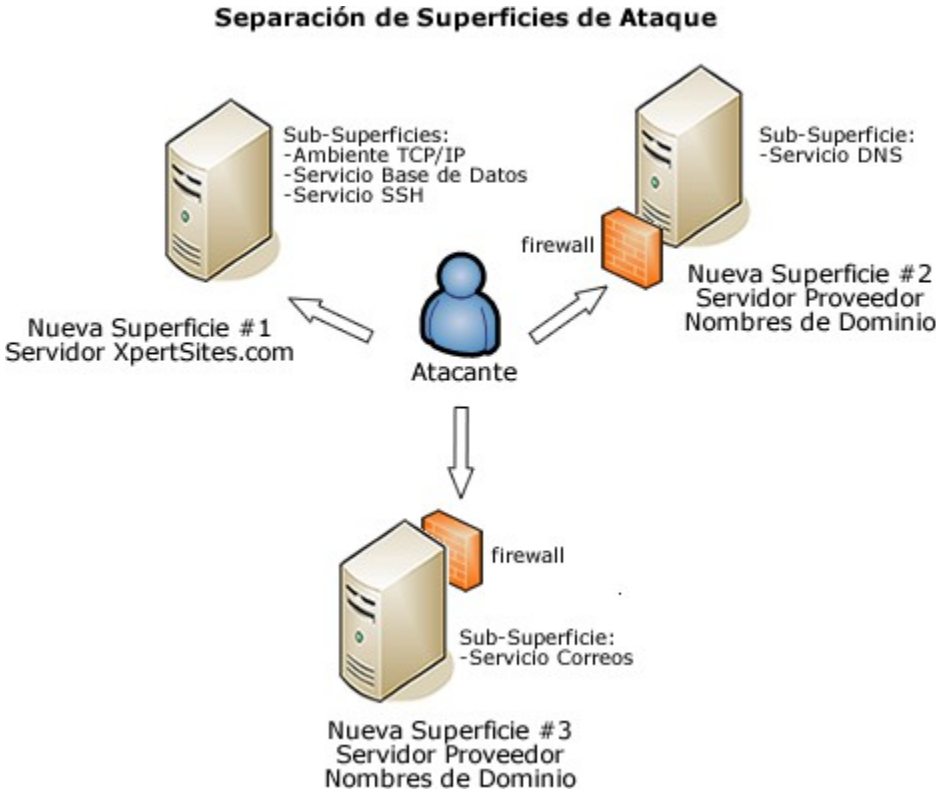
Estrategia #1 : reducción de la Superficie de Ataque Total

Como se puede observar se ha obtenido una Porosidad de 26 puntos para el Servidor XpertSites.com. Según el concepto de Seguridad Operacional, mientras más se pueda reducir el Área de la Superficie de Ataque más se reducirá la oportunidad de éxito del atacante.

Siguiendo el anterior concepto, veamos que sucedería si a la Superficie de Ataque del Servidor XpertSites.com le quitamos las Superficies de Ataque del Servicio DNS y el Servicio de Correos delegándolas al proveedor del nombre de dominio. También le quitamos la superficie del Servicio FTP, ya que se puede realizar la misma operación de transmisión de archivos al servidor a través del Servicio SSH. Es decir dejamos únicamente la Superficie de Ataque del Ambiente TCP/IP, la de la Base de Datos y del Servicio SSH, que básicamente representan la publicación y acceso a los archivos de las aplicaciones y sitios web con sus bases de datos respectivas.

Se puede observar esta estrategia #1 en la siguiente figura:

Figura 4.2: Separación de Superficies de Ataque sin Controles en Porosidades



Este planteamiento quedaría representado de la siguiente manera:

Tabla 4.9 - Nueva Superficie de Ataque Servidor XpertSites.com			
Superficie de Ataque	Sub Superficies	Porosidad	Ubicación
Nueva Superficie de Ataque #1	Ambiente TCP/IP + Base de Datos + SSH	10 + 3 + 3 = 16	Servidor XpertSites.com
Nueva Superficie de Ataque #2	Servicio DNS	n/a = fuera del Alcance	Servidor Proveedor de Nombres de Dominio
Nueva Superficie de Ataque #3	Servicio de Correos	n/a = fuera del Alcance	Servidor Proveedor de Nombres de Dominio

Porosidad en Nueva Superficie de Ataque #1 = 16 puntos.

Podemos ver en esta Tabla que la nueva superficie de Ataque del Servidor XpertSites.com quedaría reducida como mínimo a una Porosidad de 16 puntos. Es decir que este valor puede reducirse aún más si planteamos los Controles a

las Porosidades que van quedando, así iremos encontrando la Seguridad en Equilibrio, basándonos en los conceptos de la OSSTMM3.

Ahora necesitamos conocer la Ganancia en Seguridad Operacional obtenida. Para esto representemos a este concepto con las siglas GSO.

Ganancia en Seguridad Operacional = GSO

$$\%\ \text{mínimo de GSO} = \frac{(PorosidadActual - NuevaPorosidad)\,x100}{PorosidadActual}$$

$$\%\ \text{mínimo de GSO} = \frac{26-16}{26}\,x100 = 38.46\ \%\ \text{de ganancia en seguridad.}$$

Se puede observar que aplicando la Estrategia #1 se puede obtener un porcentaje mínimo de Ganancia en Seguridad Operacional del 38.46%

Las Nuevas Superficies de Ataque #2 y #3 al plantear su separación del Servidor XpertSites.com quedarían delegadas su seguridad al Datacenter de la compañía proveedora de los Nombres de Dominio.

Es así como de ésta manera estas Superficies de Ataque ya no quedarían dentro de la Superficie de Ataque del Servidor XpertSites.com. Ya no quedaría dentro de su responsabilidad, es decir estarían fuera de su Alcance, debido a que serían delegadas a la compañía proveedora de los Nombres de Dominio. De esta manera también se reduce el Riesgo en Seguridad del servidor XpertSites.com, al reducir su Superficie de Ataque. Esto nos permitiría concentrarnos en reducir al máximo la Porosidad de la Nueva Superfice de Ataque # 1, que es donde se encuentra la información vital a proteger.

A continuación la siguiente Tabla muestra en detalle como quedaría la Porosidad de la Nueva Superficie de Ataque #1.

Tabla 4.10 - Porosidad de la Nueva Superficie de Ataque #1 : Servidor XpertSites.com		
Factor: Visibilidad		
Sub Superficies	Descripción	P
Ambiente TCP/IP	- puerto 80 abierto: Servicio Apache - puerto 1000 abierto: Panel de Control VirtualMin abierto. (Se recomienda usar otro puerto no común) - versión desactualizada de Apache - versión desactualizada de PHP - versión desactualizada de Python - versión desactualizada de Django Framework - posible vulnerabilidad de Denegación de Servicio (DoS)	7
Servicio Base de Datos	- Puerto Mysql abierto - Versión desactualizada de Mysql	2
Servicio SSH	- Puerto SSH abierto - Versión desactualizada de OpenSSH	2
Factor: Acceso		
Ambiente TCP/IP	- login de usuario Wordpress - login de usuario Django - no existe un Firewall activo	3
Servicio Base de Datos	- Apache se conecta internamente con el servicio de base de datos, usando un login y password.	1
Servicio SSH	- acceso SSH usa login y password para ingresar. No existe control para ataque de fuerza bruta.	1
Factor: Confianza		
Ambiente TCP/IP	- no existen porosidades	0
Servicio Base de Datos	- no existen porosidades	0
Servicio SSH	- no existen porosidades	0
Porosidad Total		**16**

Estrategia #2: implementación de Controles en las Porosidades de la Nueva Superficie de Ataque #1. Una vez identificada la Nueva Superficie de Ataque del Servidor Xpertsites.com como lo es la Superficie de Ataque #1, podemos proceder a implementar Controles para su Porosidad.

4.5 Muestra Prototipo de Porosidades detectadas en la Nueva Superficie de Ataque #1 y sus Controles de Seguridad a Implementarse.

A continuación se describen las Porosidades detectadas, están organizadas por Sub Superficies de Ataque. El detalle de como finalmente se implementaron los Controles se encuentra en la sección de Implementación del Plan de Seguridad.

Donde P = Porosidad

Sub Superficie Ambiente TCP/IP

P1: Puerto 80 abierto: Servicio Apache

Situación de Control: la simple existencia de este servicio al Internet lo hace vulnerable, haciendo que exista permanentemente esta Porosidad. Los Controles que se implementen a las Porosidades relacionadas con este Servicio ayudarán a mitigar su vulnerabilidad.

P2: Puerto 1000 abierto: Panel de Control VirtualMin abierto.

Situación de Control: se propone cambiar el número de puerto a un valor no tan común para este Servicio. Se controla número de intentos de autenticación de usuario.

P3, P4, P5, P6: Versión desactualizada de Apache, PHP, Python, Django Framework

Situación de Control: manteniendo la última versión del software disminuye su riesgo, haciendo inoperables la variedad de vulnerabilidades encontradas anteriormente por los usuarios en Internet.

