ESCUELA POLITÉCNICA DEL EJÉRCITO

EXTENSIÓN LATACUNGA

CARRERA DE

INGENIERÍA EN SISTEMAS E INFORMÁTICA

"IMPLEMENTACIÓN DE UN SISTEMA DE PROTECCIÓN Y SEGURIDAD DE SERVIDORES WEB Y DE CORREO ELECTRÓNICO BASADO EN EL SISTEMA OPERATIVO LINUX, APLICADO AL SITIO WEB DE LA ESCUELA POLITÉCNICA DEL EJÉRCITO - SEDE LATACUNGA"

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE

INGENIERO EN SISTEMAS E INFORMÁTICA

MAYRA ELIZABETH GALLO CÁRDENAS

ZORAYA LIVEYA VILLACÍS ZUMBA

Latacunga, enero 2012

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por las señoritas Mayra Elizabeth Gallo Cárdenas y Zoraya Liveya Villacís Zumba, bajo nuestra supervisión.

_____

Ing. Fabián Montaluisa

DIRECTOR DE PROYECTO

_____

Ing. Raúl Cajas

CODIRECTOR DE PROYECTO

_____

Ing. Santiago Jácome

DIRECTOR DE CARRERA

_____

Dr. Rodrigo Vaca

SECRETARIO ACADÉMICO

ESCUELA POLITÉCNICA DEL EJÉRCITO

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICADO

ING. FABIÁN MONTALUISA (DIRECTOR)

ING. RAÚL CAJAS (CODIRECTOR)

CERTIFICAN:

Que el trabajo titulado "IMPLEMENTACIÓN DE UN SISTEMA DE PROTECCIÓN Y SEGURIDAD DE SERVIDORES WEB Y DE CORREO ELECTRÓNICO BASADO EN EL SISTEMA OPERATIVO LINUX, APLICADO AL SITIO WEB DE LA ESCUELA POLITÉCNICA DEL EJÉRCITO – SEDE LATACUNGA" realizado por las señoritas: MAYRA ELIZABETH GALLO CÁRDENAS y ZORAYA LIVEYA VILLACÍS ZUMBA ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Escuela Politécnica del Ejército.

Debido a que constituye un trabajo de excelente contenido científico que coadyuvará a la aplicación de conocimientos y al desarrollo profesional, **SI** recomiendan su publicación.

Latacunga, enero del 2012.

---	---
Ing. Fabián Montaluisa	Ing. Raúl Cajas
DIRECTOR	**CODIRECTOR**

ESCUELA POLITÉCNICA DEL EJÉRCITO

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

DECLARACIÓN DE RESPONSABILIDAD

Nosotras, MAYRA ELIZABETH GALLO CÁRDENAS, y;

ZORAYA LIVEYA VILLACÍS ZUMBA

DECLARAMOS QUE:

El proyecto de grado denominado "IMPLEMENTACIÓN DE UN SISTEMA DE PROTECCIÓN Y SEGURIDAD DE SERVIDORES WEB Y DE CORREO ELECTRÓNICO BASADO EN EL SISTEMA OPERATIVO LINUX, APLICADO AL SITIO WEB DE LA ESCUELA POLITÉCNICA DEL EJÉRCITO – SEDE LATACUNGA" ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, enero del 2012.

_____	_____
Elizabeth Gallo C.	Zoraya Villacís Z.
C.I. 050263447-0	C.I. 150080799-3

ESCUELA POLITÉCNICA DEL EJÉRCITO

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Nosotras, MAYRA ELIZABETH GALLO CÁRDENAS, y;

ZORAYA LIVEYA VILLACÍS ZUMBA

Autorizamos a la Escuela Politécnica del Ejército la publicación, en la biblioteca virtual de la Institución del trabajo "IMPLEMENTACIÓN DE UN SISTEMA DE PROTECCIÓN Y SEGURIDAD DE SERVIDORES WEB Y DE CORREO ELECTRÓNICO BASADO EN EL SISTEMA OPERATIVO LINUX, APLICADO AL SITIO WEB DE LA ESCUELA POLITÉCNICA DEL EJÉRCITO – SEDE LATACUNGA" cuyo contenido, ideas y criterios son de NUESTRA exclusiva responsabilidad y autoría.

Latacunga, enero del 2012.

_____	_____
Elizabeth Gallo C.	Zoraya Villacís Z.
C.I. 050263447-0	C.I. 150080799-3

DEDICATORIA

El presente proyecto está dedicado con todo mi amor…

A Dios, quien es el dueño de mi vida, por fortalecerme en aquellos momentos de desaliento y debilidad y por engrandecer mi espíritu con cada obstáculo presentado.

A mis queridos padres Jorge y Piedad, por confiar en mí y porque con su ejemplo de superación y entrega fueron parte fundamental para que hoy pueda ver alcanzada mi meta.

A mi adorada Mami Laurita, por ser mi fortaleza y mi maestra y porque sé que desde el cielo siempre cuidas de mi.

A mi razón de ser… mi hija Davita, por ser mi motor, mi ejemplo, la fuerza que me motiva para vencer todos los obstáculos e impulsarme para salir adelante victoriosa. Este logro es para ti y por ti mi amor!!

A mi esposo Patricio, que en los momentos difíciles me demostró comprensión.

A mi hermano Cristian, por ser mi compañero inseparable de aquellos momentos importantes en mi vida.

A toda mi familia, que me apoya siempre, por haber fomentado en mi el anhelo de triunfo y porque el orgullo que sienten por mi fue lo que me impulsó a llegar hasta el final.

Elizabeth Gallo

AGRADECIMIENTO

Primero y sobre todo a Dios, por iluminar mis pasos, por darme la sabiduría para culminar esta etapa tan importante de mi vida y por acompañarme todos los días de mi existencia.

A mis padres, por guiar mis pasos desde pequeña, por todo su amor, su dedicación, su inagotable apoyo y su esfuerzo diario para brindarme la mejor formación y educación.

A mi pequeña hija Davianny, por tu ternura y tu comprensión porque a tu corta edad supiste entenderme con amor cuando debía ausentarme por mis estudios y para la realización de este proyecto.

A mi esposo Patricio, gracias por tu infinita paciencia y todo tu apoyo brindado en los buenos y malos momentos.

A Cristian, por ser mi guardián y amigo, eres el mejor hermano.

A la ESPE por abrirme sus puertas y acogerme como si fuese mi segundo hogar. A todos mis amigos, compañeros y maestros, porque formaron parte de esta aventura y siempre se quedarán en mis recuerdos.

Al Ing. Fabián Montaluisa y al Ing. Raúl Cajas, Director y Codirector de Tesis respectivamente, gracias por su amistad, por su aporte académico y colaboración constante para ayudarnos a sacar adelante este proyecto.

Gracias a todos por compartir mi vida y mis logros!!

Elizabeth Gallo

DEDICATORIA

El presente proyecto de tesis se lo dedico a Dios, quien me dio la fe, la fortaleza necesaria para siempre salir adelante pese a las dificultades, por colocarme en el mejor camino, iluminando cada paso de mi vida, por darme salud y esperanza para terminar este trabajo además por llenar de bendiciones y alegrías dentro de nuestra familia, porque gracias a ti, mis padres con esfuerzo y añoranza me brindaron el apoyo, la confianza y amor para estudiar aunque desde muy lejos ellos, con mucha tristeza me apoyaron durante toda mi carrera, pues aquí está el fruto de sus esfuerzos.

A mi padre por haberse esforzado y buscar la manera de ofrecerme mejor calidad de estudio, trabajando día y noche sin una hora de descanso.

A mi madre por brindarme valores, respeto, el enseñarme a ser humilde ante todo, y brindarme mucho amor a la distancia.

A mis dos Hermanos Mackenna y Stalin porque estuvieron día y noche conmigo compartiendo malos y buenos momentos siempre juntos.

A mi novio Cristian por brindarme el apoyo incondicional, porque continuamente hubo una palabra cuando más lo necesite, siempre has estado a mi lado en lo bueno y lo adverso, llenándome de consejos y empujándome para ser mejor, durante todo este tiempo.

Padres hoy, no solo estoy cumpliendo mi sueño sino el de ustedes el de ser una profesional, ustedes quienes lucharon incansablemente sin importar las dificultades de la vida, lucharon por hacerme una mujer de bien, una mujer preparada, una mujer profesional; HE AQUÍ SU SUEÑO Y EL MIO HECHO REALIDAD.

Zoraya Liveya Villacís Zumba

AGRADECIMIENTO

Agradezco a todos por el tiempo que nos dedicaron.

A la ESPE- Latacunga que fue mi segundo hogar, donde allí me encontré con maestros que me enseñaron sus amplios conocimientos para desarrollarme en el campo laboral y profesional y han cumplido con sus metas y sus objetivos.

Zoraya Liveya Villacís Zumba

ÍNDICE DE CONTENIDO

INDICE DE ILUSTRACIONES

RESUMEN

El creciente número de vulnerabilidades encontradas en las aplicaciones de software las convierte en una potencial víctima. El sitio web de la ESPEL no es la excepción y con mayor razón por la importancia de la información que allí se maneja. Este problema nos ha motivado a diseñar y desarrollar esta aplicación de seguridad denominado Honeynet.

En el Primer Capítulo se describe un rápido análisis del problema de seguridad de la información, las amenazas así como las técnicas y estrategias para fortalecerla. El Segundo Capítulo es un referente sobre el uso de los Honeypots como medida de seguridad, mediante la implementación de una Honeynet, la misma que es un recurso de red destinado a ser atacado, de modo que proporcione información sobre las técnicas utilizadas por los intrusos. En el Tercer Capítulo se efectúa la configuración del Honeypot mediante la utilización de la Tecnología UML (User Mode Linux), así como la configuración de la red para UML. En el Cuarto Capítulo se realizan las respectivas configuraciones a los servidores Web y de Correo electrónico. En el Quinto Capítulo se lleva a cabo la simulación de ataques al honeypot y un análisis de los resultados obtenidos producto de la intrusión. Finalmente en el Capítulo 6 se detallan las conclusiones y recomendaciones en base a los conocimientos adquiridos durante el desarrollo de este proyecto.

ABSTRACT

The increasing number of vulnerabilities in most Software Applications makes it a potential victim. ESPEL's website is not an exception, and it is reasonable, because important information is managed there. This problem has motivated us to design and develop this security application, Honeynet Software.

The first chapter describes a quick analysis to the problem of information security, threats, techniques and strategies to strengthen it. Second chapter is a reference to the Honeypots uses as a security way, by Honeynet implementation, which is a network resource exposed to hackers attacks, so it will provide information about techniques used by intruders. On third chapter there are topics about Honeypot configuration using UML (User Mode Linux) Technology and network configuration for UML. On chapter fourth there is the respective Web servers and e-mail configurations. On fifth chapter are performed honeypot attacks, simulations and results analysis. Finally in chapter sixth, conclusions and recommendations are detailed according to knowledge increased during this project.

1 ANALISIS DEL PROBLEMA DE SEGURIDAD DE LA INFORMACIÓN

1.1 SEGURIDAD INFORMÁTICA

1.1.1 INTRODUCCIÓN

Desde el surgimiento de la raza humana en el planeta, la información estuvo presente bajo diversas formas y técnicas. El hombre buscaba representar sus hábitos, costumbres e intenciones en diversos medios que pudiesen ser utilizados por él y por otras personas, además de la posibilidad de ser llevados de un lugar a otro. La información valiosa era registrada en objetos preciosos y sofisticados, pinturas magníficas, entre otros, que se almacenaban con mucho cuidado en locales de difícil acceso, a cuya forma y contenido sólo tenían acceso quienes estuviesen autorizados o listos para interpretarla.

En la actualidad la información es el **objeto de mayor valor para las empresas**. El progreso de la informática y de las redes de comunicación nos presenta un nuevo escenario, donde **los objetos del mundo real están representados por bits y bytes**, que ocupan lugar en otra dimensión y poseen formas diferentes de las originales, no dejando de tener el mismo valor que sus objetos reales, y, en muchos casos, llegando a tener un valor superior.

Por esto y otros motivos, **la seguridad de la información es un asunto tan importante para todos, pues afecta directamente a los negocios de una empresa o de un individuo**. (Piramide Digital)

“La seguridad informática es un tema que mucha gente no le da la importancia que realmente tiene; muchas veces por el hecho de considerar que es inútil o que jamás la utilizará. Pero en el mundo moderno, cada día

más y más personas mal intencionadas pretenden tener acceso a los datos de nuestros ordenadores."

El acceso no autorizado a una red informática o a los equipos que en ella se encuentran puede ocasionar en la gran mayoría de los casos graves problemas. Una de las posibles consecuencias de una intrusión es la **pérdida de datos**. Es un hecho frecuente y ocasiona muchos trastornos, sobre todo si no estamos al día de las copias de seguridad. Y aunque estemos al día, no siempre es posible recuperar la totalidad de los datos.

Otro de los problemas más dañinos es el **robo de información** sensible y confidencial. La divulgación de la información que posee una empresa sobre sus clientes puede acarrear demandas millonarias contra esta, o un ejemplo más cercano es el de las contraseñas de las cuentas de correo por las que intercambiamos información con otros. (Wikibooks, 2009)

1.1.2 DEFINICIÓN

La seguridad informática consiste en asegurar que los recursos del sistema de información (material informático o programas) de una organización sean utilizados de la manera que se decidió y que el acceso a la información allí contenida, así como su modificación, sólo sea posible a las personas que se encuentren acreditadas y dentro de los límites de su autorización.

Podemos entender como seguridad un estado de cualquier tipo de información (informática o no) o la que nos indica que ese sistema está libre de peligro, daño o riesgo. Se entiende como peligro o daño todo aquello que pueda afectar su funcionamiento directo o los resultados que se obtienen del mismo. Para la mayoría de los expertos el concepto de seguridad en la informática es utópico porque no existe un sistema 100% seguro.

En estos momentos la seguridad informática es un tema de dominio obligado por cualquier usuario de Internet, para no permitir que su información sea comprometida. (Wikipedia, 2010)

"La Seguridad Informática es la disciplina que se ocupa de diseñar las normas, procedimientos, métodos y técnicas, orientados a proveer condiciones seguras y confiables, para el procesamiento de datos en sistemas informáticos.

La decisión de aplicarlos es responsabilidad de cada usuario.

Las consecuencias de no hacerlo… también".

Debido a que el uso de Internet se encuentra en aumento, cada vez más compañías permiten a sus socios y proveedores acceder a sus sistemas de información. Por lo tanto, es fundamental saber qué recursos de la compañía necesitan protección para así controlar el acceso al sistema y los derechos de los usuarios del sistema de información. Los mismos procedimientos se aplican cuando se permite el acceso a la compañía a través de Internet.

Además, debido a la tendencia creciente hacia un estilo de vida nómada de hoy en día, el cual permite a los empleados conectarse a los sistemas de información casi desde cualquier lugar, se pide a los empleados que lleven consigo parte del sistema de información fuera de la infraestructura segura de la compañía. (Kioskea, 2008)

1.1.3 OBJETIVOS DE LA SEGURIDAD INFORMÁTICA

Generalmente, los sistemas de información incluyen todos los datos de una compañía y también en el material y los recursos de software que permiten a una compañía almacenar y hacer circular estos datos. Los sistemas de información son fundamentales para las compañías y deben ser protegidos.

Generalmente, la seguridad informática consiste en garantizar que el material y los recursos de software de una organización se usen únicamente para los propósitos para los que fueron creados y dentro del marco previsto.

La seguridad informática se resume, por lo general, en tres objetivos principales:

Integridad: Es necesario asegurar que los datos no sufran cambios no autorizados, la pérdida de integridad puede acabar en fraudes, decisiones erróneas o como paso a otros ataques. El sistema contiene información que debe ser protegida de modificaciones imprevistas, no autorizas o accidentales, como información de censo o sistemas de transacciones financieras.

“Basándose en este principio, las herramientas de seguridad informática deben asegurar que los procesos de actualización estén sincronizados y no se dupliquen, de forma que todos los elementos del sistema manipulen adecuadamente los mismos datos.

Este principio es particularmente importante en sistemas descentralizados, es decir, aquellos en los que diferentes usuarios, computadores y procesos comparten la misma información”.

Disponibilidad: Se refiere a la continuidad operativa de la entidad, la pérdida de disponibilidad puede implicar, la pérdida de productividad o de credibilidad de la entidad. El sistema contiene información o proporciona servicios que deben estar disponibles a tiempo para satisfacer requisitos o evitar pérdidas importantes, como sistemas esenciales de seguridad y protección de la vida.

“Basándose en este principio, las herramientas de Seguridad Informática deben reforzar la permanencia del sistema informático, en condiciones de actividad adecuadas para que los usuarios accedan a los datos con la frecuencia y dedicación que requieran.

Este principio es particularmente importante en sistemas informáticos cuyo compromiso con el usuario, es prestar servicio permanente".

Confidencialidad: Se refiere a la protección de datos frente a la difusión no autorizada, la pérdida de confidencialidad puede resultar en problemas legales, pérdida del negocio o de credibilidad. El sistema contiene información que necesita protección contra la divulgación no autorizada, como información parcial de informes, información personal o información comercial patentada.

Estos aspectos además de lidiar con el riesgo que representan los atacantes remotos, se ven amenazados también por los riesgos por desastres naturales, empleados desleales, virus y sabotaje, entre otros.

"Basándose en este principio, las herramientas de seguridad informática deben proteger al sistema de invasiones, intrusiones y accesos, por parte de personas o programas no autorizados.

Este principio es particularmente importante en sistemas distribuidos, es decir, aquellos en los que usuarios, computadores y datos residen en localidades diferentes, pero están física y lógicamente interconectados".

1.1.4 FACTORES DE RIESGO

1.1.4.1 FACTORES TECNOLÓGICOS DE RIESGO

VIRUS INFORMÁTICOS

Definición

Un virus informático es un programa (código) que se replica, añadiendo una copia de sí mismo a otro(s) programa(s).

Los virus informáticos son particularmente dañinos porque pasan desapercibidos hasta que los usuarios sufren las consecuencias, que pueden ir desde anuncios inocuos hasta la pérdida total del sistema.

Características

Sus principales características son:

- ✓ **Auto-reproducción:** Es la capacidad que tiene el programa de replicarse (hacer copias de sí mismo), sin intervención o consentimiento del usuario.
- ✓ **Infección:** Es la capacidad que tiene el código de alojarse en otros programas, diferentes al portador original.

Propósitos

- ✓ **Afectar el software:** Sus instrucciones agregan nuevos archivos al sistema o manipulan el contenido de los archivos existentes, eliminándolo parcial o totalmente.
- ✓ **Afectar el hardware:** Sus instrucciones manipulan los componentes físicos. Su principal objetivo son los dispositivos de almacenamiento secundario y pueden sobrecalentar las unidades, disminuir la vida útil del medio, destruir la estructura lógica para recuperación de archivos (FAT) y otras consecuencias.

Clasificación

La inmensa cantidad de virus existentes, sus diferentes propósitos, sus variados comportamientos y sus diversas consecuencias, convierten su clasificación en un proceso complejo y polémico.

A continuación se presentan las categorías que agrupan a la mayoría de los virus conocidos. Sin embargo, es importante considerar que la aparición diaria de virus cada vez más sofisticados, puede llevar al surgimiento de nuevas categorías en cualquier momento.

- ✓ **Virus genérico o de archivo:** Se aloja como un parásito dentro de un archivo ejecutable y se replica en otros programas durante la ejecución. Los genéricos acechan al sistema esperando que se satisfaga alguna condición (fecha del sistema o número de archivos en un disco). Cuando esta condición "catalizadora" se presenta, el virus inicia su rutina de destrucción.
- ✓ **Virus mutante:** En general se comporta igual que el virus genérico, pero en lugar de replicarse exactamente, genera copias modificadas de sí mismo.
- ✓ **Virus recombinables:** Se unen, intercambian sus códigos y crean nuevos virus.
- ✓ **Virus "Bounty Hunter" (caza-recompensas):** Están diseñados para atacar un producto antivirus particular.
- ✓ **Virus específicos para redes:** Coleccionan contraseñas de acceso a la red, para luego reproducirse y dispersar sus rutinas destructivas en todos los computadores conectados.
- ✓ **Virus de sector de arranque:** Se alojan en la sección del disco cuyas instrucciones se cargan en memoria al inicializar el sistema. El virus alcanza la memoria antes que otros programas sean cargados e infecta cada nuevo disquete que se coloque en la unidad.
- ✓ **Virus de macro:** Se diseñan para infectar las macros que acompañan a una aplicación específica.

Una macro es un conjunto de instrucciones que ejecutan una tarea particular, activada por alguna aplicación específica como MS – Word o MS – Excel.

Son virus muy fáciles de programar y se dispersan rápidamente a través de anexos a e-mail, copia de archivos usando disquetes, etc.

- ✓ **Virus de Internet:** Se alojan en el código subyacente de las páginas web. Cuando el usuario accede a esos sitios en Internet, el virus se descarga y ejecuta en su sistema, pudiendo modificar o destruir la información almacenada.

  Son de rápida y fácil dispersión, puesto que se alojan y viajan en un medio de acceso multitudinario: Internet.

1.1.4.2 FACTORES HUMANOS DE RIESGO

Hackers

Los hackers son personas con avanzados conocimientos técnicos en el área informática y que enfocan sus habilidades hacia la invasión de sistemas a los que no tienen acceso autorizado. (Jemarinoi)

El hacker es una persona con grandes conocimientos de Internet, de programación en C y de sistemas operativos robustos como Linux y Unix y posee también mucho conocimiento en herramientas de seguridad como Firewalls entre otros. Los hackers son violadores de seguridad, son los piratas modernos pero al muy puro estilo cibernético.

En algunas ocasiones son contratados por las mismas empresas para mejorar sus procesos de seguridad y en otras alteran las operaciones de las empresas o roban la información de estas sin su consentimiento. (Monografias, 2004)

En general, los hackers persiguen dos objetivos:

- ✓ Probar que tienen las competencias para invadir un sistema protegido.
- ✓ Probar que la seguridad de un sistema tiene fallas.

Crackers

El término **cracker** (del inglés *crack*, romper) tiene varias acepciones, entre las que podemos observar las siguientes:

Es una persona que mediante ingeniería inversa realiza: seriales, keygens y cracks, los cuales sirven para modificar el comportamiento o ampliar la funcionalidad del software o hardware original al que se aplican, sin que en absoluto pretenda ser dañino para el usuario del mismo.

Es cualquier persona que viola la seguridad de un sistema informático de forma similar a como lo haría un hacker, sólo que a diferencia de este último, el cracker realiza la intrusión con fines de beneficio personal, no para hacer daño.

El término deriva de la expresión "criminal hacker", y fue creado alrededor de 1985 por contraposición al término hacker, en defensa de éstos últimos por el uso incorrecto del término. Se considera que la actividad realizada por esta clase de *cracker* es dañina e ilegal.

Por ello los crackers son criticados por la mayoría de hackers, por el desprestigio que les supone ante la opinión pública y las empresas, son aquellos que utilizan sus conocimientos técnicos para perturbar procesos

informáticos (Haffner y Markoff, 1995). Pueden considerarse un subgrupo marginal de la comunidad de hackers.

En ocasiones el *cracking* es la única manera de realizar cambios sobre software para el que su fabricante no presta soporte, especialmente cuando lo que se quiere es, o corregir defectos, o exportar datos a nuevas aplicaciones, en estos casos (sólo en estos casos) en la mayoría de legislaciones no se considera el *cracking* como actividad ilegal.

En muchos países existen crackers mercenarios que se ofrecen para romper la seguridad de cualquier programa informático que se le solicite y que contenga alguna protección para su instalación o ejecución. (Wikipedia, 2010)

“En general, los crackers persiguen dos objetivos:

- ✓ Destruir parcial o totalmente el sistema.
- ✓ Obtener un beneficio personal (tangible o intangible) como consecuencia de sus actividades”.

1.2 FORMAS DE VIOLAR LA SEGURIDAD DE LA INFORMACION

1.2.1 INGENIERIA SOCIAL

En el campo de la inseguridad informática, ingeniería social es la práctica de obtener información confidencial a través de la manipulación de usuarios legítimos. Es una técnica que pueden usar ciertas personas, tales como investigadores privados, criminales, o delincuentes computacionales, para obtener información, acceso o privilegios en sistemas de

información que les permitan realizar algún acto que perjudique o exponga la persona u organismo comprometido a riesgo o abusos.

El principio que sustenta la ingeniería social es el que en cualquier sistema "los usuarios son el eslabón débil". En la práctica, un ingeniero social usará comúnmente el teléfono o Internet para engañar a la gente, fingiendo ser, por ejemplo, un empleado de algún banco o alguna otra empresa, un compañero de trabajo, un técnico o un cliente. Vía Internet o la web se usa, adicionalmente, el envío de solicitudes de renovación de permisos de acceso a páginas web o memos falsos que solicitan respuestas e incluso las famosas "cadenas", llevando así a revelar información sensible, o a violar las políticas de seguridad típicas. Con este método, los ingenieros sociales aprovechan la tendencia natural de la gente a reaccionar de manera predecible en ciertas situaciones, -por ejemplo proporcionando detalles financieros a un aparente funcionario de un banco- en lugar de tener que encontrar agujeros de seguridad en los sistemas informáticos.

Quizá el ataque más simple pero muy efectivo sea engañar a un usuario llevándolo a pensar que un administrador del sistema está solicitando una contraseña para varios propósitos legítimos. Los usuarios de sistemas de Internet frecuentemente reciben mensajes que solicitan contraseñas o información de tarjeta de crédito, con el motivo de "crear una cuenta", "reactivar una configuración", u otra operación benigna; a este tipo de ataques se los llama phishing (pesca). Los usuarios de estos sistemas deberían ser advertidos temprana y frecuentemente para que no divulguen contraseñas u otra información sensible a personas que dicen ser administradores. En realidad, los administradores de sistemas informáticos raramente (o nunca) necesitan saber la contraseña de los usuarios para llevar a cabo sus tareas. Sin embargo incluso este tipo de ataque podría no ser necesario — en una

encuesta realizada por la empresa Boixnet, el 90% de los empleados de oficina de la estación Waterloo de Londres reveló sus contraseñas a cambio de un bolígrafo barato.

Otro ejemplo contemporáneo de un ataque de ingeniería social es el uso de archivos adjuntos en e-mails, ofreciendo, por ejemplo, fotos "íntimas" de alguna persona famosa o algún programa "gratis" (a menudo aparentemente provenientes de alguna persona conocida) pero que ejecutan código malicioso (por ejemplo, usar la máquina de la víctima para enviar cantidades masivas de spam). Ahora, después de que los primeros e-mails maliciosos llevaran a los proveedores de software a deshabilitar la ejecución automática de archivos adjuntos, los usuarios deben activar esos archivos de forma explícita para que ocurra una acción maliciosa. Muchos usuarios, sin embargo, abren casi ciegamente cualquier archivo adjunto recibido, concretando de esta forma el ataque.

La ingeniería social también se aplica al acto de manipulación cara a cara para obtener acceso a los sistemas computacionales.

La principal defensa contra la ingeniería social es educar y entrenar a los usuarios en el uso de políticas de seguridad y asegurarse de que estas sean seguidas.

Uno de los ingenieros sociales más famosos de los últimos tiempos es Kevin Mitnick. Según su opinión, la ingeniería social se basa en estos cuatro principios:

1. Todos queremos ayudar.
2. El primer movimiento es siempre de confianza hacia el otro.
3. No nos gusta decir No.
4. A todos nos gusta que nos alaben. (Wikipedia, 2010)

1.2.2 AMENAZAS A LA SEGURIDAD DE LA INFORMACIÓN

ATAQUES

Existen varios tipos de ataques en un sistema informático, los podemos dividir en dos grandes grupos:

- ✓ **Ataques pasivos:** No modifican la información contenida en los sistemas. Ni el estado del sistema ni su operación son alterados.
- ✓ **Ataques activos:** Estos implican la modificación de la información contenida en un sistema. Esto puede alterar el estado del sistema o su operación.

Entre los atacantes activos están los hackers y los crackers. (Monografias, 2010)

Los tres elementos principales a proteger en cualquier sistema informático son el *software*, el *hardware* y los datos. Contra cualquiera de los tres elementos dichos anteriormente (pero principalmente sobre los datos) se pueden realizar multitud de ataques o, dicho de otra forma, están expuestos a diferentes amenazas. Generalmente, la taxonomía más elemental de estas amenazas las divide en cuatro grandes grupos: interrupción, interceptación, modificación y fabricación. Un ataque se clasifica como **interrupción** si hace que un objeto del sistema se pierda, quede inutilizable o no disponible. Se tratará de una **interceptación** si un elemento no autorizado consigue un acceso a un determinado objeto del sistema, y de una **modificación** si además de conseguir el acceso consigue modificar el objeto; algunos autores **[Olovsson, 1992]** consideran un caso especial de la modificación: la **destrucción**, entendiéndola como una modificación que inutiliza al objeto afectado. Por último, se dice que un ataque es una **fabricación** si se trata de una

modificación destinada a conseguir un objeto similar al atacado de forma que sea difícil distinguir entre el objeto original y el "fabricado". En la figura se muestran estos tipos de ataque de una forma gráfica.

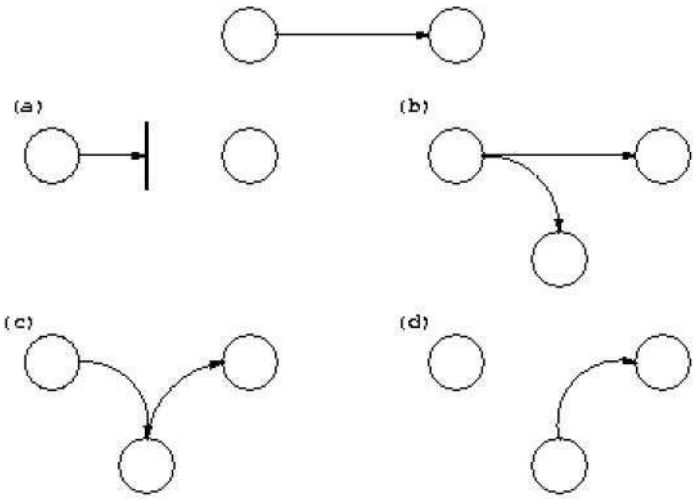


Ilustración 1-1 Flujo normal de información entre emisor y receptor y posibles amenazas: (a) interrupción, (b) interceptación, (c) modificación y (d) fabricación

Fuente: http://mmc.geofisica.unam.mx/

En la gran mayoría de publicaciones relativas a la seguridad informática en general, tarde o temprano se intenta clasificar en grupos a los posibles elementos que pueden atacar nuestro sistema. Con frecuencia, especialmente en las obras menos técnicas y más orientadas a otros aspectos de la seguridad **[Icove, 1995] [Meyer, 1989]**, se suele identificar a los atacantes únicamente como personas; esto tiene sentido si hablamos por ejemplo de responsabilidades por un delito informático. Pero en este trabajo es preferible hablar de "elementos" y no de personas: aunque a veces lo olvidemos, nuestro sistema puede verse perjudicado por múltiples entidades aparte de humanos. A continuación se presenta una relación de los elementos que potencialmente pueden amenazar a nuestro sistema. No pretende ser exhaustiva, ni por

supuesto una taxonomía formal; simplemente trata de proporcionar una idea acerca de qué o quién amenaza un sistema.

a) Personas

No podemos engañarnos, la mayoría de ataques a nuestro sistema van a provenir en última instancia de personas que, intencionada o inintencionadamente, pueden causarnos enormes pérdidas.

Aquí se listan los diferentes tipos de personas que de una u otra forma pueden constituir un riesgo para nuestros sistemas; generalmente se dividen en dos grandes grupos: los atacantes **pasivos**, aquellos que fisgonean por el sistema pero no lo modifican o destruyen, y los **activos**, aquellos que dañan el objetivo atacado, o lo modifican en su favor. Generalmente los curiosos y los *crackers* realizan ataques pasivos (que se pueden convertir en activos), mientras que los terroristas y ex-empleados realizan ataques activos puros; los intrusos remunerados suelen ser atacantes pasivos si nuestra red o equipo no es su objetivo, y activos en caso contrario, y el personal realiza ambos tipos indistintamente, dependiendo de la situación concreta.

- ✓ Personal.
- ✓ Ex-empleados.
- ✓ Curiosos.
- ✓ Crackers.
- ✓ Terroristas.
- ✓ Intrusos (remunerados).

b) Amenazas lógicas.

Bajo la etiqueta de “amenazas lógicas” encontramos todo tipo de programas que de una forma u otra pueden dañar a nuestro sistema, creados de forma intencionada para ello (*software* malicioso, también conocido como *malware*) o simplemente por error (*bugs* o agujeros). Una excelente lectura que estudia

las definiciones de algunas de estas amenazas y su implicación se presenta en **[Garfinkel, 1996]**; otra buena descripción, pero a un nivel más general, se puede encontrar en **[Parker, 1981].**

- ✓ **Software incorrecto:** Las amenazas más habituales a un sistema provienen de errores cometidos de forma involuntaria por los programadores de sistemas o de aplicaciones.
- ✓ **Herramientas de Seguridad:** Cualquier herramienta de seguridad representa un arma de doble filo: de la misma forma que un administrador las utiliza para detectar y solucionar fallos en sus sistemas o en la subred completa, un potencial intruso las puede utilizar para detectar esos mismos fallos y aprovecharlos para atacar los equipos.
- ✓ **Puertas traseras:** Durante el desarrollo de aplicaciones grandes o de sistemas operativos es habitual entre los programadores insertar "atajos". A estos atajos se les denomina puertas traseras. Algunos programadores pueden dejar estos atajos en las versiones definitivas de su *software*; la cuestión es que si un atacante descubre una de estas puertas traseras (no nos importa el método que utilice para hacerlo) va a tener un acceso global a datos que no debería poder leer.
- ✓ **Bombas lógicas:** Son partes de código de ciertos programas que permanecen sin realizar ninguna función hasta que son activadas. Los activadores más comunes de estas bombas lógicas pueden ser la ausencia o presencia de ciertos ficheros, la llegada de una fecha concreta; cuando la bomba se activa va a poder realizar cualquier tarea que pueda realizar la persona, los efectos pueden ser fatales.
- ✓ **Virus:** Un virus es una secuencia de código que se inserta en un fichero ejecutable (denominado *huésped*), de forma que cuando el archivo se

ejecuta, el virus también lo hace, insertándose a sí mismo en otros programas.

- ✓ **Gusanos:** Un gusano es un programa capaz de ejecutarse y propagarse por sí mismo a través de redes, en ocasiones portando virus o aprovechando *bugs* de los sistemas a los que se conecta para dañarlos.
- ✓ **Caballos de Troya:** Los troyanos o caballos de Troya son instrucciones escondidas en un programa de forma que éste parezca realizar las tareas que un usuario espera de él, pero que realmente ejecuta funciones ocultas sin el conocimiento del usuario.

c) Catástrofes

Las catástrofes (naturales o artificiales) son la amenaza menos probable contra los entornos habituales: simplemente por su ubicación geográfica, a nadie se le escapa que la probabilidad de sufrir un terremoto o una inundación que afecte a los sistemas informáticos en una gran ciudad, es relativamente baja, al menos en comparación con el riesgo de sufrir un intento de acceso por parte de un pirata o una infección por virus. Sin embargo, el hecho de que las catástrofes sean amenazas poco probables no implica que contra ellas no se tomen unas medidas básicas, ya que sí se produjeran generarían los mayores daños.

1.3 ESTRATEGIAS PARA FORTALECER LA SEGURIDAD DE LOS DATOS Y LA INFORMACIÓN

La metodología de seguridad está diseñada para ayudar a los profesionales de la seguridad a desarrollar una estrategia para proteger la *disponibilidad, integridad y confidencialidad* de los datos de los sistemas informáticos (IT)

de las organizaciones. Es de interés para los administradores de recursos de información, los directores de seguridad informática y los administradores; y tiene un valor especial para todos aquellos que intentan establecer directivas de seguridad.

La metodología ofrece un acercamiento sistemático a esta importante tarea y, como precaución final, también implica el establecimiento de planes de contingencia en caso de desastre.

Los administradores de seguridad tienen que decidir el tiempo, dinero y esfuerzo que hay que invertir para desarrollar las directivas y controles de seguridad apropiados.

Cada organización debe analizar sus necesidades específicas y determinar sus requisitos y limitaciones en cuanto a recursos y programación. Cada sistema informático, entorno y directiva organizativa es distinta, lo que hace que cada servicio y cada estrategia de seguridad sean únicos. Sin embargo, los fundamentos de una buena seguridad siguen siendo los mismos. **[Benson, 2001].**

a) Identificar métodos, herramientas y técnicas de ataques probables

Las listas de amenazas, de las que disponen la mayoría de las organizaciones, ayudan a los administradores de seguridad a identificar los distintos métodos, herramientas y técnicas de ataque que se pueden utilizar en los ataques. Los métodos pueden abarcar desde virus y gusanos a la adivinación de contraseñas y la interceptación del correo electrónico. Es importante que los administradores actualicen constantemente sus conocimientos en esta área, ya que los nuevos métodos, herramientas y técnicas para sortear las medidas de seguridad evolucionan de forma continua.

b) Establecer estrategias proactivas y reactivas

En cada método, el plan de seguridad debe incluir una estrategia *proactiva* y otra *reactiva*. La estrategia *proactiva* o de previsión de ataques es un conjunto de pasos que ayuda a reducir al mínimo la cantidad de puntos vulnerables existentes en las directivas de seguridad y a desarrollar planes de contingencia. La determinación del daño que un ataque va a provocar en un sistema y las debilidades y puntos vulnerables explotados durante este ataque ayudará a desarrollar la estrategia proactiva.

La estrategia *reactiva* o estrategia posterior al ataque ayuda al personal de seguridad a evaluar el daño que ha causado el ataque, a repararlo o a implementar el plan de contingencia desarrollado en la estrategia proactiva, a documentar y aprender de la experiencia, y a conseguir que las funciones comerciales se normalicen lo antes posible.

c) Pruebas

El último elemento de las estrategias de seguridad, las pruebas y el estudio de sus resultados, se lleva a cabo después de que se han puesto en marcha las estrategias reactiva y proactiva. La realización de ataques simulados en sistemas de pruebas o en laboratorios permite evaluar los lugares en los que hay puntos vulnerables y ajustar las directivas y los controles de seguridad en consecuencia. Estas pruebas no se deben llevar a cabo en los sistemas de producción real, ya que el resultado puede ser desastroso. La carencia de laboratorios y equipos de pruebas a causa de restricciones presupuestarias puede imposibilitar la realización de ataques simulados. Para asegurar los fondos necesarios para las pruebas, es importante que los directivos sean conscientes de los riesgos y consecuencias de los ataques, así como de las medidas de seguridad que se pueden adoptar para proteger al sistema, incluidos los procedimientos de las pruebas. Si es posible, se deben probar

físicamente y documentar todos los casos de ataque para determinar las mejores directivas y controles de seguridad posibles que se van a implementar.

d) Equipos de respuestas a incidentes

Es aconsejable formar un equipo de respuesta a incidentes. Este equipo debe estar implicado en los trabajos proactivos del profesional de la seguridad. Entre éstos se incluyen:

- ✓ El desarrollo de instrucciones para controlar incidentes.
- ✓ La identificación de las herramientas de software para responder a incidentes y eventos.
- ✓ La investigación y desarrollo de otras herramientas de seguridad informática.
- ✓ La realización de actividades formativas y de motivación.
- ✓ La realización de investigaciones acerca de virus.
- ✓ La ejecución de estudios relativos a ataques al sistema.

Estos trabajos proporcionarán los conocimientos que la organización puede utilizar y la información que hay que distribuir antes y durante los incidentes. Una vez que el administrador de seguridad y el equipo de respuesta a incidentes han realizado estas funciones proactivas, el administrador debe delegar la responsabilidad del control de incidentes al equipo de respuesta a incidentes.

Esto no significa que el administrador no deba seguir implicado o formar parte del equipo, sino que no tenga que estar siempre disponible, necesariamente, y que el equipo debe ser capaz de controlar los incidentes por sí mismo. El equipo será el responsable de responder a incidentes como virus, gusanos o cualquier otro código dañino, invasión, engaños, desastres naturales y ataques del personal interno. El equipo también debe participar en el

análisis de cualquier evento inusual que pueda estar implicado en la seguridad de los equipos o de la red.

1.3.1 METODOLOGÍA PARA LA DEFINICIÓN DE UNA ESTRATEGIA DE SEGURIDAD

La figura explica una metodología para definir una estrategia de seguridad informática que se puede utilizar para implementar directivas y controles de seguridad con el objeto de aminorar los posibles ataques y amenazas. Los métodos se pueden utilizar en todos los tipos de ataques a sistemas, independientemente de que sean intencionados, no intencionados o desastres naturales, y, por consiguiente, se puedan volver a utilizar en distintos casos de ataque.

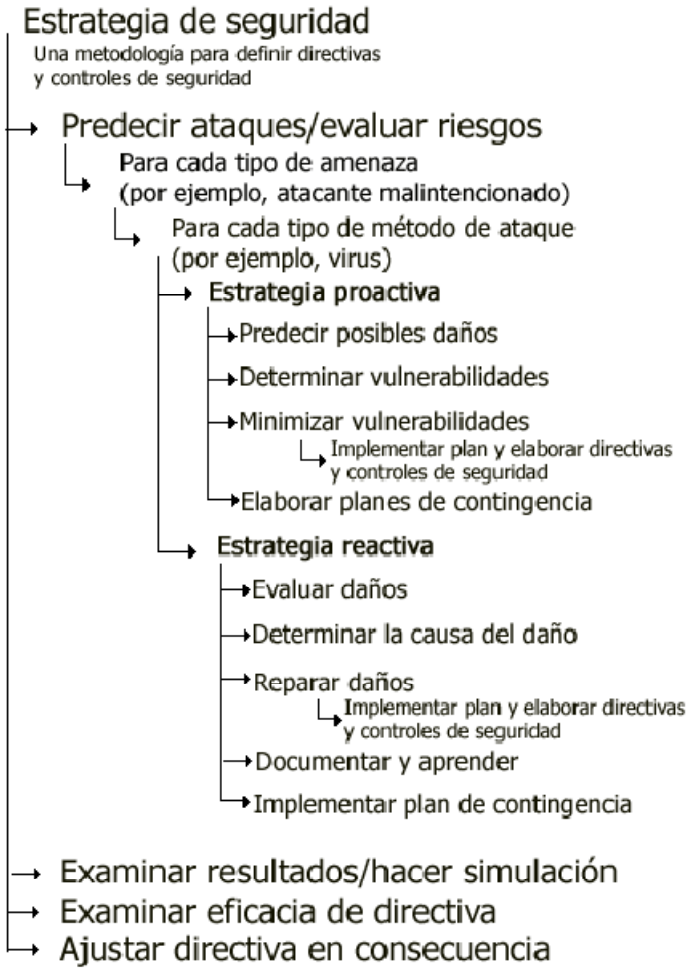


Ilustración 1-2 Metodología de estrategias de seguridad

Fuente: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/jerez_l_ca/capitulo1.pdf

a) Predecir posibles ataques y analizar riesgo

La primera fase de la metodología esquematizada en la figura, es determinar los ataques que se pueden esperar y las formas de defenderse contra ellos. Es imposible estar preparado contra todos los ataques; por lo tanto, hay que prepararse para los que tiene más probabilidad de sufrir la organización. Siempre es mejor prevenir o aminorar los ataques que reparar el daño que han causado.

Para mitigar los ataques es necesario conocer las distintas amenazas que ponen en peligro los sistemas, las técnicas correspondientes que se pueden

utilizar para comprometer los controles de seguridad y los puntos vulnerables que existen en las directivas de seguridad. El conocimiento de estos tres elementos de los ataques ayuda a predecir su aparición e, incluso, su duración ubicación. La predicción de los ataques trata de pronosticar su probabilidad, lo que depende del conocimiento de sus distintos aspectos. Los diferentes aspectos de un ataque se pueden mostrar en la siguiente ecuación:

AMENAZAS + MOTIVOS + HERRAMIENTAS Y TÉCNICAS + PUNTOS VULNERABLES = **ATAQUE**

b) Para cada tipo de amenaza

Considere todas las amenazas posibles que causan ataques en los sistemas. Entre éstas se incluyen los agresores malintencionados, las amenazas no intencionadas y los desastres naturales. La siguiente figura clasifica las distintas amenazas a los sistemas:

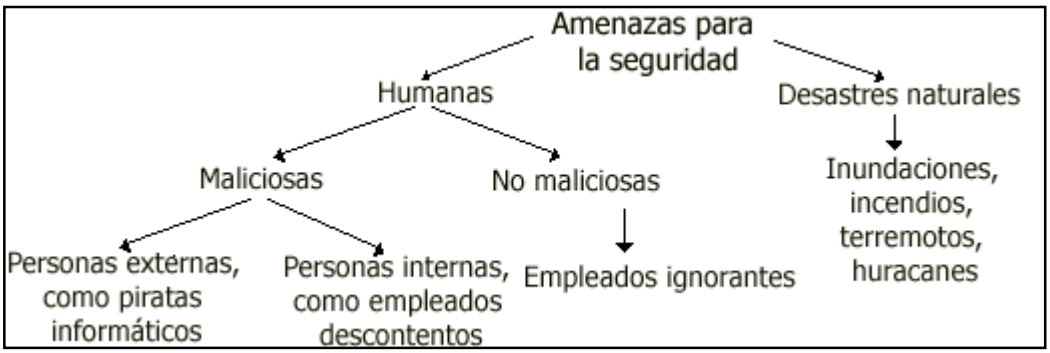


Ilustración 1-3 Amenazas para la Seguridad

Fuente: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/jerez_l_ca/capitulo1.pdf

Amenazas como empleados ignorantes o descuidados, y los desastres naturales no implican motivos u objetivos; por lo tanto, no se utilizan métodos, herramientas o técnicas predeterminadas para iniciar los ataques.

Casi todos estos ataques o infiltraciones en la seguridad se generan internamente; raras veces los va a iniciar alguien ajeno a la organización.
Para estos tipos de amenazas, el personal de seguridad necesita implementar estrategias proactivas o reactivas siguiendo las instrucciones de la figura.

c) Para cada tipo de método de ataque

Para iniciar un ataque, se necesita un método, una herramienta o una técnica para explotar los distintos puntos vulnerables de los sistemas, de las directivas de seguridad y de los controles. Los agresores pueden utilizar varios métodos para iniciar el mismo ataque. Por lo tanto, la estrategia defensiva debe personalizarse para cada tipo de método utilizado en cada tipo de amenaza.
De nuevo, es importante que los profesionales de la seguridad estén al día en los diferentes métodos, herramientas y técnicas que utilizan los agresores. La siguiente es una lista breve de esta técnica:

- ✓ Ataques de denegación de servicio.
- ✓ Ataques de invasión.
- ✓ Ingeniería social.
- ✓ Virus.
- ✓ Gusanos.
- ✓ Caballos de Troya.
- ✓ Modificación de paquetes.
- ✓ Adivinación de contraseñas.
- ✓ Interceptación de correo electrónico.

d) Estrategia proactiva

La estrategia proactiva es un conjunto de pasos predefinidos que deben seguirse para evitar ataques antes de que ocurran. Entre estos pasos se incluye observar cómo podría afectar o dañar el sistema, y los puntos vulnerables. Los

conocimientos adquiridos en estas evaluaciones pueden ayudar a implementar las directivas de seguridad que controlarán o aminorarán los ataques.

Éstos son los tres pasos de la estrategia proactiva:

- ✓ Determinar el daño que causará el ataque.
- ✓ Establecer los puntos vulnerables y las debilidades que explotará el ataque.
- ✓ Reducir los puntos vulnerables y las debilidades que se ha determinado en el sistema para ese tipo de ataque específico.

El seguimiento de estos pasos para analizar los distintos tipos de ataques tiene una ventaja adicional: comenzará a emerger un modelo, ya que en los diferentes factores se superponen para diferentes ataques. Este modelo puede ser útil al determinar las áreas de vulnerabilidad que plantean el mayor riesgo para la empresa. También es necesario tomar nota del costo que supone la pérdida de los datos frente al de la implementación de controles de seguridad. La ponderación de los riesgos y los costos forma parte de un análisis de riesgos del sistema que se explica en el documento técnico acerca del diseño de la seguridad. Las directivas y controles de seguridad no serán, en ningún caso, totalmente eficaces al eliminar los ataques. Éste es el motivo por el que es necesario desarrollar planes de recuperación y de contingencia en caso de que se quebranten los controles de seguridad.

e) Determinar el daño posible que puede causar un ataque

Los daños posibles pueden oscilar entre pequeños fallos del equipo y la pérdida catastrófica de los datos. El daño causado al sistema dependerá del tipo de ataque. Si es posible, utilice un entorno de prueba o de laboratorio para clarificar los daños que provocan los diferentes tipos de ataques. Ello permitirá al personal de seguridad ver el daño físico que causan los ataques experimentales. No todos los ataques causan el mismo daño.

f) Determinar los puntos vulnerables o las debilidades que pueden explotar los ataques

Si se pueden descubrir los puntos vulnerables que explota un ataque específico, se pueden modificar las directivas y los controles de seguridad actuales o implementar otras nuevas para reducir estos puntos vulnerables. La determinación del tipo de ataque, amenaza y método facilita el descubrimiento de los puntos vulnerables existentes. Esto se puede reconocer por medio de una prueba real. Se deben determinar los puntos vulnerables o debilidades en las áreas de seguridad física, de datos y de red.

g) Reducir los puntos vulnerables y debilidades que puede explotar un posible ataque

La reducción de los puntos vulnerables y las debilidades del sistema de seguridad que se determinaron en la evaluación anterior es el primer paso para desarrollar directivas y controles de seguridad eficaces. Ésta es la compensación de la estrategia proactiva.

Mediante la reducción de los puntos vulnerables, el personal de seguridad puede hacer disminuir tanto la probabilidad de un ataque como su eficacia, si se produce alguno.

Tenga cuidado de no implementar controles demasiado estrictos, ya que la disponibilidad de la información se convertiría en un problema. Debe haber un cuidado equilibrio entre los controles de seguridad y el acceso a la información. Los usuarios deben tener la mayor libertad posible para tener acceso a la información.

h) Elaborar planes de contingencia

Un plan de contingencia es un plan alternativo que debe desarrollarse en caso de que algún ataque penetre en el sistema y dañe los datos o cualquier otro

activo, detenga las operaciones comerciales habituales y reste productividad. El plan se sigue si el sistema no se puede restaurar a tiempo. Su objetivo final es mantener la disponibilidad, integridad y confidencialidad de los datos (es el proverbial "Plan B").

Debe haber un plan para cada tipo de ataque y tipo de amenaza. Cada plan consta de un conjunto de pasos que se han de emprender en el caso de que un ataque logre pasar las directivas de seguridad.

i) Estrategia reactiva

La estrategia reactiva se implementa cuando ha fallado la estrategia proactiva y define los pasos que deben adoptarse después o durante un ataque. Ayuda a identificar el daño causado y los puntos vulnerables que se explotaron en el ataque, a determinar por qué tuvo lugar, a reparar el daño que causó y a implementar un plan de contingencia, si existe. Tanto la estrategia reactiva como la proactiva funcionan conjuntamente para desarrollar directivas y controles de seguridad con el fin de reducir los ataques y el daño que causan. El equipo de respuesta a incidentes debe incluirse en los pasos adoptados durante o después del ataque para ayudar a evaluarlo, a documentar el evento y a aprender de él.

j) Evaluar el daño

Determine el daño causado durante el ataque. Esto debe hacerse lo antes posible para que puedan comenzar las operaciones de restauración. Si no se puede evaluar el daño a tiempo, debe implementarse un plan de contingencia para que puedan proseguir las operaciones comerciales y la productividad normales.

k) Determinar la causa del daño

Para determinar la causa del daño, es necesario saber a qué recursos iba dirigido el ataque y qué puntos vulnerables se explotaron para obtener acceso o perturbar los servicios. Revise los registros del sistema, los registros de auditoría y las pistas de auditoría. Estas revisiones suelen ayudar a descubrir el lugar del sistema en el que se originó el ataque y qué otros recursos resultaron afectados.

l) Revisar el resultado y hacer simulaciones

Tras el ataque o tras defenderse de él, revise su resultado con respecto al sistema. La revisión debe incluir la pérdida de productividad, la pérdida de datos o de hardware, y el tiempo que se tarda en recuperarlos.

Documente también el ataque y, si es posible, haga un seguimiento del lugar en el que se originó, qué métodos se utilizaron para iniciarlo y qué puntos vulnerables se explotaron. Para obtener los mejores resultados posibles, realice simulaciones en un entorno de prueba.

m) Reparar el daño

Es muy importante que el daño se repare lo antes posible para restaurar las operaciones comerciales normales y todos los datos perdidos durante el ataque. Los planes y procedimientos para la recuperación de desastres de la organización deben cubrir la estrategia de restauración. El equipo de respuesta a incidentes también debe poder controlar el proceso de restauración y recuperación, y ayudar en este último.

n) Documentar y aprender

Es importante documentar el ataque una vez que se ha producido. La documentación debe abarcar todos los aspectos que se conozcan del mismo,

entre los que se incluyen el daño que ha causado (en hardware y software, pérdida de datos o pérdida de productividad), los puntos vulnerables y las debilidades que se explotaron durante el ataque, la cantidad de tiempo de producción perdido y los procedimientos tomados para reparar el daño. La documentación ayudará a modificar las estrategias proactivas para evitar ataques futuros o mermar los daños.

o) Implementar un plan de contingencia

Si ya existe algún plan de contingencia, se puede implementar para ahorrar tiempo y mantener el buen funcionamiento de las operaciones comerciales. Si no hay ningún plan de contingencia, desarrolle un plan apropiado basado de la documentación del paso anterior.

p) Revisar la eficacia de las directivas

Si hay directivas para defenderse de un ataque que se ha producido, hay que revisar y comprobar su eficacia. Si no hay directivas, se deben redactar para aminorar o impedir ataques futuros.

q) Ajustar las directivas en consecuencia

Si la eficacia de la directiva no llega al estándar, hay que ajustarla en consecuencia. Las actualizaciones de las directivas debe realizarlas el personal directivo relevante, los responsables de seguridad, los administradores y el equipo de respuesta a incidentes. Todas las directivas deben seguir las reglas e instrucciones generales de la organización. (Catarina)

1.4 TÉCNICAS PARA ASEGURAR EL SISTEMA

El activo más importante que se posee es la información y, por lo tanto, deben existir técnicas que la aseguren, más allá de la seguridad física que se establezca sobre los equipos en los cuales se almacena. Estas técnicas las brinda la seguridad lógica que consiste en la aplicación de *barreras y procedimientos* que resguardan el acceso a los datos y sólo permiten acceder a ellos a las personas autorizadas para hacerlo.

Existe un viejo dicho en la seguridad informática que dicta: *"lo que no está permitido debe estar prohibido"* y ésta debe ser la meta perseguida.

Los medios para conseguirlo son:

1. Restringir el acceso (de personas de la organización y de las que no lo son) a los programas y archivos.
2. Asegurar que los operadores puedan trabajar pero que no puedan modificar los programas ni los archivos que no correspondan (sin una supervisión minuciosa).
3. Asegurar que se utilicen los datos, archivos y programas correctos en/y/por el procedimiento elegido.
4. Asegurar que la información transmitida sea la misma que reciba el destinatario al cual se ha enviado y que no le llegue a otro.
5. Asegurar que existan sistemas y pasos de emergencia alternativos de transmisión entre diferentes puntos.
6. Organizar a cada uno de los empleados por jerarquía informática, con claves distintas y permisos bien establecidos, en todos y cada uno de los sistemas o aplicaciones empleadas.
7. Actualizar constantemente las contraseñas de acceso a los sistemas de cómputo. (Wikipedia, 2010)

1.4.1 CÓMO IMPLEMENTAR UNA POLÍTICA DE SEGURIDAD

Generalmente, la seguridad de los sistemas informáticos se concentra en garantizar el derecho a acceder a datos y recursos del sistema configurando los mecanismos de autentificación y control que aseguran que los usuarios de estos recursos sólo posean los derechos que se les han otorgado.

Los mecanismos de seguridad pueden sin embargo, causar inconvenientes a los usuarios. Con frecuencia, las instrucciones y las reglas se vuelven cada vez más complicadas a medida que la red crece. Por consiguiente, la seguridad informática debe estudiarse de modo que no evite que los usuarios desarrollen usos necesarios y así puedan utilizar los sistemas de información en forma segura.

Por esta razón, uno de los primeros pasos que debe dar una compañía es definir una **política de seguridad** que pueda implementar en función a las siguientes cuatro etapas:

- ✓ Identificar las necesidades de seguridad y los riesgos informáticos que enfrenta la compañía así como sus posibles consecuencias.
- ✓ Proporcionar una perspectiva general de las reglas y los procedimientos que deben implementarse para afrontar los riesgos identificados en los diferentes departamentos de la organización.
- ✓ Controlar y detectar las vulnerabilidades del sistema de información, y mantenerse informado acerca de las falencias en las aplicaciones y en los materiales que se usan.
- ✓ Definir las acciones a realizar y las personas a contactar en caso de detectar una amenaza.

La política de seguridad comprende todas las reglas de seguridad que sigue una organización (en el sentido general de la palabra). Por lo tanto, la

administración de la organización en cuestión debe encargarse de definirla, ya que afecta a todos los usuarios del sistema.

En este sentido, no son sólo los administradores de informática los encargados de definir los derechos de acceso sino sus superiores. El rol de un administrador de informática es el de asegurar que los recursos de informática y los derechos de acceso a estos recursos coincidan con la política de seguridad definida por la organización.

Es más, dado que el/la administrador/a es la única persona que conoce perfectamente el sistema, deberá proporcionar información acerca de la seguridad a sus superiores, eventualmente aconsejar a quienes toman las decisiones con respecto a las estrategias que deben implementarse, y constituir el punto de entrada de las comunicaciones destinadas a los usuarios en relación con los problemas y las recomendaciones de seguridad.

La seguridad informática de una compañía depende de que los empleados (usuarios) aprendan las reglas a través de sesiones de capacitación y de concientización. Sin embargo, la seguridad debe ir más allá del conocimiento de los empleados y cubrir las siguientes áreas:

- ✓ Un mecanismo de seguridad física y lógica que se adapte a las necesidades de la compañía y al uso de los empleados.
- ✓ Un procedimiento para administrar las actualizaciones.
- ✓ Una estrategia de realización de copias de seguridad (backup) planificada adecuadamente.
- ✓ Un plan de recuperación luego de un incidente.
- ✓ Un sistema documentado actualizado. (Kioskea, 2008)

1.5 ALGUNAS AFIRMACIONES ERRÓNEAS COMUNES ACERCA DE LA SEGURIDAD

- ✓ **Mi sistema no es importante para un cracker**

Esta afirmación se basa en la idea de que no introducir contraseñas seguras en una empresa no entraña riesgos, pues ¿quién va a querer obtener información mía? Sin embargo, dado que los métodos de contagio se realizan por medio de programas *automáticos*, desde unas máquinas a otras, estos no distinguen buenos de malos, interesantes de no interesantes, etc. Por tanto abrir sistemas y dejarlos sin claves es facilitar la vida a los virus.

- ✓ **Estoy protegido pues no abro archivos que no conozco**

Esto es falso, pues existen múltiples formas de contagio, además los programas realizan acciones sin la supervisión del usuario poniendo en riesgo los sistemas.

- ✓ **Como tengo antivirus estoy protegido**

En general los programas antivirus no son capaces de detectar todas las posibles formas de contagio existentes, ni las nuevas que pudieran aparecer conforme los ordenadores aumenten las capacidades de comunicación, además los antivirus son vulnerables a desbordamientos de búfer que hacen que la seguridad del sistema operativo se vea más afectada aún.

- ✓ **Como dispongo de un firewall no me contagio**

Esto únicamente proporciona una limitada capacidad de respuesta. Las formas de infectarse en una red son múltiples. Unas provienen directamente de accesos al sistema (de lo que protege un firewall) y otras de conexiones que se realizan (de las que no me protege). Emplear usuarios con altos privilegios

para realizar conexiones puede entrañar riesgos, además los firewalls de aplicación (los más usados) no brindan protección suficiente contra el spoofing (uso de técnicas de suplantación de identidad generalmente con usos maliciosos o de investigación).

- ✓ **Tengo un servidor web cuyo sistema operativo es un Unix actualizado a la fecha**

Puede que esté protegido contra ataques directamente hacia el núcleo, pero si alguna de las aplicaciones web (PHP, Perl, Cpanel, etc.) está desactualizada, un ataque sobre algún script de dicha aplicación puede permitir que el atacante abra una shell y por ende ejecutar comandos en el Unix. (Wikipedia, 2010)

1.6 ORGANISMOS OFICIALES DE SEGURIDAD INFORMÁTICA

Existen organismos oficiales encargados de asegurar servicios de prevención de riesgos y asistencia a los tratamientos de incidencias, tales como el **CERT/CC** (*Computer Emergency Response Team Coordination Center*) del *SEI* (Software Engineering Institute) de la Carnegie Mellon University el cual es un centro de alerta y reacción frente a los ataques informáticos, destinados a las empresas o administradores, pero generalmente estas informaciones son accesibles a todo el mundo. (Wikipedia, 2010)

2 EL HONEYPOT COMO UNA HERRAMIENTA DE SEGURIDAD

2.1 HONEYPOT

El Honeypot es una herramienta de apoyo que permite mejorar la seguridad de la institución, cuyo objetivo está en atraer a los atacantes, haciéndoles pretender que el sistema de seguridad de la red es muy vulnerable o débil a los ataques, este objetivo se cumple con un conjunto de computadores conectados a la red. (Wikipedia, Honeypot)

Esta herramienta informática "honeypot" también es una herramienta de investigación, la cual cumple otra función, en la que permite recoger información acerca de los atacantes y las técnicas que utilizan para ingresar a la red, esto nos permite conocer a un nivel más profundo sobre el enemigo para poder combatirlo, además de monitorizar y dar un aviso al administrador informándole que está siendo atacado. (el-directorio)- (Blanco Ramos, et al.)- (Hack, Honeypots-Definición)

Un honeypot es un sistema a ser comprometido, ya que pertenece a la red, definiéndose así como un sistema trampa con el fin de recolectar información de gran valor dentro de la seguridad informática, es un sistema reutilizable ya que no necesita de actualización para su defensa, mientras tanto otros sistemas si la necesitan y que cuando no se renuevan pierden su utilidad y su funcionalidad. (UTPL)

Existen dos tipos de honeypot:

- ✓ Los honeypot de baja interacción, son aquellos que se limitan a simular sistemas operativos que no son existentes en la realidad, son usados como medidas de seguridad.
- ✓ Los honeypot de alta interacción son aquellos que trabajan sobre sistemas reales y son capaces de reunir mucha información, estos también se le suele utilizar para la investigación. (Wikipedia, Tipos de Honeypots)

El honeypot imita el comportamiento de los sistemas que pueden ser de interés para el intruso, además el honeypot se lo configura condicionándole que cualquier tráfico entrante o saliente sea considerado como sospechoso, por lo tanto el monitoreo se lo realiza constantemente y sin ninguna distinción de usuarios.

2.1.1 VENTAJAS

- ✓ **Encriptación**

A diferencia de la mayoría de las tecnologías de seguridad (como los IDS "Sistema de detección de intrusos") los honeypot funcionan bien en el cifrado. No importa que los intrusos traten de violar la seguridad de un honeypot, el honeypot detectará y capturará a los atacantes, con esto permitirá fortalecer al sistema. (Compute-rs)

- ✓ **Nuevas Herramientas y tácticas**

Los Honeypots son una herramienta para la recolección de nuevas formas y perfiles de ataque, con en el fin de cumplir su objetivo que es el de penetrar

los servicios informáticos de la institución u organización, y así generar daños a dichos servicios ofrecidos en una organización. (Vinklud A.)

- ✓ **Protocolo IPv6**

La mayoría de las herramientas de seguridad de la información presentan un problema y es el que no soportan el protocolo IPv6 sucesor del actual IPv4 ampliamente utilizado en internet. Este protocolo está siendo principalmente utilizado en los países asiáticos como Japón, pero los sistemas trampa logran identificar ataques provenientes de intrusos que utilizan este protocolo para usarlo como una ventaja de sus ataques. (Vinklud A. , Honeypots-Protocolo-IPv6)

- ✓ **Recursos**

El honeypot no utiliza mucho ancho de banda y tampoco tiene la necesidad de utilizar máquinas de última tecnología, para cumplir su función principal que es el análisis y seguridad a la red, con lo cual capturan lo que viene hacia ellos. (Vinklud A. , Honeypots-Recursos)

- ✓ **Reutilización**

La mayoría de los sistemas de seguridad necesita que se actualicen diariamente los mecanismos de detección y defensa para el mantenimiento de su efectividad, en cuanto al honeypot mientras pase el tiempo no necesita de ninguna actualización y no pierde su efectividad desde el día que se instaló, son de gran ayuda ya que los intrusos querrán penetrarse a la red y comprometer información de la institución u organización. (Vinklud A. , Honeypots-Reutilización)

- ✓ **Recolección de información**

El Honeypot recoge pequeñas cantidades de información pero de alto valor cuando el atacante interactúa con ellos. El honeypot captura la actividad de cualquier mal y su respectiva interacción, de una actividad no autorizada o maliciosa. (Comput-rs)

- ✓ **Simplicidad**

El honeypot "sistema trampa" posee un punto muy importante que es la sencillez, por lo cual no utiliza algoritmos complicados de análisis, ni técnicas convencionales para el registro de las actividades de los intrusos, simplemente hay que instalar y realizar la prueba en la red. (Vinklud A. , Honeypots-Simplicidad)

- ✓ **Universalidad**

Este sistema también se puede utilizar para los atacantes internos como externos. Además se debe evitar poner los nombres a las máquinas como es "honeypot". El objetivo principal de este sistema trampa es pasar desapercibido.

2.1.2 DESVENTAJAS

- ✓ **Vistas limitadas**

Sólo puede realizar el seguimiento y captura de actividad que interactúa directamente con ellos. Honeypot no captará los ataques contra otros sistemas, a menos que el atacante o la amenaza interactúen con los honeypots también. (Compute-rs, Honeypots-Vistas_Limitadas)

- ✓ **Riesgo**

El sistema trampa “honeypot” es una fuente de mucho riesgo, en el cual se puede presentar que el intruso se apodere de la red y genere daños a otros sistemas. (Vinklur) - (Compute-rs, Honeypots-Riesgo)

- ✓ **Perspectiva ilimitada**

Los Honeypots pierden su valor si no reciben ataques. Si un atacante logra identificar uno de estos sistemas, puede anular toda su funcionalidad y efectividad al evitarlos, concentrando sus fuerzas en realizar un ataque en otro lugar de la misma red. (Vinklud A.)

2.1.3 CLASIFICACIÓN DE LOS HONEYPOT

De acuerdo a la clasificación existen dos tipos:

- ✓ Según el ambiente de implementación.
- ✓ Según su nivel de interacción.

2.1.3.1 SEGÚN EL AMBIENTE DE IMPLEMENTACIÓN

Bajo esta categoría podemos definir dos tipos de Honeypots:

- ✓ Para la producción
- ✓ Para la investigación.

2.1.3.1.1 HONEYPOTS PARA LA PRODUCCIÓN

Son aquellos honeypots que se instalan en las empresas para desviar la atención de máquinas mucho más importantes. Están dentro de la propia red de la empresa. Se instalan servicios vulnerables (o aparentemente vulnerables), suelen tener muchísimos puertos abiertos, datos falsos, etc. Así los hacen más atractivos y más deseables para un posible intruso. (Segovia)- (UTPL, Honeypot de Investigación)

2.1.3.1.2 HONEYPOTS PARA LA INVESTIGACIÓN

Estos Honeypots no son implementados con la finalidad de proteger redes, sino que constituyen recursos educativos de naturaleza demostrativa y de investigación cuyo objetivo se centra en estudiar patrones de ataque y amenazas de todo tipo. Gran parte de la atención actual se centra en los Honeypots para la investigación, que se utilizan para recolectar información sobre las acciones de los intrusos. (Jara, Gaete, & Villalón, Honeypot_Investigación)- (Verdejo Alvarez)

2.1.3.2 SEGÚN SU NIVEL DE INTERACCIÓN

2.1.3.2.1 HONEYPOTS DE BAJA INTERACCIÓN

Normalmente, éstos honeypots trabajan únicamente emulando servicios y sistemas operativos. La actividad del atacante se encuentra limitada al nivel de emulación del honeypot. La ventaja de un honeypot de baja interacción

radica principalmente en su simplicidad, ya que estos tienden a ser fáciles de utilizar y mantener con un riesgo mínimo. Ejemplos Specter[1], Honeyd, y KFSensor[2]. (Segovia, Honeynets II: Honeypots) - (Cano Nuñez)

- ✓ **Honeyd:** Es un demonio que crea hosts virtuales en una red. Los anfitriones pueden ser configurados para ejecutar servicios arbitrarios, y su comportamiento puede ser adaptado para que simule estar en ejecución en ciertos sistemas operativos. (Xombra)
- ✓ **HoneyC:** El objetivo es identificar servidores Web maliciosos en la red. Para ello emula varios clientes y recaba la mayor cantidad posible de información de las respuestas de los servidores cuando estos contestan a sus solicitudes de conexión. HoneyC es ampliable de diversas formas: pueden utilizarse diferentes clientes, sistemas de búsqueda y algoritmos de análisis. (Proyect.Honeynet)
- ✓ **Nephentes:** Es un honeypot de baja interacción que pretende emular vulnerabilidades conocidas para recopilar información sobre posibles ataques. Nephentes está diseñado para emular vulnerabilidades que los gusanos utilizan para propagarse y cuando estos intentan aprovecharlas, captura su código para su posterior análisis. (Hack, Honeypots-Nephentes)
- ✓ **Honeytrap:** Este honeypot está destinado a la observación de ataques contra servicios de red. En contraste con otros honeypots, que se suelen centrar en la recogida de malware, el objetivo de Honeytrap es la captura de exploits. (Segovia, Honeynets II:Honeypots-Honeytrap)
- ✓ **Glastopf:** Emula miles de vulnerabilidades para recopilar datos de los ataques contra aplicaciones Web. La base para la recolección de

[1] Es un Honeypot o un señuelo que simula una máquina completa, proporcionando un interesante objetivo para los hackers, lejos de las máquinas de producción.
[2] Es un sistema de detección de intrusos basado en host, que actua como un señuelo para atraer potencial.

información es la respuesta correcta que se le ofrece al atacante cuando intenta explotar la aplicación Web, su configuración es fácil y una vez indexado por los buscadores, los intentos de explotación de sus vulnerabilidades se multiplican. (Segovia A.)