P7: Posible vulnerabilidad de Denegación de Servicio (DoS)

Situación de Control: implementar el módulo mod_evasive de Apache. Cambiar comportamiento de TCP en archivo de configuración del Kernel. Investigar que reglas en el firewall de Linux ayudan a disminuir el riesgo de este ataque.

P8: Login de usuario Wordpress

Situación de Control: implementar Captcha para evitar intentos automatizados de autenticación de usuario que busquen adivinar la clave (ataques de fuerza bruta). Redireccionar automáticamente url de autenticación de usuario, a una conección SSL (de http a https) para evitar interpretación de la información sobre usuario y claves enviadas a través del Canal en el caso de una interceptación de la comunicación..

P9: Login de usuario Django

Situación de Control: Redireccionar url de autenticación de usuario, a una conección SSL, de Http a Https, para evitar interpretación de la información sobre usuario y claves enviadas a través del Canal en el caso de una intercepción de la

comunicación.

P10: No existe un Firewall activo

Situación de Control: implementar un Firewall solo con los puertos permitidos, dejando sin efecto si un intruso levanta un puerto desde otro usuario que no sea root. Aparte el firewall puede servir para detener ataques del tipo Denial of Service y cualquier tráfico que sea configurable en IPTables para su bloqueo.

Sub Superficie Servicio Base de Datos

P11: Puerto Mysql abierto

Situación de Control: la simple existencia de este servicio al Internet lo hace vulnerable, haciendo que exista permanentemente esta Porosidad. Los Controles que se implementen a las Porosidades (P12, P13) relacionadas con este Servicio ayudarán a mitigar su vulnerabilidad.

P12: Versión desactualizada de Mysql

Situación de Control: manteniendo la última versión del software disminuye su riesgo, haciendo inoperables una gran mayoría de vulnerabilidades encontradas anteriormente por los usuarios en Internet. Muchos de los ataques de “SQL injection” funcionan en versiones desactualizadas del software que todavían no han corregido ese tipo de vulnerabilidades encontradas.

P13: Apache se conecta internamente con el servicio de base de datos, usando un login y password.

Situación de Control: revisar privilegios y cuentas de usuario Mysql. Revisar que no exista permiso para el ingreso de usuarios "guest" o similares. Revisar que Mysql no esté siendo ejecutado como usuario "root". En cuanto a los ataques de “SQL injection” muchos de ellos necesitan que Mysql esté funcionando como usuario root o con privilegios similares para poder realizar sus operaciones.

Sub Superficie Servicio SSH

P14: Puerto SSH abierto

Situación de Control: la simple existencia de este servicio al Internet lo hace vulnerable. El Control que se implemente en P16 hará que su Porosidad desaparezca y el Control P15 ayudará a mitigar riesgos por uso de una versión desactualizada.

P15: Versión desactualizada de OpenSSH

Situación de Control: manteniendo la última versión del software disminuye su riesgo, haciendo inoperables una gran mayoría de vulnerabilidades encontradas anteriormente por los usuarios en Internet.

P16: Usuarios usan login y password para acceder a SSH. No existe control para ataque de fuerza bruta.

Situación de Control: se recomienda desactivar la autenticación de usuarios cortando así cualquier ataque de fuerza bruta contra este servicio, implementando el acceso a SSH únicamente a través del uso del par de llaves pública y privada con encriptación RSA que debe tener solo el administrador del servidor en

su computador de acceso al sistema.

4.5.1 Simulación de la Implementación de Controles en la Nueva Superficie de Ataque.

Tomando en cuenta que se aplicaría la Alternativa #2 de la sección 4.1.2 que consiste en la actualización de todo el software utilizado a través de la contratación de un nuevo servidor e implementamos la Estrategia #1 y Estrategia #2 de la presente sección, tendríamos los resultados de Porosidad como muestra la siguiente simulación de la tabla a continuación:.

Tabla 4.11 - Resultado de Porosidades Después de la Simulación de Controles Implementados en nueva Superficie de Ataque #1

Factor: Visibilidad				
Sub superficies	Antes	P	Después	P
Ambiente TCP/IP	- puerto 80 abierto: Servicio Apache - puerto 1000 abierto: Panel de Control VirtualMin abierto. (Se recomienda usar otro puerto no común) - versión desactualizada de Apache - versión desactualizada de PHP - versión desactualizada de Python - versión desactualizada de Django Framework - posible vulnerabilidad de Denegación de Servicio (DoS)	7	- puerto 80 abierto: Servicio Apache - se realiza cambio en el # del puerto, se limita # de intentos de autenticación de usuario - versión actualizada - versión actualizada - versión actualizada - versión actualizada - versión actualizada - implementación de controles (DoS) en Apache, Linux Kernel, Linux Firewall.	1
Servicio Base de Datos	- Puerto Mysql abierto - Versión desactualizada de Mysql	2	- Puerto Mysql abierto - versión actualizada	1
Servicio SSH	- Puerto SSH abierto - Versión desactualizada de OpenSSH	2	- Puerto SSH accesible únicamente al administrador a través de llaves de encriptación. - versión actualizada	0
Factor: Acceso				
Ambiente TCP/IP	- login de usuario Wordpress - login de usuario Django - no existe un Firewall activo	3	- se instala captcha y se redirecciona a url con SSL - se redirecciona a url con SSL - se activa un firewall solo con los	0

			puertos necesarios.	
Servicio Base de Datos	- Apache se conecta internamente con el servicio de base de datos, usando un login y password.	1	- se controla privilegios de usuario y acceso remoto a mysql	0
Servicio SSH	- acceso SSH usa login y password para ingresar. No existe control para ataque de fuerza bruta.	0	- se habilita ingreso únicamente a través de llaves de encriptación.	0
Factor: Confianza				
Ambiente TCP/IP	- no existen porosidades.	0	- no existen porosidades.	0
Servicio Base de Datos	- no existen porosidadas.	0	- no existen porosidades.	0
Servicio SSH	- no existen porosidades.	1	- no existen porosidades.	0
Total Porosidad		**16**		**2**

Se puede observar en la anterior tabla, que se ha llegado a obtener una Porosidad de 2 puntos. Esto debido a que los servicios Apache y Mysql al estar activos siempre estarán expuestos a nuevas vulnerabilidades descubiertas en la comunidad del Internet, por lo tanto la Porosidad mínima de la Superficie de Ataque #1 va a ser siempre de 2 puntos.

Porosidad Estrategia #1 = Visibilidad + Acceso + Confianza

Porosidad Estrategia #1 = 2 + 0 + 0 = 2

Ganancia en Seguridad Operacional = GSO

$$\%\ \text{mínimo de GSO} = \frac{(PorosidadActual - NuevaPorosidad)\,x100}{PorosidadActual}$$

$$\%\ \text{mínimo de GSO} = \frac{26-2}{26} = 92.3\ \%$$

Se puede observar que aplicando la Estrategia #2 se puede obtener un porcentaje mínimo de Ganancia en Seguridad Operacional del 92.3%.

4.6 Plan de Seguridad Resultante

Observando los resultados de la simulación anterior, a continuación se sugiere la implementación de las siguientes soluciones planteadas para el Plan de Seguridad.

- Se sugiere implementar la Alternativa #2 según sección 4.1.2 que consiste en contratar un nuevo servidor, donde se realizaría una “instalación en fresco”, instalando todo el software requerido en sus últimas versiones.
- Se sugiere implementar Estrategia #1 y #2 según sección 4.4 que consiste en reducir la Superficie de Ataque Total del Servidor XpertSites.com, conformando una nueva Superficie de Ataque mas reducida, a la cual a su vez se sugiere implementar Controles para la muestra de Porosidades resultante de la conformación de esta nueva Superficie de Ataque #1.

4.7 Implementación del Plan de Seguridad

El primer paso de la implementación del Plan de Seguridad, es la creación de la nueva Superficie de Ataque: Servidor XpertSites.com. Para esto necesitamos primero crear el ambiente de la nueva Superficie de Ataque, contratando un nuevo Servidor. Luego se procede a implementar los respectivos Controles a las Porosidades.

4.7.1 Colocación de la Nueva Superficie de Ataque: servidor XpertSites.com

Al observar que las versiones de software de los principales componentes del Servidor se encuentran bastante desactualizadas, implicando esto un alto riesgo de seguridad por las posibles vulnerabilidades ya detectadas para esas

versiones, se determina que la mejor opción sería contratar un nuevo servidor, en el cual se instala todo el software requerido en sus últimas versiones.

A continuación las características del servidor que se buscaba:

Tabla 4.12 - Características Requeridas para el Nuevo Servidor	
Características	**Requerimientos**
Memoria Ram	mayor o igual a 1gb RAM
Disco Duro	mayor o igual a 40 gigas
Sistema Operativo	versión de Debian Linux o derivada
Procesador	preferiblemente de 2 o más procesadores.
Sistema de respaldo de datos	que ofrezca una variedad flexible en la capacidad de almacenamiento para los respaldos.
Capacidad de expansión	que exista la opción de aumentar las capacidades del Servidor sin tener que realizar complejas migraciones.