2.1.3.2.2 HONEYPOTS DE MEDIA INTERACCIÓN

El honeypot de media interacción, brinda mayor interacción pero sin llegar a proveer un sistema operativo sobre el cual interactuar. El atacante obtiene una mejor ilusión de un sistema operativo real y mayores posibilidades de interactuar y permite escanear el sistema. Además el desarrollo e implementación es más complejo y consume más tiempo. (Jara, Gaete, & Villalón, Honeypots de Media Interacíon)

2.1.3.2.3 HONEYPOTS DE ALTA INTERACCIÓN

Este tipo de honeypots constituyen una solución compleja, ya que implica la utilización de sistemas operativos y aplicaciones reales montados en hardware real sin la utilización de software de emulación e involucrando aplicaciones reales que se ejecutan de manera normal, muchas veces en directa relación a servicios como bases de datos y directorios de archivos compartidos. Ejemplos:

- ✓ **HI-HAT (High Interaction Honeypot Analysis Toolkit):** Es una herramienta que transforma aplicaciones php en aplicaciones honeypot de alta interacción. Además ofrece una interfaz web que permite

consultar y monitorizar los datos registrados. (HI-HAT) - (Segovia A. , Honeynet II:Honeypot HI-HAT)

- ✓ **HoneyBow:** Esta herramienta de recopilación de malware que puede integrarse con el honeypot de baja interacción Nephentes para crear una herramienta de recolección mucho más completa. (Segovia A. , Honeynet II:Honeypot_HoneyBow)
- ✓ **Sebek:** Funciona como un HIDS (Host-based Intrusion Detection System) permitiendo capturar una gran variedad de información sobre la actividad en un sistema ya que actúa a muy bajo nivel. Es una arquitectura cliente-servidor, con capacidad multiplataforma, que permite desplegar honeypots cliente en sistemas Windows, Linux, Solaris, *BSD, etc., que se encargan de la captura y el envío de la actividad recopilada hacia el servidor Sebek, se podría decir que forma parte de una tercera generación de honeypots. (Segovia A. , Honeynet II:Honeypot_Sebek)
- ✓ **Capture-HPC:** Es de tipo cliente, como HoneyC, identifica servidores potencialmente maliciosos interactuando con ellos, utilizando una máquina virtual dedicada y observando cambios de sistema no previstos o autorizados. (Segovia A. , Honeynet II:Honeypot_Capture-HPC)

2.1.4 UBICACIÓN DE HONEYPOTS

Las honeypots se pueden ubicar en distintas partes de la red:

2.1.4.1 HONEYPOTS ANTES DEL FIREWALL

Es la ubicación con menos riegos a la red, ya que está fuera de la zona protegida por el firewall, puede ser atacado sin ningún tipo de peligro para el resto de la red. (Cócaro, García, Jose, & Rouiller) - (UTPL, Honeypot) - (Jara, Gaete, & Villalón, Antes del firewall (Front of firewall))

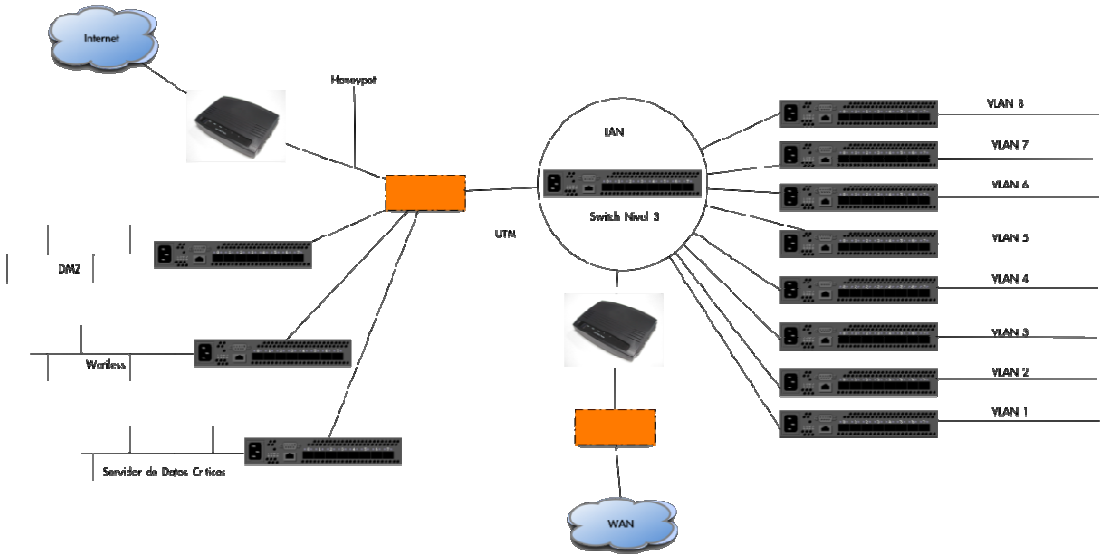


Ilustración 2-1 Ubicación de una Honeypot antes del firewall

Fuente: http://www.utpl.edu.ec/honeynet/?p=159

2.1.4.2 HONEYPOT DESPUÉS DEL FIREWALL

Esta ubicación en el acceso al Honeypot está dirigido por las reglas de filtrado del firewall, su ubicación permite la detección de los atacantes internos. (Cócaro, García, Jose, & Rouiller) - (UTPL, Honeypot) - (Jara, Gaete, & Villalón, Detrás del firewall (Behind the firewall))

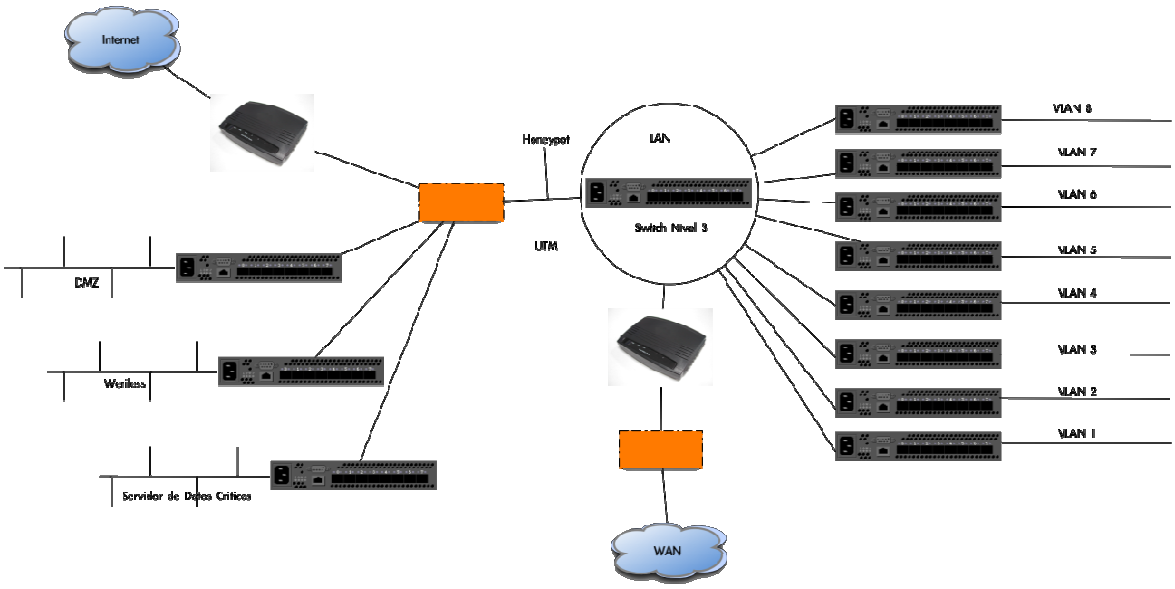


Ilustración 2-2 Ubicación de una Honeypot después del Firewall

Fuente: http://www.utpl.edu.ec/honeynet/?p=159

2.1.4.3 HONEYPOT EN LA ZONA DESMILITARIZADA

La ubicación en esta zona es la mejor porque detecta ataques tantos internos como externos, para esto se requiere una reconfiguración del firewall. (Cócaro, García, Jose, & Rouiller) - (UTPL, Honeypot) - (Jara, Gaeta, & Villalón, En la zona desmilitarizada (into DMZ))

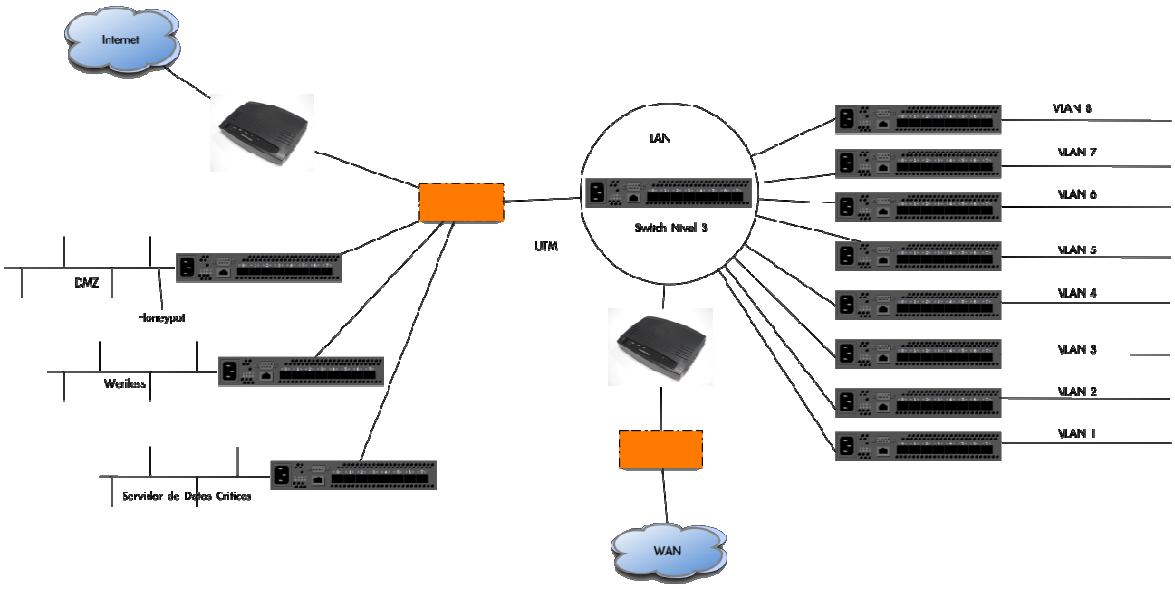


Ilustración 2-3 Ubicación de una Honeypot en la zona desmilitarizada

Fuente: http://www.utpl.edu.ec/honeynet/?p=159

2.1.5 HERRAMIENTAS DE HONEYPOTS

2.1.5.1 COMERCIALES

- ✓ **Back Officier Friendly:** Permite detectar cuando alguien intenta un escaneo Back Orifice contra una computadora.
- ✓ **KFSensor:** Sistema de detección de intrusos honeypot para plataformas Windows.
- ✓ **NetFacade:** Crea una Honeynet que permite dar una alerta al personal de red cuando alguien quiere ingresar a la red.
- ✓ **Symantec Decoy Server (formerly ManTrap):** Es una herramienta que alerta sobre ataques internos y externos de cualquier origen, desautoriza el uso de contraseñas y accesos a servidores para ayudar a priorizar procesos, etc.

2.1.5.2 LIBRES

- ✓ **Sebek:** Es una herramienta que captura los datos, diseñada para capturar actividad maliciosa y ataques en honeypots.
- ✓ **Honeybee:** Herramienta para la creación de manera semiautomatizada de emuladores de servidores de aplicaciones de red.
- ✓ **Brcontrol:** Conjunto de parches que permiten la interacción de IDS[3] y cortafuegos (snort y linux netfilter) y que ayudan a la creación de honeypot agresivos y otras avanzadas configuraciones de cortafuegos.
- ✓ **FakeAP:** Black Alchemy's Fake AP genera millares de falsos puntos de acceso 802.11b.
- ✓ **GHH – The "Google Hack" Honeypot:** GHH es un motor de búsqueda hacker. Está diseñado para proveer reconocimientos contra hackers que usan herramientas de Hawking contra los recursos de la organización.
- ✓ **HoneyBot:** Es una solución de honeypot de interacción media para plataformas Windows.
- ✓ **Honeyd:** Es un demonio que crea sistemas virtuales en una red. Existe una versión para Windows.
- ✓ **HoneyMole:** Su objetivo es actuar como Bridge Ethernet sobre TCP/IP completamente transparente, tunelizando de forma fácil y segura el tráfico de red a un sitio remoto sin la necesidad de utilizar parches del kernel o módulos, e incluso sin la necesidad de ocultar el enrutamiento en los honeypots.
- ✓ **Honeynet Security Console para Windows 2000 /XP:** Es una herramienta para el análisis de los eventos producidos en su red

[3] **IDS:** Sistema de detección de intrusos "Intrusion Detection System"

personal o honeynet, además permite ver eventos procedentes de Snort, TCPDump, Firewall, Syslog y Sebek logs.

- ✓ **HoneyPerl:** Honeypot software basado en Perl[4] con multitud de plugins como fakehttp, fakesmtp, fakesquid, faketelnet,etc.
- ✓ **HoneyWeb:** Es un servidor web que puede ser utilizado como servidor autónomo, o en enlace con HoneyD para proveer peticiones basadas en spoofing de cabeceras http y servicio de páginas.
- ✓ **Impost:** Es una herramienta de auditoría de seguridad de redes diseñada para analizar los datos forenses que hay detrás de demonios comprometidos y/o vulnerables.
- ✓ **Kojoney:** Es un honeypot de baja interacción que emula un servidor SSH.
- ✓ **LaBrea Tarpit:** Es un programa que crea un "Stick honeypot".
- ✓ **OpenBSD's spamd:** Falso demonio sendmail que rechaza correos falsos.
- ✓ **ProxyPot:** Es un servidor que pretende ser un servidor Proxy abierto, aceptando peticiones maliciosas y respondiendo con una respuesta simulada.
- ✓ **Single-Honeypot:** Es un singular y pequeño honeypot para probar sus redes ante visitantes hostiles.
- ✓ **SMTPot.py:** Autónomo honeypot SMTP escrito en Python.
- ✓ **Spamhole:** Falso sustituto de SMTP, que intenta paralizar algunos spams de spammers convincentes.
- ✓ **Sampot.py:** Servidor SMTP que actúa como un honeypot orientado a Spam.

[4] **Perl:** Es un lenguaje de programación, tiene las caracteriscas del Lenguaje C, del lenguaje interpretado Shell (sh) AWK, sed, Lisp, y un grado inferir de muchos otros lenguajes de programación.

- ✓ **Specter:** Es un honeypot que simula una máquina completa, proporcionando un interesante objetivo para los hackers, lejos de las máquinas de producción.
- ✓ **SwiSH:** Es un básico honeypot SMTP multiprocesos diseñado para correr en sistemas Windows.
- ✓ **Tiny Honeypot (thp):** Escucha en todos los puertos, proporcionando falsas respuestas a los atacantes.
- ✓ **The Deception Toolkit:** Es una herramienta diseñada para dar a defensores ventaja frente a atacantes. (García, Jess)

2.2 HONEYNET

La honeynet es un tipo de honeypot pero con una alta complejidad de interacción, la honeynet permite recopilar mayor cantidad de información, además de ser una red completa contiene un conjunto de sistemas para ser atacados. (Gallego & Lopez de Vergara) - (honeynet.org)

La honeynet está compuesta por un conjunto de dispositivos como son los routers, switchs, con esto permite replicar a la red de cualquier organización, además contiene sistemas reales con servicios y configuraciones habituales, hace que los riesgos y las vulnerabilidades que permiten descubrir sean exactamente las mismas que se pueden encontrar en cualquier organización que cuente con los sistemas similares a los expuestos. (honeynet.org, Conoce a tu enemigo)

El objetivo de la honeynet es estudiar las técnicas, tácticas y motivos de los atacantes y compartir las lecciones aprendidas, los dispositivos con los que

cuenta la honeynet permiten detectar, filtrar y registrar tanto el tráfico que entra como el que sale de la red, todo esto se lo realiza en forma pasiva, para que el intruso no note ningún comportamiento extraño que le induzca a pensar que está siendo vigilando.

A continuación se muestran las funciones de los elementos de una honeynet:

Control de intruso

La honeynet es comprometida cuando el intruso ataca, y será necesario tener la garantía de que no pueda ser utilizado para atacar a otros sistemas que no pertenezcan a la red, con el fin de controlar cada una de las conexiones que el atacante trate de ingresar al sistema, si filtraría esto sería dañino para la organización.

Captura de datos

El éxito de la honeynet es la capacidad de capturar mayor información ya que estos datos permitirán realizar el estudio y dar a conocer cada una de las tácticas que utiliza el atacante, es fundamental capturar todo el tráfico que entre y salga de la honeynet, así mismo como cualquier actividad que realice el intruso.

Centralización de información

Cuando se tienen varias honeynet dispersas por el internet, es preferible que la información obtenida se lo envié a un servidor centralizado para su almacenamiento y análisis, con esto se puede tener mayor control sobre los datos recogidos para tener una visión más clara sobre los diferentes ataques

que se presentan en la red. (Wikipedia, Honeynet) - (Gallego & Lopez de Vergara)

2.2.1 ARQUITECTURAS

Un honeynet no es un modelo de arquitectura cerrado, existe libertad para su desarrollo tanto para la topología y las herramientas que se utilizan para realizar tareas de control, registro y análisis de acciones del intruso en el interior.

Existen dos tipos de arquitecturas:

- ✓ Arquitectura de Primera Generación (GEN I)
- ✓ Arquitectura de Segunda Generación (GEN II)

2.2.1.1 ARQUITECTURA DE PRIMERA GENERACIÓN (GEN I)

Consta de una red de sistemas señuelos, dispuestos a ser atacados o honeypots, un cortafuegos, un router, un detector de intrusiones basado en red o NIDS y un servidor centralizado de logs y alarmas.

Además de controlar los intrusos conjuntamente con el router y un cortafuegos.

El cortafuegos es un filtro de paquetes a nivel de red que constituye la unión entre el interior del honeynet e internet y a la vez divide a la honeynet en dos segmentos de red:

Honeypots y los Administrativos que contiene el servidor remoto de log y el sistema de detección de intrusiones, está configurado para permitir cualquier conexión desde el honeynet a la red del honeypot, además protege los equipos de la subred administrativa y controla las conexiones que se traten de establecer desde los honeypot hacia el exterior.

Para el control de intrusos hay un contador de intentos de conexión desde cualquier honeypot hacia cualquier equipo del exterior, a partir de que supere el número de intentos, se bloqueará cualquier nuevo intento que se realice.

El inconveniente que tienen los honeynet de la Primera Generación son las limitaciones en el control de atacantes. Si se le permite ciertos intentos para realizar la conexión, uno de los casos terribles sería que el atacante tenga éxito en la conexión. (Gallego & Lopez de Vergara)

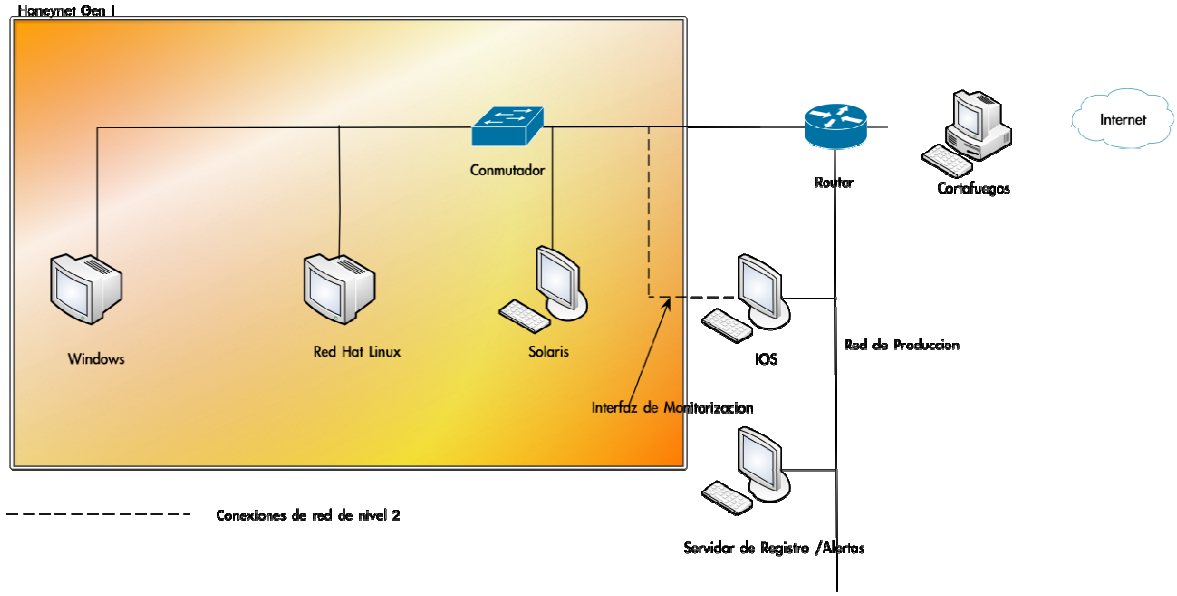


Ilustración 2-4 Esquema de Honeynet de Primera Generación

Fuente: http://jungla.dit.upm.es/~jlopez/publicaciones/mundointernet04.pdf

2.2.1.2 ARQUITECTURA DE SEGUNDA GENERACIÓN (GEN II)

La arquitectura de la segunda generación llamada Honeynet Project tiene como objetivo un entorno más difícil de identificar por parte de los atacantes, controlando sus acciones en forma más estrecha con el sistema comprometido.

Esta arquitectura es más sencilla, realizando las tareas de control y recolección de datos en un único sistema que el Honeynet Project denomina honeywall, con esto se simplificarán también los procesos de desarrollo y administración de la honeynet.

El honeywall dispone de tres interfaces de red, conectado a un router para la administración remota, con las dos interfaces del sistema este se comportará como un puente (bridge) ya que carecen de direcciones IP y MAC, con esto el sistema no hará encaminamiento de tráfico ni disminuirá el tiempo de vida de los paquetes que lo atraviesan.

Con el honeywall el honeynet se puede integrar a la red, además de compartir una VLAN (Red de Área Local Virtual), esto permitirá el estudio de las amenazas internas como externas de una organización.

Para realizar el control de acciones de un intruso el honeywall a la acción de un cortafuego se va a unir a un sistema de prevención de intrusos NIPS, esta herramienta analiza el tráfico en tiempo real, al mínimo intento de ataque este tiene la capacidad de impedir que el atacante tenga éxito, siempre y cuando el ataque esté registrado en la configuración. Mientras tanto que con el honeywall se trabajará para aquellos ataques que no puedan ser descubiertos por el sistema de prevención de intrusos.

A diferencia del procedimiento de control de las honeynet de primera generación este modelo incrementará las conexiones permitidas, en la segunda generación el Honeynet Project permite un máximo de 15 conexiones TCP, 20 UDP, 50 ICMP[5]. La captura de datos se lo realizará de igual manera que la primera generación, con la diferencia que la información se recopilará en forma centralizada desde honeywall. (Gallego & Lopez de Vergara)

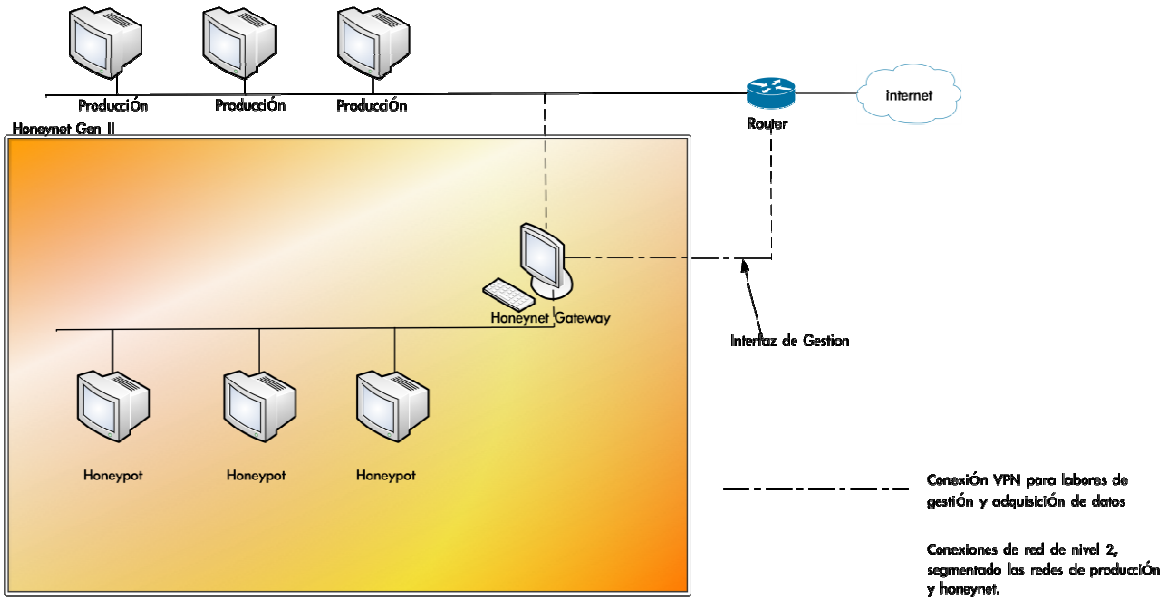


Ilustración 2-5 Esquema de una Honeynet de Segunda Generación

Fuente: http://jungla.dit.upm.es/~jlopez/publicaciones/mundointernet04.pdf

2.2.2 HONEYNET VIRTUALES

Una honeynet virtual permite que varios sistemas operativos se ejecuten en una sola máquina física, comparte recursos del sistema anfitrión e incluso aparentan estar en máquinas distintas e independientes. Estas redes virtuales tienen la misma arquitectura de la primera y segunda generación, con la cual

[5] **Internet Control Message Protocol:** Es el sub protocolo de control y notificación de errores del Protocolo de Internet, se utiliza para enviar mensajes de error indicando un servicio no disponible, router, host no puede ser localizado.

ofrecen ventajas e incluso sus respectivas limitaciones para el diseño de las honeynets. (Gallego & Lopez de Vergara, Honeynet: Aprendiendo del atacante "Honeynet Virtual") - (Jara, Gaeta, & Villalón, Honeypot: Honeynets Virtuales)

La herramienta de virtualización ofrece ciertas características comunes, con la utilización de este software para el desarrollo de las honeynets, se tiene las siguientes ventajas y desventajas:

Ventajas:

- ✓ Coste reducido y más fácil manejo ya que está combinado en un único sistema.
- ✓ Administra en forma centralizada el sistema de honeynet desde el equipo anfitrión.
- ✓ La Honeynet se convierte en una solución “plug and play” ya que se ejecuta todo en un solo equipo.
- ✓ Los discos duros de los diferentes sistemas instalados pueden ser virtuales, en este caso archivos en el sistema anfitrión, cada vez que se instala se crea una copia de seguridad, con esto podemos reemplazar los ficheros del sistema atacado.

Desventajas:

- ✓ Con la utilización de la virtualización se limita los sistemas operativos con la cual se puede desarrollar la honeynet debido al hardware y al programa virtual.
- ✓ Con la utilización de máquinas virtuales se pueden delatar este tipo de software del hecho que sea un sistema virtual, esto no implica que se

trate de una honeypot, así el intruso se daría cuenta y perdería su interés en la máquina.

- ✓ Las Honeynets virtuales traen un riesgo, específicamente que un atacante puede salirse del programa virtual y tomar el sistema Honeynet, saltándose el mecanismo de Control de datos y de Captura de datos

2.2.2.1 TIPOS DE HONEYNET VIRTUALES

Una honeynet virtual puede ser Autocontenida o Híbrida.

2.2.2.1.1 HONEYNET VIRTUAL AUTOCONTENIDA

La honeynet virtual autocontenida comprende a una honeynet en un solo equipo, la red entera está virtualmente contenida en un solo equipo físico, además la Honeynet consiste de un cortafuegos para el control de datos y captura de datos, y los honeypots dentro de la Honeynet. (honeynet.org, Conoce a tu enemigo- Honeynet Virtual Auto-contenida) - (Jara, Gaeta, & Villalón, Honeypots-Honeynets Virtuales Auto-contenidas)

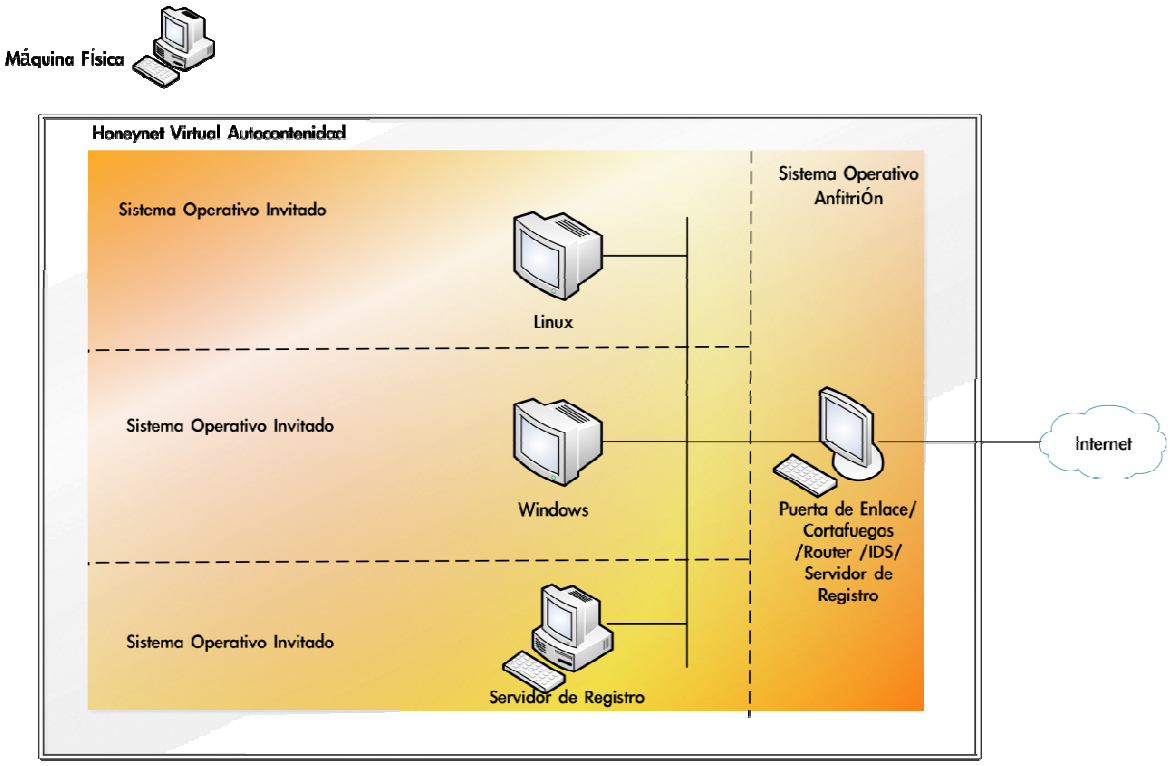


Ilustración 2-6 Esquema de una Honeynet Virtual Autocontenida

Fuente: http://dgonzalez.net/papers/ids/html/cap04.htm

Ventajas

- ✓ Facilidad de transportación, se puede instalar hasta en una laptop.
- ✓ Funcionamiento rápido. Una vez instalada, sólo hay que conectarla a la red y configurarla en pocos minutos.
- ✓ Ocupa poco espacio y el costo es reducido.

Desventajas

- ✓ Si falla el hardware, la Honeynet podría dejar de funcionar.
- ✓ El equipo debe tener suficiente memoria y capacidad de procesador.

- ✓ Inseguridad ya que como todos los sistemas comparten el mismo hardware, puede que un atacante acceda a otras partes del sistema. Tiene mucha dependencia del software virtual.
- ✓ Limitación por software. Como todo tiene que ejecutarse en una sola máquina, hay software que no se podrá utilizar por problemas de incompatibilidad.

2.2.2.1.2 HONEYNET VIRTUAL HÍBRIDA

Una Honeynet virtual híbrida es una combinación de la clásica Honeynet y del software virtual. Captura de Datos, como por ejemplo cortafuegos, y Control de datos, es decir, los sensores de IDS y el almacenamiento de registros, están en un sistema separado y aislado, para reducir el riesgo de compromiso. Sin embargo, todos los honeypots son ejecutados en una única máquina. (Jara, Gaeta, & Villalón, Honeypots: Honeynets Virtual Híbrida) - (honeynet.org, Conoce a tu enemigo: Honeynet Virtuales Híbridas)

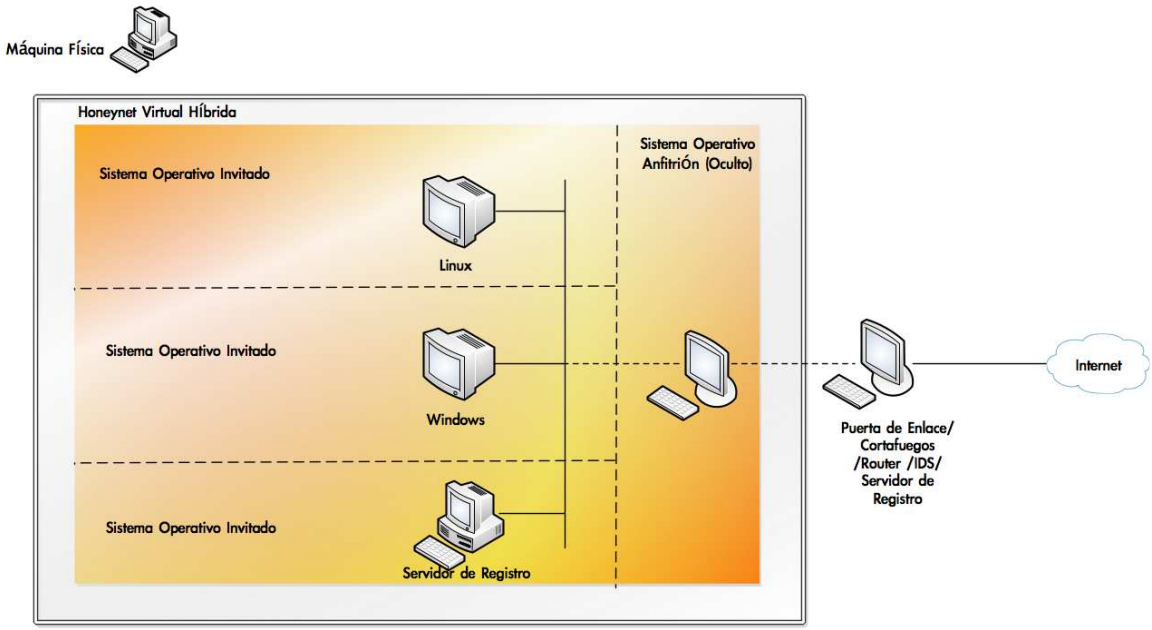


Ilustración 2-7 Esquema de Honeynet Virtual Híbrida

Fuente: http://dgonzalez.net/papers/ids/html/cap04.htm

Ventajas

- ✓ Seguridad. El único peligro sería que el atacante accediera a otro Honeypot.
- ✓ Hay mayor flexibilidad a la hora de utilizar software para el control y captura de datos de la red.

Desventajas

- ✓ Al implicar a más de una máquina, la movilidad es más reducida.
- ✓ Es más cara y ocupa más espacio que la Autocontenida.

2.3 TECNOLOGÍAS PARA IMPLEMENTAR HONEYNET VIRTUALES

Algunas tecnologías para implementar una honeynet virtual son: User Mode Linux, VMware Workstation y GSX Server, o Microsoft Virtual PC.

2.3.1 USER MODE LINUX

User Mode Linux permite trabajar en su propia interfaz probando versiones inestables del núcleo sobre un sistema en funcionamiento. Por ejemplo un núcleo que está a prueba es solo un proceso de usuario, si se cuelga no compromete al sistema que lo aloja. (Wikipedia, User Mode Linux)

Ventajas

- ✓ Creación de honeypots (sistema para probar la seguridad de una máquina sin comprometerla).
- ✓ Ejecución de servicios de red, al ejecutar servicios de red en diferentes procesos UML de una misma máquina se los aíslan unos de otros, de forma que no pueden comprometer mutuamente la estabilidad o seguridad de los demás.
- ✓ Realizar pruebas con software inestable o incompatible con la versión del núcleo del sistema que aloja el UML (las versiones del núcleo de ambos sistemas pueden ser distintas).
- ✓ Permite tener acceso (aparentemente) de administrador a una máquina (principalmente interesante para servidores virtuales o para entornos educativos en los que se limita las capacidades de un root).

- ✓ Es una herramienta de código abierto con lo que se puede corregir, revisar y adaptar su código a las necesidades del honeypot.
- ✓ Es un software de libre distribución, se puede utilizar sin pagar licencias.
- ✓ Permite capturar las sesiones de los intrusos en forma pasiva a través del kernel del sistema anfitrión.
- ✓ Inicialmente se desarrolló para la arquitectura x86 aunque hoy en día está disponible en otras como ia64 y PowerPC. (Gallego & Lopez de Vergara, Honeypots: Aprendiendo del Atacante -Ventajas de UML)

Desventajas

- ✓ Solo permite el funcionamiento de máquinas Linux.
- ✓ UML no manipula interfaz gráfica y su utilización no resulta demasiado intuitiva, tampoco existe una documentación clara detallada sobre su modo de empleo y en un principio no es sencilla de manejar.
- ✓ Como herramienta de código abierto, carece de soporte técnico. (Gallego & Lopez de Vergara, Honeypots: Aprendiendo del Atacante-Desventajas UML)

2.3.2 VMWARE WORKSTATION

Es una herramienta comercial que permite a los administradores y desarrolladores implementar herramientas de software más complejas de tipo servidor en red que se ejecuten en Microsoft, Linux o Netware todo esto desde un solo computador, sus características esenciales son:

- ✓ Funcionamiento de red Virtual.
- ✓ Copias puntuales activas.
- ✓ Arrastrar y soltar carpetas, archivos, etc.
- ✓ Carpetas compartidas.
- ✓ Soporte PXE[6] convierte a VMWare Workstation, herramienta indispensable para los desarrolladores y administradores de sistemas TI empresariales.
- ✓ Reduce los costos, aumenta la flexibilidad y la capacidad de respuesta. (Alegsa.com.ar) - (Granados)

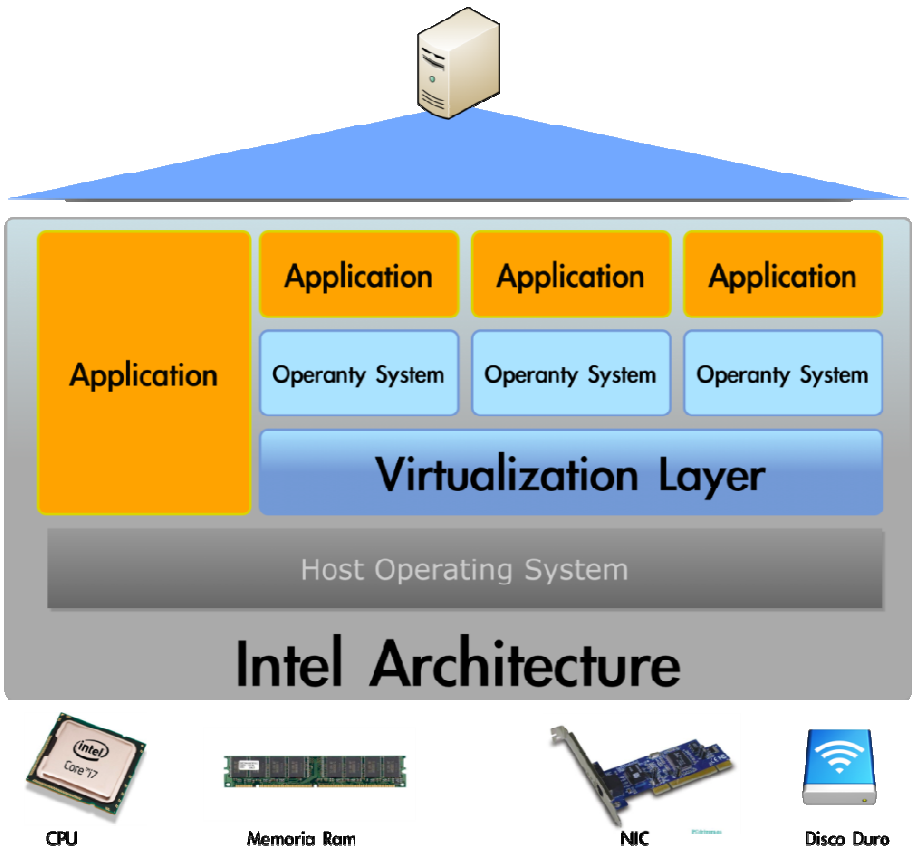


Ilustración 2-8 Arquitectura de VMWARE WORKSTATION

Fuente: http://www.avansis.es/vmware/productos-workstation.htm

[6] **PXE** "Preboot Excution Enviroment": Entorno de ejecución de prearraque, es un entorno para arrancar e instalar el sistema operativo en ordenadores a través de la red, de manera independiente de los dispositivos de almacenamiento de datos disponibles, discos duros o de los sistemas operativos instalados.