Para esto se hizo una búsqueda de opciones de servidores Linux.

Opciones de ubicación geográfica para el nuevo Servidor

Al momento de contratar el nuevo Servidor con el Proveedor elegido se nos daba la opción de escoger su ubicación geográfica.

A continuación los puntos donde el Proveedor ofrece Datacenters:

Tabla 4.13 - Localización de los Datacenters	
Ubicación del Data Center	**Dirección IP**
Newark - New Jersey	50.116.57.237
Dallas - Texas	50.116.25.154
Fremont - California	50.116.14.9
Atlanta - Georgia	50.116.39.117
Tokyo - Japón	106.187.96.148
Londres - Inglaterra	176.58.107.39

Factores determinantes

Se ha considerado analizar las siguientes relaciones como factores determinantes para seleccionar la ubicación geográfica del nuevo servidor.

- Factor 1: relación entre la ubicación geográfica y el público a quien está principalmente dirigido el contenido hospedado en el Servidor: principalmente las aplicaciones y contenido del Servidor es orientado al mercado ecuatoriano referentes a los Bienes Raíces en el país. Esta relación se hace debido a que mientras mas cerca esté ubicado el Servidor al público que estamos orientados, tendrán que esperar un menor tiempo de respuesta en la descarga del contenido almacenado en el Servidor.

Método de medición del Factor

Evaluamos identificando el número de saltos (hops) que se requiere para llegar al Servidor Web desde la ciudad de Quito así como el tiempo de latencia de los paquetes enviados hacia el servidor.

- Factor 2: relación entre la ubicación geográfica y la actividad principal que se realiza en el Servidor: al ser un servidor principalmente orientado al desarrollo de aplicaciones para el Internet, donde constantemente se está haciendo modificaciones y pruebas en vivo a las aplicaciones, se requiere una conexión remota frecuente desde el Ecuador hacia el Servidor que menor tiempo de acceso tenga.

Método de medición del Factor

El método de medición es el mismo que el del Factor 1

- Factor 3: relación entre la ubicación geográfica y la identificación de una

zona física menos propensa a desastres naturales: es decir se tratará de no elegir un lugar donde se esté muy cerca a zonas costeras, fallas geológicas o situaciones climáticas adversas.

Evaluación del Factor

Para conocer cuantos saltos entre puntos de conexión del Internet necesitamos para llegar al Servidor, utilizamos en Linux el siguiente comando desde la ciudad de Quito:

traceroute NombredelServidorDestino

Para conocer la latencia, es decir el tiempo que demora en ir y venir un paquete de datos desde Quito hacia el data center, se ha envíado 50 paquetes a cada data center, usando el siguiente comando:

ping -c 50 NombredelServidorDestino

Con la finalidad de ver si es que la hora en que se realizaba las pruebas podría darnos datos no fidedignos debido a algún congestionamiento de tráfico de datos, se decidió realizar tres pruebas en tres días diferentes a horas distintas y de estos valores sacar un promedio, como muestra la tabla a continuación.

Tabla 4.14 - # de Saltos y Latencia

Ubicación del Data Center	# de saltos (promedio)	Latencia (promedio) milisegundos (ms)	IP
Newark - New Jersey	(22+18+18)/3 = 19	(127+118+119)/3 = 121	50.116.57.237
Dallas - Texas	(19+18+18)/3 = 18	(134+148+148)/3 = 143	50.116.25.154
Fremont - California	(17+16+16)/3 = 16	(193+191+192)/3 = 192	50.116.14.9
Atlanta - Georgia	(16+16+16)/3 = 16	(106+118+120)/3 = 115	50.116.39.117
Tokyo - Japón	(16+16+16)/3 = 16	(275+275+276)/3 = 275	106.187.96.148
Londres - Inglaterra	(14+16+14)/3 = 15	(191+189+194)/3 = 191	176.58.107.39

Análisis de resultados:

El data center que obtiene el menor número de saltos es el de Londres con 15 saltos.

El data center que obtiene el menor tiempo de respuesta es el de Atlanta con 115 ms.

El data center de Japón y Fremont tienen igual número de saltos que el de Atlanta, pero su tiempo de respuesta es mas alto que el de Atlanta.

Se puede observar que si bien para llegar al data center en Londres se tiene el menor número de saltos (15 hops), su tiempo de respuesta necesariamente no llega a ser el menor (191ms).

En el análisis anterior se puede ver que el data center de Atlanta cumple con los requerimientos de los Factores 1 y 2.

Para el Factor 3, en cuanto a la zona física de menor riesgo a desastres naturales, se puede ver que si bien es cierto que en Atlanta se han registrado tornados, otros factores como la incidencia de terremotos es bastante baja, peor aún la posibilidad de inundación, esto según los datos encontrados en el portal USA.-COM (Atlanta Natural Disasters).

Por lo tanto el Data Center de Atlanta al cumplir con los tres factores, éste es el lugar escogido para colocar el nuevo servidor.

Tabla 4.15 - Saltos y Latencia: Anterior Servidor vs Nuevo Servidor

Ubicación del Data Center	# de saltos	Latencia	IP
Anterior Servidor (Panama)	16	196 ms	190.120.226.249
Nuevo Servidor (Atlanta)	16	106 ms	198.74.53.138

Veamos el porcentaje de ganancia en latencia del nuevo servidor en relación al anterior.

% ganancia en latencia =

$$= \frac{(LatenciaPaquetesServidorAnterior - LatenciaPaquetesNuevoServidor)\,x100}{LatenciaPaquetesServidorAnterior}$$

$$\% \text{ ganancia en latencia} = \frac{196-106}{196}\,x100 = 46\ \%$$

Se puede ver que la latencia de los paquetes de datos que se transmiten desde Quito hacia el nuevo servidor ha mejorado en un 46% más en relación con el anterior servidor.

4.7.2 Tipo de Distribución del Sistema Operativo y su relación con la Seguridad.

El servidor ha estado funcionando con la distribución de Debian 5. Actualmente ésta distribución ya no tiene soporte, es decir ya no se liberan nuevas actualizaciones con parches de seguridad para esta distribución, lo cual implica un alto riesgo de Seguridad. Como se ha venido trabajando con Debian y se tiene ya una gran familiaridad y confianza con este sistema operativo, se desea buscar una distribución basada en Debian.

Al momento de esta Tesis está vigente la versión Debian 6.0 Se plantea como otra opción la instalación de una distribución alternativa basada en Debian, que en este caso sería Ubuntu 12.10 LTS.

La versión Ubuntu LTS ofrece un soporte y paquetes de actualización de las seguridades del sistema operativo por más años que la de Debian.

Otra característica interesante es, que para que Debian apruebe que un programa deba considerarse “estable” y por lo tanto pueda ser puesto disponible

para toda la comunidad en su repositorio de “archivos estables”, deberán de pasar alrededor de unos 2 años. A diferencia de Ubuntu que publica en sus repositorios estables las versiones últimas de un programa cada 6 meses.

Para algunos esto será una ventaja por la rápida disponibilidad del software y para otros lo entenderán como una desventaja al pensar que el software todavía no ha sido bien probado. En todo caso me atrevería a pensar que un software se lo determina como bien probado por el número de personas que lo han usado, antes que por el tiempo que ha pasado desde que se lo hizo disponible.

También se puede observar que en Internet hay más gente interesada en buscar información sobre Ubuntu que Debian, de esta manera podríamos decir que la comunidad alrededor de Ubuntu está creciendo mas aceleradamente que la de Debian, según las estadísticas de búsqueda en Google Trends (Web Search Interest: ubuntu, debian).

Para nuestro caso particular de XpertSites.com, se desea trabajar en lo mas posible con las últimas versiones estables de los herramientas de software para su pronta integración en el proceso de prueba y desarrollo de las aplicaciones web y también contar en el menor tiempo posible con las actualizaciones de seguridad en el software. También nos interesa que exista una mayor comunidad que utilice este Sistema Operativo, generando naturalmente más documentación, soporte y reporte de vulnerabilidades.

Por todas estas razones se ha decidido instalar Ubuntu LTS 12.10 en vez de Debian 6.0.

4.7.3 Implementación de Controles a las Porosidades

A continuación se describen las implementaciones realizadas, están organizadas por Sub Superficies de Ataque.

Sub Superficie Ambiente TCP/IP

P1: Puerto 80 abierto: Servicio Apache

Se han implementado controles a esta porosidad, al cubrir las porosidades en P3, P7 y P10. Pero se decide dejar esta porosidad marcada como abierta, ya el puerto de este Servicio queda siempre abierto al público, además que la posibilidad de que en algún momento se descubran nuevas vulnerabilidades para Apache queda siempre latente. Hay que tomar en cuenta que éste es un servicio el cual no se lo puede apagar, ya que es el servicio que publica las páginas del servidor en Internet.