2.3.2.1 CARACTERISTICAS DE VMWARE WORKSTATION

OPTIMIZA EL DESARROLLO Y LAS PRUEBAS DE SOFTWARE

Modo de Uso

- ✓ Crear múltiples entornos de desarrollo y prueba en un único sistema.
- ✓ Crear aplicaciones de misión crítica basadas en Windows y/o Linux.
- ✓ Archivar entornos de prueba en File Servers (servidores de archivos) y resultados rápidamente, según sea lo necesario.
- ✓ Probar nuevas actualizaciones de aplicaciones, correcciones y service packs de sistemas operativos en un solo computador.

Beneficios

- ✓ Aceleración de los ciclos de desarrollo y disminución del tiempo de salida al mercado.
- ✓ Disminución de los costos de hardware.
- ✓ Disminución del costoso tiempo de configuración.
- ✓ Mejora de la calidad de los proyectos mediante pruebas más rigurosas.
- ✓ Eliminación de los costosos problemas de implementación y mantenimiento.

ACELERA EL DESARROLLO DE LAS APLICACIONES

Modo de uso

- ✓ Probar , configurar y realizar el provisionamiento de servidores de clase empresarial como máquinas virtuales de VMWare Workstation y luego

implementarlos en un servidor físico o servidor VMWare GSX o VMWare ESX.

- ✓ Crear una completa red de aplicaciones compuesta de múltiples computadores y switches de red en un conjunto de máquinas virtuales y probarlas sin afectar la red de producción.
- ✓ Probar migraciones de entornos físicos a virtuales para la consolidación de servidores y migraciones de aplicaciones antiguas.

Beneficios

- ✓ Disminución de los costos en hardware.
- ✓ Mejora de la calidad de las implementaciones.
- ✓ Mejora en la productividad.
- ✓ Disminución del riesgo para las redes corporativas al crear redes virtuales complejas, seguras y aisladas que reflejan las redes de las empresas.

GARANTIZA LA COMPATIBILIDAD DE LA APLICACIÓN Y REALIZA MIGRACIONES DE SISTEMAS OPERATIVOS

Modo de Uso

- ✓ Soporta aplicaciones antiguas mientras se realiza la migración a un nuevo sistema operativo.
- ✓ Permite probar nuevos sistemas operativos en las máquinas virtuales seguras y válidas antes de la implementación.
- ✓ Elimina la necesidad de modificar las aplicaciones antiguas para ejecutarlas en otras plataformas.

Beneficios

- ✓ Realización de proyectos complejos sin excederse en el plazo y en el presupuesto.
- ✓ Aumento de las eficiencias en un 50%.
- ✓ Disminución de los costos de capital de computadores en un 50%.
- ✓ Minimización de los problemas de usuario.

2.3.3 GSX SERVER

Es una herramienta de software empresarial para servidores x86, permite el fortalecimiento de los servidores, la recuperación ante desastres y para optimizar los procesos de desarrollo de herramientas de software, GSX Server ofrece la capacidad de administración y escalabilidad incomparables.

Permite que las máquinas virtuales se administren en forma remota, transforma las computadoras físicas en un repositorio de máquinas virtuales, las aplicaciones y los sistemas operativos se aíslan en múltiples máquinas virtuales que residen en un solo hardware (CPU).

Su sólida arquitectura y la capacidad para integrarse con entornos de Microsoft Windows y Linux, hace que sea más sencillo y rápido de implementar y administrar. GSX Server se ejecuta como una aplicación que permite implementar, administrar y controlar en forma remota múltiples servidores que se ejecutan en máquinas virtuales. (Granados, VMWare GSX)

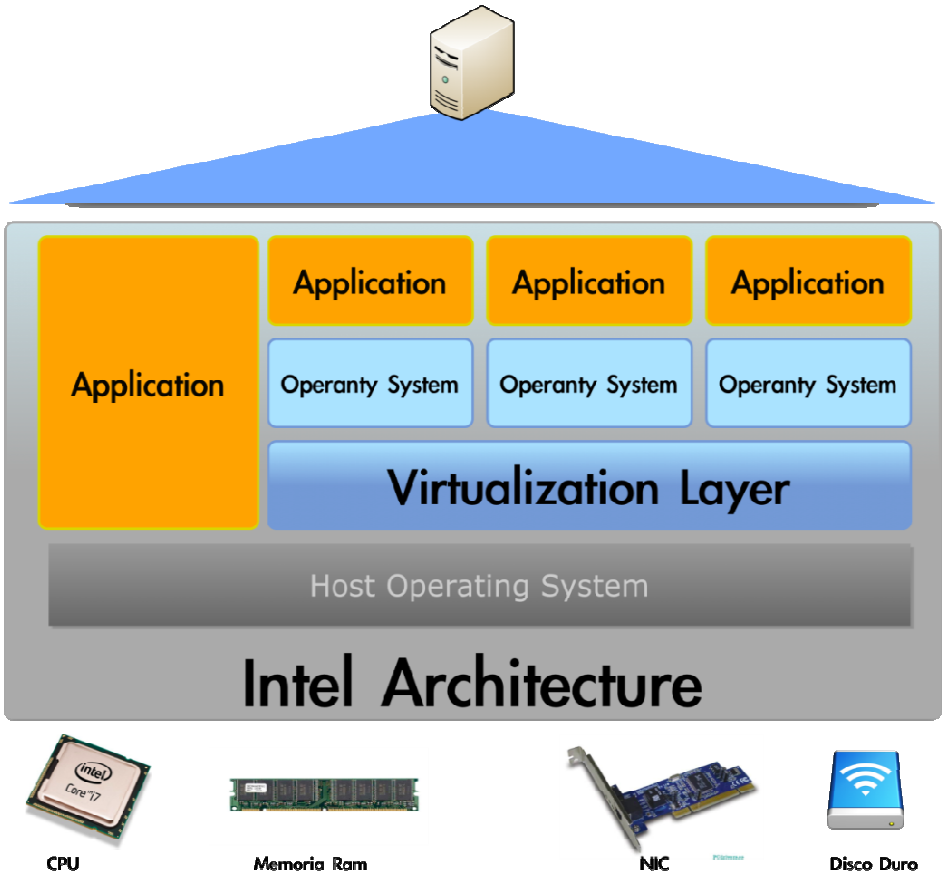


Ilustración 2-9 Arquitectura de GSX SERVER

Fuente: http://www.avansis.es/vmware/productos-workstation.htm

2.3.3.1 CARACTERISTICAS DE GSX SERVER

OPTIMIZA EL DESARROLLO Y LAS PRUEBAS DE SOFTWARE

Modo de Uso

- ✓ Administración de entornos con grandes cantidades de máquinas de desarrollo, prueba y múltiples sistemas operativos en máquinas virtuales basadas en servidores.

Beneficios

- ✓ Hacer provisionamiento de nuevas máquinas para desarrollo y prueba en minutos en lugar de horas o días.
- ✓ Reducir considerablemente los tiempos de ciclos de prueba.
- ✓ Mantener bibliotecas de entorno de máquinas en archivos de disco virtuales encapsulados e independientes del hardware.
- ✓ Integrarse con herramientas líderes de automatización de pruebas, tales como IBM, Rational Test Manager.

IMPLEMENTA LA CONSOLIDACIÓN DE SERVIDORES DE MANERA RENTABLE

Modo de uso

- ✓ Consolidar aplicaciones y servicios de infraestructura en menos servidores de clase empresarial altamente confiables y escalables.

Beneficios

- ✓ Reducir el costo total de propiedad en toda la infraestructura computacional hasta en un 64%.
- ✓ Espacio para expansión y escalabilidad.
- ✓ Maximizar la utilización de hardware
- ✓ Simplificar la administración de los sistemas.
- ✓ Justificar los costos de hardware de servidores de calidad superior.
- ✓ VMWare P2V Assistant convierte rápidamente los servidores físicos en máquinas virtuales.

PROVISIONAMIENTO RÁPIDO DE SERVIDORES

Modo de Uso

- ✓ Los servidores de máquinas virtuales configuradas previamente se pueden crear con rapidez e implementar de inmediato en cualquier lugar, efectuar provisionamiento de un nuevo servidor es muy sencillo.

Beneficios

- ✓ Satisfacer la demanda de los servidores, creaciones y service packs nuevos, al tiempo que se controlan los costos.
- ✓ Failover más rápido con servidores configurados y probados previamente en máquinas virtuales.
- ✓ El soporte de PXE permite que sus actuales herramientas de provisionamiento se puedan utilizar con máquinas virtuales.

2.3.4 MICROSOFT VIRTUAL PC

Ahora en la actualidad llamado Windows Virtual PC, anteriormente llamado Microsoft Virtual PC desarrollado por Connectix y comprado por Microsoft para la creación de equipos virtuales, su función principal es permitir que varios sistemas trabajen en un solo equipo físico y se comuniquen entre ellos. (Wikipedia, Microsoft Virtual PC)

La emulación en el propio hardware, Virtual PC deja que el mismo procesador ejecute instrucciones en el mismo entorno emulado, por lo contrario para MacOS emula un procesador Intel Pentium III de 32Bits, además del procesador emula otras partes del hardware que son:

- ✓ La placa madre con chipset Intel 440BX
- ✓ Tarjeta de video SVGA[7] VESA[8] Estándar S3 Trio 32/64 con
- ✓ 8 MB de memoria VRAM
- ✓ Chip de BIOS de American Megatrends
- ✓ Tarjeta de Sonido SoundBlaster 16
- ✓ Tarjeta de Red DEC 21140

Tiene una desventaja, no soporta todos los programas debido a que pueden existir fallos de sincronización, en las Mac con recopilación dinámica y procesador Intel no existe una versión de Virtual PC, para ello se tiene que utilizar otro tipo de herramientas para la virtualización, mientras que en Windows la recopilación dinámica se lo traduce en modo kernel y en modo real X86 a código de usuario X86, mientras que el usuario original corre en forma nativa.

2.3.4.1 COMPLEMENTOS QUE TIENE VIRTUAL PC

Permite el intercambio de archivos, ficheros, carpetas, etc. entre el anfitrión y el huésped, Virtual PC proporciona una opción llamada Virtual Machine Additions, se lo puede instalar en el sistema operativo huésped para facilitar las siguientes funcionalidades a través del anfitrión y el huésped:

- ✓ Un mejor rendimiento del sistema operativo Huésped.
- ✓ Integración con el teclado y Mouse.
- ✓ Controlador de video optimizado

[7] **SVGA** - Super Video Graphics Array: Es una amplia gama de estándares de visualización de gráfica de computadores, esto incluye tarjetas de video y monitores.
[8] **VESA –** Video Electronics Standards Association: Su objetivo es desarrollar pantallas de video con una resolución 800x600 pixeles con alta velocidad de video.

- ✓ Resolución de la pantalla dinámica.
- ✓ Sincronización de tiempo con el anfitrión
- ✓ Sincronización con el portapapeles.
- ✓ Capacidad de arrastrar archivos entre el Sistema Operativo huésped y anfitrión.
- ✓ Carpetas compartidas.

2.3.4.2 REQUISITOS PARA LA UTILIZACIÓN DE VIRTUAL PC EN EL SISTEMA OPERATIVO

- ✓ Procesador AMD Athlon Dual Core X2 a 1.50Ghz o Intel Celeron
- ✓ 2GB De Memoria RAM
- ✓ Hardware de Virtualización Activado
- ✓ Tarjeta de video con por lo menos 64MB de VRAM[9]
- ✓ Resolución de pantalla 800x600
- ✓ Conexión a Internet de Banda Ancha
- ✓ 48.5MB Libres En el Disco Duro (Se Recomienda 2GB para la instalación de Sistemas Virtuales)
- ✓ No se Soporta Windows Vista Starter, Home Basic y Home Premium. En el caso de Windows 7 es igual que en Windows Vista

Virtual PC 2007 no posee las funcionalidades para los sistemas operativos como son Windows 95, MS-DOS 6.22, mientras que en Virtual PC 2004 si se puede utilizar este tipo de sistemas operativos.

[9] **Video Random Electronics Standards Association:** Es un tipo de memoria RAM que utiliza el controlador gráfico para manejar toda la información visual que envia el CPU del sistema.

Para Windows Vista como huésped en la Virtual PC, el tema gráfico Aero de Windows Vista esta deshabilitada debido a las limitaciones de la tarjeta emulada de los gráficos S3, pero este tema puede ser ejecutado conectado con la máquina huésped por conexión de escritorio remoto, iniciada desde un anfitrión que soporta Aero de Vista.

2.3.4.3 EMULACIÓN EN UN ENTORNO LINUX

En si Windows Virtual PC no soporta Linux como huésped pero otras versiones como Microsoft Virtual Server si soportan Linux, algunos sistemas operativos de Linux se deben instalar en modo texto debido a que Virtual PC emula gráficos de 16 Bits o 32 Bits, no de 24 Bits, para que funcione tenemos que configurar X Windows, esto es para obtener color de 16 Bits especificando el archivo de configuración xorg.conf del sistema huésped, un ejemplo el Ubuntu 8.10 se lo debe instalar SafeMode pero necesita ciertas modificaciones.

3 CONFIGURACION DEL HONEYPOT UTILIZANDO USER MODE LINUX (UML)

3.1 USER MODE LINUX (UML)

3.1.1 DEFINICIÓN

“UML fue creado por Jeff Dike en el año de 1999, es distribuido sin costo, y mejorado cada vez más gracias a las comunidades de software libre en todo el mundo, es muy poco difundido debido al desconocimiento de su utilidad y a la fuerte competencia comercial de otras máquinas virtuales, además porque actualmente está limitada a Linux y no posee interfaz gráfica, aunque ahora se está desarrollando una versión para que trabaje en Windows”.

User-mode Linux (UML) es una modificación del núcleo Linux para que funcione sobre su propia interfaz de llamadas al sistema. De este modo, un núcleo compilado para la arquitectura **um** puede operar como un proceso de usuario más de otro núcleo Linux que hace las veces de anfitrión.

Inicialmente UML se creó para que los desarrolladores del núcleo pudieran probar versiones inestables del núcleo sobre un sistema en funcionamiento. Como el núcleo en prueba es sólo un proceso de usuario, si se cuelga, no compromete al sistema que lo aloja.

Pero además, el uso de UML permite muchas posibilidades:

- Creación de honeypots (sistemas para probar la seguridad de una máquina sin comprometerla).
- Ejecución de servicios de red. Al ejecutar servicios de red en diferentes procesos UML de una misma máquina se los aisla unos de otros, de

forma que no pueden comprometer mutuamente la estabilidad o seguridad de los demás.

- Realizar pruebas con software inestable o incompatible con la versión del núcleo del sistema que aloja el UML (las versiones del núcleo de ambos sistemas pueden ser distintas).
- Permite tener acceso (aparentemente) de administrador a una máquina (principalmente interesante en servidores virtuales o para entornos educativos en los que se limita las capacidades de un root). (Wikipedia, 2010)

El objetivo principal de UML es probar un código kernel nuevo, si este falla o se cuelga tendríamos que volver a reiniciar el sistema, con User Mode Linux este trabajo es innecesario, además es posible realizar Redes virtuales, ofreciendo servicios de hostings y consolidación de servidores de una forma segura y controlada.

El kernel del sistema operativo que corre en el host utilizado es llamado “sistema anfitrión” y cualquier sistema operativo añadido es llamado como “virtual”, como el kernel UML son conocidos como “invitados”.

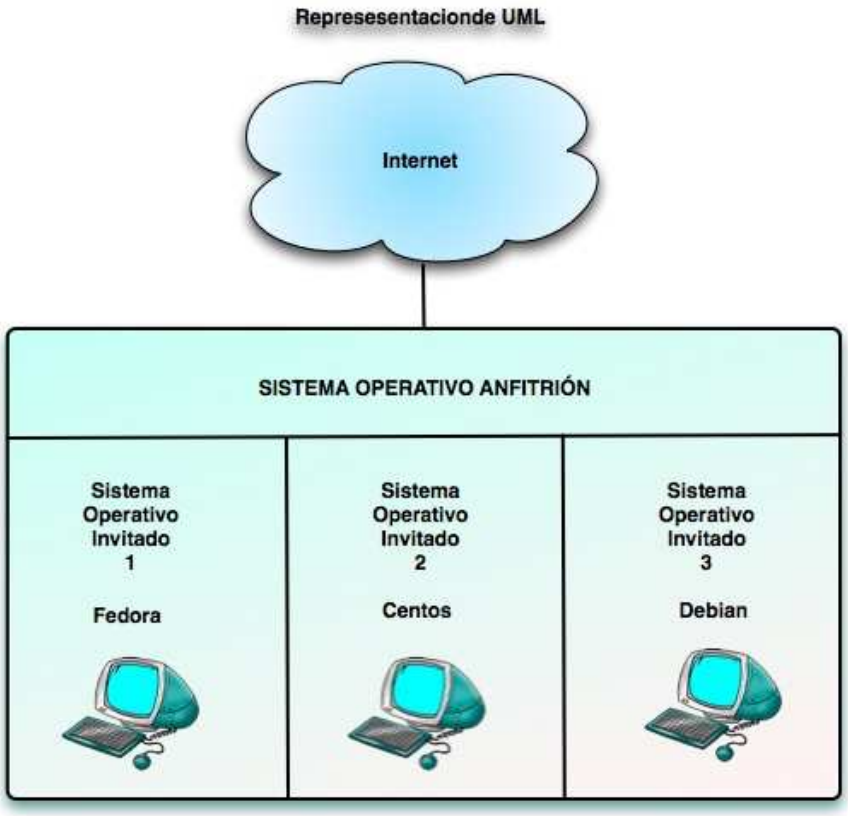


Ilustración 3-1 Diagrama de un Sistema Anfitrión con varios Sistemas Invitados

Fuente: http://bytecoders.net/content/tarros-de-miel-para-los-malos.html

3.1.2 CARACTERÍSTICAS

Algunas de las mejores características que han sido recientemente añadidas a UML están diseñadas específicamente para honeypots. Estas opciones mejoran significativamente nuestra Honeynet UML. Nos centraremos en tres de estas opciones:

- ✓ **TTY Logging:** UML tiene la opción de capturar todas las teclas pulsadas por el atacante, incluso si usa encriptación, como SSH (Secure **SH**ell, en español: Intérprete de Órdenes Segura- es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a

máquinas remotas a través de una red), para comunicarse con el honeypot UML. UML lo hace gracias a un parche al controlador de la TTY (acrónimo que deriva antiguamente de las máquinas de escribir, fue usado para denominar los teletipos. UNIX/Linux fue originalmente un OS filosofía cliente servidor, y todos los recursos fueron originalmente distribuidos, entonces clientes y operadores se conectaban remotamente a través de una consola TTY, para asuntos por medio de escritura), que guarda todo el tráfico a través de dispositivos TTY. En contraste a los mecanismos físicos para guardar datos del honeypot, esto es indetectable. No causa tráfico en la red o cualquier cosa que pueda ser detectada desde el honeypot. Está incluido en el núcleo UML, lo que significa que no puede ser eliminado por nada que haga el intruso.

- ✓ **hppfs:** Una de las preocupaciones con un honeypot virtual es el reconocimiento. Una vez que el atacante ha accedido al sistema operativo virtual, puede ser capaz de determinar que es un honeypot. UML mitiga este riesgo con la opción de modificar el sistema de ficheros /proc para que parezca que es un sistema operativo verdadero.
- ✓ **Modo skas:** UML fue recientemente cambiado para permitir que se ejecute en un modo en el que el núcleo UML está en un espacio de direcciones totalmente diferente de sus procesos. Esto hace al binario del núcleo UML y a los datos totalmente invisibles a sus procesos, y a cualquier persona del sistema. También hace a los datos del núcleo UML seguros de alteración de sus procesos. (Honeynet Project)

3.1.3 VENTAJAS

A continuación enumeramos algunas de las principales ventajas que presenta UML.

- ✓ En el caso que UML se cuelgue, el kernel principal (máquina host) no se verá afectado.
- ✓ UML puede correrse como un usuario cualquiera, lo cual es sumamente recomendable, ya que se evitan posibles modificaciones que se puedan realizar si montamos sistemas de ficheros de la máquina host en la máquina virtual.
- ✓ Se pueden realizar procesos de Debug, profiling, entre otros como si fueran procesos normales (con la ventaja de que si hay cuelgues nuestro sistema sigue funcionando).
- ✓ Se pueden probar nuevas versiones de kernels, quizá hasta tratar de desarrollar módulos para el kernel. Además se puede probar diferentes distribuciones de linux, lo cual para muchos "adictos" a linux les resulta bastante interesante y divertido.

3.1.4 APLICACIONES

***Máquinas de desarrollo o pruebas*:** Sin duda esta es la aplicación por defecto, dado que siempre es mejor probar las cosas en una máquina que no es crítica, como en el caso del uso de máquinas virtuales, se puede recuperar en muy poco tiempo.

***Consolidación de Servidores*:** Se trata de agrupar todos los servidores de una empresa en una sola máquina (que debe tener cierta solvencia de recursos,

evidentemente). La idea se basa en aprovechar mejor los recursos del servidor, ya que es habitual el desaprovechamiento de recursos de hardware. En estos casos, como siempre que se usan máquinas virtuales, la realización de copias de seguridad de cada una de las máquinas resulta muy fácil, puesto que en general supondrá la copia de un solo fichero.

***Hosting*:** Cada vez son más los ISP que ofrecen servidores virtuales usando estas tecnologías.

Honeypots: Máquinas puestas en internet para que los hackers "jueguen" con ellas. Se usan en general para aprender los comportamientos y las nuevas técnicas que usan los intrusos informáticos. (Suárez)

3.2 INSTALACION Y CONFIGURACION DE USER MODE LINUX

Construiremos un kernel modular, es decir que se ajustará a las necesidades de la aplicación, en este caso al sistema honeypot. El kernel User Mode Linux tendremos que descargarlo de la fuente para configurarlo y luego compilarlo.

Antes de obtener el kernel debemos saber que versión posee nuestro sistema operativo, utilizamos el comando: uname –r

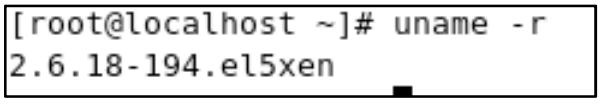


Ilustración 3-2 Sistema Anfitrión “Centos”

Nuestro sistema operativo tiene la versión **2.6.18-194** por lo que lo más apropiado es usar una fuente de kernel **2.6.19-XX** o superior.

Previamente nos descargamos el kernel que utilizaremos durante todo el proceso, es recomendable descargarse el más actual y estable.

En la página http://www.kernel.org/pub/linux/kernel/v2.6 encontraremos una extensa lista de kernels y parches:

Ilustración 3-3 Sistema Anfitrión "Centos"

También podemos descargarnos directamente mediante el comando **wget** pero necesariamente tendremos que saber la versión del kernel a descargarse para poder obtenerlo.

Dependiendo de la fuente obtendremos un archivo ya sea del tipo ***gz** o ***bz2,** estos son archivos comprimidos. Para evitar confusiones se debe descomprimir en el mismo directorio que se va trabajar por lo que es recomendable crear una carpeta, para este caso crearemos una llamada **build**. Debemos utilizar un directorio que no sea **root**, por ejemplo dentro de /home/proyecto, esto permitirá construir dicha carpeta y darle todos los permisos dentro del usuario **proyecto.**

```
[proyecto@localhost ~]$ mkdir build
[proyecto@localhost ~]$ chmod 777 build
[proyecto@localhost ~]$ ls -l
total 8
drwxrwxrwx 2 proyecto proyecto 4096 mar 25 15:55 build
[proyecto@localhost ~]$
```

Ilustración 3-4 Sistema Anfitrión "Centos"

Ingresamos a la carpeta **build** y ejecutamos **wget** para descargarnos el kernel directamente.

```
[proyecto@localhost ~]$ cd build/
[proyecto@localhost build]$ wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.38.tar.bz2
--2011-03-25 15:58:17--  http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.38.tar.bz2
Resolviendo www.kernel.org... 199.6.1.164, 130.239.17.4, 149.20.4.69, ...
Connecting to www.kernel.org|199.6.1.164|:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 74739098 (71M) [application/x-bzip2]
Saving to: `linux-2.6.38.tar.bz2'

29% [===============>                                  ] 22.161.327   169K/s  eta 5m 40s
```

Ilustración 3-5 Sistema Anfitrión "Centos"

Una vez descargados les damos los permisos de lectura y escritura a los archivos que ahora se encuentran dentro de **build** y visualizamos con el

comando **ls**, deben estar el Sistema de Archivos "**Fedora7-x86-root_fs.bz2**" y el Kernel **"linux-2.6.38.tar.bz2"**

```
[proyecto@localhost build]$ chmod 777 linux-2.6.38.tar.bz2
[proyecto@localhost build]$ ls
Fedora7-x86-root_fs.bz2  linux-2.6.38.tar.bz2
[proyecto@localhost build]$
```

Ilustración 3-6 Sistema Anfitrión "Centos"

Descomprimimos el SISTEMA DE ARCHIVOS con el comando **bunzip2**, podemos utilizar el comando **bunzip2 –k** si queremos conservar el original

```
[proyecto@localhost build]$ bunzip2 Fedora7-x86-root_fs.bz2
```

Ilustración 3-7 Sistema Anfitrión "Centos"

Para el KERNEL debemos realizar doble descompresión, utilizamos los comandos **bunzip2** y **tar.**

```
[proyecto@localhost build]$ bunzip2 linux-2.6.38.tar.bz2
[proyecto@localhost build]$ tar -xf linux*
```

Ilustración 3-8 Sistema Anfitrión "Centos"

Los parámetros **xf** adjuntos al comando **tar** indican que debe extraer todos los archivos y empaquetar el contenido del archivo.

x: extraer el archivo

f: empaquetar contenidos del archivo

Con el comando **ls** visualizamos que ya tenemos los archivos descomprimidos.

Ilustración 3-9 Sistema Anfitrión "Centos"

Debemos evitar que el directorio de trabajo sea /usr/src/linux, debido a que en esta área existen un conjunto de kernel headers (encabezados de kernel) y que por lo normal están incompletos, además son utilizados por los archivos de la librería headers. Estos deberían sincronizar las librerías del nuevo sistema en lugar de desordenarlas.

También se puede actualizar los kernel **2.6.XX** mediante parches, los cuales son distribuidos en formato **gzip o bzip2**. Para instalar un parche obtenemos el más actualizado, ingresamos al directorio principal de la fuente del kernel (**linux-2.6.XX)** y ejecutamos **../patch-2.6.xx.gz gzip-cd | patch-p1** o **bzip2-dc ../patch-2.6.xx.bz2 | patch-p1.**

Para evitar utilizar uno o más parches debemos descargarnos la última versión, o saber exactamente qué clase de aplicación necesitamos, en este caso no vamos a parchar nuestro kernel porque vamos a configurar una versión para el uso del honeypot, y además porque tenemos la última versión disponible.

No es recomendable usar versiones muy antiguas debido a que ocasionarían errores y que no se solucionarían solo con actualizar, por lo cual recomendamos utilizar versiones de kernel **2.6.XX** en adelante, además se debe verificar la compatibilidad con las librerías del kernel que se va a utilizar, si no tiene compatibilidad al momento de compilar se producirán

errores y pedirá librerías más actuales, para evitar esto se debe utilizar la versión más actual disponible.

3.2.1 DESCRIPCIÓN DE LIBRERÍAS QUE DEBEN ESTAR INSTALADAS

- ✓ **binutils:** maneja, enlaza, ensambla paquetes binarios
- ✓ **util-linux**: ayuda a visualizar mensajes de kernel, crea nuevos sistemas de archivos.
- ✓ **module-init-tools:** carga, inserta, remueve módulos del kernel.
- ✓ **e2fsprogs:** crea, verifica y mantiene sistemas de archivos.
- ✓ **pcmciautils:** facilita expansión de memoria, conexiones de red.
- ✓ **nfs-utils**: provee un modo de servicios NFS kernel.
- ✓ **procps:** provee información acerca del estado de los procesos.
- ✓ **udev:** es un servicio que crea y traslada nodos del directorio /dev, maneja eventos, y carga driver el momento de arrancar un sistema.

Para evitar errores durante la instalación de utilidades debemos verificar que existan ciertas librerías de desarrollo así como sus dependencias. Las librerías que se deben constatar son:

- ✓ gcc
- ✓ glibc –devel
- ✓ ncurses-devel
- ✓ rpm-build
- ✓ qt-devel
- ✓ fuse-devel

En el caso de no tenerlas debemos instalarlas con el comando **yum –y install [nombre de la librería]**; ejemplo: **yum –y install qt-devel,** con esto también tienen que estar las dependencias de cada librería.

3.2.2 CONSTRUCCION Y CONFIGURACION DEL KERNEL UML

Ingresar al directorio de la fuente del kernel

```
[proyecto@localhost linux-2.6.38]$
```

Ilustración 3-10 Sistema Anfitrión “Centos”

Antes de proceder a la configuración detallamos los comandos que podríamos utilizar:

make menuconfig: un programa gráfico, compuesto por menús, donde los componentes son presentados en listas de categorías, siendo posible seleccionar los componentes deseados de la misma manera que son presentados en el programa de instalación del Conectiva Linux. Basta con seleccionar el elemento correspondiente al ítem deseado: Y (si), N (no) o M (módulo).

make config: un programa texto interactivo, donde los componentes son presentados uno a uno. Basta con presionar Y (si), N (no) o M (módulo).

make xconfig: Es un programa X Windows, donde los componentes son listados en diferentes niveles de menús y los componentes son seleccionados utilizándose el ratón. Las selecciones posibles son Y (si), N (no) y M (módulo).

make oldconfig: Realiza una configuración basada en el archivo .confg existente y solo realiza preguntas, sobre puntos específicos de nueva configuración.

make defconfig: produce una configuración por defecto.

make allXXconfig: donde XX es yes, mod, no; crea una configuración donde establece la mayoría XX como sea posible, en las fases de configuración. (Sourceforge) (Linuxlots)

Utilizamos la configuración por defecto, con esto evitaremos la búsqueda de controladores que no incluyan la arquitectura, y para no establecer configuraciones fuera del alcance de la aplicación para honeypot, por lo tanto ejecutamos todos los comandos como usuario "**root**", de esta manera nos evitaremos tener restricciones a la hora de usar instrucciones y librerías.

Ejecutamos el comando **su** para cambiar de usuario y poder trabajar como superusuario.

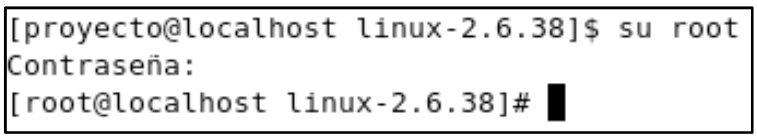


Ilustración 3-11 Sistema Anfitrión "Centos"

Una vez que ingresamos al directorio de la fuente con el comando **cd /home/proyecto/build/Linux-2.6.38** iniciamos la configuración por defecto, para esto utilizamos el comando **make defconfig**, si no se inicia con **defconfig** el kernel que se implementará va a utilizar la configuración por defecto del kernel del host anfitrión, es decir el kernel del Sistema Centos y como consecuencia es posible que UML no funcione. Además debemos colocar el signo **">"** para direccionar los datos de la compilación a un archivo

de texto llamado **mdefco.bk.txt** que se encuentra dentro del directorio **/home/proyecto/build**, este nos servirá como respaldo y allí podremos verificar esta compilación.

```
[root@localhost linux-2.6.38]# make defconfig ARCH=um > /home/proyecto/build/mdefco.bk.txt
```

Ilustración 3-12 Sistema Anfitrión "Centos"

Debemos tener mucha precaución al momento de utilizar el comando **make** mientras se construye UML, siempre se debe colocar **"ARCH=um"** junto con cada comando **make** ya que con esto definimos una arquitectura propia. Esto lo hacemos debido a que los archivos de configuración de la construcción del sistema anfitrión podrían interferir en la construcción de UML.

Si al momento de ejecutar el comando **make** por error omitimos escribir **ARCH=um** procedemos a limpiar el directorio de la fuente con el comando **make mrproper**.

```
[root@localhost linux-2.6.38]# make mrproper
```

```
[root@localhost linux-2.6.38]# make mrproper ARCH=um
```

Ilustración 3-13 Sistema Anfitrión "Centos"

E iniciamos nuevamente el proceso de compilación sin olvidar **ARCH=um** en cada **make**.

Ejecutamos el comando **make menuconfig,** aparecerá una ventana que nos permitirá configurar o cambiar cualquier parámetro en la compilación del kernel.

En el caso de los kernel 2.6.24.3 – 2.6.28 existe la opción **"Automatic kernel module loading"** y se pueden cargar los módulos del kernel al iniciar UML, mientras que para los kernel 2.6.29 hasta el actual que es el 2.6.38 no existe esta opción, razón por la cual tendremos que utilizar el comando **make all** para cargar los módulos.

```
[root@localhost linux-2.6.38]# make menuconfig ARCH=um
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
```

Ilustración 3-14 Sistema Anfitrión "Centos"

Nos aparecerá un cuadro de diálogo el cual nos permitirá modificar cualquier parámetro en la compilación del kernel.

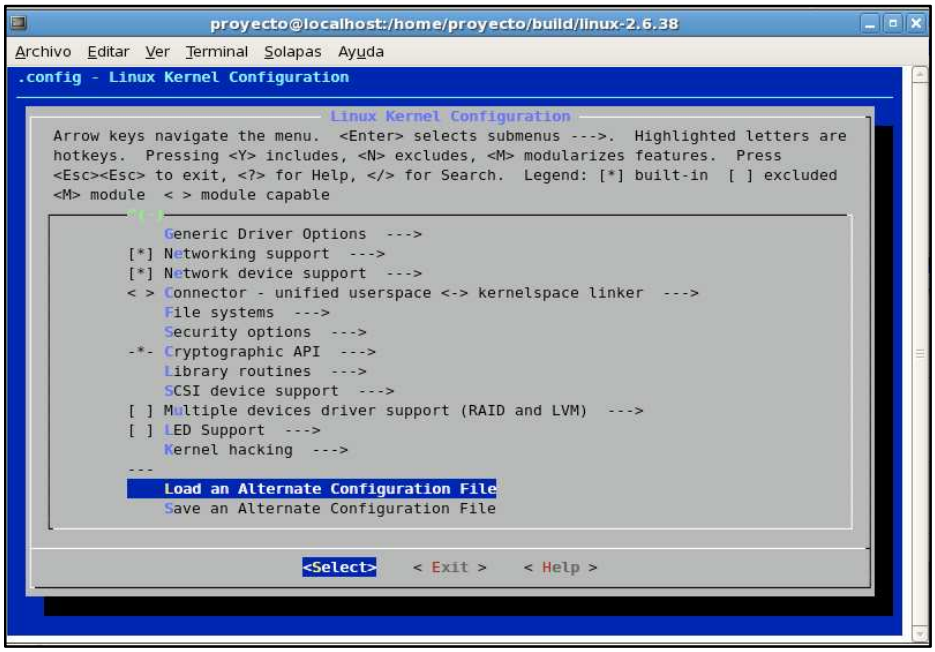


Ilustración 3-15 Sistema Anfitrión "Centos"

Buscamos la opción **Load an Alternate Configuration File** y marcamos la opción para cargar el archivo **.config**. Esto nos permitirá compilar y cargar los módulos del kernel. Guardar los cambios y salir del menú.

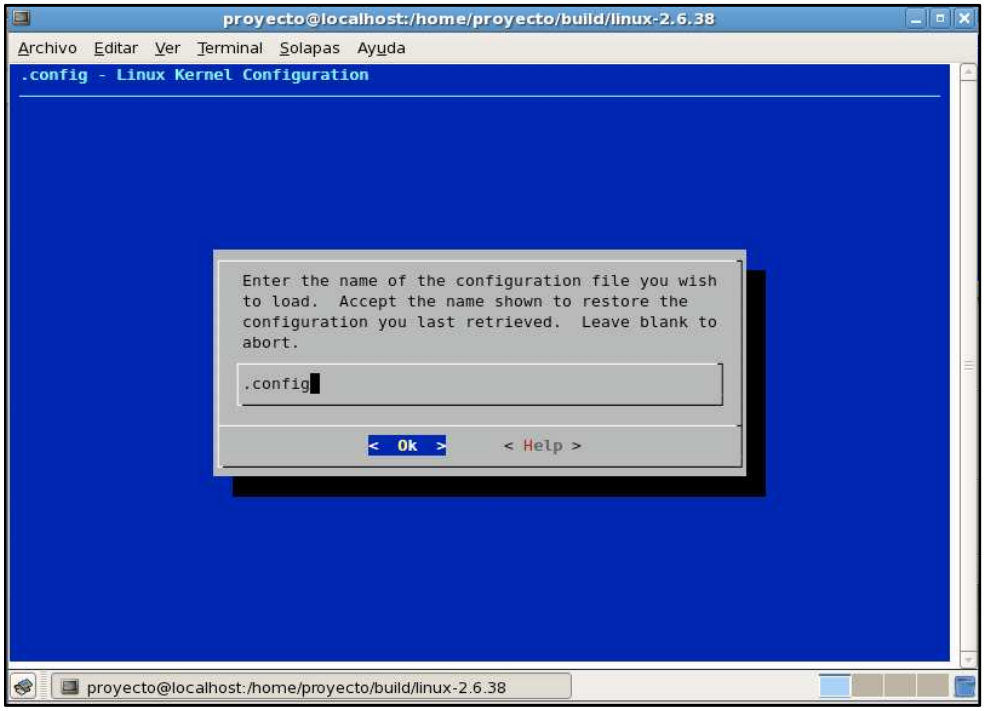


Ilustración 3-16 Sistema Anfitrión "Centos"

Al salir del menú se visualizará la siguiente pantalla:

```
scripts/kconfig/mconf arch/um/Kconfig.x86
#
# configuration written to .config
#


*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
```

Ilustración 3-17 Sistema Anfitrión "Centos"

Ahora instalamos los módulos del kernel en el sistema de archivos a utilizar, para lo cual creamos una carpeta dentro de **/mnt** llamada **fedo7,** le damos los permisos respectivos.

```
[root@localhost ~]# cd /mnt/
[root@localhost mnt]# mkdir fedo7
[root@localhost mnt]# chmod 777 fedo7
[root@localhost mnt]# ls
fedo7
[root@localhost mnt]#
```

Ilustración 3-18 Sistema Anfitrión "Centos"

Montamos el sistema de archivos a utilizar:

```
[root@localhost ~]# mount -o loop /home/proyecto/Fedora7-x86-root_fs /mnt/fedo7
[root@localhost ~]#
```

Ilustración 3-19 Sistema Anfitrión "Centos"

Verificamos si está montado dentro de **/mnt/fedo7:**

```
[root@localhost ~]# cd /mnt/fedo7/
[root@localhost fedo7]# ls
bin   dev  home  lost+found  mnt  proc  sbin     srv  tmp  var
boot  etc  lib   media       opt  root  selinux  sys  usr
```

Ilustración 3-20 Sistema Anfitrión "Centos"

Ejecutamos el comando **make all** para cargar los módulos del kernel, esto tomará un tiempo.

```
[root@localhost linux-2.6.38]# make all ARCH=um
scripts/kconfig/conf --silentoldconfig arch/um/Kconfig.x86
  CC      arch/um/sys-i386/user-offsets.s
  CHK     arch/um/include/shared/user_constants.h
  UPD     arch/um/include/shared/user_constants.h
  CHK     include/linux/version.h
  UPD     include/linux/version.h
  CHK     include/generated/utsrelease.h
  UPD     include/generated/utsrelease.h
  CC      kernel/bounds.s
  GEN     include/generated/bounds.h
  CC      arch/um/kernel/asm-offsets.s
  GEN     include/generated/asm-offsets.h
  CALL    scripts/checksyscalls.sh
  CC      scripts/mod/empty.o
  HOSTCC  scripts/mod/mk_elfconfig
  MKELF   scripts/mod/elfconfig.h
  HOSTCC  scripts/mod/file2alias.o
  HOSTCC  scripts/mod/modpost.o
  HOSTCC  scripts/mod/sumversion.o
```

Ilustración 3-21 Sistema Anfitrión "Centos"

Instalamos los módulos del kernel en el sistema de archivos a utilizar:

```
[root@localhost linux-2.6.38]# make modules_install INSTALL_MOD_PATH=/mnt/fedo7 ARCH=um
  INSTALL arch/um/drivers/hostaudio.ko
  INSTALL crypto/aes_generic.ko
  INSTALL crypto/ansi_cprng.ko
  INSTALL crypto/crypto_algapi.ko
  INSTALL crypto/krng.ko
  INSTALL crypto/rng.ko
  INSTALL drivers/block/loop.ko
  INSTALL drivers/block/nbd.ko
  INSTALL drivers/net/dummy.ko
  INSTALL drivers/net/ppp_generic.ko
  INSTALL drivers/net/slhc.ko
  INSTALL drivers/net/slip.ko
  INSTALL drivers/net/tun.ko
  INSTALL fs/autofs4/autofs4.ko
  INSTALL fs/binfmt_misc.ko
  INSTALL fs/isofs/isofs.ko
  INSTALL sound/soundcore.ko
  DEPMOD  2.6.38
```

Ilustración 3-22 Sistema Anfitrión "Centos"

Iniciamos la construcción ejecutando el comando **make** sin olvidar **ARCH=um,** tal como lo hicimos anteriormente respaldamos el proceso en un archivo llamado **make.bk.txt** ubicado dentro del directorio **/home/proyecto/build** y se iniciará un proceso de depuración con el cual

obtendremos un archivo UML binario llamado **linux**, el cual es el ejecutable de User Mode Linux.

```
[root@localhost linux-2.6.38]# make ARCH=um > /home/proyecto/build/make.bk.txt
```

Ilustración 3-23 Sistema Anfitrión "Centos"

Visualizamos con **ls**

```
[root@localhost linux-2.6.38]# ls -l lin*
-rwxr-xr-x 2 root root 26375993 mar 25 20:02 linux
```

Ilustración 3-24 Sistema Anfitrión "Centos"

Ejecutamos la instancia UML como cualquier ejecutable de Linux anteponiendo **./** al nombre. La directiva **ubda** nos permitirá elegir un sistema de archivos para que arranque el sistema operativo invitado, en este caso **"Fedora".**

```
[root@localhost linux-2.6.38]# ./linux ubda=/home/proyecto/build/Fedora7-x86-roo
t_fs
```

Ilustración 3-25 Sistema Anfitrión "Centos"

Arrancará el nuevo sistema operativo con imagen de Fedora, el cual se está ejecutando dentro del sistema operativo anfitrión Centos.

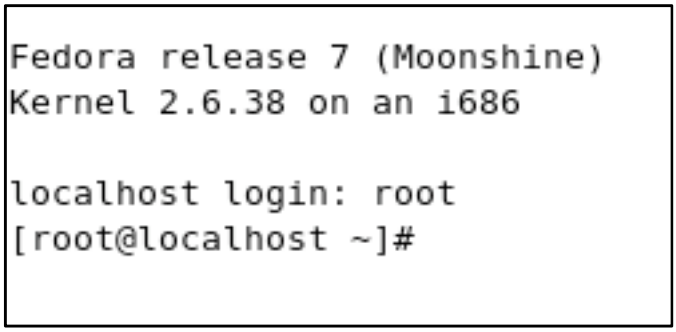


Ilustración 3-26 Sistema Anfitrión "Centos"

Fedora, que es el sistema invitado tendrá su propia consola por lo que no tenemos el mínimo riesgo de alterar el sistema anfitrión Centos ni su sistema de archivos.

El mismo kernel puede ser utilizado por diferentes tipos de sistemas de archivos, para iniciar se ejecuta un Fedora7 como honeypot que es donde se implementa el servidor web y mail honeypot, adicionalmente tendremos la honeynet conformada por una UML con un sistema de archivos Debian, otro con Centos5. Aquí podremos colocar todo tipo de información trampa para atraer atacantes, por ejemplo: bases de datos o cualquier clase de fichero que pueda ser útil para estos fines.

3.2.3 CREACIÓN DE UNA IMAGEN DE SISTEMA DE ARCHIVOS PARA SER UTILIZADA COMO HONEYPOT

Tenemos dos formas para obtener un sistema de archivos.

a.- La primera forma es construir un sistema de archivos y colocar los ficheros ajustándose a las necesidades de la aplicación que vamos a ejecutar. Esto lo hacemos a través de una conexión a internet.

Dentro del directorio **/var/local** creamos un archivo llamado **uml** y le damos los permisos de lectura y escritura.

```
[root@localhost ~]# cd /var/local/
[root@localhost local]# mkdir uml
[root@localhost local]# chmod 777 uml/
[root@localhost local]# cd uml/
[root@localhost uml]#
```

Ilustración 3-27 Sistema Anfitrión "Centos"

Ingresamos a **uml**. Con el comando **dd** se crea y se convierte el fichero **root_fs**, este será nuestro sistema de archivos y determinamos un tamaño de 750M.

```
[root@localhost uml]# dd if=/dev/zero of=/var/local/uml/root_fs=1M count=
1 seek=750
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0,000363 seconds, 1,4 MB/s
```

Ilustración 3-28 Sistema Anfitrión "Centos"

También creamos un área de intercambio o swap, con un tamaño de 250M.

```
[root@localhost uml]# dd if=/dev/zero of=/var/local/uml/swap_fs=1M count=
1 seek=250
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0,000193 seconds, 2,7 MB/s
```

Ilustración 3-29 Sistema Anfitrión "Centos"

Establecemos el tipo ext3 de linux a la raíz del sistema de archivos creado mediante el comando **mkfs.**

```
[root@localhost uml]# mkfs.ext3 -F root_fs=1M
mke2fs 1.39 (29-May-2006)
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=1024 (bitácora=0)
Tamaño del fragmento=1024 (bitácora=0)
48 nodos i, 372 bloques
18 bloques (4.84%) reservados para el súper usuario
Primer bloque de datos=1
Maximum filesystem blocks=524288
1 bloque de grupo
8192 bloques por grupo, 8192 fragmentos por grupo
48 nodos i por grupo

Mientras se escribían las tablas de nodos i: terminado


El sistema de ficheros es demasiado pequeño para un fichero de transaccio
nes
Escribiendo superbloques y la información contable del sistema de fichero
s: hecho

Este sistema de ficheros se revisará automáticamente cada 20 meses o
180 dias, lo que suceda primero.  Utilice tune2fs -c o -i para cambiarlo.
```

Ilustración 3-30 Sistema Anfitrión "Centos"

Definimos el área de intercambio swap_fs, para el sistema de archivos root_fs, con el comando **mkswap**.

```
[root@localhost uml]# /sbin/mkswap swap_fs=1M
Configurando espacio de intercambio versión 1, tamaño = 122 kB
[root@localhost uml]#
```

Ilustración 3-31 Sistema Anfitrión "Centos"

Creamos un archivo llamado **centinst** dentro del directorio **/mnt** al cual le damos los permisos respectivos.

```
[root@localhost ~]# cd /mnt/
[root@localhost mnt]# mkdir ceninst
[root@localhost mnt]# chmod 777 ceninst
[root@localhost mnt]# ls
ceninst  fedo7
```

Ilustración 3-32 Sistema Anfitrión "Centos"

Montamos desde la raíz, allí cargaremos un sistema de archivos vía ftp.

```
[root@localhost ~]# mount -o loop /var/local/uml/root_fs=1M /mnt/ceninst
```

Ilustración 3-33 Sistema Anfitrión “Centos”

Vía ftp nos descargamos el sistema de archivos Centos, al que modificaremos según las necesidades, respaldando el proceso en el archivo respaft.txt

```
[root@localhost ~]# debootstrap --arch i386 etch /mnt/ceninst/ http://ftp.nl
.debian.org/debian > respatf.txt
```

Ilustración 3-34 Sistema Anfitrión “Centos”

Una vez hecha la descarga podemos modificar cualquier archivo y añadir cualquier software de acuerdo a nuestras necesidades.

b.- Si no tenemos conexión a internet para la descarga vía ftp, la elección es descargar un sistema de archivos pre-construido. Para este caso ya vienen pre-instaladas todas las librerías y utilidades de un sistema básico de linux, por lo que solo se requiere añadir los servicios y aplicaciones necesarios. Como en este caso vamos a implementar un honeypot que proporcionará servicios web y mail llamado honeyserver, con el sistema de archivos montado se copia desde un cd repositorio o desde el disco duro del sistema anfitrión los archivos requeridos como por ejemplo los paquetes de sendmail, apache, dovecot, etc.

En el sistema virtual Fedora7 una dependencia para dovecot es la librería de mysql, **mysql-libs-5.0.37-2.fc7.i386.rpm**, por lo que con el comando **mount**

montamos el sistema de archivos reconstruido en un directorio llamado **fedo7**, el cual fue creado dentro de **/mnt**

```
[root@localhost ~]# mount -o loop /home/proyecto/Fedora7-x86-root_fs /mnt/fe
do7/
```

Ilustración 3-35 Sistema Anfitrión "Centos"

Realizamos la copia de los archivos requeridos, desde un DVD repositorio. Es importante tener en cuenta que antes de realizar la copia debemos darle los permisos necesarios al **home** que se encuentra dentro de **fedo7**

```
[root@localhost ~]# cd /mnt/fedo7/
[root@localhost fedo7]# ls
bin   dev  home  lost+found  mnt  proc  sbin     srv  tmp  var
boot  etc  lib   media       opt  root  selinux  sys  usr
[root@localhost fedo7]# chmod 777 home/
[root@localhost fedo7]# ls -l
total 88
drwxr-xr-x  2 root root  4096 jun 18  2008 bin
drwxr-xr-x  2 root root  4096 abr 17  2007 boot
drwxr-xr-x  3 root root  4096 jun  1  2007 dev
drwxr-xr-x 44 root root  4096 oct 17  2008 etc
drwxrwxrwx  2 root root  4096 ene 14  2009 home
```

Ilustración 3-36 Sistema Anfitrión "Centos"

Realizamos la copia de todos los paquetes necesarios:

```
[root@localhost ~]# cd /media/Fedora\ 7\ i386\ DVD/Fedora/
[root@localhost Fedora]# cp mysql-libs-5.0.37-2.fc7.i386.rpm /mnt/fedo7/home/
[root@localhost Fedora]# cp dovecot-1.0.0-11.fc7.i386.rpm /mnt/fedo7/home/
[root@localhost Fedora]# cp postgresql-libs-8.2.3-2.fc7.i386.rpm /mnt/fedo7/home
/
```

Ilustración 3-37 Sistema Anfitrión "Centos"

Y de la misma manera que en la primera forma ejecutamos uml

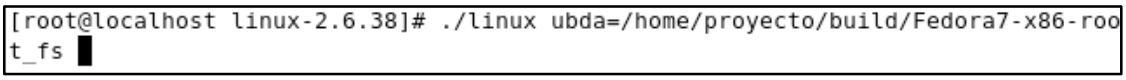


Ilustración 3-38 Sistema Anfitrión "Centos"

Al momento de arrancar el sistema invitado UML, se instalan los paquetes como lo haríamos en cualquier Sistema Fedora 7. Con esto se cargará una imagen Fedora 7 con todo lo necesario para configurar un servidor web y de correo electrónico virtuales a ser utilizados por el honeypot.

Una vez instalados los paquetes es recomendable borrarlos para que el fichero no crezca indiscriminada e innecesariamente y pierda su movilidad que es una de las principales características de los honeypot virtuales.

El kernel UML no puede trabajar independientemente, necesita un sistema de archivos, esta imagen se la obtiene creándola, o se la descarga pre-construida, esta imagen es tratada por el sistema anfitrión como un archivo más del sistema, puede estar colocada en cualquier directorio pero es preferible que sea en uno de usuario (excepto usuario root), para que no afecte al rendimiento y funcionamiento del sistema anfitrión. Una vez ejecutado en conjunto con UML como sistema invitado, su configuración, administración y mantenimiento es como cualquier otro sistema Linux.

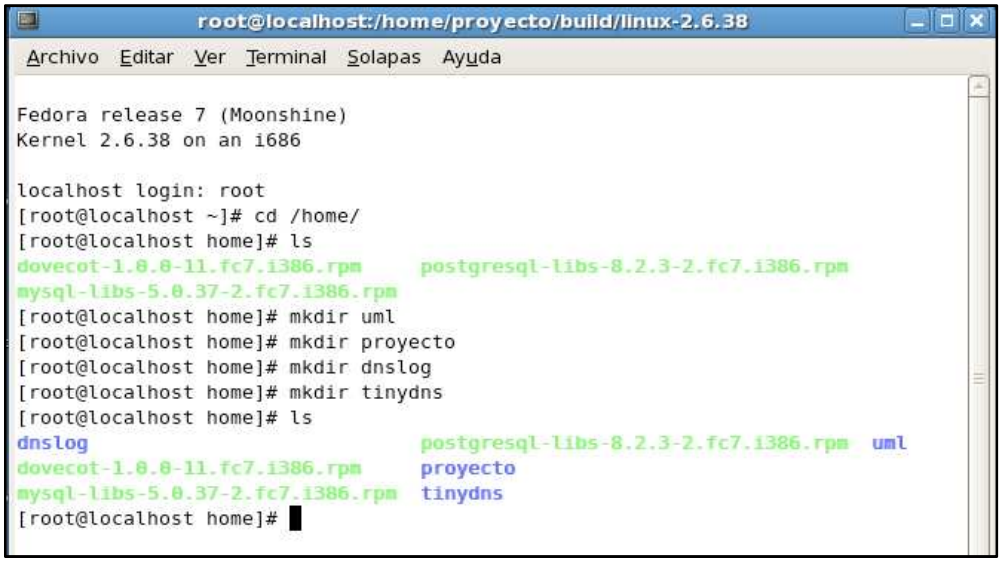


Ilustración 3-39 Sistema Invitado "Fedora7"

3.2.4 INSTALACION DE HERRAMIENTAS Y UTILIDADES DE UML

User Mode Linux tiene un sin número de utilidades, entre ellas la de permitirnos interactuar con el sistema de archivos del host anfitrión.

Antes de ejecutarlas necesitamos descargarlas e instarlas.

Nos descargamos utilizando el comando **wget** dentro de la carpeta **build** para luego descomprimirlas con los comandos **bunzip2** y **tar.**

```
[root@localhost build]# wget http://user-mode-linux.sourceforge.net/uml_utilitie
s_20070815.tar.bz2
```

```
[root@localhost build]# bunzip2 uml_utilities_20070815.tar.bz2
[root@localhost build]# tar -xf uml_utilities_20070815.tar
[root@localhost build]# ls
Fedora7-x86-root_fs  make.bk.txt     uml_utilities_20070815.tar
linux-2.6.38         mdefco.bk.txt
linux-2.6.38.tar     tools-20070815
```

Ilustración 3-40 Sistema Anfitrión "Centos"

Ingresamos al directorio tool-20070815

```
[root@localhost build]# cd tools-20070815
[root@localhost tools-20070815]#
```

Ilustración 3-41 Sistema Anfitrión "Centos"

Ejecutamos el comando **make all** y el comando **make install** respaldando dicho proceso como ya lo hemos hecho anteriormente

```
[root@localhost tools-20070815]# make all > /home/proyecto/anexoutil.txt
```

```
[root@localhost tools-20070815]# make install > /home/proyecto/anexoutil2.txt
```

Ilustración 3-42 Sistema Anfitrión "Centos"

Verificamos la correcta instalación en el directorio /usr/bin

```
[root@localhost tools-20070815]# ls /usr/bin/um*
/usr/bin/umbrello      /usr/bin/uml_moo     /usr/bin/uml_switch
/usr/bin/uml_mconsole  /usr/bin/uml_mount   /usr/bin/uml_watchdog
/usr/bin/uml_mkcow     /usr/bin/uml_net
[root@localhost tools-20070815]# ls /usr/lib/um*
port-helper
```

Ilustración 3-43 Sistema Anfitrión "Centos"

3.2.5 DESCRIPCION DE UTILIDADES Y CONFIGURACION DE LA MAQUINA VIRTUAL USER MODE LINUX

Hay dos archivos del sistema de UML que permiten el fácil acceso al sistema de archivos del sistema anfitrión, estos son: hostfs y humfs. En ambos casos,

se monta un sistema de archivos dentro de UML y bajo este estará contenido un directorio del host.

El **hostfs** trabaja muy bien montando directorios del host anfitrión, siempre y cuando:

- ✓ No se creen sockets Unix o nodos de dispositivo.
- ✓ No sea un punto de arranque del sistema.
- ✓ No se escriban archivos del host como un usuario diferente a root dentro de UML.

El **humfs** no muestra este tipo de problemas, pero es menos recomendable usarlo, porque para esto se necesita configurar el directorio del host a ser montado como un directorio humfs, antes de montarlo dentro de UML.

Hay una restricción común para ambos métodos, si un archivo es escrito en UML, los cambios en el sistema anfitrión serán efectivos en un período de tiempo que no se puede precisar.

Esto significa que UML no verá los cambios realizados en el host, incluso podría sobrescribirlos si algo dentro de UML ha cambiado los mismos archivos. Además los cambios realizados dentro de UML no serán visibles en el host de inmediato. Se podría corregir que las actualizaciones aparezcan en el host mediante el montaje del sistema de archivos de forma sincronizada con la opción "**-o sync**", pero esto nos resultaría muy difícil.

3.2.6 CONFIGURACIÓN DE HOSTFS

Ejecutamos dentro de UML y creamos una carpeta llamada **host**

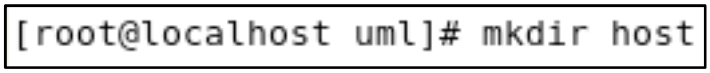


Ilustración 3-44 Sistema Invitado "Fedora7"

Montamos el directorio de root del sistema anfitrión en UML en el directorio **/host**

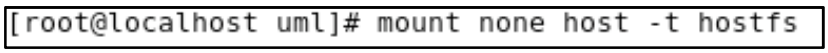


Ilustración 3-45 Sistema Invitado "Fedora7"

Ingresamos a la carpeta **host** y visualizamos con **ls,** aquí están todos los archivos de sistema anfitrión **Centos**, incluso podemos ver al usuario que contiene.

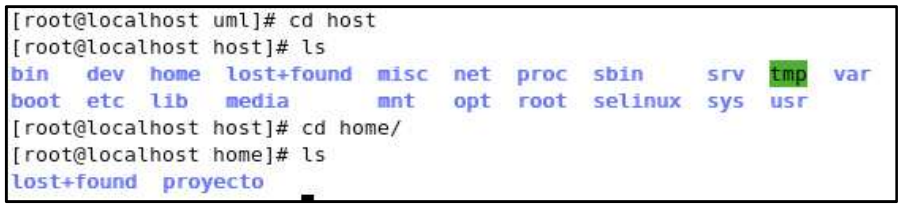


Ilustración 3-46 Sistema Invitado "Fedora7"

Si queremos montar otro subdirectorio, como un subdirectorio del usuario dentro de **home**, especificamos con la opción **–o**, aquí montaremos el subdirectorio del host anfitrión **/home/user** en el directorio **/host** dentro de UML.

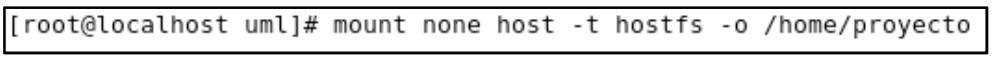


Ilustración 3-47 Sistema Invitado "Fedora7"

3.2.7 CONFIGURACIÓN DE HUMFS

Con el **humfs** podemos montar cualquier directorio del host anfitrión **Centos** dentro de UML, para ello el directorio a ser montado debe estar configurado antes de continuar. Esto lo hacemos con la herramienta **humfsify**, que forma parte del paquete **uml_utilities**.

Dentro del sistema anfitrión Centos creamos un directorio en el root llamado **humfs-mount** y luego ingresamos en el mismo

```
[root@localhost ~]# mkdir humfs-mount
[root@localhost ~]# cd humfs-mount/
```

Ilustración 3-48 Sistema Anfitrión "Centos"

Dentro de **humfs-mount** creamos un directorio llamado **mnt** para luego montar el sistema de archivos **Fedora7**

```
[root@localhost humfs-mount]# mkdir mnt
[root@localhost humfs-mount]# mount /home/proyecto/Fedora7-x86-root_fs mnt -o loop
```

Ilustración 3-49 Sistema Anfitrión "Centos"

Realizamos una copia del contenido de la imagen del sistema de archivos UML existente a un subdirectorio llamado **data**, esta copia se la debe realizar como usuario root, adicionalmente colocamos la opción **–a** que nos permitirá preservar la estructura, atributos y permisos de los archivos originales.

```
[root@localhost humfs-mount]# cp -a mnt data
[root@localhost humfs-mount]# ls
data  mnt
```

Ilustración 3-50 Sistema Anfitrión "Centos"

Como usuario root, ejecutamos la utilidad **humfsify** para convertir este directorio al formato que necesita el sistema de ficheros UML humfs, aquí debemos especificar el nombre de usuario y el grupo que ejecutará UML, el último argumento es el tamaño del sistema de archivos visto desde UML.

```
[root@localhost humfs-mount]# humfsify proyecto bin 4G
```

Ilustración 3-51 Sistema Anfitrión "Centos"

Ahora ya se lo puede montar dentro de UML, adicionando la opción **–o** ya que especifica la dirección del archivo **host** que deseamos montar.

```
[root@localhost uml]# mount none host -t hostfs -o /home/proyecto
```

Ilustración 3-52 Sistema Invitado "Fedora 7"

Para arrancar un sistema de archivos **humfs**, se debe acoplar a **humfsify** el sistema de archivos UML.

```
rootstype=humfs rootflags=/path/to/humfs/root
```

Ilustración 3-53 Sistema Anfitrión "Centos"

3.2.8 CONFIGURACION DE RED PARA UML

Primero debemos verificar la configuración de red del sistema anfitrión **Centos** ya que UML utilizará como intermediario la red del sistema anfitrión para poder conectarse a internet.

Cabe resaltar que en la configuración de la red para la implementación de este Proyecto se utilizarán direcciones IP virtuales por motivos de seguridad.

Para visualizar todas las interfaces de red que tiene el host hasta el momento utilizamos el comando **ifconfig**:

```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:97:13:EE
          inet addr:192.168.0.120  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe97:13ee/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7541 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4394 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1743939 (1.6 MiB)  TX bytes:244320 (238.5 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:5718 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5718 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:11421526 (10.8 MiB)  TX bytes:11421526 (10.8 MiB)

peth0     Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          inet6 addr: fe80::fcff:ffff:feff:ffff/64 Scope:Link
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:33535 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4456 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5135337 (4.8 MiB)  TX bytes:253696 (247.7 KiB)
          Interrupt:18 Base address:0x2000

tap0      Link encap:Ethernet  HWaddr 2E:24:DD:9F:8F:7C

vif0.0    Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          inet6 addr: fe80::fcff:ffff:feff:ffff/64 Scope:Link
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:4394 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7541 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:244320 (238.5 KiB)  TX bytes:1743939 (1.6 MiB)

virbr0    Link encap:Ethernet  HWaddr 00:00:00:00:00:00
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::200:ff:fe00:0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:44 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:8292 (8.0 KiB)

xenbr0    Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:6924 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:697180 (680.8 KiB)  TX bytes:0 (0.0 b)
```

Ilustración 3-54 Sistema Anfitrión "Centos"

Asignamos un ID Unique Machine**,** con el comando **umid=my-uml** que es un comando interno diseñado para la máquina virtual, esto nos generará una identificación en el directorio de usuario que esté ejecutando UML.

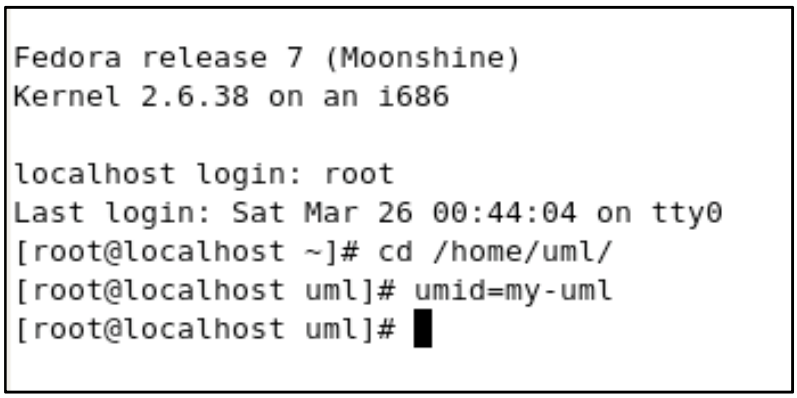
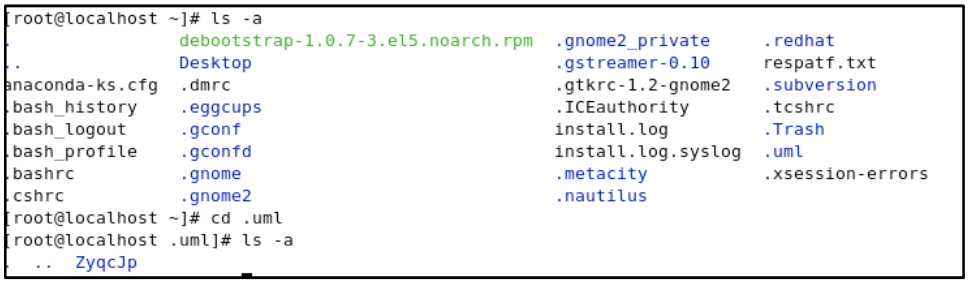


Ilustración 3-55 Sistema Invitado "Fedora7"

Ingresamos al root del sistema anfitrión Centos y visualizamos con **ls** si existe el directorio **.uml,** dentro de este directorio verificamos el ID con el comando **ls –a,** y **cd** al **ID** para verificar la existencia de los archivos.



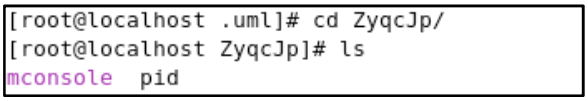


Ilustración 3-56 Sistema Anfitrión "Centos"

Ingresamos nuevamente a la consola UML Fedora7 y asignamos la dirección IP a UML que nos permitirá tener comunicación.

```
[root@serverespe ~]# eth0=tuntap,,,192.168.0.20
```

Ilustración 3-57 Sistema Invitado "Fedora7"

Para poder comunicarnos con el sistema invitado, ejecutamos dentro del host anfitrión Centos el siguiente comando:

```
[root@localhost ~]# cd /home/tesis/build/linux-2.6.38
[root@localhost linux-2.6.38]# uml_mconsole ZyqcJp eth0=tuntap,,,192.168.0.20
```

Ilustración 3-58 Sistema Anfitrión "Centos"

Hemos creado una interfaz lógica en el sistema anfitrión Centos llamada **tap0**, esta es la pasarela por defecto del sistema invitado Fedora7 y es **192.168.0.20**. Comprobamos que las direcciones IP's son diferentes a las otras interfaces de red.

```
[root@localhost linux-2.6.38]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:97:13:EE
          inet addr:192.168.0.120  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe97:13ee/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9929 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4394 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2046306 (1.9 MiB)  TX bytes:244320 (238.5 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:5718 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5718 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:11421526 (10.8 MiB)  TX bytes:11421526 (10.8 MiB)

peth0     Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          inet6 addr: fe80::fcff:ffff:feff:ffff/64 Scope:Link
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:37292 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4456 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5576679 (5.3 MiB)  TX bytes:253696 (247.7 KiB)
          Interrupt:18 Base address:0x2000

tap0      Link encap:Ethernet  HWaddr 2E:24:DD:9F:8F:7C
          inet addr:192.168.0.20  Bcast:192.168.0.20  Mask:255.255.255.255
          inet6 addr: fe80::2c24:ddff:fe9f:8f7c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:18 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:1204 (1.1 KiB)  TX bytes:8818 (8.6 KiB)

tap1      Link encap:Ethernet  HWaddr FE:3B:8E:D1:FF:BF
          inet addr:192.168.0.26  Bcast:192.168.0.26  Mask:255.255.255.255
          inet6 addr: fe80::fc3b:8eff:fed1:ffbf/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:378 (378.0 b)  TX bytes:8143 (7.9 KiB)

tap2      Link encap:Ethernet  HWaddr 4E:8D:BD:03:46:AA
          inet addr:192.168.0.24  Bcast:192.168.0.24  Mask:255.255.255.255
          inet6 addr: fe80::4c8d:bdff:fe03:46aa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:672 (672.0 b)  TX bytes:8484 (8.2 KiB)
```

```
vif0.0    Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          inet6 addr: fe80::fcff:ffff:feff:ffff/64 Scope:Link
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:4394 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9929 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:244320 (238.5 KiB)  TX bytes:2046306 (1.9 MiB)

virbr0    Link encap:Ethernet  HWaddr 00:00:00:00:00:00
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::200:ff:fe00:0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:44 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:8292 (8.0 KiB)

xenbr0    Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:9277 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:960297 (937.7 KiB)  TX bytes:0 (0.0 b)
```

Ilustración 3-59 Sistema Anfitrión "Centos"

La parte eth0 del comando crea la interfaz eth0 con el sistema operativo invitado**,** y envía paquetes a través de tap0 en el sistema operativo anfitrión Centos, ahora asignamos una dirección IP en el sistema invitado Fedora7.

Para esto ingresamos al directorio **/etc/sysconfig/network-scripts** e insertamos la IP y la máscara de red con el comando **vi ifcfg-eth0**, como podemos darnos cuenta esto lo realizamos como se lo hace en cualquier sistema Linux.

```
[root@localhost network-scripts]# cd /etc/sysconfig/
[root@localhost sysconfig]# cd network-scripts/
[root@localhost network-scripts]# vi ifcfg-eth0
```

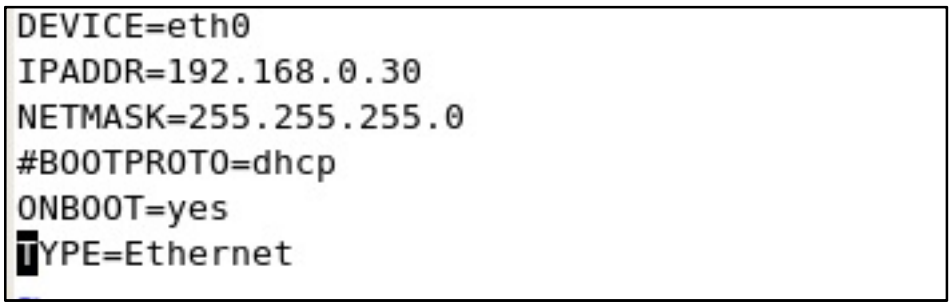


Ilustración 3-60 Sistema Invitado "Fedora7"

Habilitamos eth0 dentro del sistema invitado Fedora7

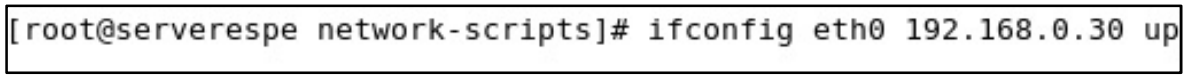


Ilustración 3-61 Sistema Invitado "Fedora7"

Comprobamos la conexión haciendo **ping** hacia el sistema anfitrión Centos desde el sistema invitado

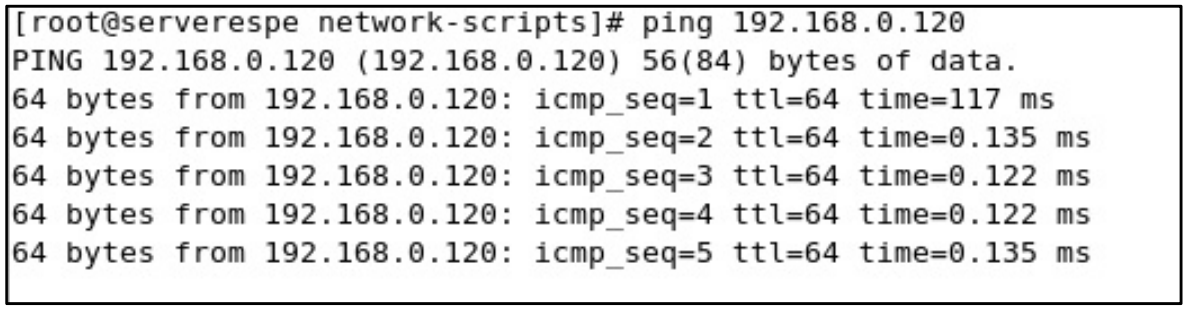


Ilustración 3-62 Sistema Invitado "Fedora7"

Podemos ver que tenemos respuesta.