P2: puerto 10000 abierto - VirtualMin Panel de Control del Servidor

El panel de control de Webmin se encuentra dentro del software VirtualMin, por lo que se usará este último nombre. Se procede a instalar la última versión del panel de control VirtualMin para cubrir los últimas brechas de seguridad encontradas. En cuanto al puerto de acceso se lo configura para que escuche en otro puerto no común. Aparte se configura un límite de intentos de ingreso al Panel de Control cuando el usuario no ingresa la clave o usuario correcto.

Para poder operar el servidor y activar el resto de sus Servicios se procede a instalar el Panel de Control VirtualMin realizando las siguientes operaciones.

Comprobamos que nombre del servidor esté correctamente configurado:

hostname -f

Descargamos el instalador de Virtualmin y luego lo ejectuamos:

wget http://software.virtualmin.com/gpl/scripts/install.sh

chmod 755 install.sh

./install.sh

Virtualmin instalará automáticamente los componentes y servicios típicos del Servidor para su funcionamiento en Internet como Apache, Mysql, PHP, Pro-FTP. Luego desde el Panel de Control, se procede a activar solo los servicios que se han planificado, según el Plan de Seguridad establecido.

En Virtualmin se configuró claves de encriptación “hash" en vez de claves de texto aumentando la seguridad.

Se realiza el cambio de puerto de VirtualMin a un puerto popularmente no conocido, para mayor seguridad. Esto a través de la modificación del archivo de configuración en:

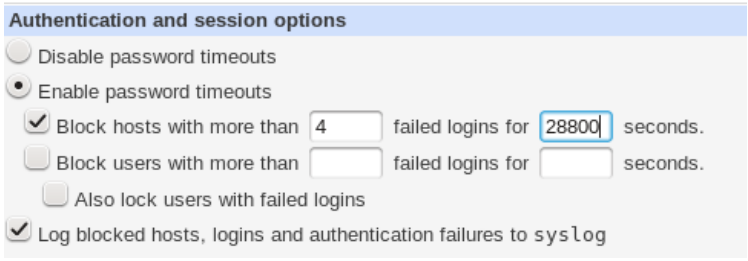
/etc/VirtualMin/miniserv.conf

Se ha configurado a Virtualmin para que asuma las IPtables que están funcionando actualmente en el sistema, y luego puedan ser administradas desde este panel de control.

Así mismo se ha configurado en el panel de control, que en caso de ser reiniciado el servidor, el firewall también sea activado al arrancar el sistema.

En cuanto al ingreso al panel de control de VirtualMin, se limita el número de intentos de ingreso al Panel de Control a 4 intentos, cuando el usuario no ingresa la clave o usuario correcto, bloqueando su dirección ip origen, por 8 horas.

Figura 4.3: Control de intentos de ingreso a VirtualMin



P3: versión desactualizada de Apache

El Control de esta porosidad quedó implementado al instalar el software de Apache actualizado al realizar el procedimiento en P2.

P4: versión desactualizada PHP

El Control de esta porosidad quedó implementado al instalar el software PHP actualizado al realizar el procedimiento en P2.

P5: versión desactualizada de Python

El Control de esta porosidad quedó implementado al instalar Python actualizado al momento en que se contrató e instaló el nuevo servidor.

P6: versión desactualizada de Django Framework

Empezamos instalando el instalador pip desde el shell de Linux:

curl https://raw.github.com/pypa/pip/master/contrib/get-pip.py | python

Luego instalamos última versión de Django con el siguiente comando: *pip install django*

P7: posible vulnerabilidad de Denegación de Servicio en Apache Web Service

Se procede a implementar una combinación de controles en el firewall IPta-

bles, configuración del Kernel y configuración de Apache para disminuir los riesgos de un ataque de denegación de servicio. Estas implementaciones son el producto de investigar cuales son los Controles más recomendados para este tipo de Porosidades y que a su vez se aplican a nuestro servidor.

IPTables

En IPtables se procede a realizar la siguiente implementación:

Se crea una nueva cadena llamada “syn-flood”: *iptables -N syn-flood*

Limitamos las peticiones por segundo: *iptables -A syn-flood -m limit --limit 10/second --limit-burst 50 -j RETURN*

Graba ataques en el log del sistema: *iptables -A syn-flood -j LOG --log-prefix "SYN flood: "*

iptables -A syn-flood -j DROP

Grabamos los cambios en el archivo de reglas del firewall: *sh -c "iptables-save > /etc/iptables.up.rules"*

Linux Kernel

En cuanto a la configuración del Kernel de Linux se investigó que la implementación de “syncookies" es un control bastante importane para prevenir los ataques DoS. "SYN cookies es un elemento clave de la técnica utilizada para defenderse en contra de los ataques de SYN flood" (SYN Cookies).

en el shell de Linux:

vim /etc/sysctl.conf

net.ipv4.tcp_syncookies=1

Apache

Protección en Apache a través de mod_evasive. Esta implementación impedirá que un ataque DoS solicite datos al Servicio Apache de forma exhaustiva

agotando los recursos del Servicio y del servidor, a continuación el procedimiento que fue posible en el servidor:

Instalación mod_evasive:

apt-get install apache2-utils

apt-get install libapache2-mod-evasive

mkdir /var/log/mod_evasive

chown www-data:www-data /var/log/mod_evasive/

vim /etc/apache2/mods-available/mod-evasive.conf

<IfModule mod_evasive20.c>

DOSHashTableSize 3097

DOSPageCount 2

DOSSiteCount 50

DOSPageInterval 1

DOSSiteInterval 1

DOSBlockingPeriod 60

DOSEmailNotify dp@xpertsites.com

</IfModule>

Se revisa si modulo se ha cargado en Apache: *sudo a2enmod mod-evasive*

Se reinicia el servicio Apache: *service apache2 restart*

Esta implementación fue realizada según recomendaciones de compañías con una gran trayectoria de experiencia en implementar Seguridades en servidores web, como lo es Linode.com (mod_evasive on Apache). Así como foros de Seguridad Informática ampliamente reconocidos como lo es FaqForge.com, basándome en las sugerencias que han publicado en sus sitios web (Prevent DOS attacks on apache webserver).

No se considera implementar otro módulo de seguridad para Apache como modSecurity ya que éste módulo busca proteger a los sistemas a través de reglas

preestablecidas sobre vulnerabilidades previamente encontradas, concepto similar a lo que el sistema Snort maneja(Flexible Rule Engine ModSecurity). En XpertSites no se dispone de horas/hombre disponibles ni presupuesto para invertir en este tipo de sistemas de detección que requieren bastante mantenimiento de su bases de datos de reglas, por lo tanto esta fuera de nuestro alcance. En XpertSites se ha preferido siempre implementar métodos de Seguridad que no requieran mucho mantenimiento y que su efectividad esté influenciada por su simplicidad.

Respecto a pruebas en el servidor para verificar si este no es vulnerable a ataques de Denial of Service, no se las realiza dentro de esta Tesis, debido a que las pruebas de DoS en el mismo servidor de producción no es adecuado realizarlas por la congestión y daños al servicio que pueda causar a los clientes del servidor y a las redes implicadas. Es recomendable realizar estas pruebas de DoS en un segundo servidor que sea un clon del anterior, para que en ese servidor de pruebas se hagan los respectivas pruebas exhaustivas de Denial of Service.

Este tipo de procedimiento de clonación del servidor de XpertSites.com para crear uno similar para pruebas, requeriría de presupuesto extra así como también se convertiría en un proyecto aparte de investigación exclusivamente sobre este tipo de ataques DoS, lo cual no está dentro del alcance de la presente Tesis, pero que en algún momento XpertSites lo puede considerar como un proyecto de investigación a futuro, cuando crean conveniente.

P8: login de usuario Wordpress

Tenemos dos situaciones de seguridad una que el login en Wordpress por default permite que se hagan intentos indefinidos para ingresar un usuario y una

clave pudiéndose realizarse un ataque de fuerza bruta, y la otra situación es que el ingreso de usuario y clave en la página puede realizarse sin utilizar un canal de comunicación encriptado.

Se procede a implementar el modulo captcha en Wordpress para evitar un ataque de fuerza bruta. La instalación del plugin SI-Captcha se procede a realizar-la a través del panel de control del manejador de contenidos.

El plugin SI-Captcha para su funcionamiento, necesita que instalemos la librería gráfica GD para php, a continuación el procedimiento:

```
apt-get install php5-gd

sudo /etc/init.d/apache2 restart
```

En cuanto a la encriptación del canal, se ha procedido a configurar Apache para que cuando se digite la dirección web que contenga la URL “/wp-admin/” automáticamente sea redireccionada a una conección segura, para esto se ha implementado la siguiente regla en el archivo de configuración de Apache.

```
RewriteCond %{HTTPS} !=on

RewriteRule ^/?wp-admin/(.*) https://%{SERVER_NAME}/wp-admin/$1 [R,L]
```

Figura 4.4: Implementación de Control en login de Wordpress

P9: login de usuario Django

El ingreso de usuario y clave en la página de login puede realizarse sin utilizar un canal de comunicación encriptado. Para esto se procede a redireccionar automáticamente a una conexión encriptada cada vez que el cliente digite una url que contenga la dirección "/admin/", esto modificando el archivo de configuración

de Apache con las siguientes reglas.

```
RewriteCond %{HTTPS} !=on

RewriteRule ^/?admin/(.*) https://%{SERVER_NAME}/admin/$1 [R,L]
```

P10: no existe un firewall activo

Para resolver esto, levantamos las siguientes reglas de firewall del Servidor bajo ipTables de Netcraft que nos ayudarán a restringir mejor el tráfico de datos.

```
#  Pemite todo el tráfico loopback (lo0)  y descarta el tráfico hacia 127/8 que no utiliza la interfaz lo0
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 -j REJECT
#  Acepta todas las conexiones establecidas interiormente
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#  Permite todo el tráfico saliente del servidor
-A OUTPUT -j ACCEPT
#  Permite conexiones HTTP y HTTPS para el Servicio Web
-A INPUT -p tcp --dport 80 -j ACCEPT
-A INPUT -p tcp --dport 443 -j ACCEPT
# Permite conexion al panel de control VirtualMin
-A INPUT -p tcp --dport 7777 -j ACCEPT
#  Permite conecciones SSH
-A INPUT -p tcp -m state --state NEW --dport 22 -j ACCEPT
#  Permite ping
-A INPUT -p icmp -j ACCEPT
#  Registra las llamadas denegadas por iptables
-A INPUT -m limit --limit 5/min -j LOG --log-prefix "iptables denied: " --log-level 7
```

```
#  Descarta todo el resto de tráfico entrante
-A INPUT -j DROP
-A FORWARD -j DROP
COMMIT
```

Activamos las IPtables con el siguiente comando:

```
sudo iptables-restore < /etc/iptables.up.rules
```

Comprobamos que los cambios se hayan implementado:

```
sudo iptables -L
```

Configuramos que los errores captados en el log del sistema sean enviados a mi correo electrónico, como una pronta alerta.

Para esto definimos un redireccionamiento de email para el usuario root, agregando en el archivo /etc/aliases lo siguiente:

```
root: dp@xpertsites.com
```

Sub Superficie Servicio Base de Datos

P11: puerto Mysql abierto

Este es el servicio de la base de datos, y su puerto permanece abierto para ofrecer su servicio. Se lo declara como porosidad abierta ya que siempre existirá el riesgo latente de que se encuentre una nueva vulnerabilidad aunque sea en esta última versión instalada.

P12: versión desactualizada de Mysql

El Control de esta porosidad quedó implementado al instalar el software de Mysql actualizado al realizar el procedimiento en P2.

P13: Apache se conecta internamente con el servicio de base de datos, usando un login y password.

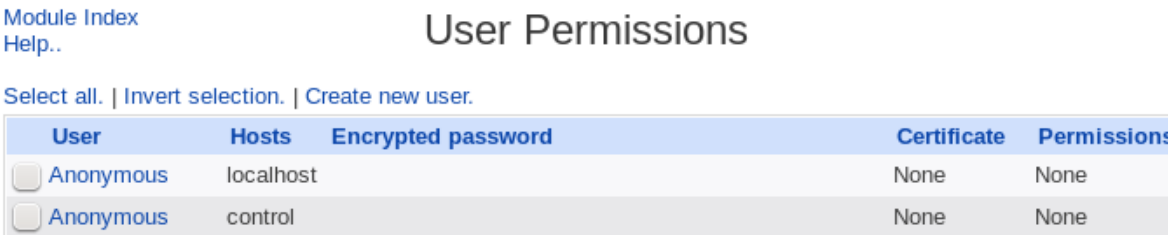
Comprobamos que el servicio Mysql esté ejecutado por el usuario “mysql” y no por el usuario “root”, ya que este tipo de vulnerabilidad permitiría a un ataque del tipo “SQL injection”correr con los privilegios de root:

```
ps aux | grep mysql
resultado en pantalla:
mysql     2175  0.0  3.0 291028 31148 ?        Ssl  Oct28   7:17 /usr/sbin/mysqld
```

Revisamos en el panel de control VirtualMin que los usuarios que han sido creados para Mysql sean los que realmente necesitamos, verificando principalmente que no existan usuarios "guest", "Anonymous" o similares.

Se encontró que existe el usuario "Anonymous" y este no requiere clave de acceso para entrar, en todo caso los permisos que tiene este usuario es de ninguno. Para evitar cualquier riesgo se procede a eliminar este usuario de Mysql desde el panel de control.

Figura 4.5: Revisión de priviliegios de usuario en Mysql



Sub Superficie Servicio SSH

P14: puerto SSH abierto.

El puerto SSH no es necesario que permanezca siempre abierto, por lo tanto solamente cuando se vaya a utilizar este servicio se lo puede activar a través del Panel de Control de VirtualMin, de esta manera esta porosidad queda controlada, aparte de los Controles implementados en P15 y P16.

P15: versión desactualizada de OpenSSH

El Control de esta porosidad quedó implementado al instalar el software OpenSSH actualizado al realizar el procedimiento en P2.

P16: usuarios usan login y password para acceder a SSH. No existe control para ataque de fuerza bruta.

Se procede a desactivar el acceso a SSH a través del ingreso de un usuario y clave, esto con la finalidad de impedir totalmente un ataque de fuerza bruta o adivinanza de claves. Podemos decir que se impide totalmente un ataque de fuerza bruta de esta manera, ya que estos ataques requieren de que el sistema les pregunte por un login y un password, al implementar el Control de acceso a través del par de llaves pública y privada, desaparece la interacción de tener que ingresar un login y un password, eliminando esta interacción desaparece este tipo de ataque. En vez de este método de ingreso, se implementa el ingreso a través de la generación de una llave pública y privada con encriptación RSA, entre el servidor y el computador que realiza el acceso. Es decir solo el computador del administrador y que tenga almacenada la llave codificada puede acceder al servidor a

través de este servicio, es decir no existe interacción de acceso a SSH para aquel que no tenga el computador del administrador en sus manos.

De esta manera generamos Seguridad sobre este asset, basándonos en el principio de separación según la OpSec, separando el login de cualquier visitante hacia el servicio SSH. El posible Vector de interacción en este caso sería: Atacante vs login SSH. Después de la implementación de Seguridad este Vector desaparecería y por lo tanto su porosidad.

Se procede a generar el par de llaves pública y privada en el computador del administrador con el siguiente comando: *ssh-keygen*

En pantalla se muestra lo siguiente:

Generating public/private rsa key pair.

Enter file in which to save the key (/home/david/.ssh/id_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/david/.ssh/id_rsa.

Your public key has been saved in /home/david/.ssh/id_rsa.pub.

Luego se procede a copiar la llave pública en el servidor:

scp ~/.ssh/id_rsa.pub root@xpertsites.com:/root/.ssh/uploaded_key.pub

ssh root@xpertsites.com "cat ~/.ssh/uploaded_key.pub >>

~/.ssh/authorized_keys"

Una vez funcionando el acceso a SSH a través de la utilización de la llave pública y privada, se procede a desactivar el acceso a SSH a través de un login y password realizando la siguiente operación:

sudo vim /etc/ssh/sshd_config

en el archivo de configuración revisamos que esté la siguiente sentencia:

PasswordAuthentication no

Con este procedimiento solo el computador del administrador puede intentar ingresar al servicio SSH.