Ahora lo hacemos desde el sistema anfitrión hacia el sistema invitado

```
[root@localhost linux-2.6.38]# ping 192.168.0.30
PING 192.168.0.30 (192.168.0.30) 56(84) bytes of data.
64 bytes from 192.168.0.30: icmp_seq=1 ttl=64 time=0.260 ms
64 bytes from 192.168.0.30: icmp_seq=2 ttl=64 time=0.124 ms
64 bytes from 192.168.0.30: icmp_seq=3 ttl=64 time=0.204 ms
64 bytes from 192.168.0.30: icmp_seq=4 ttl=64 time=0.128 ms
64 bytes from 192.168.0.30: icmp_seq=5 ttl=64 time=0.119 ms
64 bytes from 192.168.0.30: icmp_seq=6 ttl=64 time=0.244 ms
```

Ilustración 3-63 Sistema Anfitrión "Centos"

También obtenemos respuesta.

Para conectarnos fuera de la red establecemos una ruta por defecto al host por tuntap ya configurado:

```
[root@serverespe network-scripts]# route add default gw 192.168.0.30
```

Ilustración 3-64 Sistema Invitado "Fedora7"

3.3 COMPONENTES ADICIONALES DEL SISTEMA HONEYPOT

El propósito de las honeynet es al igual que el honeypot, investigar el uso de las técnicas y herramientas que hacen los atacantes en internet. Se diferencia básicamente de un honeypot en que no supone una sola máquina, sino múltiples sistemas y aplicaciones que emulan otras tantas, imitan vulnerabilidades o servicios conocidos o crean entornos “jaula” donde es posible una mejor observación y análisis de los ataques. Los requerimientos básicos e imprescindibles para construir una honeynet son dos, los llamados Data Control (Control de datos) y Data Capture (Captura de datos).

3.3.1 ANALISIS DEL CONTROL DE DATOS

Una vez que hemos configurado UML y la red, el siguiente paso es el Control de Datos. El objetivo del Control de Datos es controlar qué es lo que el atacante puede hacer en la entrada y salida de la honeypot. Típicamente, permitimos cualquier cosa en la entrada a los sistemas honeypot, pero limitamos la salida.

Utilizaremos Iptables, que es un cortafuegos OpenSource que viene con Linux. Iptables es un cortafuegos altamente flexible, que incluye la habilidad de limitar conexiones, traducción de direcciones de red (NAT), registro y muchas otras características. Iptables actúa como un filtro en nuestro host, contando los paquetes de salida. Una vez que se haya alcanzado el límite de conexiones de salida, cualquier intento posterior será bloqueado, previniendo que el honeypot comprometido dañe a otros sistemas. Configurar e implementar esas características puede ser extremadamente complejo. No obstante, el honeypot Project ha desarrollado un **script Iptables** llamado **rc.firewall** que hace el trabajo por nosotros. Simplemente tenemos que modificar las variables del script para que se adapten a nuestra honeypot, y luego ejecutar el script. Esta configuración la realizaremos en el sistema operativo Centos 5, ya que es el gateway de la honeynet.

El firewall se puede configurar en modo de enrutamiento de nivel tres ("layer three routing mode"), o en modo puente de nivel dos ("layer two bridging mode"). El modo puente de nivel dos (también conocido como GenII, o segunda generación) es el método preferido. Cuando la pasarela actúa como puente, no hay enrutamiento o decrementos de paquetes TTL (time to live), actúa como un dispositivo de filtro invisible, haciéndola más difícil de detectar ante atacantes.

Hay dos áreas críticas que configurar, los asuntos de red y los de conexión, identificaremos las direcciones IP y las redes pertenecientes al sistema anfitrión y al sistema invitado UML.
En realidad, la red es más simple en modo puente que en modo de enrutamiento. En modo puente no hay enrutamiento, ni asuntos de Traducción de Direcciones de Red (NAT).
Una vez que se haya configurado el script **honeywall.sh** se lo implementa ejecutándolo en el sistema anfitrión con el comando: **./honeywall.sh**
Aplicando cualquiera de las herramientas tendríamos varios controles como la detección y modificación de ataques, limitaciones del ancho de banda además de impedir ciertas aplicaciones.

3.3.2 ANALISIS DE CAPTURA DE DATOS

Es la captura y almacenamiento de la actividad, tanto de entrada como de salida, es una tarea fundamental en un honeypot. Un análisis de los datos, ya sea con el objetivo de la formación o aprendizaje, como si es el de analizar un posible ataque fuera de los controles habituales de la red, necesita de una buena recolección de información. Es recomendable efectuar una captura de datos a varios niveles. Podemos diferenciar entre 3 niveles diferentes y complementarios:

El registro del Firewall: "Logear" todo el tráfico controlado por el cortafuegos. Es el punto crítico y donde es posible obtener mayor información sobre las actividades llevadas a cabo por el atacante.

El tráfico de red: Captura de cada paquete, junto con su contenido (payload), que entra y sale de la red. Para esta tarea, la herramienta más indicada es un

IDS (Sistema de Detección de Intrusiones), concretamente se aconseja el uso de Snort, un IDS de software libre usado masivamente por los administradores de sistemas y grupos de seguridad, y que ofrece una configuración y clasificación envidiable. Puede usarse la aplicación BASE (Basic Analysis and Security Engine) para acceder y analizar la información capturada de una forma ágil y sencilla.

Actividad del sistema: La captura de la actividad ocurrida en el honeypot es otra parte fundamental para la comprensión del ataque. Esta puede resultar más complicada de lo que en un principio pueda parecer ya que actualmente se utilizan conexiones cifradas. Aún así, si el honeypot lleva a cabo correctamente su cometido, la comunicación será descifrada y podrá ser analizada. Como ya comentamos en el post de la serie que trataba los diferentes honeypots, uno de los de alta interacción, Sebek, está orientado a registrar las pulsaciones de teclado del atacante directamente desde el kernel, lo que nos permite seguir de una forma muy efectiva su actividad en el honeypot. (Honeynet Project, 2003)

3.4 LEGALIDAD DEL USO DE LOS HONEYPOTS

Algunos de los problemas que se asocian al uso de Honeynets se centran en el aspecto legal, sin dejar a un lado el alto riesgo de que se comprometan más sistemas desde la Honeynet, la legalidad de las Honeynet siempre estará en entredicho. La supuesta tentativa generada por el uso de Honeypots es uno de estos problemas. Se podría pensar que el hecho de colocar un equipo en la red con la finalidad de que sea comprometido, puede ser tomado como tentativa para comprometer otros sistemas, y fomentar la actividad maliciosa en la red,

es decir nosotros habilitamos el medio por el cual el intruso puede atacar a otros sistemas, sin embargo esto no es cierto. Debido a la naturaleza de la red cualquier equipo conectado a ella cae en este rubro, cualquier equipo puede ser el medio por el cual, un intruso puede comprometer a otros sistemas, y esto no significa que el propietario del equipo lo haga con ese fin. Los intrusos comprometen el equipo por su propia iniciativa, se puede considerar un Honeypot, como un objetivo alternativo en la organización destinado a la investigación de las amenazas a las cuales está expuesta.

3.4.1 PERMISOS Y SANSIONES

Uno de los requisitos necesarios para la implantación de una Honeynet en una organización, es el permiso de la misma para dicho proyecto. Se deberá de contar con un permiso explícito de la organización para monitorear, capturar, y analizar el tráfico que llega a la Honeynet, debido a que esto puede incluir información ajena a la detección de amenazas en la red, como por ejemplo: conversaciones por mensajero instantáneo, transferencia de archivos, etc.(UNAM)

3.4.2 REPERCUSIONES LEGALES

La ley norteamericana (al igual que la europea) protege la inviolabilidad de las comunicaciones personales (interception of communications) de una forma muy estricta. El punto de discusión se basa en que dependiendo del tipo de Honeynet que se utilice, se puede violar esta ley al recoger una serie de información sobre el atacante que la ley protege. Los Estados Unidos suelen ser los primeros en regular todo lo referente a la seguridad informática, y en

especial a cualquier cosa que afecte internet. Muchas leyes estatales y/o europeas se basan en las americanas.

Tal y como se ha explicado anteriormente, los honeynets en producción tienen un objetivo mucho más concreto (proteger la red), que los honeynets de investigación (cuyo interés se centra en conseguir tanta información de los atacantes como sea posible para comprender/estudiar su comportamiento y técnicas).

Lance Spitzner quien es uno de los fundadores del Proyecto de Investigación de Honeynet y que a su vez trabaja como Arquitecto Especialista de Seguridad para Sun Microsystems, desglosa las posibles responsabilidades legales derivadas del uso de honeypots y las honeynet en tres cuestiones básicas:

a. Trampa: Es el proceso realizado por los Cuerpos policiales (Law Enforcement) de "inducir" a alguien a cometer un acto punible con el objetivo de iniciar la acción judicial pertinente.

"En nuestro caso a pesar de que el Honeypot es un elemento pasivo creado por cuenta propia para ser atacado no busca perseguir judicialmente la intrusión en el Honeypot, tampoco se pretende realizar una trampa. El objetivo es simplemente recibir los ataques más no recoger información para demandar a los atacantes"

b. Privacidad: Que el Honeypot recoge información es innegable. Sin embargo, la información recogida puede dividirse en información transaccional (transactional) e información de contenido (content). La información transaccional (meta-información) no hace referencia a la información en sí, sino a aspectos de esta como la dirección IP, la fecha y hora, valores de las cabeceras de los paquetes IP.

La información de contenido es propiamente la comunicación que realiza el atacante con terceros. Precisamente este es el objetivo del debate (y también el de los Honeypots de investigación). La intercepción de una comunicación privada es la piedra angular que puede permitir a un atacante demandar ante un juzgado y probablemente ganar, en el caso de las leyes aplicables en países como Brasil, Estados Unidos y España; pero en El Salvador es un punto que no tiene validez ya que no existe una legislación que lo apruebe o lo prohíba – por el momento.

En cualquier caso, todos los autores están de acuerdo que se deberían incluir mensajes de advertencia y renuncia (disclaimer). Sin embargo esto no exime del problema, ya que el hecho de que se ponga un aviso no significa que un eventual atacante lo vea o lo lea.

c. Responsabilidad (Liability): Este aspecto hace referencia a las posibles demandas que se puede recibir en el caso de que un atacante utilice el Honeypot como plataforma de lanzamiento de ataques. Las demandas se basarían en que se ha realizado unos mínimos esfuerzos de seguridad en la red propia, sino que al contrario, se ha facilitado el acceso a los recursos para que sean utilizados en todo tipo de ataques. (UFG)

3.4.3 MARCO REGULATORIO DE LAS TELECOMUNICACIONES EN EL ECUADOR APLICADO A LA IMPLEMENTACIÓN DE HONEYPOTS

En el Ecuador existe regulación sobre los servicios de telecomunicaciones, pero no existe ninguna ley o reglamento que controle las redes de telecomunicaciones, o mucho menos hable de la intercepción de paquetes en redes informáticas. Sin embargo en la Ley Especial de Telecomunicaciones, constan artículos que hablan del derecho a la privacidad.

Art. 11._ USO PROHIBIDO._ Es prohibido usar los medios de telecomunicación contra la seguridad del estado, el orden público, la moral y las buenas costumbres.

La contravención en esta disposición será sancionada de conformidad con el Código Penal y más leyes pertinentes.

Art. 14._ DERECHO AL SECRETO DE LAS TELECOMUNICACIONES._ El estado garantiza el derecho al secreto, y a la privacidad de las telecomunicaciones. Es prohibido a terceras personas interceptar, interferir, publicar o divulgar sin consentimiento de las partes la información cursada mediante los servicios de telecomunicaciones.

En el artículo 11 se podría faltar a la moral y las buenas costumbres, lo que según el artículo sería sancionado por el Código Penal lo cual es un agravante. En el artículo 14 se prohíbe la intercepción de información, pero no detalla qué tipo de redes, sanciones a aplicar, y aun más si la red honeypot es implementada por una corporación de uso privado, y sufre un ataque desde internet, no quedaría claro si el delito es del atacante, o de la intercepción del paquete por la red privada. Lo que daría en un conflicto legal, similar al analizado en las leyes norteamericanas en cuanto a la **privacidad,** ya que el atacante puede revertir el proceso y simplemente implantar una demanda por intrusión a su privacidad. Sin embargo se puede concluir que existe un vacío legal en el Marco regulatorio ecuatoriano que aun no tiene los suficientes detalles en cuanto a intrusiones en redes de telecomunicaciones e informáticas. En cuanto a la nueva Constitución de la República del Ecuador aprobada en el Referéndum del 29 de Septiembre de 2008, en el Título II que habla sobre “Derechos”, en el Capítulo Sexto sobre Derechos de Libertad en el Artículo 66 incisos 19 y 21:

19. El derecho a la protección de datos de carácter personal, que incluye el acceso y la decisión sobre información y datos de este carácter, así como su

correspondiente protección. La recolección, archivo, procesamiento, distribución o difusión de estos datos o información requerirán la autorización del titular o el mandato de la ley.

21. El derecho a la inviolabilidad y al secreto de la correspondencia física y virtual; ésta no podrá ser retenida, abierta ni examinada, excepto en los casos previstos en la ley, previa intervención judicial y con la obligación de guardar el secreto de los asuntos ajenos al hecho que motive su examen. Este derecho protege cualquier otro tipo o forma de comunicación.

En este caso nuevamente no hay una prohibición explícita, y como sucede en muchos ámbitos legales del país quedaría de acuerdo a como se haga la interpretación: La recolección de información realizada en el *honeypot* no estaría respetando el inciso 19 del artículo 66 descrito anteriormente. Y se tendría que aclarar si un atacante el momento de apoderarse de un sistema y empieza a generar información esta se la podría considerar como de "carácter personal". En lo concerniente al inciso 21 habla de la inviolabilidad y secreto de la correspondencia virtual, lo que podría ser aplicado si se usara el *honeyserver* con el fin de enviar correos no deseados (spam) o tratar de difundir *software* malicioso, y bajo esta sospecha el administrador de la honeynet, espíe contenidos, y los bloquee con el fin de evitar que se use el honeyserver como plataforma para atacar otras redes y host. Al final de este inciso también se especifica que esto es aplicable a otros tipos o formas de comunicación. Por lo que cualquier tipo de rastreo realizado con el software "sniffer" estaría violando este inciso. (EPN, 2009)

4 CONFIGURACION E IMPLEMENTACIÓN DE LOS SERVIDORES WEB Y CORREO ELECTRONICO EN BASE A HERRAMIENTAS DEL SISTEMA OPERATIVO LINUX

4.1 CONFIGURACION Y COMPROBACION DE SERVICIOS REQUERIDOS PARA EL FUNCIONAMIENTO DE LOS SERVIDORES WEB Y DE CORREO ELECTRONICO

Dado que tanto Fedora7 como Centos5 están basados en el código fuente de la distribución Red Hat permite que los comandos y configuraciones utilizados sean iguales, razón por la cual la configuración e implementación hecha para el servidor web y de correo electrónico real en el sistema operativo Centos será la misma para el servidor virtual honeypot UML que está implementado bajo una imagen de Fedora7.

4.1.1 CONFIGURACIÓN DEL SERVICIO DE RED

En la configuración básica, se debe tener correctamente configurado el servicio de red del host para ser utilizado como servidor, también se debe comprobar que el servicio de red esté activo, esto podemos hacerlo con la ayuda del comando service, éste nos permite iniciar, detener o verificar el estado del servicio. (Portatiles) - (Mis respuestas.com, 2005)

Ingresar al directorio **cd /etc/** y digitar **service network status**

```
[root@server ~]# cd /etc
[root@server etc]# service network status
Dispositivos configurados:
lo eth0
Dispositivos activos en el momento:
lo peth0 virbr0 vif0.0 eth0 xenbr0
```

Ilustración 4-1 Sistema Operativo "Centos"

Con el comando **vi** editar el archivo de interfaz eth0, dentro del mismo directorio **/etc** ingresar a los subdirectorios "**sysconfig"** y **"network-scripts"**, nuevamente podemos utilizar el comando vi para editar la configuración actual.

```
[root@server etc]# cd sysconfig
[root@server sysconfig]# cd network-scripts/
[root@server network-scripts]# vi ifcfg-eth0
```

Ilustración 4-2 Sistema Operativo "Centos"

Asignar manualmente la dirección IP, éste va a ser el servidor DHCP de la red. Además debemos establecer las opciones para las interfaces de red, así como configurar para que el dispositivo de red se encienda cuando el sistema operativo se inicie.

Es muy importante tener en cuenta la coherencia de los próximos host para ser añadidos

```
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
HWADDR=00:0c:29:1e:d7:14
NETMASK=255.255.255.0
IPADDR=192.168.0.109
GATEWAY=192.168.0.1
```

Ilustración 4-3 Sistema Operativo "Centos"

Verificar que la interfaz esté habilitada con el comando **ifconfig**

```
[root@server ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:1E:D7:14
          inet addr:192.168.0.109  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe1e:d714/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14716 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4346 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:11427481 (10.8 MiB)  TX bytes:545033 (532.2 KiB)
```

Ilustración 4-4 Sistema Operativo "Centos"

Si la interfaz no está habilitada debemos restablecerla con el comando **“ifconfig <dispositivo de red> <dirección IP> netmask <dirección de máscara de red>”** ó dentro del directorio /etc con el comando **“/etc/rc.d/init.d/servicenetworkrestart**”.

```
[root@server ~]# ifconfig eth0 192.168.0.109 netmask 255.255.255.0
[root@server ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:1E:D7:14
          inet addr:192.168.0.109  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe1e:d714/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14745 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4346 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:11431618 (10.9 MiB)  TX bytes:545033 (532.2 KiB)
```

Ilustración 4-5 Sistema Operativo "Centos"

4.1.2 CONFIGURACIÓN DEL SERVICIO DHCP

DHCP (acrónimo de Dynamic Host Configuration Protocol que se traduce como Protocolo de Configuración Dinámica de Servidores) es un protocolo

que permite a dispositivos individuales en una red de direcciones IP obtener su propia información de configuración de red (dirección IP, máscara de sub-red, puerta de enlace, etc.) a partir de un servidor DHCP. Su propósito principal es hacer más fáciles de administrar las redes grandes.(linuxparatodos, 2010)

Existen tres métodos de asignación en el protocolo **DHCP**:

Asignación manual: La asignación utiliza una tabla con direcciones **MAC** (Media Access Control Address, que se traduce como Dirección de Control de Acceso al Medio). Sólo los anfitriones con una dirección MAC definida en dicha tabla recibirán el IP asignada en la misma tabla. Esto se hace a través de los parámetros hardware ethernet y fixed-address.

Asignación automática: Una dirección IP disponible dentro de un rango determinado se asigna permanentemente al anfitrión que la requiera.

Asignación dinámica: Se determina arbitrariamente un rango de direcciones IP y cada anfitrión conectado a la red está configurado para solicitar su dirección IP al servidor cuando se inicia el dispositivo de red, utilizando un intervalo de tiempo controlable (parámetros **default-lease-time** y **max-lease-time**) de modo que las direcciones IP no son permanentes y se reutilizan de forma dinámica. (Linuxparatodos)

Comprobar si el servidor DHCP está instalado ejecutando el comando el **service** dentro del directorio **/etc**

```
[root@server ~]# cd /etc
[root@server etc]# service dhcpd status
dhcpd: service desconocido
[root@server etc]#
```

Ilustración 4-6 Sistema Operativo "Centos"

En caso de no tener instalado el servidor, descargarse el paquete DHCP ya que este servicio se encarga de asignar direcciones IP dinámicas a la red.

Instalar el DHCP con el comando **yum –y install dhcp**

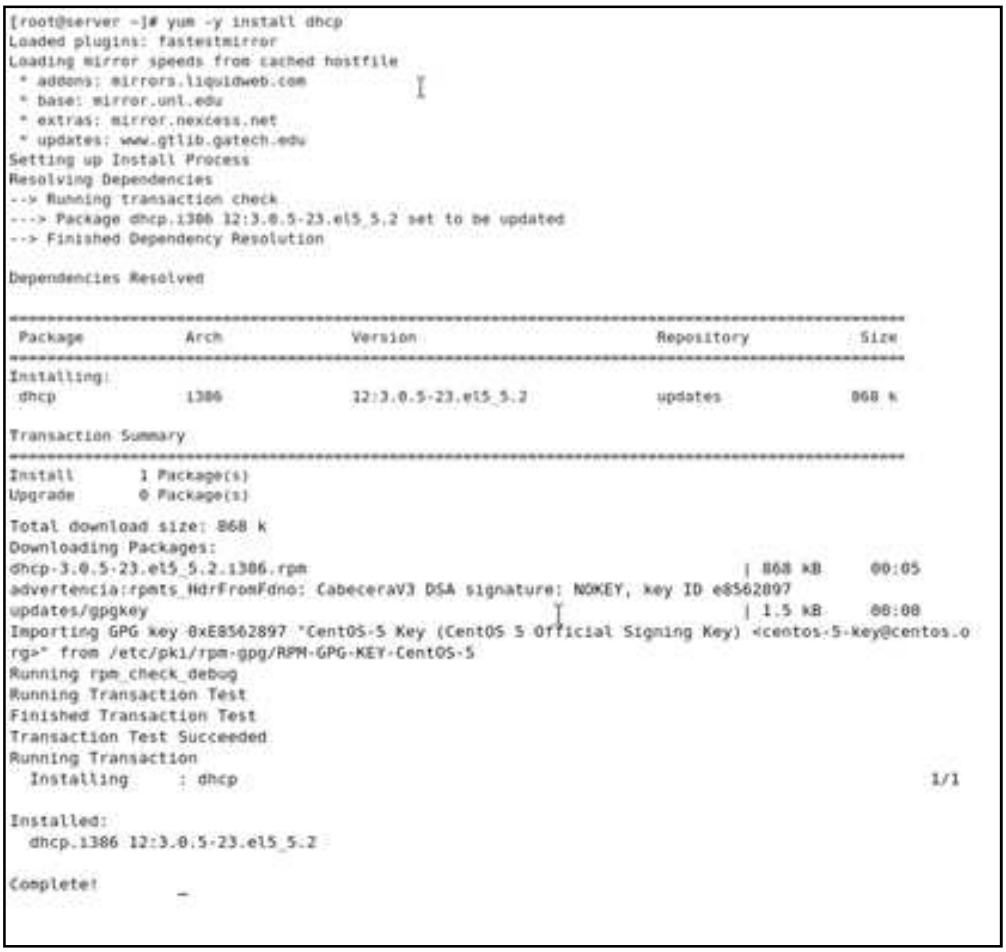


Ilustración 4-7 Sistema Operativo "Centos"

Comprobar si está instalado DHCP

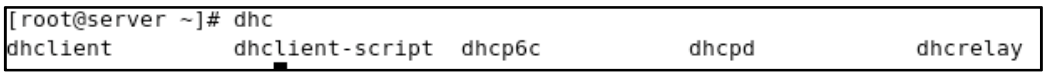


Ilustración 4-8 Sistema Operativo "Centos"

El archivo que se necesita para configurar se encuentra en el directorio **/usr/share/doc/dhcp-3.0.5**, aquí debemos copiar el archivo **dhcpd.conf.sample** en el directorio **/etc** para posteriormente cambiarlo de nombre por **dhcpd.conf**

```
[root@server ~]# cd /usr/share/doc/dhcp-3.0.5
[root@server dhcp-3.0.5]# cp dhcpd.conf.sample /etc
```

Ilustración 4-9 Sistema Operativo "Centos"

Con el comando **rename** renombramos el archivo **dhcpd.conf.sample**

```
[root@server ~]# cd /etc
[root@server etc]# ls -l dhc*
-rw-r--r-- 1 root root 178 mar 31  2010 dhcp6c.conf
-rw-r--r-- 1 root root  86 jul 29  2005 dhcpd.conf
-rw-r--r-- 1 root root 852 may  1 00:23 dhcpd.conf.sample
[root@server etc]# rm -rf dhcpd.conf
[root@server etc]# rename dhcpd.conf.sample dhcpd.conf dhcpd.conf.sample
```

Ilustración 4-10 Sistema Operativo "Centos"

Visualizar el cambio

```
[root@server etc]# ls -l dhcp*
-rw-r--r-- 1 root root 178 mar 31  2010 dhcp6c.conf
-rw-r--r-- 1 root root 852 may  1 00:23 dhcpd.conf
```

Ilustración 4-11 Sistema Operativo "Centos"

Para poder editarlo utilizamos el comando **vi dhcpd.conf**

```
[root@server etc]# vi dhcpd.conf
```

Ilustración 4-12 Sistema Operativo "Centos"

Modificamos los valores pertinentes ya que como habíamos mencionado debe existir coherencia con la información de red del servidor

```
ddns-update-style interim;
ignore client-updates;

subnet 192.168.0.0 netmask 255.255.255.0 {

# --- default gateway
        option routers                  192.168.0.1;
        option subnet-mask              255.255.255.0;

#       option nis-domain               "domain.org";
        option domain-name              "proyecto.eztesis.espe";
        option domain-name-servers      192.168.0.109;

        option time-offset              -18000; # Eastern Standard Time
#       option ntp-servers              192.168.1.1;
#       option netbios-name-servers     192.168.1.1;
# --- Selects point-to-point node (default is hybrid). Don't change this unless
# -- you understand Netbios very well
#       option netbios-node-type 2;

        range dynamic-bootp 192.168.0.0 192.168.0.254;
        default-lease-time 21600;
        max-lease-time 43200;

        # we want the nameserver to appear at a fixed address
        host ns {
                next-server marvin.redhat.com;
                hardware ethernet 12:34:56:78:AB:CD;
```

Ilustración 4-13 Sistema Operativo "Centos"

En la parte de **range dynamic** vamos a especificar el rango de IPs que se van a asignar, también podemos especificar la puerta de enlace, el Gateway, el servidor DNS y netbios en caso de tener clientes Windows.

En la opción **default-lease-time** podemos especificar el tiempo dado en segundos, tiempo durante el cual se reserva una dirección IP para el mismo cliente. Este registro se puede verificar en **/var/lib/dhcp/dhcpd.leases**. (Balboa)

Una vez instalado volver a ejecutar el comando **service dhcpd status** para comprobar su estado

```
[root@server ~]# service dhcpd status
Se está ejecutando dhcpd (pid  2833)...
[root@server ~]#
```

Ilustración 4-14 Sistema Operativo "Centos"

4.1.3 CONFIGURACION DEL SERVIDOR DNS

Antes de configurar el servidor web Apache debemos configurar el servidor DNS, una de las mejores opciones para implementarlo es el utilizar el software BIND (Berkeley Internet Name Domain).

Este software provee una implementación libre de los principales componentes del sistema de nombres de dominio, los cuales incluyen:

- ✓ Un servidor de sistema de nombres de dominio (named).
- ✓ Una biblioteca resolutoria de sistema de nombres de dominio.
- ✓ Herramientas para verificar la operación adecuada del servidor DNS (bind-utils).

DNS (acrónimo de Domain Name System) es una base de datos distribuida y jerárquica que almacena la información necesaria para los nombres de

dominio. Sus usos principales son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico correspondientes para cada dominio. El DNS nació de la necesidad de facilitar a los seres humanos el acceso hacia los servidores disponibles a través de Internet permitiendo hacerlo por un nombre, algo más fácil de recordar que una dirección IP, los servidores DNS utilizan TCP y UDP en el puerto 53 para responder las consultas. (Wikipedia, 2011)

"El servidor DNS BIND es ampliamente utilizado en la Internet (99% de los servidores DNS lo usan) proporcionando una robusta y estable solución.
Su configuración es muy complicada y además es considerado inseguro, razón por la cual utilizaremos TinyDNS, este software llega a cubrir las falencias que se tiene y su configuración es menos compleja".

Jerarquía DNS ó Estructura de Dominios DNS

A cada nivel de la estructura le asignó un nombre o etiqueta. El nivel cero, o raíz, no tiene nombre, el **nivel 1** puede ser alguno de los que se muestran en la figura, .mx, .uk, .com o .net, el cual se conoce como Top Level Domain – TLD. A su vez, éstos pueden tener subclasificaciones, como en el caso de .mx que tiene debajo a .com.mx, .net.mx, .gob.mx, etc. A este nivel se le conoce como Second Level Domain – SLD.

De esta forma, los nombres de dominio se construyen por una secuencia de etiquetas separadas por un punto, empezando en el nivel más profundo hasta llegar al nivel superior.

Por ejemplo, en la figura se puede apreciar que el nombre de dominio **empresa.com.mx.** se forma desde el último nivel llamado "empresa",

después el SLD “com” y por último el TLD “mx”

Las etiquetas pueden tener letras, números y el guión medio “-”, pero no puede iniciar ni terminar con guión. Cada etiqueta puede llevar hasta 63 caracteres, el nombre de dominio en total puede tener hasta 255 (cualquier combinación de letras, números y guión medio). Y puede haber hasta 127 niveles (siempre y cuando no se rebase el límite de 255 caracteres). (Inegi.gob)

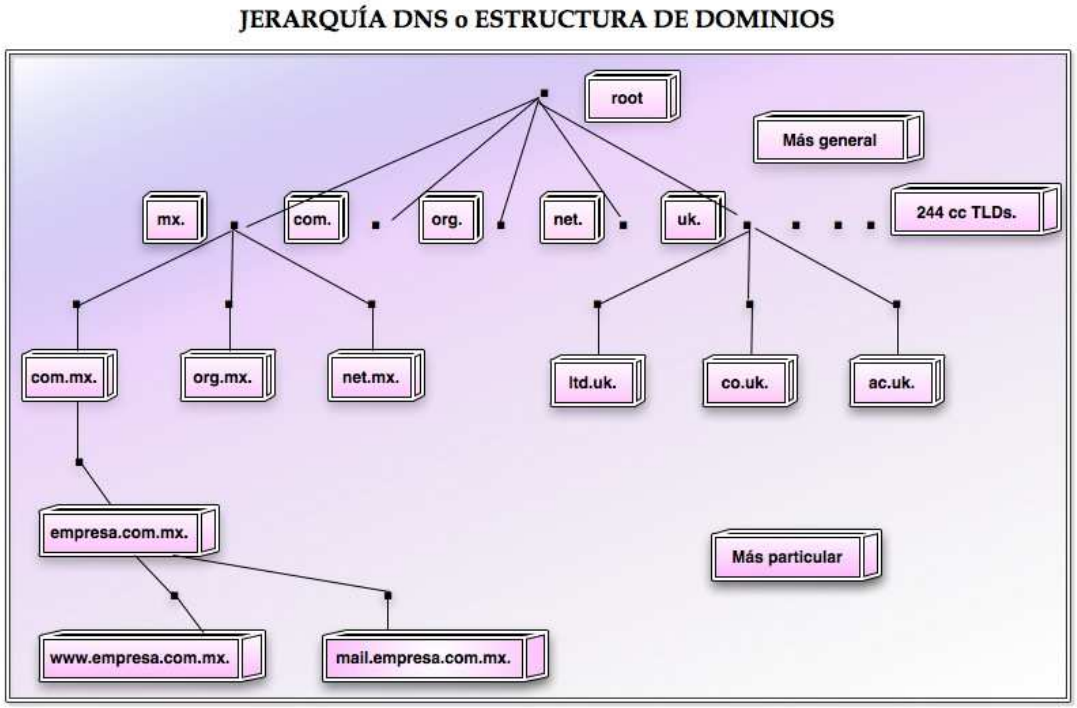


Ilustración 4-15 Estructura de Dominios

Fuente: http://www.inegi.org.mx/inegi/contenidos/espanol/prensa/contenidos/articulos/tecnologia/dns03.pdf

Tipos de dominios

.com: Son los dominios más extendidos en el mundo. Sirven para cualquier tipo de página web, temática, persona o entidad. Organizaciones comerciales.

.net: Originalmente orientado a empresas relacionadas con internet y la tecnología, en la actualidad se usa como alternativa a los dominios .com, y puede ser usado para cualquier tipo de página web. Pasarelas y otras redes administrativas.

.org: Diminutivo de “organización”, este tipo de dominios están orientados a organizaciones sin ánimo de lucro, asociaciones o fundaciones.

.es: Es un tipo de dominio territorial y se usa para páginas web que tengan alguna relación con España o la cultura española. Puede contratarlo cualquier persona o entidad sin necesidad de que tenga residencia en España.

.eu: Al igual que el .es o el .cat, es un dominio territorial cuyo ámbito son los países de la Unión Europea.

.info: Se utilizan para páginas de información general o puntual. Se puede usar como alternativa a los .com o .net.

.biz: Creados originalmente para empresas, se puede usar para cualquier tipo de página web y es al igual que los .info, una alternativa a los .com o .net.

.tv: Utilizados por páginas web que tienen secciones con vídeos o que estén relacionadas con el cine, televisión o medios de comunicación.

.tel: Esta extensión de dominio sirve para albergar los datos de contacto de una persona o entidad, email, dirección, teléfono fijo, móvil, Skype, fax, personas de contacto, dirección web, blogs, redes sociales o como llegar a una empresa o dirección concreta (geolocalización).

.cc: Este tipo de dominio es de ámbito global, y se utiliza como alternativa a los .com o a los .net. Significa "Compañía de Comercio".

.mobi: Este dominio se utiliza para páginas web especialmente construidas para funcionar tanto como una web tradicional, como en dispositivos móviles.

.cat: Esta extensión está orientada a páginas en Catalán o relacionadas con la cultura catalana.

.ws: Diminutivo de Web Site, se utiliza como dominio genérico para cualquier tipo de página, y es una alternativa a los .com o .net.

.be: al igual que los .es en España, los .be son la extensión para Bélgica, aunque se pueden contratar para cualquier tipo de página web. También puede ser interesante esta extensión para hacer juegos de palabras del tipo be.be, ara.be o escri.be. (Hispavista, 2011) - (Raxasolutions, 2007)

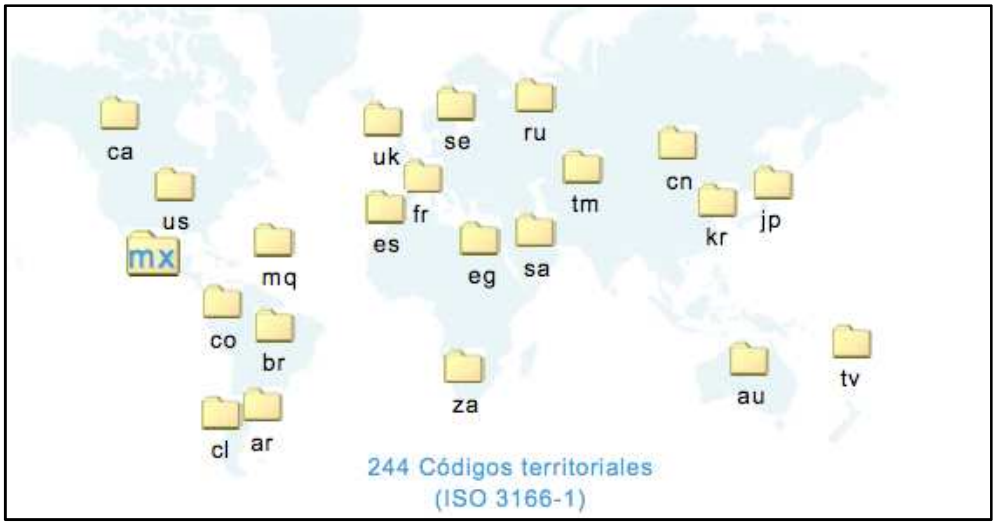


Ilustración 4-16 Dominios a nivel mundial

Fuente: http://www.inegi.org.mx/inegi/contenidos/espanol/prensa/contenidos/articulos/tecnologia/dns03.pdf

Cada país suele tener su propio dominio de primer nivel codificado con las dos letras de su país.