4.7.4 Resultado de la Implementación

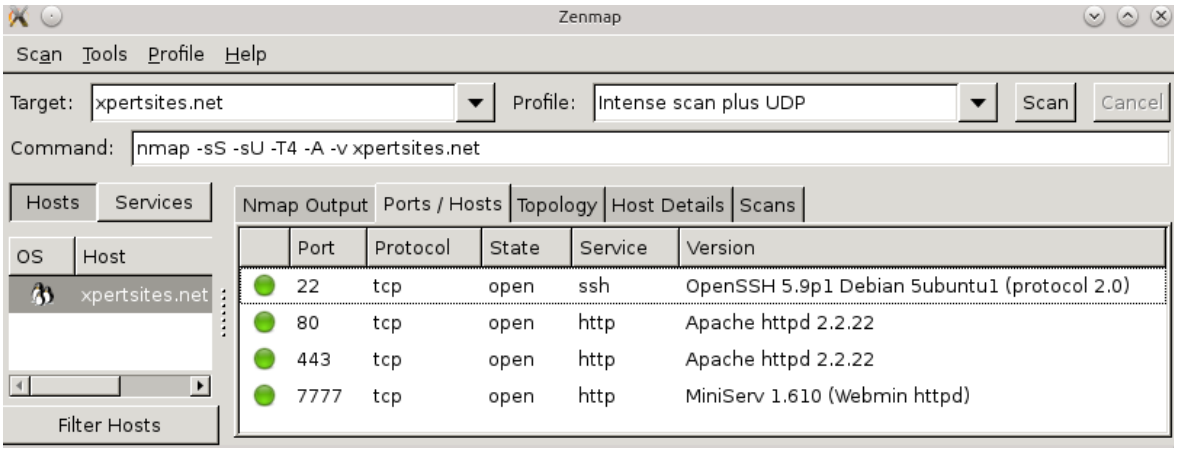
La implementación de Controles de Seguridad a las Porosidades detectadas, ha sido realizada con éxito y se ha logrado implementar lo propuesto en el Plan de Seguridad, tal como lo muestra la simulación en la sección 4.5.1. Es decir el resultado de Porosidades resultantes después de la implementación actualmente realizada es de 2 Porosidades, las cuales son el puerto abierto de Apache y el puerto abierto de Mysql. El otro puerto que queda abierto es el puerto 443 de Secure Socket Layer del mismo servicio de Apache, el cual no representa una Porosidad por sus propias características de seguridad, aparte también queda abierto el puerto del Panel de Control VirtualMin, pero con los controles implementados ya no representa una Porosidad.

Al inicio de este estudio el servidor se encontraba con 26 Porosidades no cubiertas. Esto quiere decir que la ganancia en seguridad obtenida ha sido de un 92.3% en comparación a como el servidor XpertSites.com se encontraba en un inicio.

Escaneo de Seguridad con Zenmap

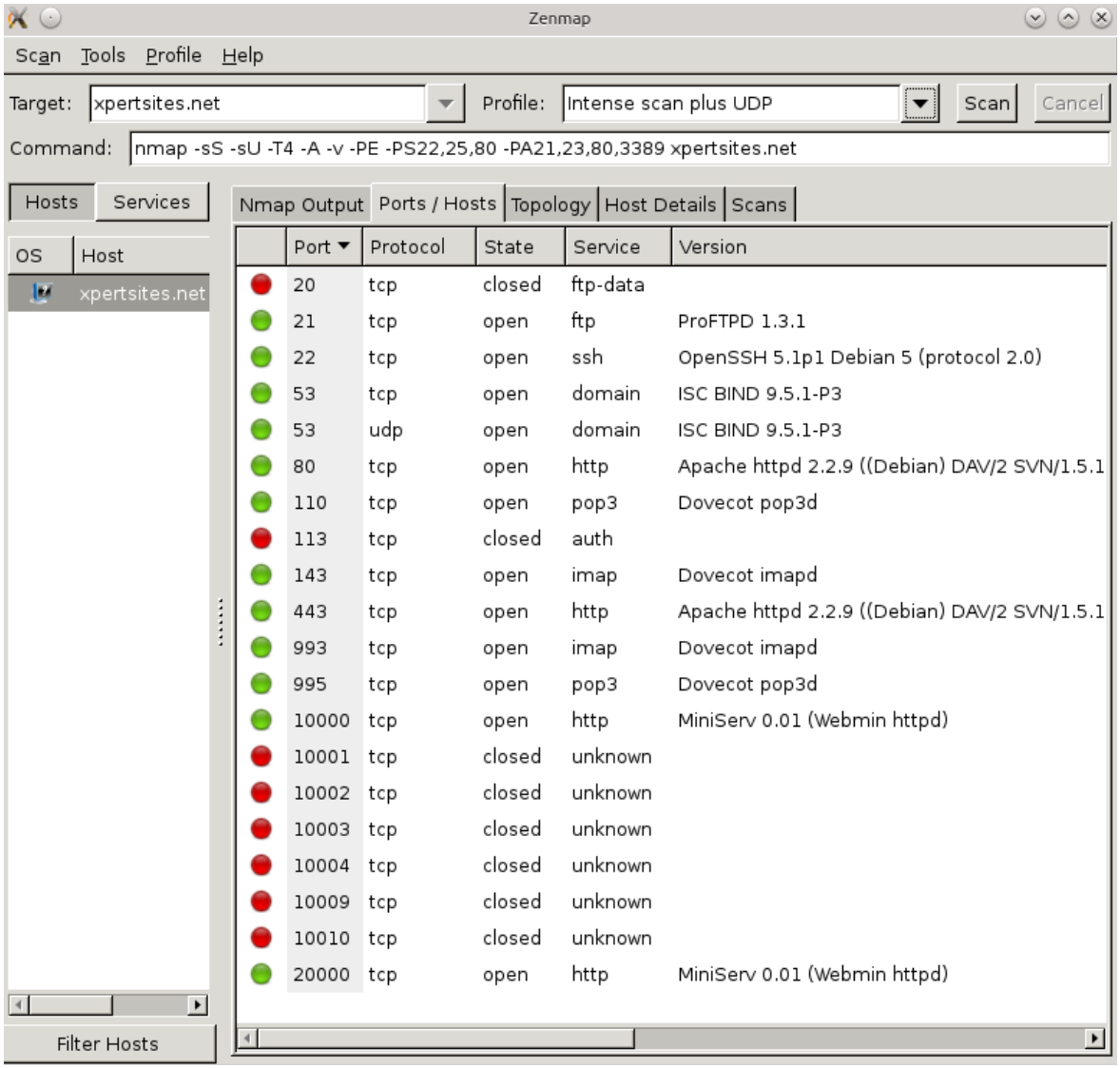
A continuación se muestran los puertos abiertos que encuentra Zenmap después de la implementación del Plan de Seguridad y que reflejan la Nueva Superficie de Ataque.

Figura 4.6: Puertos abiertos detectados por Zenmap después de la implementación del Plan de Seguridad en la nueva Superficie de Ataque.



Ahora comparemos este nuevo resultado con el escaneo que se realizó al servidor antes de implementar las seguridades del Plan de Seguridad, como lo muestra la siguiente figura 4.7

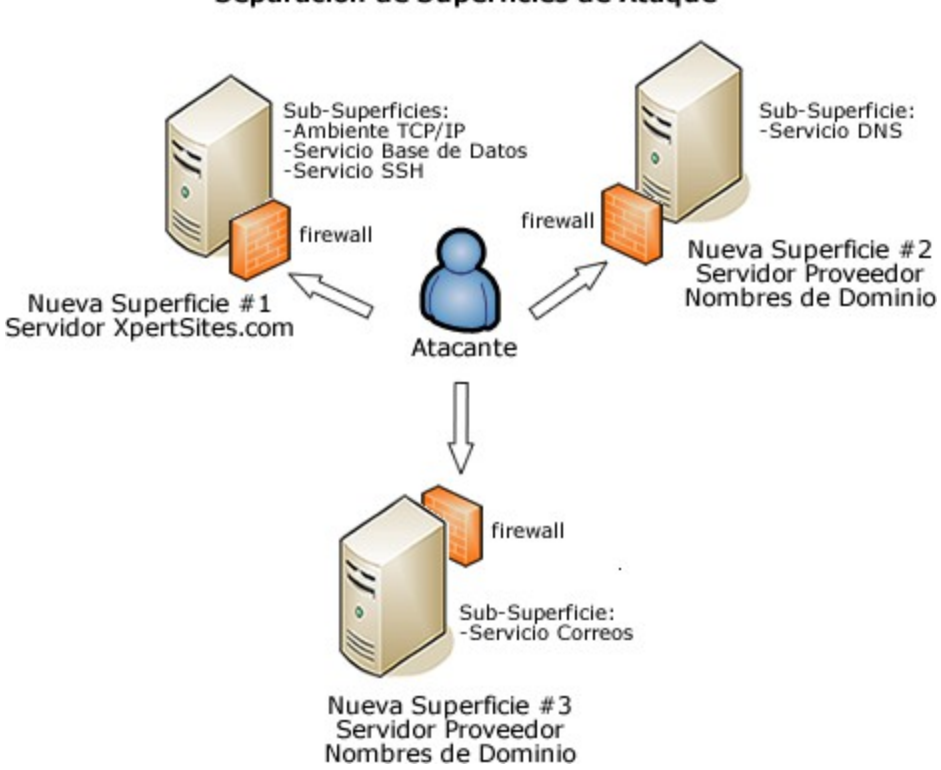
Figura 4.7: Puertos abiertos detectados por Zenmap al inicio del proyecto



A continuación se muestra una tabla comparativa en base a los escaneos de Zenmap antes y después de la implementación del Plan de Seguridad.