Concluyendo DNS define:

- ✓ Espacios jerárquicos por host.
- ✓ Una tabla de host implementada como una base datos distribuida.
- ✓ Rutinas de biblioteca para hacer preguntas a la base de datos.
- ✓ Implementa ruteo para e-mail.
- ✓ Un protocolo de intercambio de información de nombres.

Para especificar la ruta a seguir y anexar el nombre el dominio local se usa el archivo **/etc/resolv.conf,** en la línea **search** se coloca la IP del servidor DNS en la configuración del cliente, con lo que se evita desvíos a otras zonas posibles. (Inegi.gob)

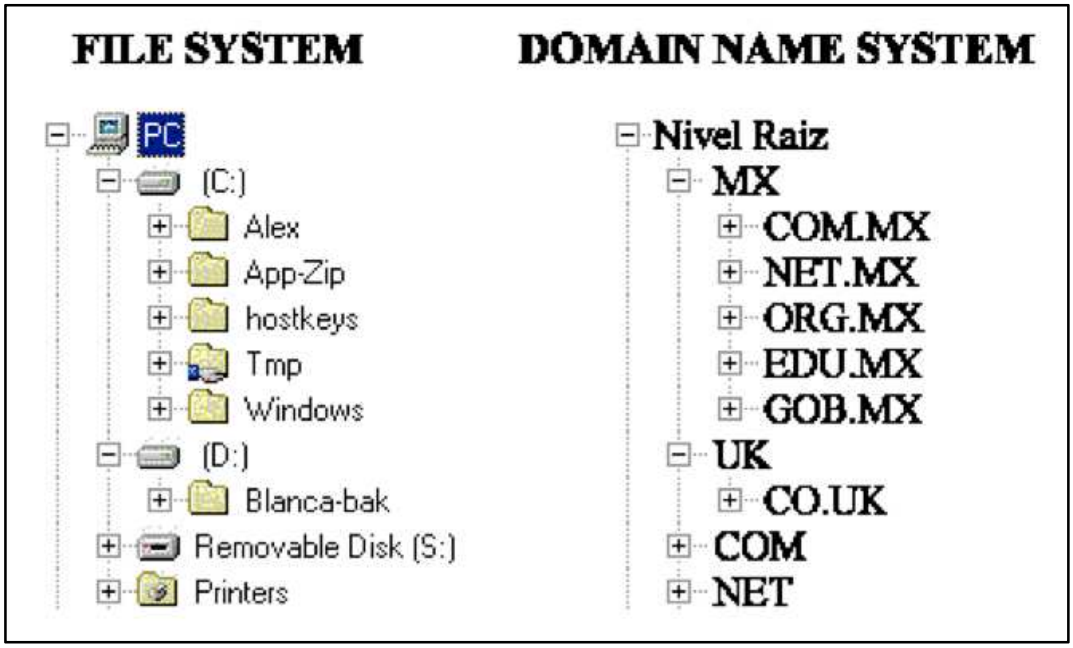


Ilustración 4-17 Comparación entre un sistema de archivos y el sistema de nombres de dominio (DNS)

Fuente: http://www.inegi.org.mx/inegi/contenidos/espanol/prensa/contenidos/articulos/tecnologia/dns03.pdf

Software Requerido

Daemontools es una herramienta para la gestión de los servicios Unix, supervisa y monitorea, además contiene una serie de paquetes como es el tinydns, djbdns, estos paquetes no pueden trabajar solos, tienen dependencia. Para la configuración djbdns utilizaremos las siguientes herramientas que son muy útiles para la configuración:

svscan: Realiza trabajos de seguimiento y asegura que sigan funcionando los servicios enlazados a directorios bajo /etc creados por programas–conf como dnscache-conf, tinydns-conf, etc.

svstat: Muestra el estado de un servicio el cual es monitoreado por **supervise.**

supervise: Es ejecutado por svscan para vigilar un servicio o demonio específico, y reiniciarlo de darse el caso de falla. (Bulma, 2007) - (Lifewithdjbdns) - (Cuquejo, 2005)

Descarga y configuración del software

Primero debemos crear un directorio llamado **/package**, le damos los permisos 1755 y adicionamos la opción –p ya que se está creando un directorio padre.

```
[root@server ~]# mkdir -p /package
[root@server ~]# chmod 1755 /package/
```

Ilustración 4-18 Sistema Operativo "Centos"

Realizar la descarga desde la página fuente, esto podemos hacerlo desde:

- ✓ http://cr.yp.to/daemontools/daemontools-0.76.tar.gz
- ✓ http://cr.yp.to/djbdns/djbdns-1.05.tar.gz

También podemos hacerlo directamente utilizando el comando **wget** como a continuación se indica.

Para el caso de Daemontools:

```
[root@server ~]# cd /package/
[root@server package]# wget http://cr.yp.to/daemontools/daemontools-0.76.tar.gz
--2011-04-30 23:54:57--  http://cr.yp.to/daemontools/daemontools-0.76.tar.gz
Resolviendo cr.yp.to... 80.101.159.118
Connecting to cr.yp.to|80.101.159.118|:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 36975 (36K) [application/x-gzip]
Saving to: `daemontools-0.76.tar.gz'

100%[======================================================>] 36.975      12,3K/s   in 2,9s

2011-04-30 23:55:05 (12,3 KB/s) - `daemontools-0.76.tar.gz' saved [36975/36975]
```

Ilustración 4-19 Sistema Operativo "Centos"

Para djbdns:

```
[root@server package]# wget http://cr.yp.to/djbdns/djbdns-1.05.tar.gz
--2011-04-30 23:56:37--  http://cr.yp.to/djbdns/djbdns-1.05.tar.gz
Resolviendo cr.yp.to... 80.101.159.118
Connecting to cr.yp.to|80.101.159.118|:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 85648 (84K) [application/x-gzip]
Saving to: `djbdns-1.05.tar.gz'

100%[======================================================>] 85.648      29,1K/s   in 2,9s

2011-04-30 23:56:43 (29,1 KB/s) - `djbdns-1.05.tar.gz' saved [85648/85648]
```

Ilustración 4-20 Sistema Operativo "Centos"

Dar los permisos correspondientes a los paquetes:

```
[root@server ~]# cd /package/
[root@server package]# ls
daemontools-0.76.tar.gz  djbdns-1.05.tar.gz
[root@server package]# chmod 777 daemontools-0.76.tar.gz
[root@server package]# chmod 777 djbdns-1.05.tar.gz
[root@server package]# ls
daemontools-0.76.tar.gz  djbdns-1.05.tar.gz
```

Ilustración 4-21 Sistema Operativo "Centos"

Descomprimir el paquete **daemon-tools** con el comando **gunzip** ya que tiene una extensión **.gz** y luego con el comando **.tar,** aquí se crea un árbol de directorios, donde se ejecutará la compilación de **daemon-tools.**

```
[root@server ~]# cd /package/
[root@server package]# ls
daemontools-0.76.tar.gz  djbdns-1.05.tar.gz
[root@server package]# chmod 777 daemontools-0.76.tar.gz
[root@server package]# chmod 777 djbdns-1.05.tar.gz
[root@server package]# ls
daemontools-0.76.tar.gz  djbdns-1.05.tar.gz
[root@server package]# gunzip daemontools-0.76.tar.gz
```

Ilustración 4-22 Sistema Operativo "Centos"

Ahora descomprimir el paquete con el comando **tar**

```
[root@server package]# tar -xvf daemontools-0.76.tar
admin
admin/daemontools-0.76
admin/daemontools-0.76/package
admin/daemontools-0.76/package/README
admin/daemontools-0.76/package/files
admin/daemontools-0.76/package/sharing
admin/daemontools-0.76/package/commands
admin/daemontools-0.76/package/install
admin/daemontools-0.76/package/compile
admin/daemontools-0.76/package/upgrade
admin/daemontools-0.76/package/run
admin/daemontools-0.76/package/run.inittab
admin/daemontools-0.76/package/boot.inittab
admin/daemontools-0.76/package/run.rclocal
admin/daemontools-0.76/package/boot.rclocal
admin/daemontools-0.76/src
admin/daemontools-0.76/src/CHANGES
admin/daemontools-0.76/src/TODO
admin/daemontools-0.76/src/Makefile
admin/daemontools-0.76/src/svscanboot.sh
admin/daemontools-0.76/src/svscan.c
admin/daemontools-0.76/src/supervise.c
```

Ilustración 4-23 Sistema Operativo "Centos"

Ingresar al directorio **admin** y luego al directorio **daemontools**

```
[root@server package]# ls
admin  daemontools-0.76.tar  djbdns-1.05.tar.gz
[root@server package]# cd admin/
[root@server admin]# ls
daemontools-0.76
[root@server admin]# cd daemontools-0.76/
```

Ilustración 4-24 Sistema Operativo "Centos"

En el directorio **src** podemos editar con el comando **vi error.h**

```
[root@server daemontools-0.76]# ls -l
total 16
drwxr-xr-x 2 root root 4096 jul 12  2001 package
drwxr-xr-x 2 root root 4096 jul 12  2001 src
[root@server daemontools-0.76]# cd src
[root@server src]# ls -l erro*
-rw-r--r-- 1 root root 1233 jul 12  2001 error.c
-rw-r--r-- 1 root root  595 jul 12  2001 error.h
-rw-r--r-- 1 root root 5502 jul 12  2001 error_str.c
[root@server src]# vi error.h
```

Ilustración 4-25 Sistema Operativo "Centos"

Aquí se reemplaza la línea **“extern int errno”** por **“#include <errno.h>”,** esto nos permitirá solucionar el problema de compilación de **daemontools,** sobre la última versión de la librería de compilación **glibc.**

Editando el archivo **error.h**

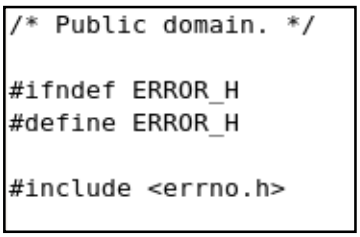


Ilustración 4-26 Sistema Operativo "Centos"

Dentro del directorio **daemontools** compilar e instalar con el ejecutable **install** dentro del directorio **/package**, el proceso se respalda en un archivo llamado **daemon.bk**

Ilustración 4-27 Sistema Operativo "Centos"

Una vez completada la instalación se creará un directorio llamado **/service,** y se mostrará el inicio de la utilidad svscan. Para comprobar utilizamos el comando **ps**, éste nos mostrará los puertos, adicionalmente utilizamos el carácter **“|”** que es el encargado de redireccionar la entrada al comando **grep**, con esto vamos a filtrar solo el servicio svscan. Digitamos: **ps –elf | grep svscan**

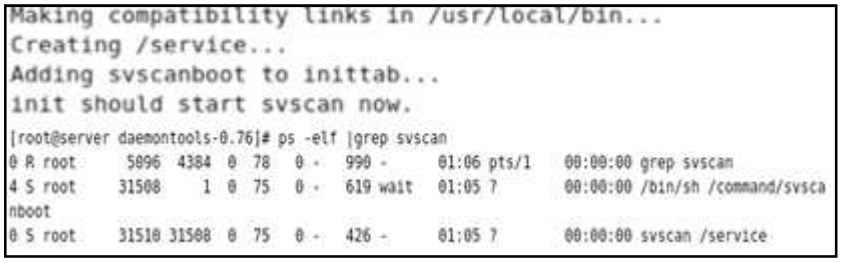


Ilustración 4-28 Sistema Operativo "Centos"

Compilamos e instalamos **dbjdns/tinydns**, para esto primero debemos descomprimir el paquete descargado con el comando **gunzip** y luego lo hacemos con el comando **tar,** respaldando este proceso en un archivo llamado **respa.txt**

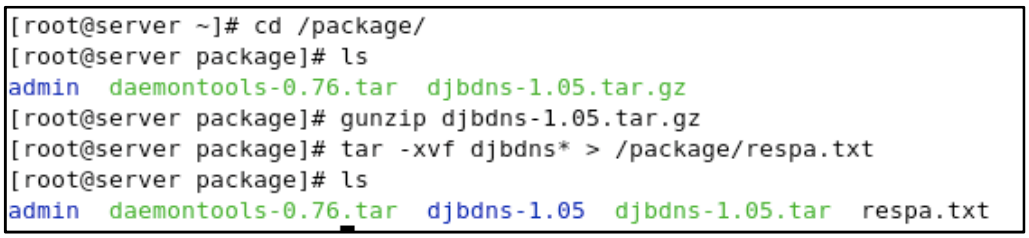


Ilustración 4-29 Sistema Operativo "Centos"

Ingresar al directorio **djbdns-1.05** y editar el archivo **error.h** con el comando vi, luego debemos reemplazar la línea **“extern int errno”** por **“#include <errno.h>”**, con esto tal como se hizo el daemontools se arreglará el problema sobre la última versión de la librería de la compilación **glibc**.

```
[root@server package]# cd djbdns-1.05
[root@server djbdns-1.05]# ls -l err*
-rw-r--r-- 1 root root 1149 feb 11  2001 error.c
-rw-r--r-- 1 root root  548 feb 11  2001 error.h
-rw-r--r-- 1 root root 5497 feb 11  2001 error_str.c

#ifndef ERROR_H
#define ERROR_H

#include <errno.h>

extern int error_intr;
extern int error_nomem;
extern int error_noent;
extern int error_txtbsy;
extern int error_io;
extern int error_exist;
extern int error_timeout;
```

Ilustración 4-30 Sistema Operativo "Centos"

Una vez solucionado el problema de compilación ya podemos compilar e instalar el servidor **djbdns**, para esto ejecutamos el comando **make** y, respaldamos en el archivo "**make.bk**" dentro del directorio **/home/proyecto/build/make.bk,** direccionando la salida de pantalla con el carácter **">".**

Para verificar la compilación utilizamos el comando **"make setup check"** y de igual manera respaldamos en el archivo**"madjsetchk.bk".**

```
[root@server djbdns-1.05]# make > /home/proyecto/build/made.bk

[root@server djbdns-1.05]# make setup check > /home/proyecto/build/madjsetch.bk
```

Ilustración 4-31 Sistema Operativo "Centos"

Configuración básica

Trabajando como administrador se añade una cuenta de usuario para ejecutar el servicio tinydns.

```
[root@server djbdns-1.05]# /usr/sbin/useradd -s /bin/false dnslog
[root@server djbdns-1.05]# /usr/sbin/useradd -s /bin/false tinydns
```

Ilustración 4-32 Sistema Operativo "Centos"

La creación de estos usuarios se puede verificar dentro del directorio **/home**

```
[root@server ~]# cd /home/
[root@server home]# ls
dnslog  lost+found  proyecto  tinydns
[root@server home]#
```

Ilustración 4-33 Sistema Operativo "Centos"

Al momento de compilar se creó una utilidad llamada "**tinydns-conf**", verificamos su existencia con **ls**

```
[root@server djbdns-1.05]# ls tiny*
tinydns       tinydns-conf.c  tinydns-data.c  tinydns-edit.c  tinydns-get.c
tinydns.c     tinydns-conf.o  tinydns-data.o  tinydns-edit.o  tinydns-get.o
tinydns-conf  tinydns-data    tinydns-edit    tinydns-get     tinydns.o
```

Ilustración 4-34 Sistema Operativo "Centos"

El archivo **tinydns-conf** es un ejecutable que ayuda a la configuración de la cuenta de usuario "**tinydns"** que se creó anteriormente para ejecutar el

servicio **tinydns**, y la cuenta de usuario "**dnslog**" que se creó para facilitar la ejecución de **DNS loggin**, adicionalmente crea el directorio **/etc/tinydns** y define que el servicio **tinydns** será el encargado de escuchar las peticiones en la dirección IP **192.168.0.109.**

```
[root@server package]# cd djbdns-1.05
[root@server djbdns-1.05]# tinydns-conf tinydns dnslog /etc/tinydns 192.168.0.109
```

Ilustración 4-35 Sistema Operativo "Centos"

Una vez ejecutada la línea de comandos verificamos en el directorio **/etc/tinydns**

```
[root@server home]# cd
[root@server ~]# cd /etc
[root@server etc]# cd tinydns/
[root@server tinydns]# ls
env  log  root  run
```

Ilustración 4-36 Sistema Operativo "Centos"

Construimos un enlace simbólico de **/etc/tinydns** al directorio **/service** para que la herramienta **svscan** de **daemontools** inicie el servicio tinydns y mantenga el monitoreo de los estados.

```
[root@server djbdns-1.05]# ln -s /etc/tinydns /service
```

Ilustración 4-37 Sistema Operativo "Centos"

Dentro del directorio **/etc/tinydns** podemos ver que se nos ha creado un directorio llamado **supervise**

```
[root@server ~]# cd /etc/
[root@server etc]# cd tinydns/
[root@server tinydns]# ls
env  log  root  run  supervise
```

Ilustración 4-38 Sistema Operativo "Centos"

Para comprobar que el servicio tinydns se está ejecutando utilizamos la herramienta **svstat.**

```
[root@server djbdns-1.05]# svstat /service/tinydns
/service/tinydns: up (pid 17597) 143 seconds
```

Ilustración 4-39 Sistema Operativo "Centos"

Ahora con la ayuda del comando **vi** debemos proveer la información IP y hostname de los computadores de red al demonio **djbdns/tinydns**, para esto primero crearemos la IP del servidor **djbdns/tinydns**, para luego indicar el nombre del host del servidor con su respectiva IP, y por último las IP de los demás elementos de la red, esto lo realizaremos dentro del directorio **/service/tinydns/root**

```
.:192.168.0.109:a:259200
@server.eztesis.espe:192.168.0.109:86400:in
~
```

Ilustración 4-40 Sistema Operativo "Centos"

Estos cambios se guardarán en el archivo **data**.

Dentro del mismo directorio ejecutamos el comando **make** para realizar la compilación de **service/tinydns/root/data** a **service/tinydns/root/data.cdb**, el mismo que es un programa usado para resolver las peticiones hostname-IP.

```
[root@server root]# make
/usr/local/bin/tinydns-data
[root@server root]# ls
add-alias  add-childns  add-host  add-mx  add-ns  data  data.cdb  Makefile
```

Ilustración 4-41 Sistema Operativo "Centos"

Por último configuramos el dns-cache que es el encargado de ayudar a resolver peticiones de manera más segura que una configuración BIND, de clientes locales como web browser o MTA.

Creamos una cuenta de usuario llamada **dnscache.**

```
[root@server root]# useradd dnscache
```

Ilustración 4-42 Sistema Operativo "Centos

Verificamos su existencia en el directorio **/home**

```
[root@server ~]# cd /home/
[root@server home]# ls
dnscache  dnslog  lost+found  proyecto  tinydns
```

Ilustración 4-43 Sistema Operativo "Centos"

Con la ayuda de la herramienta ejecutable creada en la compilación de djbdns, **dnscache-conf** creamos el directorio **/etc/dnscache** con sus respectivos

subdirectorios dependientes y configuramos la IP a la que debería escuchar, por defecto los logs del archivo se copian en **/etc/dnscache/log/main.**

```
[root@server root]# dnscache-conf dnscache dnslog /etc/dnscache 192.168.0.109
```

Ilustración 4-44 Sistema Operativo "Centos"

Verificamos dentro del directorio **/etc/dnscache**

```
[root@server ~]# cd /etc/
[root@server etc]# cd dnscache/
[root@server dnscache]# ls
env  log  root  run  seed
```

Ilustración 4-45 Sistema Operativo "Centos"

De la misma forma a **tinydns** se le comunica del inicio de **dnscache,** creando un enlace simbólico a **/etc**.

```
[root@server root]# ln -s /etc/dnscache /service/
```

Ilustración 4-46 Sistema Operativo "Centos"

Podemos observar que dentro de **/etc/dnscache** se ha creado el directorio **supervise**

```
[root@server ~]# cd /etc/
[root@server etc]# cd dnscache/
[root@server dnscache]# ls
env  log  root  run  seed  supervise
```

Ilustración 4-47 Sistema Operativo "Centos"

Con esto hemos logrado que dnscache solo acepte peticiones locales, si queremos que escuche peticiones de otros host podemos ejecutar lo siguiente:

```
[root@server root]# touch /etc/dnscache/root/ip/192.168.0
```

Ilustración 4-48 Sistema Operativo "Centos"

Con esto podemos añadir o quitar redes sin que sea afectada la ejecución de dnscache.

Observamos la configuración dentro del directorio **/etc/dnscache/root/ip**

```
[root@server ~]# cd /etc/
[root@server etc]# cd dnscache/
[root@server dnscache]# ls
env  log  root  run  seed  supervise
[root@server dnscache]# cd root/
[root@server root]# ls
ip  servers
[root@server root]# cd ip
[root@server ip]# ls
127.0.0.1  192.168.0
```

Ilustración 4-49 Sistema Operativo "Centos"

Para verificar la efectividad de la configuración utilizamos el comando **netstat** o **lsof**, que permiten comprobar los servicios de red que se están ejecutando, pero filtrando solo a servicios de dominio.

Utilizando **netstat**:

```
[root@server root]# netstat -tulpa |grep domain
tcp        0      0 localhost.localdomain:2208  *:*                         LISTEN      2939/hpio
d
tcp        0      0 eztesis.espe:domain         *:*                         LISTEN      17661/dns
cache
tcp        0      0 192.168.122.1:domain        *:*                         LISTEN      3165/dnsm
asq
tcp        0      0 localhost.localdomain:ipp   *:*                         LISTEN      2966/cups
d
tcp        0      0 localhost.localdomain:smtp  *:*                         LISTEN      2988/send
mail: acce
tcp        0      0 localhost.localdomain:2207  *:*                         LISTEN      2944/pyth
on
udp        0      0 eztesis.espe:domain         *:*                                     17661/dns
cache
udp        0      0 eztesis.espe:domain         *:*                                     17597/tin
ydns
udp        0      0 192.168.122.1:domain        *:*                                     3165/dnsm
asq
```

Ilustración 4-50 Sistema Operativo "Centos"

Utilizando **losf**:

```
[root@server root]# lsof -i |grep domain
hpiod      2939     root    0u  IPv4  11931       TCP localhost.localdomain:2208 (LISTEN)
python     2944     root    4u  IPv4  11968       TCP localhost.localdomain:2207 (LISTEN)
cupsd      2966     root    4u  IPv4  12046       TCP localhost.localdomain:ipp (LISTEN)
sendmail   2988     root    4u  IPv4  12145       TCP localhost.localdomain:smtp (LISTEN)
dnsmasq    3165   nobody    6u  IPv4  13063       TCP 192.168.122.1:domain (LISTEN)
dnsmasq    3165   nobody    7u  IPv4  13064       UDP 192.168.122.1:domain
tinydns   17597  tinydns    3u  IPv4  77095       UDP eztesis.espe:domain
dnscache  17661 dnscache    3u  IPv4  80609       UDP eztesis.espe:domain
dnscache  17661 dnscache    4u  IPv4  80610       TCP eztesis.espe:domain (LISTEN)
```

Ilustración 4-51 Sistema Operativo "Centos"

Con esto hemos concluido la configuración del servidor DNS, si el cliente es Linux se debe editar con el comando **vi** el archivo **/etc/resolv.conf**, agregando lo siguiente:

searcheztesis.espe

name server 192.168.0.109

Para clientes Windows debemos ingresar a Conexiones de Red, clic derecho en Propiedades del Protocolo TCP/IP e ingresamos la dirección del servidor DNS.

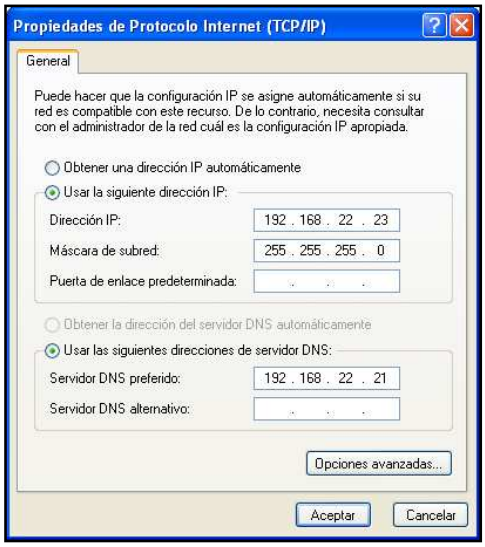


Ilustración 4-52 Sistema operativo Windows "cliente"

4.2 CONFIGURACIÓN E IMPLEMENTACIÓN DEL SERVIDOR WEB APACHE

4.2.1 INTRODUCCIÓN

Un servidor web o servidor HTTP es un programa que procesa cualquier aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI. El

término también se emplea para referirse al ordenador que ejecuta el programa

4.2.2 ARQUITECTURA

PETICIÓN GET

Un servidor web opera mediante el protocolo HTTP, de la capa de aplicación del Modelo OSI. Al protocolo HTTP se le asigna habitualmente el puerto TCP 80. Las peticiones al servidor suelen realizarse mediante HTTP utilizando el método de petición GET en el que el recurso se solicita a través de la url al servidor web.

GET /index.html HTTP/1.1 HOST: www.host.com

Esquema de una petición GET

1.- El navegador por medio de la interfaz de usuario permite al usuario realizar una o varias peticiones web.
2.- El usuario o el navegador realiza la petición web.
3.- Se produce un socket con un servidor dado en dirección IP mediante TCP.
4.- Si la dirección dada es DNS y no existe una regla en la base de datos DNS, el Host Resolver Request solicita al servidor DNS la o las direcciones IPs correspondientes. El navegador crea una nueva regla y almacena la dirección IP junto a la dirección DNS.
5.- Una vez almacenada la regla se realiza una petición a la base de datos DNS para recuperar los valores de la regla.
6.- Se produce un socket con la dirección IP mediante TCP.

7.- Se crea la petición GET estableciendo la url, un flag, la priority de la petición y el method “implícitamente GET”.

8.- Se abre y/o se crea una entrada en el http cache

9.- Se realiza la petición GET. Se leen las cabeceras HTTP de la http transaction y más tarde el cuerpo de la http transaction.

10.- Se consulta en el caché de disco si existe una entrada en el caché asociada al recurso que se ha solicitado. Los valores son created “true o false” y key “la url del recurso”.

11.- Si la entrada no existe “si el valor de created es false” se escriben los datos en el caché de disco. Si no, se lee directamente.

12.- Se concluye la operación y se muestra en pantalla “si es preciso” la información.

PETICIÓN POST

Es el segundo tipo de petición HTTP más utilizado. Los datos a enviar al servidor se incluyen en el cuerpo de la misma petición con las cabeceras HTTP asignadas correspondientemente respecto al tipo de petición. Generalmente se asocia con los formularios web en el que los datos suelen ser cifrados para enviarlos de manera segura al servidor.

Funcionamiento

El Servidor web se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente “un navegador web” y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error. A modo de ejemplo, si tecleamos www.wikipedia.org en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor

responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma. (Cibernetia)

Además de la transferencia de código HTML, los Servidores web pueden entregar aplicaciones web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- ✓ Aplicaciones en el lado del cliente, y
- ✓ Aplicaciones en el lado del servidor

Aplicaciones en el lado del cliente: El cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java "applets" o Javascript, el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones “también llamadas scripts”. Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript y java, aunque pueden añadirse más lenguajes mediante el uso de plugins.

Aplicaciones en el lado del servidor: El servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor muchas veces suelen ser la mejor opción para realizar aplicaciones web. La razón es que, al ejecutarse ésta en el servidor y

no en la máquina del cliente, éste no necesita ninguna capacidad añadida, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones.

El hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un lenguaje de marcas y HTTP es un "protocolo".

4.2.3 DESCRIPCION DEL SOFTWARE

El paquete que utilizaremos para la implementación del servidor web será **httpd** junto con **httpd-devel, php, php-common**, Para verificar su estado podemos utilizar las herramientas **service** o **chkconfig**, los log del sistema se los puede encontrar en el archivo **/var/log/httpd/acces_log** o bajo el mismo directorio en **error_log**, en éste último podemos encontrar información útil para el administrador que permitirá el mantenimiento y administración del servidor.

4.2.4 ARCHIVO DE CONFIGURACIÓN

El archivo de configuración se encuentra en el directorio **/etc/httpd/conf/httpd.conf** y es administrado por el usuario apache, este archivo pertenece al grupo apache, razón por la cual puede ser editado por el administrador root del sistema.
Abrimos una consola y digitamos lo anteriormente mencionado:

```
[root@server httpd]# cd
[root@server ~]# cd /etc
[root@server etc]# cd httpd
[root@server httpd]# cd conf
[root@server conf]# ls -l http*
-rw-r--r-- 1 root root 33726 abr  4  2010 httpd.conf
[root@server conf]# vi httpd.conf
```

Ilustración 4-53 Sistema Operativo "Centos"

Podemos observar el modelo del archivo httpd.conf

```
<Directory "/var/www/html">

</Directory>

<Files ~ "^\.ht">

</Files>

<VirtualHost *:80>

</VirtualHost>
```

Ilustración 4-54 Modelo del archivo httpd.conf

4.2.5 CONFIGURACIÓN BÁSICA

Con el comando **vi** editamos el archivo **httpd.conf** que está dentro del directorio **/etc/httpd/conf**

```
[root@server ~]# cd /etc/
[root@server etc]# cd httpd
[root@server httpd]# cd conf
[root@server conf]# vi httpd.conf

Listen 80
ServerAdmin root@server.eztesis.espe
ServerName server.eztesis.espe:80
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/var/www/html"
```

Ilustración 4-55 Sistema Operativo "Centos"

Las directivas que encontraremos son:

Listen 80

La directiva Listen le indica al servidor que acepte peticiones entrantes solamente en los puertos y en las combinaciones de puertos y direcciones que se especifiquen. Si solo se especifica un número de puerto en la directiva Listen el servidor escuchará en ese puerto, en todas las interfaces de red de la máquina. Si se especifica una dirección IP y un puerto, el servidor escuchará solamente en la interfaz de red a la que pertenezca esa dirección IP y solamente en el puerto indicado. Se pueden usar varias directivas Listen para especificar varias direcciones IP y puertos de escucha. El servidor responderá a las peticiones de todas las direcciones y puertos que se incluyan. (Apache, 2011)

ServerAdmin root@server.eztesis.espe

Especifica la dirección de correo del administrador, es aquí donde se recibirán los reportes e informes de errores. Esto también podemos observar que aparece cuando solicitamos ciertas páginas erróneas.

ServerName server.eztesis.espe:80

SeverName especifica el nombre y el puerto que el servidor utiliza para escuchar las peticiones. Con una correcta configuración, este valor se determina en forma automática, pero es recomendable especificarlo explícitamente para evitar problemas durante el arranque. Si no se pone un nombre DNS válido no funcionarán las redirecciones generadas por el

servidor. De no existir un nombre DNS registrado entonces deberemos colocar la dirección IP.

DocumentRoot "/var/www/html”

DocumentRoot indica el directorio donde se almacenan los documentos web. Existe la posibilidad de utilizar enlaces simbólicos dentro del DocumentRoot.

Existen diversas opciones tales como ingresar a un directorio del servidor, autenticación http, host virtuales, etc, las que se configuran según los requerimientos y criterios del administrador. (Bdat)

Y por último se inicia el servicio httpd como se indica a continuación.

```
[root@server conf]# service httpd start
Iniciando httpd:                                           [  OK  ]
```

Ilustración 4-56 Sistema Operativo "Centos"

4.3 CONFIGURACIÓN E IMPLEMENTACIÓN DEL SERVIDOR DE CORREO ELECTRÓNICO

4.3.1 DEFINICIÓN

El servicio de correo electrónico es un servicio de red que permite a los usuarios enviar y recibir mensajes rápidamente “también denominados mensajes electrónicos o cartas electrónicas” mediante sistemas de comunicación electrónicos. Principalmente se usa este nombre para denominar al sistema que provee este servicio en Internet, mediante el

protocolo SMTP, aunque por extensión también puede verse aplicado a sistemas análogos que usen otras tecnologías. Por medio de mensajes de correo electrónico se puede enviar, no solamente texto, sino todo tipo de documentos digitales. Su eficiencia, conveniencia y bajo coste están logrando que el correo electrónico desplace al correo ordinario para muchos usos habituales. (Via server center, 2006)

4.3.2 FUNCIONAMIENTO DE UN SERVIDOR DE CORREO

Cuando un usuario de correo de su dominio envía un correo, primero llega a su servidor de correo que luego él lo envía al servidor destinatario, donde el mensaje queda almacenado en el buzón del destinatario. Cuando el destinatario se conecte al servidor, este le enviará todos sus mensajes pendientes. (Via server center, 2006)

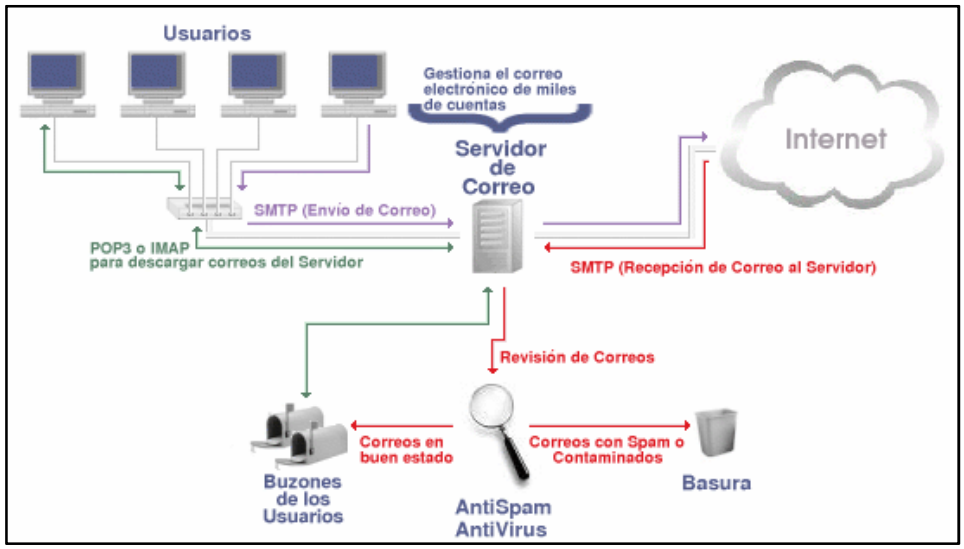


Ilustración 4-57 Esquema del servidor de correo

Fuente: http://viaservercenter.com/index.php?option=com_content&task=view&id=13&Itemid=28

4.3.3 CARACTERÍSTICAS

- ✓ Es flexible y rápido.
- ✓ Es configurable.
- ✓ Independencia total de servidores externos poco fiables.
- ✓ Creación ilimitada de cuentas de correo electrónico.
- ✓ Creación ilimitada de dominios en el mismo servidor.
- ✓ Servidor SMTP propio.
- ✓ Se puede enviar correo a través de su servidor desde cualquier parte del mundo por el webmail o un cliente de correo.
- ✓ Cantidad de espacio para cada buzón es configurable.
- ✓ Tanto para Sendmail, Postfix como para Qmail existen módulos adicionales para soporte de listas de correo, Antivirus, Anti Spam, Interface Webmail y otros. (Via server center, 2006)

Los servidores de correo a menudo realizan diferentes funciones según sea el uso que se planifique para el mismo.

Para su adecuado funcionamiento tenemos varios componentes, entre ellos:

- ✓ **MTA"Mail Transport Agent"**

Se encarga de enviar/recibir los correos desde un servidor de e-mail hacia/desde Internet, que implementa el protocolo **SMTP** "**Simple Mail Transfer Protocol"**

- ✓ **MUA "Mail User Agent"**

Un agente de usuario de correo (MUA) es sinónimo con una aplicación

cliente de correo. Un MUA es un programa que, al menos, les permite a los usuarios leer y redactar mensajes de correo.

Muchos MUAs son capaces de recuperar mensajes a través de los protocolos POP o IMAP, configurando los buzones de correo para almacenar mensajes y enviando los mensajes salientes a un MTA.

Los MUAs pueden ser de interfaz gráfica tal como **Evolution** o tener una interfaz basada en texto muy sencilla tal como **mutt**. (Via server center, 2006)

Para la recuperación de correo, se puede utilizar los protocolos POP3 o IMAP el que permite la utilización de webmails o ambos.

4.3.4 SOFTWARE UTILIZADO

El proceso de instalación de Sendmail requiere de los siguientes paquetes.