Tabla 4.16 - Comparación puertos abiertos en el Servidor		
Antes de Implementación		**Después de Implementación**
21 tcp	ProFTPD 1.3.1	cerrado
22 tcp	**OpenSSH 5.1p1**	**OpenSSH 5.9p1**
53 tcp	ISC Bind 9.5.1	cerrado
53 udp	ISC Bind 9.5.1	cerrado
80 tcp	**Apache httpd 2.2.9**	**Apache httpd 2.2.22**
110 tcp	Dovecot pop3d	cerrado
143 tcp	Dovecot imapd	cerrado
443 tcp	**Apache httpd 2.2.9**	**Apache httpd 2.2.22 SSL**
993 tcp	Dovecot imapd	cerrado
995 tcp	Dovecot pop3d	cerrado
10000 tcp	MiniServ Webmin	cerrado
20000 tcp	MiniServ Webmin	cerrado
7777 tcp	**cerrado**	**MiniServ Webmin 1.6**

Figura 4.8 Superficies de Ataque después de la Implementación del Plan de Seguridad



Observando los resultados de la tabla anterior donde se muestran los puertos abiertos que quedan, así como el gráfico anterior en donde se muestra como ha quedado reducida la Superficie de Ataque del servidor XpertSites.com protegida ahora por un firewall, mas la verificación de que se han implementado los distintos Controles de Seguridad a las Porosidades encontradas, se ha logrado comprobar que en efecto la seguridad en el servidor XpertSites.com ha sido mejorada en gran magnitud, disminuyendo su Vulnerabilidad en Internet.

CAPÍTULO 5 : Conclusiones y Recomendaciones

5.1 Conclusiones

El desarrollo de la actual Tesis, con su implementación del Plan de Seguridad en el Servidor XpertSites.com, no solo nos ha dejado con un servidor web más seguro funcionando en Internet como hemos podido ver durante el proceso de reducción de la Superficie de Ataque, la reducción de sus Porosidades y el Control a las mismas, sino que también nos ha permitido ver la Seguridad desde ángulos antes no claramente visibles.

Al inicio del presente proyecto se propuso la idea de analizar los ataques mas frecuentes que se venían dando hacia el servidor, con la idea de defendernos de éstos ataques a través de la posibilidad de llegar a implementar un Sistema de Respuesta a Intrusiones, así como también con la implementación de correctivos de seguridad.

En el proceso de esta investigación, al analizar los ataques mas frecuentes que registró el Sistema de Detección de Intrusos Snort en el servidor, se logró determinar que ofrecer protección al servidor basándonos en éstos ataques frecuentes no era el norte a seguir. Esto debido a que de los 20 ataques mas frecuentes analizados, el 65% es decir la mayoría resultaron ser falsos positivos y éste número podía seguir en aumento si continuábamos analizando una muestra mas amplia de ataques frecuentes, como se explicó en el respectivo capítulo. Por el otro lado el 35% restante representa los ataques que apuntan a elementos que el Servidor si tiene en su configuración, pero que realmente no son prueba de que sean efectivos contra estos elementos. Simplemente indica que estos elementos

existen en el servidor, mas no que éstos realmente puedan ser vulnerados. Su determinación de vulnerabilidad efectiva, se muestra incierta, ya que son comandos que han utilizado los atacantes en los paquetes de datos, pero no necesariamente estos comandos han probado tener algún sentido para el servidor.

Fue así que llegar a implementar un Sistema de Respuesta a Intrusiones al Sistema de Detección de Intrusiones Snort donde la mayoría de intentos de intrusión resultaba ser falsos positivos, dejó de tener sentido para cubrir la seguridad del servidor. Así es como se continúo con la búsqueda de implementar Seguridad al servidor a través de una metodología ya reconocida en el mundo de la seguridad en Internet.

De esta manera se procedió a evaluar la seguridad del servidor a través de la metodología OSSTMM3, la cual nos permitió analizar las Operaciones que existen dentro de la Superficie de Ataque que es el servidor XpertSites.com, para identificar cuales son innecesarias y determinar que se puede hacer con ellas, así mismo identificar que Operaciones se pueden separar del servidor y determinar las Porosidades de las Operaciones que se llevan a cabo en el mismo, para a su vez llegar a implementar Controles que cubran de alguna manera éstas Porosidades. En el servidor éstas Operaciones estaban representadas por las interacciones que realizaban los distintos servicios del servidor, como los servicios TCP/IP, Apache Web, MySQL, SSH, FTP y Correo. A esta forma de analizar la seguridad es lo que la OSSTMM3 llama Seguridad Operacional (OpSec).

Así es como con la metodología OSSTMM3 se logró reducir la Superficie de Ataque del servidor XpertSites.com y por ende su Porosidad. En un inicio se encontró en el servidor 26 Porosidades, donde mas luego se procedió a reducir la

Superficie de Ataque XpertSites.com, creando la Nueva Superficie de Ataque #1 logrando de esta manera quedarnos con 16 Porosidades es decir ya obteniendo una ganancia en Seguridad del 38.46%, sin haber todavía implementado Controles de Seguridad a éstas Porosidades. Finalmente luego de implementar los Controles a estas Porosidades resultantes nos quedamos con tan solo 2 Porosidades que a pesar de que igualmente se les implementó Controles de Seguridad, se decidió declararles como persistentes ya que su acceso está disponible de forma directa a los visitantes en Internet, esto es el servicio Apache y MySql. Esto fue el resultado de la implementación del Plan de Seguridad, obteniendo así finalmente una ganancia en seguridad del 92.3% comparado a como se encontraba en un inicio el servidor.

Para la implementación de Controles a las Porosidades, se investigó en Internet los Controles más comunmente probados que se utilizan para cubrir a este tipo de Porosidades y que a su vez se aplicaban a la realidad del servidor XpertSites.com ya que correspondían a las Porosidades resultantes encontradas.

En el Plan de Seguridad, se dió bastante importancia a la actualización del software de todo el servidor, ya que un software desactualizado significa que éste ha tenido mucho más tiempo para ser descubierto sus vulnerabilidades y éstas ya podrían ser utilizadas para un ataque, la última versión del software tiende a ofrecer correcciones a las posibles vulnerabilidades encontradas anteriormente. La actualización del software se llevó a cabo al contratar un nuevo servidor con todo su software actualizado, desde donde se creó la nueva Superficie de Ataque reducida, evitando así también tener que intervenir en el servidor XpertSites.com que se encontraba en estado de producción ofreciendo su servicio. Una vez confi-

gurado el nuevo servidor, es decir la nueva y reducida Superficie de Ataque, se migró la información del anterior servidor al nuevo.

Por otro lado, de 12 puertos abiertos que se encontraban inicialmente en el servidor, después de implementar el Plan de Seguridad, nos quedamos con tan solo 4 puertos abiertos, de los cuales se decidió que solo 2 puertos representan ser realmente una posible Porosidad a futuro. Estos dos puertos representan al Servicio Web Apache y al Servicio de Base de Datos MySQL, que se comprende que los visitantes en Internet tienen acceso directo a estos servicios ya que son los que publican el contenido y que siempre estarán en la mira de los atacantes para seguir encontrando vulnerabilidades aún en sus mas nuevas versiones corregidas del software y a pesar de los Controles de Seguridad que se hayan implementado. Esto no sucede con el puerto abierto de VirtualMin, porque para su acceso se necesita ingresar un login y password con un número limitado de intentos, y para el caso del servicio SSH sólo el computador del administrador del sevidor puede accederlo ya que sólo ahí se encuentra almacenada la llave privada de acceso a este servicio.

5.2 Recomendaciones

Recordemos que después de implementar el Plan de Seguridad, quedaron 2 Porosidades latentes, que corresponden al Servicio Apache y al Servicio de la base de datos MySQL. Aparte hay que considerar que en el futuro se puedan encontrar nuevas Porosidades en el servidor debido a nuevas configuraciones que se hagan en el mismo, inclusive Porosidades que se encontraban antes pero que no fueron detectadas.

Para todo recomiendo trazar una estrategia de monitoreo de estos servicios, estudiar el comportamiento de los atacantes a través de herramientas y así mismo trazar una estrategia de contingencia en el caso de que un atacante logre vulnerar los mismos.

En el análisis de ataques frecuentes al servidor, varios de estos ataques tenían que ver con comandos enviados a través de una dirección URL del Servicio Apache, comandos que buscan vulnerar alguna de las aplicaciones web que puedan estar instaladas en el servidor como PhpMyAdmin tal como lo intentaba el ataque "WEB-PHP Setup.php access", o incluso a la misma aplicación Apache como lo muestran los ataques "OVERSIZE REQUEST-URI DIRECTORY" y “WEB-MISC http directory traversal". En los 20 ataques frecuentes no se encontró actividad de ataques frecuentes sobre la base de datos MySQL, pero de todas maneras es un Blanco Objetivo muy común en Internet y por lo tanto también requiere un monitoreo al igual que el servicio Apache.

A continuación recomiendo las siguientes estrategias.

Estrategias de Monitoreo:

- Configurar en el servidor un sistema de alertas, el cual le notifica a su correo electrónico o al servicio SMS de su celular, cuando uno de los Servicios del servidor ha sido dado de baja. Para el caso del presente servidor, se puede configurar en el panel de control VirtualMin, bajo la sección "System and Server Status" este tipo de alertas.
- Aparte se puede dejar activo el Sistema de Detección de Intrusiones Snort,

configurado solo con reglas de alarmas que tengan que ver con la versión del Servicio Apache y Mysql utilizado. Esto puede requerir bastante investigación y tiempo dedicado para llegar a escoger y pulir bien las alarmas y así no tener una infinidad de ellas que se tapan entre sí. Esta implementación de Snort, personalmente le encuentro que su valor está mas en la oportunidad que nos da para aprender como los atacantes realizan sus procedimientos, aún así estos ataques no representen ningún riesgo al servidor.

- Personalmente no recomiendo usar Snort como un sistema de Respuesta a Intrusiones, principalmente recomiendo que su uso sea con fines de aprendizaje sobre los ataques, es decir que su uso sea como un Sistema de Detección de Intrusiones, esto al menos hasta que no haya pasado un buen tiempo en donde se haya logrado el nivel necesario en donde se tengan ya pulidas las reglas de alertas más adecuadas, porque lo que menos se quiere es llegar a tener un servidor que responda o bloquee ya sea paquetes o direcciones ip que son inofensivas.
- Mantener las versiones de software en el servidor lo mas actualizadas posibles.
- Igualmente se recomienda seguir investigando sobre nuevas formas de implementar Controles a las Porosidades que han quedado y a las que se sigan encontrando en un futuro.

Estrategias de Contingencia:

- Configurar en el servidor un sistema de respaldo de los archivos del servidor así como su base de datos, de forma automatizada, donde se deposite

automáticamente los respaldos en un servidor de respaldos. Esto se puede llegar a implementar configurando las opciones de VirtualMin “Filesystem Backup”, “Backup Configuration File", “Backup Databases” o usar un sistema integrado de respaldos como el “Bacula Backup System" que también puede ser configurado a través de VirtualMin.

- Con un sistema adecuado de respaldos, se puede lograr reestablecer los datos dañados o perdidos del servidor no solo producto de un ataque, sino que también producto de algún fallo en el disco de almacenamiento.

Aparte de todas las anteriores recomendaciones, se recomienda tener un clon del servidor, que vendría a ser un servidor de pruebas, donde se puedan realizar todas las pruebas de vulnerabilidad necesarias, así como modificaciones en el servidor, sin correr el riesgo de perjudicar al servidor de producción, teniendo así mas libertad en ese aspecto.

GLOSARIO

Dirección IP: dirección del Protocolo de internet. Cada equipo que se conecta a internet tiene una dirección ip que lo identifica, su número está relacionado a la red a la cual se conecta, esta dirección puede ser dinámica o estática.

DoS: Denial of Service attack, ataque de Denegación de Servicio.

Gateway: es un nodo en la red que sirve como entrada a otra red. En una conexión de Internet en casa, el gateway sería el servidor del proveedor de Internet que conecta el usuario al Internet.

ICMP: Internet Control Message

IDS: Intrusion Detection System. Sistema de Detección de Intrusiones (SDI)

Interface loopback: (l0) es un dispositivo de red virtual creado a través de software. Todo el tráfico enviado a esta interfaz se redirecciona a los servicios locales de una máquina.

IRS: Intrusion Response System. Sistema de Respuesta a Intrusiones.

Iptables: es un poderoso firewall compilado dentro del kernel, y es parte de el proyecto netfilter. iptables es usado para IPv4 y ip6tables es usado para IPv6.

Número de Saltos: (hops) representa el número de dispositivos ruteadores que un paquete de datos necesita pasar para llegar a su destino.

Shim: pequeño conjunto de instrucciones que corren en el mismo espacio de proceso de la aplicación a la cual se unen.

Sniffer: término en inglés que se refiere a la acción de únicamente espiar los paquetes de red, sin tomar ninguna acción con ellos.

TCP: Transmission Control Protocol

REFERENCIAS BIBLIOGRÁFICAS

"About". *HackerHighScool.org.* ISECOM, s.f. Web. 28 dic. 2012. <http://www.hackerhighschool.org/about.html>

"About US". *Isecom.org.* ISECOM, s.f. Web. 28 dic. 2012. <http://www.isecom.org/about-us.html>

"Atlanta Natural Disasters". *USA.COM*. World Media Group, s.f. Web. 28 sep. 2012. <http://www.usa.com/atlanta-ga-natural-disasters-extremes.htm>

"Bacula, the Open Source, Network Backup Tool for Linux, Unix, Mac and Windows". *Bacula.org. Sp, s.f.* Web. 2 ene. 2013. <http://www.bacula.org/en/>

"Bind". *Wikipedia Enciclopedia*. WikiMedia Foundation, s.f. Web. 15 dic. 2012. <http://es.wikipedia.org/wiki/Bind>

"Case-sensitivity in 2570.6 (WEB-MISC Invalid HTTP Version String)". *Marc.info.* KoreLogic Security, 13 nov. 2004. Web. 14. ago. 2012.<http://marc.info/?l=snort-sigs&m=110030510624482>

Ciufo, Chris. "mod_evasive on Apache". *library.linode.com*. Linode, 14 nov. 2011. Web. 12 oct. 2012. <http://library.linode.com/web-servers/apache/mod-evasive>

"Dovecot". *Wikipedia Enciclopedia*. WikiMedia Foundation, s.f. Web. 15 dic. 2012. <http://es.wikipedia.org/wiki/Dovecot>

Esler, Joel. *Snort IDS and IPS Toolkit*. Burlington: Syngress Publishing, 2007. Impreso.

"Flexible Rule Engine ModSecurity". *TrustWave.com*. Trustwave Holdings, s.f. Web. 20 oct. 2012. <https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual#wiki-Current_as_of_v2513_v26_and_v27>

Herzog, Pete. *The Open Source Security Testing Methodology Manual OSSTMM 3*. ISECOM, 2010. Web. 2 de sep. 2012. <http://www.isecom.org/research/osstmm.html>

“Mysql". *Wikipedia Enciclopedia*. WikiMedia Foundation, s.f. Web. 15 dic. 2012. <http://es.wikipedia.org/wiki/Mysql>

"Nmap". *Wikipedia Enciclopedia*. WikiMedia Foundation, s.f. Web. 15 dic. 2012. <http://es.wikipedia.org/wiki/Nmap>

"OpenSSH". *Wikipedia Enciclopedia*. WikiMedia Foundation,s.f. Web. 15 dic. 2012. <http://es.wikipedia.org/wiki/OpenSSH>

"Over 21,000 Snort Rules". *ActiveWorx Blog*. ActiveWorx, 5 dic.2008. Web. 10 ago. 2012. <http://activeworx.blogspot.com/2008/12/over-21000-snort-rules.html>

"Postfix". *Wikipedia Enciclopedia*. WikiMedia Foundation, s.f. Web. 15 dic. 2012. <http://es.wikipedia.org/wiki/Postfix>

"Prevent DOS attacks on apache webserver for DEBIAN linux with mod_evasive". *FaqForge.com. Projektfarm ,* 7 mar. 2011. Web. 10 oct. 2012. <http://www.faqforge.com/linux/prevent-dos-attacks-on-apache-webserver-for-debian-linux-with-mod_evasive/>

"ProFTPd". *Wikipedia Enciclopedia*. WikiMedia Foundation, s.f. Web. 15 dic. 2012. Diciembre 2012. <http://es.wikipedia.org/wiki/ProFTPd>

Rash, Michael y Angela Orebaugh. *Intrusion Prevention and Active Response*. Rockland: Syngress Publishing, 2005. Impreso.

Scarfone, Karen y Peter Mell. *Guide to Intrusion Detection and Prevention Systems (IDPS), NIST SP800-94*. Gaithersburg: National Institute of Standards and Technology, 2007. Impreso.

"Servidor HTTP Apache". *Wikipedia Enciclopedia*. WikiMedia Foundation, s.f. Web. 15 dic. 2012. <http://es.wikipedia.org/wiki/Servidor_HTTP_Apache>

Somerville, Liam. *SnortId.com.* Cook Computing, s.f. Web. 14. ago. 2012. <http://www.snortid.com/>

"SYN Cookies". *Wikipedia Enciclopedia*. WikiMedia Foundation, sf. Web. 4 nov. 2012 <http://en.wikipedia.org/wiki/SYN_cookies>

"Training". *Isecom.org.* ISECOM, s.f. Web. 28 dic. 2012. <http://www.isecom.org/training.html>

"Vulnerability Summary for CVE-2004-0646". *NIST.gov*. NIST, 14 ago. 2008. Web. ago. 2012. <http://web.nvd.nist.gov/view/vuln/detail?vulnId=CAN-2004-0646>

"Web Search Interest: ubuntu, debian". *Google Trends.* Google Inc, s.f. Web. 5 oct. 2012. <http://www.google.com/trends/explore#q=ubuntu%2C%20debian&cmpt=q>