- ✓ Sendmail
- ✓ Imap
- ✓ Pop3
- ✓ Make
- ✓ Cyrus-sasl
- ✓ Cyrus-sasl-md5
- ✓ Cyrus-sasl-plain (Via server center, 2006)

Sendmail en la mayoría de las distribuciones Linux viene instalado en la configuración básica, en las distribuciones basadas en Red Hat podemos comprobar su instalación utilizando el comando **rpm**.

```
[root@server ~]# rpm -qa |grep sendmail*
sendmail-cf-8.13.8-8.el5
sendmail-8.13.8-8.el5
```

Ilustración 4-58 Sistema Operativo "Centos"

Es muy importante que el paquete sendmail-cf esté instalado, caso contrario no se podrá compilar los archivos necesarios para configurar Sendmail. Al igual que como verificamos con los servicios podemos utilizar la herramienta **chkconfig** y **service** para comprobar su ejecución.

```
[root@server ~]# service sendmail status
Se está ejecutando sendmail (pid  2904)...
```

Ilustración 4-59 Sistema Operativo "Centos"

Además podemos utilizar los servidores dovecot, cyrus o imap para realizar la recuperación de correo.

En la versión de Centos que estamos utilizando ya viene instalado por defecto dovecot, similar al caso de sendmail utilizamos **chkconfig** para su inicio, y **service** para su comprobación.

```
[root@server ~]# chkconfig --level 345 dovecot on
[root@server ~]# service dovecot status
dovecot está parado
[root@server ~]# service dovecot start
Iniciando Dovecot Imap:                                    [  OK  ]
[root@server ~]# service dovecot status
Se está ejecutando dovecot (pid  18502)...
```

Ilustración 4-60 Sistema Operativo "Centos"

4.3.5 CONFIGURACIÓN DE SENDMAIL

Antes de empezar la configuración debemos comprobar que en la configuración de DNS, exista al menos un Mail Exchanger en la zona definida como de reenvío. (Duarte)

Dentro del directorio **/service/tinydns/root,** editamos el archivo **data** con el comando **vi** para añadir lo siguiente:

```
.:192.168.0.109:a:259200
@server.eztesis.espe:192.168.0.109:86400:in
```

Ilustración 4-61 Sistema Operativo "Centos"

Ahora editamos el fichero **/etc/mail/local-host-names,** aquí debemos ingresar todos y cada uno de los alias que tenga el servidor que estamos configurando, así como sus posibles subdominios, para esto vamos a añadir todos los dominios para los cuales se recibirá correo en determinado momento.

```
[root@server root]# cd
[root@server ~]# cd /etc
[root@server etc]# cd mail
[root@server mail]# vi local-host-names

# local-host-names - include all aliases for your machine here.
server.eztesis.espe
eztesis.espe
```

Ilustración 4-62 Sistema Operativo "Centos"

Debemos sacar un respaldo del archivo sendmail.mc que se encuentra dentro del directorio **/etc/mail,** y editamos el original.

```
[root@server ~]# cd /etc
[root@server etc]# cd mail
[root@server mail]# ls -l sendm*
-rw-r--r-- 1 root root 58290 mar 30  2010 sendmail.cf
-rw-r--r-- 1 root root  7205 mar 30  2010 sendmail.mc
[root@server mail]# cp sendmail.mc /home/proyecto/build/
[root@server mail]# vi sendmail.mc
```

Ilustración 4-63 Sistema Operativo "Centos"

Es importante tener en cuenta que dentro de la configuración de sendmail la instrucción **dnl** inicia un comentario.

Sendmail solo nos permitirá enviar correo desde el mismo servidor o lo que es lo mismo desde la interfaz loopback (127.0.0.1). Si queremos enviar correos desde los host de la red local debemos comentar la siguiente línea:

DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl

```
dnl #
#DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
dnl #
```

Ilustración 4-64 Sistema Operativo "Centos"

También podemos añadir las interfaces desde las cuales se quiere que sendmail escuche peticiones, u omita las que no debe.

Para iniciar el rechazo al Spam debemos bloquear el correo que proviene de dominios que no estén registrados en el DNS o que no están resueltos. Para este propósito a menos que se requiera todo lo contrario, es necesario mantener comentada la siguiente línea

```
dnl #
#FEATURE(`accept_unresolvable_domains')dnl
dnl #
```

Ilustración 4-65 Sistema Operativo "Centos"

Además debemos definir la mascará que se va utilizar para el envío de correo desde el servidor y guardamos los cambios.

```
dnl #
dnl MASQUERADE_AS(`eztesis.espe')dnl
dnl #
```

Ilustración 4-66 Sistema Operativo "Centos"

Para actualizar el archivo **/etc/mail/sendmail.cf** a partir del archivo sendmail.mc utilizamos la herramienta **m4**

```
[root@server mail]# m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

Ilustración 4-67 Sistema Operativo "Centos"

Establecemos los dominios a los cuales queremos permitir enviar correos electrónicos, para lo cual creamos el archivo **relay-domains** dentro del directorio **/etc/mail/**

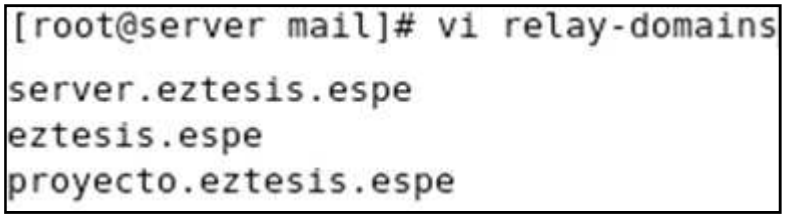


Ilustración 4-68 Sistema Operativo "Centos"

Cada uno de los equipos debe tener el permiso para enviar mensajes hacia el exterior, es decir el MTA deberá aceptar los mensajes y redirigirlos hacia el internet.

Sendmail debe ser **relay** de nuestras estaciones pero no de otros computadores. Por defecto Sendmail no permite el relay a nadie, pero podemos activarlo para nuestras estaciones mediante el archivo "**access**".

Dentro del directorio **/etc/mail/access** definimos quienes pueden hacer uso del servidor de correo para enviar mensajes usando la directiva "relay", u opcionalmente podemos establecer que IP's o hostname no están permitidas enviar correo bajo la directiva "Reject".

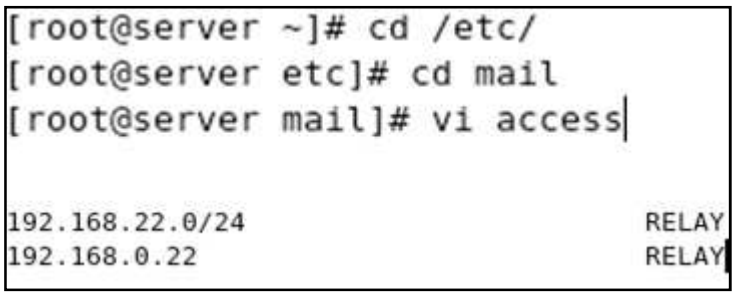


Ilustración 4-69 Sistema Operativo "Centos"

Sendmail utiliza una versión compilada del archivo Access para un rápido acceso, éste se encuentra almacenado en el directorio **/etc/mail/access.db**, la cual debe ser restablecida mediante el comando **make**.

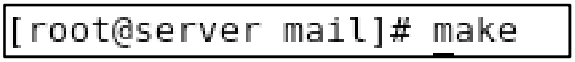


Ilustración 4-70 Sistema Operativo "Centos"

Opcionalmente podemos designar un alias a la cuenta de root, con esto podremos recibir mensajes generados por el sistema en una cuenta común de usuario.

Dentro del directorio **/etc** editamos el archivo **aliases** con el comando **vi** y visualizaremos el siguiente mensaje:

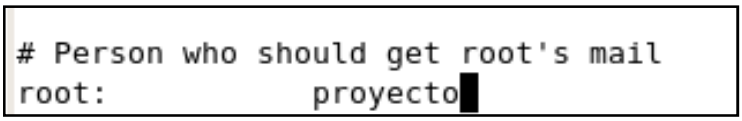


Ilustración 4-71 Sistema Operativo "Centos"

Ejecutamos el comando **newaliases** para actualizar los cambios.

Ilustración 4-72 Sistema Operativo "Centos"

Por último iniciamos el servicio Sendmail

```
[root@server mail]# service sendmail start
```

Ilustración 4-73 Sistema Operativo "Centos"

Para poder utilizar el servidor de correo electrónico debemos estar autenticados en el servidor.

Para esto añadimos un usuario con su respectivo password:

```
[root@server mail]# useradd proyecto
useradd: el usuario proyecto existe
[root@server mail]# useradd liveya
[root@server mail]# passwd  liveya
Changing password for user liveya.
New UNIX password:
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

Ilustración 4-74 Sistema Operativo "Centos"

4.3.6 CONFIGURACIÓN DE MAIL USER AGENT

MUA es el programa que interactúa directamente con el usuario para el envió y recepción de correo. Para la configuración de MUA utilizaremos Outlook Express que es una aplicación de Windows.

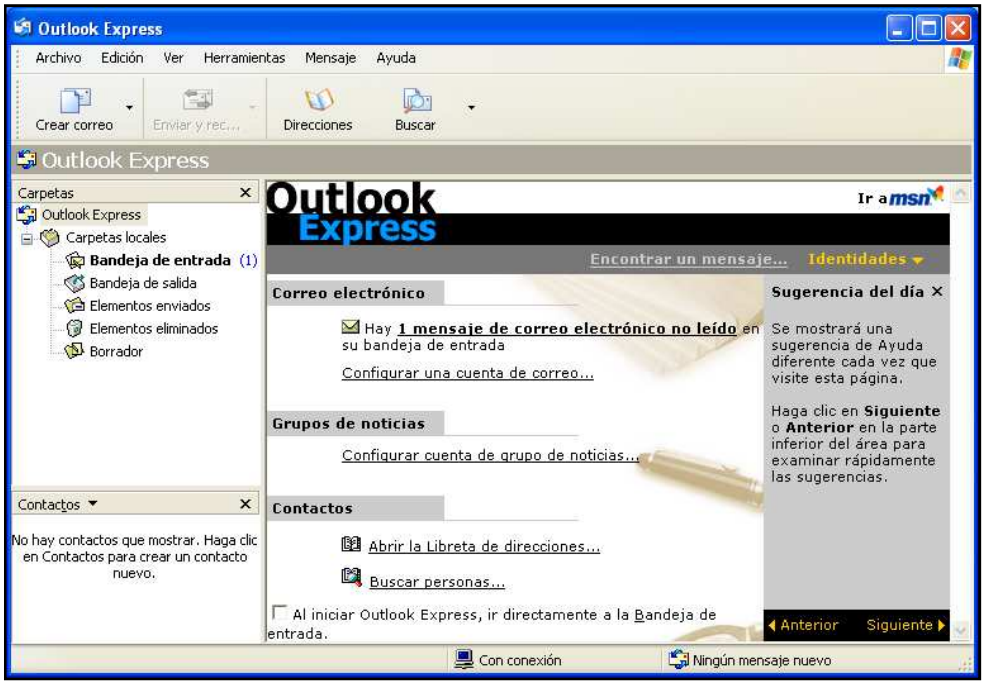


Ilustración 4-75 Sistema operativo "Windows"

En la pestaña **Herramientas > Cuentas** seleccionamos la opción **Agregar cuenta correo**

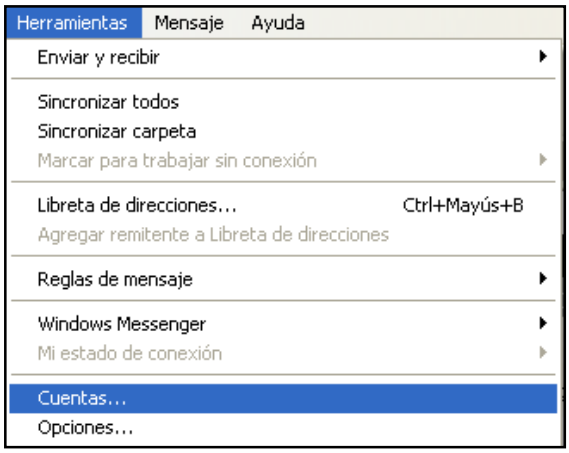


Ilustración 4-76 Sistema operativo "Windows"

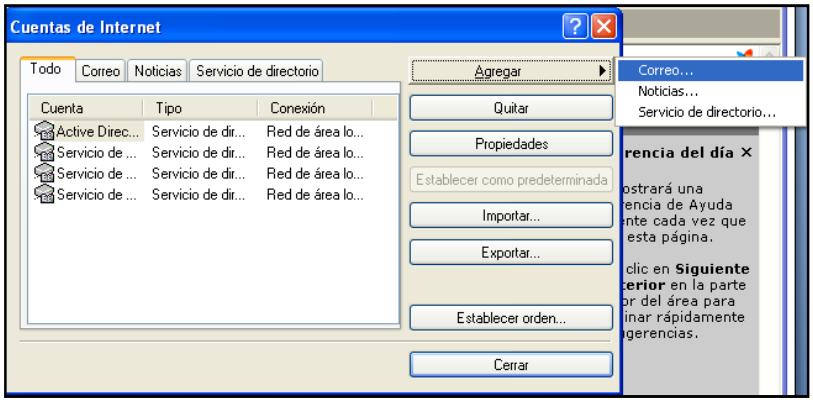


Ilustración 4-77 Sistema operativo "Windows"

Llenamos el campo del nombre del usuario que queremos mostrar. Clic en **siguiente**

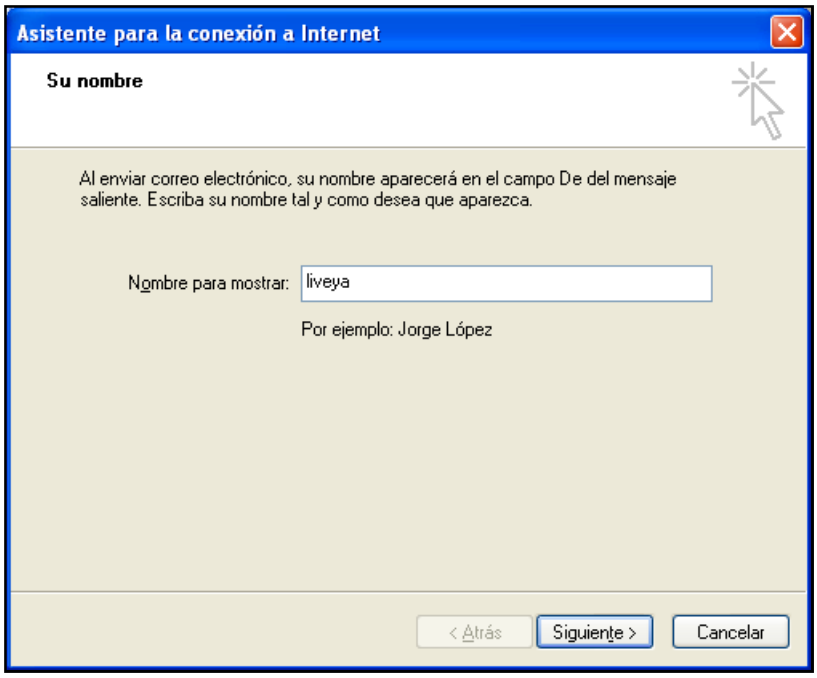


Ilustración 4-78 Sistema operativo "Windows"

Ingresar la cuenta de correo electrónico, clic en **siguiente**

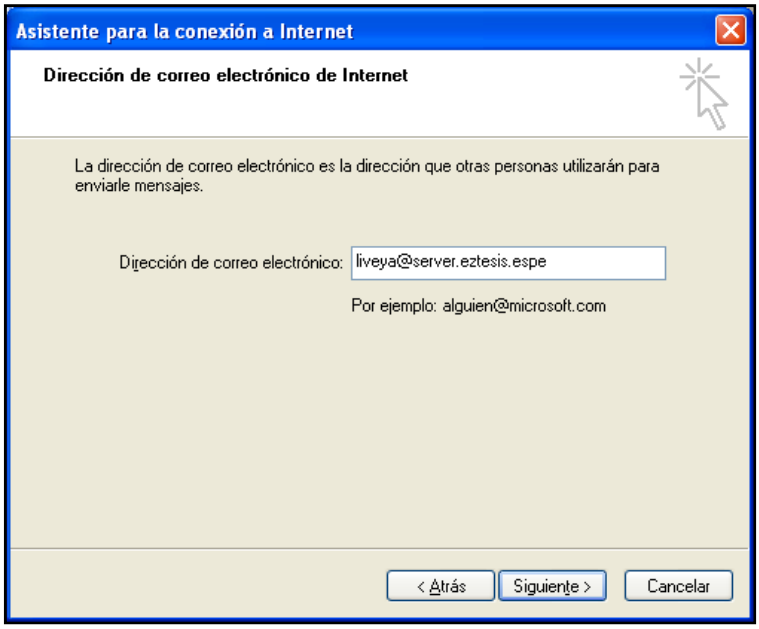


Ilustración 4-79 Sistema operativo "Windows"

Seleccionar el tipo de servicio de correo entrante, este será IMAP “puerto 143”, y el servicio de correo saliente será SMTP “puerto 125”, estos dos servicios son proporcionados por **server.eztesis.espe**, clic en **siguiente**

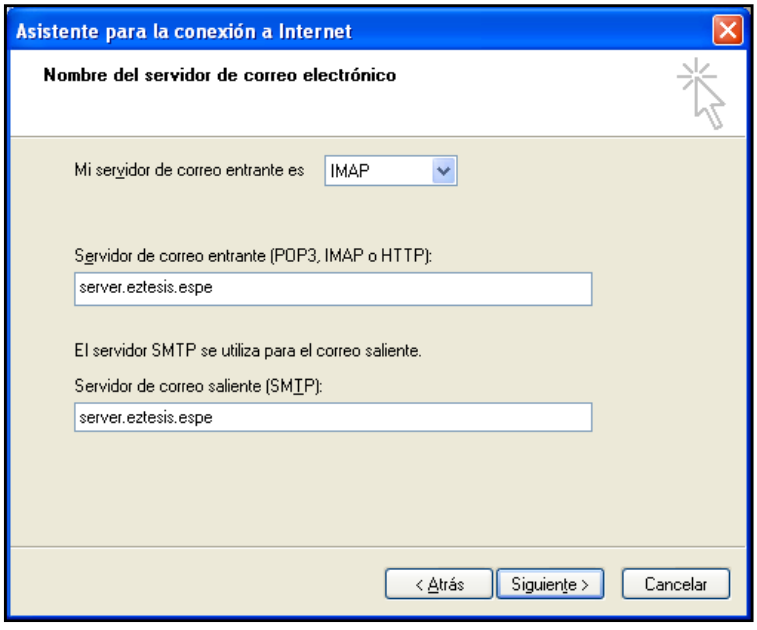


Ilustración 4-80 Sistema operativo "Windows"

Por último el usuario deberá hacer login en la cuenta, con el usuario y password que el administrador le asignó previamente

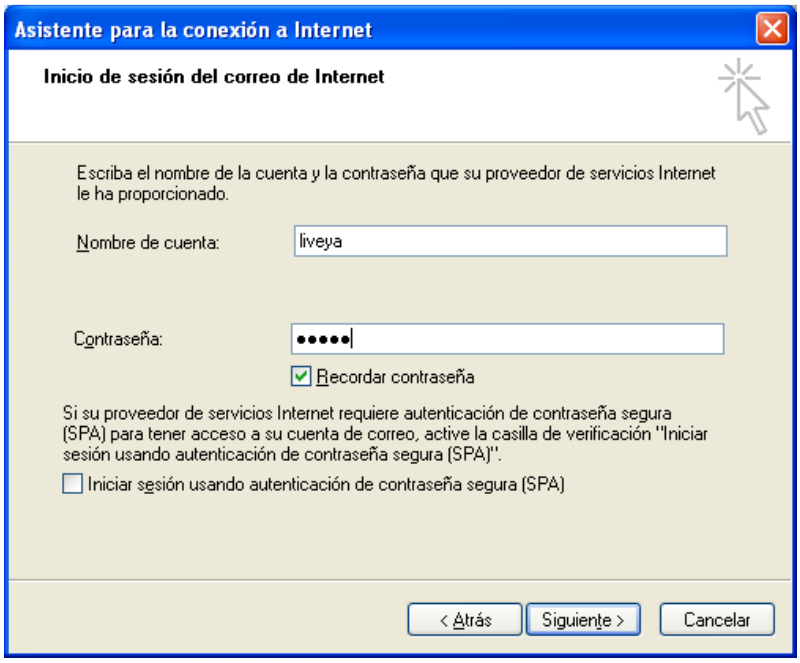


Ilustración 4-81 Sistema operativo "Windows"

Con esto el usuario ya puede enviar y recibir correo electrónico

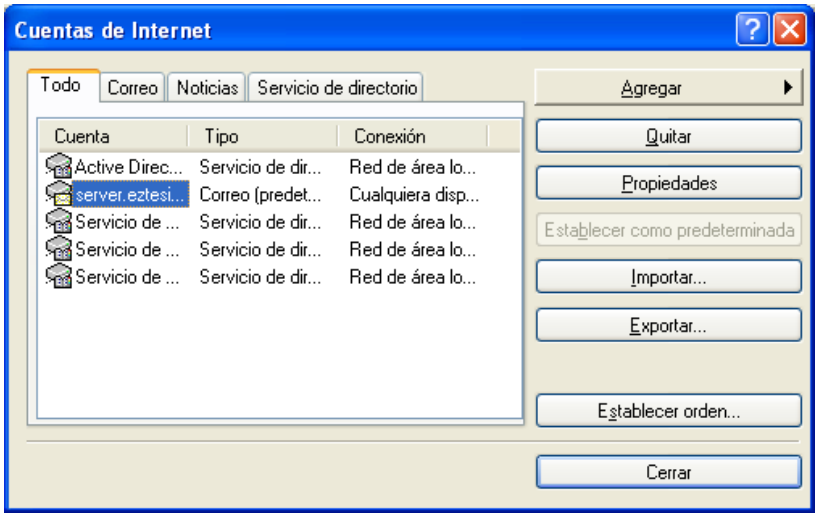


Ilustración 4-82 Sistema operativo "Windows"

4.3.7 CONTROL DE CORREO NO DESEADO

SPAMASSASSIN es un programa de análisis y filtraje de correos electrónicos que sobre la base de más de 100 test, trata de descubrir si un determinado correo es SPAM antes de que llegue a su buzón de mensajes.

SpamAssassin corre a nivel de servidor. Cada mensaje que llega al servidor, viene analizado en busca de características comunes al SPAM. Busca por ejemplo, palabras como "Mortgage", "Viagra" o "PenisEnlargement" y si las encuentra, los marcará como SPAM.

Cada vez que el puntaje sea más alto, el correo tiene más probabilidad de ser SPAM. A puntaje bajo, corresponde casi siempre un mensaje de correo que no es SPAM. Todo este proceso es añadido al mismo mensaje, de modo que el usuario pueda darse cuenta del por qué un determinado mensaje ha sido identificado como SPAM. Normalmente marca el mensaje con la cadena ** SPAM ** o [SPAM] en el asunto y explicando en el cuerpo porque el correo recibido tiene aspecto de spam. El usuario decidirá qué hacer con esos correos. Por ejemplo puede decidir borrarlos directamente a nivel de servidor sin ni siquiera enterarse de su existencia. En detalle, SpamAssassin toma el mensaje de correo (cabeceras y cuerpo) y busca determinados patrones. Por cada patrón que encuentra suma una determinada cantidad de puntos. Cuando los puntos superan un umbral fijado por el usuario, que se define REQUIRED POINTS, “por defecto es 5” el correo se marca como spam. (Ole-web, 2005)

Primero debemos instalar el paquete spamssassin, y configurar el archivo que se encuentra en el directorio **/etc/mail/spamssassin/local.cf**, también se pueden añadir las direcciones IP’s consideradas seguras.

```
# These values can be overridden by editing ~/.spamassassin/user_prefs.cf
# (see spamassassin(1) for details)

# These should be safe assumptions and allow for simple visual sifting
# without risking lost emails.

required_hits 5
report_safe 0
rewrite_header Subject [SPAM]

#redes confiables
trusted_network 192.168.0.0/24
#confirmación de las IPs
trusted_network 192.168.0.0/24
```

Ilustración 4-83 Sistema Operativo Centos5 “Servidor”

Una vez realizados los cambios debemos guardarlos, y para que estos tengan efecto debemos reiniciar el servicio spamassassin.

```
[root@server spamassassin]# service spamassassin restart
Parando spamd:                                             [  OK  ]
Iniciando spamd:                                           [  OK  ]
```

Ilustración 4-84 Sistema Operativo Centos5 “Servidor”

4.3.8 CONTROL DE ANTIVIRUS, UTILIZANDO CLAMAV

El objetivo primario de ClamAV es la consecución de un conjunto de herramientas que identifiquen y bloqueen el malware proveniente del correo electrónico. Uno de los puntos fundamentales en este tipo de software es la rápida localización e inclusión en la herramienta de los nuevos virus encontrados y escaneados. El paquete provee un demonio administrador flexible, el cual se puede actualizar automáticamente vía internet. (Clamav, 2002) (Wikipedia, 2011)

Su instalación puede ser desde la fuente que es el más aconsejable, pues no hay que preocuparse de las dependencias, teniendo presente los siguientes paquetes. (Guateweriless) - (Encuentro Alternativo)

- ✓ Clamav: Software antivirus.
- ✓ Clamd: Demonio administrador del servicio.
- ✓ Clamav-milter: paquete de integración con MTA (Sendmail).

Descargamos la última versión estable que es **clamav-0.97**, y vamos a descomprimirla en el directorio **/usr/local/src**

```
[root@server ~]# ls
anaconda-ks.cfg  clamav-0.97.tar  Desktop  install.log  install.log.syslog
[root@server ~]# chmod 777 clamav-0.97.tar
[root@server ~]# ls
anaconda-ks.cfg  clamav-0.97.tar  Desktop  install.log  install.log.syslog
[root@server ~]# cp clamav-0.97.tar /usr/local/src/

[root@server ~]# cd /usr/local/src/
[root@server src]# tar -zxvf clamav-0.97.tar.gz
```

Ilustración 4-85 Sistema Operativo Centos5 "Servidor"

Añadimos un grupo y un usuario llamado clamav, para que los procesos que realicemos sean asignados a estos.

```
[root@server src]# groupadd clamav
[root@server src]# useradd -g clamav -s /bin/false -c "Antivirus" clamav
```

Ilustración 4-86 Sistema Operativo Centos5 "Servidor"

Ingresamos al directorio **/usr/local/src/clamav-0.97**, y configuramos estableciendo como destino de la configuración el directorio **/etc.**

```
[root@server src]# ls
clamav-0.97  clamav-0.97.tar  index.html
[root@server src]# cd clamav-0.97
[root@server clamav-0.97]# ./configure -sysconfdir=/etc
```

Ilustración 4-87 Sistema Operativo Centos5 “Servidor”

Si configuramos correctamente, Clamav no mostrará errores, entonces podemos proceder a la compilación e instalación.

```
[root@server clamav-0.97]# make > /home/proyecto/tesis/cp_cuatro/clamav/make.txt
[root@server clamav-0.97]# make install > /home/proyecto/tesis/cp_cuatro/clamav/makeinst.txt
```

Ilustración 4-88 Sistema Operativo Centos5 “Servidor”

Para configurar debemos editar el archivo **/etc/clamd.conf,** dado que Clamav se actualiza mediante la ejecución de la herramienta freshclam debemos comentar la línea que contiene Example, caso contrario freshclam no se ejecutará. Comentamos la línea “Example” dentro de clamd.conf de la siguiente manera:

 # Example

Buscamos y descomentamos la línea “LocalSocket” que contiene la dirección del socket clamd, este nos ayudará a ejecutar y administrar el servicio.

LocalSocket /tmp/clamd.socket

Finalmente descomentamos la línea “#AllowSupplementaryGroups no” de modo que otros usuarios que no pertenezcan al grupo clamav puedan hacer solicitudes a clamd.

Guardamos los cambios y salimos del archivo.

Para actualizar clamav debemos editar el fichero **/etc/freshclam.conf,** y borrar la línea que dice **"Example"**, guardamos los cambios y ejecutamos el comando **clamd** para comprobar la configuración. Aquí se iniciará la aplicación clamav.

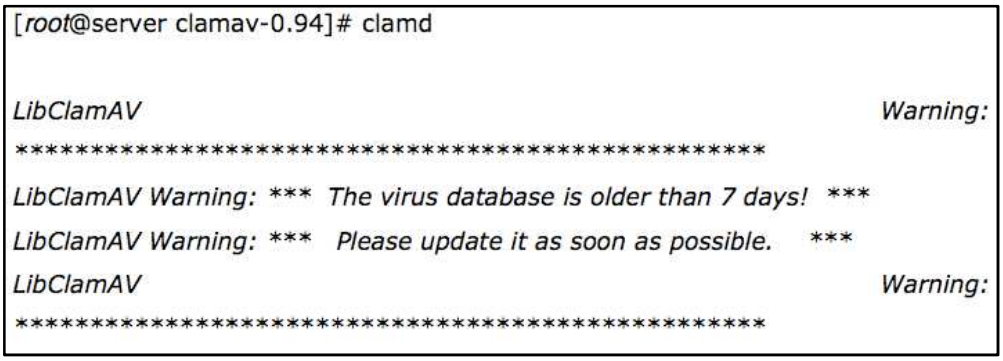


Ilustración 4-89 Sistema Operativo Centos5 "Servidor"

Actualizamos la base de datos ejecutando **freshclam**, es importante tener en cuenta que para esto debemos tener conexión a internet, finalmente reiniciamos Sendmail.

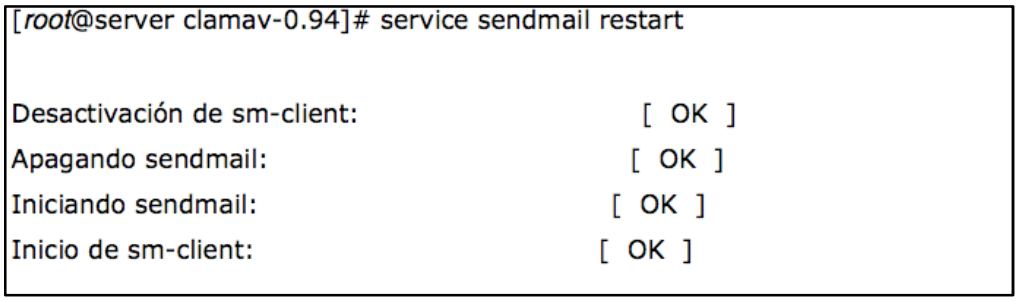


Ilustración 4-90 Sistema Operativo Centos5 "Servidor"

5 SIMULACIÓN DE ATAQUES AL HONEYPOT Y ANÁLISIS DE RESULTADOS.

5.1 COMPONENTES DE LA RED PARA LA SIMULACIÓN

Para realizar la simulación necesitamos un computador con sistema operativo Centos 5, éste será el servidor web y de correo electrónico, y un segundo computador con el mismo sistema operativo Centos 5 que será el encargado de cumplir la función de un firewall de la red así como de ser el sistema anfitrión de la instancia UML, en donde además será el Gateway de la honeynet. Adicionalmente necesitaremos un tercer computador el cual representará a un usuario ajeno a la empresa, quien será el encargado de simular el ataque.

El sistema operativo Centos5 utilizado para ser la instancia de UML posee dos interfaces de red, la primera **eth0** que es la que se conecta con el mundo exterior y es la puerta de entrada de peticiones DNS**,** web y de correo electrónico, tiene una dirección IP **192.168.0.1,** la segunda **eth1** con dirección IP **192.168.0.109** es la conexión directa con el servidor web y de correo electrónico sobre Centos 5, la cual tiene implementado un firewall interno que es bastante restrictivo el cual solo dará entrada a solicitudes DNS, web y de correo electrónico, las demás solicitudes se negarán por defecto.

Por último para comunicarse con la honeynet y de manera especial con el honeyserver posee una interfaz virtual **tap0** con dirección IP **192.168.0.20**. La IP del honeyserver es **192.168.0.30** que está implementado sobre UML ya que éste no posee un firewall interno y el único filtro que encontrará a las peticiones es el principal llamado **honeywall**. Las interfaces virtuales **tap1** y **tap2** han sido asignadas a la UML Centos 5 con sus respectivas IP

192.168.0.26 y **192.168.0.24**.

Aquí el usuario ajeno a la empresa o también llamado atacante verá dos posibilidades de acceso, la una que es el servidor real y la otra que es el honeyserver como parte de la honeynet. Para desviar la atención del intruso, el camino de la honeynet aparentemente será más vulnerable, de esta manera evitaremos atentados al servidor real, teniendo además la opción de estudiar las técnicas y comportamientos del atacante.

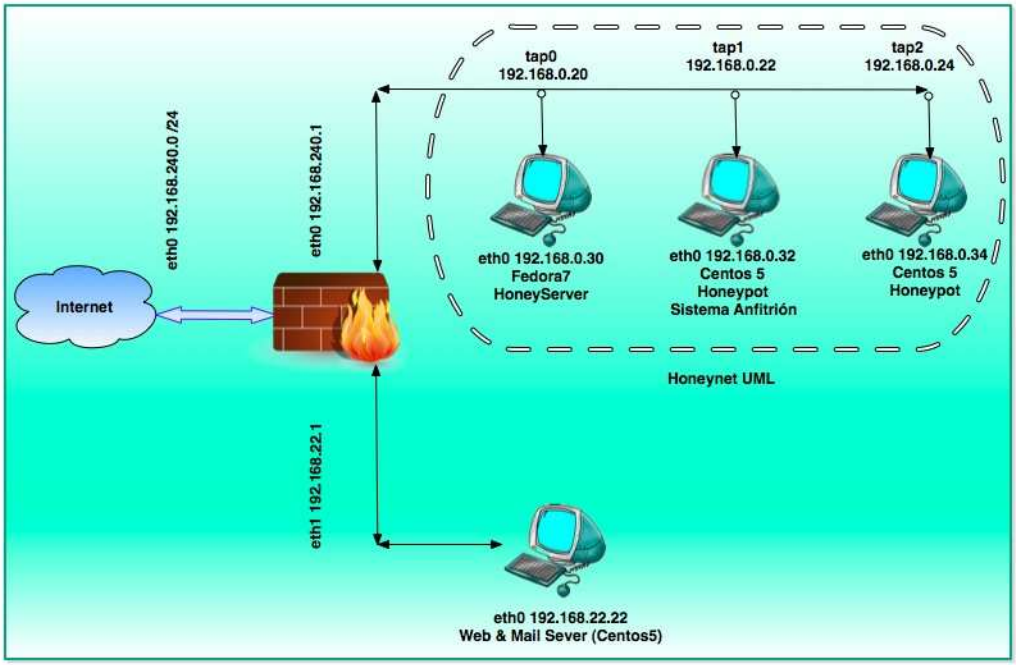


Ilustración 5-1 Esquema de la Red

Fuente: http://bibdigital.epn.edu.ec/bitstream/15000/1523/1/CD-2231.pdf

Para la simulación vamos a utilizar un tercer computador que será quien se conecte al firewall, en éste implementaremos un software basado en Linux llamado Nessus, será el encargado de explorar los denominados “huecos de seguridad” tanto en redes como en servidores. De esta manera podemos evaluar la efectividad de las configuraciones realizadas del firewall y sniffer, además de las vulnerabilidades del verdadero servidor que es al cual buscamos proteger.

5.2 CONFIGURACIÓN DE IPTABLES PARA EL CONTROL DE DATOS

Antes de proceder a la configuración del firewall debemos tener clara su definición así como su utilidad. Un firewall es un dispositivo que filtra el tráfico entre redes, como mínimo dos. El firewall puede ser un dispositivo físico o un software sobre un sistema operativo. En general debemos verlo como una caja con dos o más interfaces de red en la que se establecen unas reglas de filtrado con las que se decide si una conexión determinada puede establecerse o no, incluso puede ir más allá y realizar modificaciones sobre las comunicaciones. (Kioskea)

La configuración del firewall va a ser realizada sobre el sistema operativo Centos5, este firewall se lo denomina honeywall[10] para el caso de aplicaciones de honeynet. Una configuración común permitirá cualquier tipo de acceso hacia la honeynet, esto nos ayudará a que la atención del atacante se desvíe hacia el servidor virtual comprometiéndole pero sin afectar el sistema real.

Un firewall filtra el tráfico TCP/UDP/ICMP/IP y decide si un paquete pasa, se modifica, se convierte o se descarta.

Hay dos maneras de implementar un firewall:

1.- Política por defecto ACEPTAR:

En principio todo lo que entra y sale por el firewall se acepta y solo se denegará lo que se diga explícitamente.

[10] **Honeywall:** Una honeywall es un ordenador configurado para filtrar y observar el tráfico que generan uno o varios honeypots protegiendo al resto de la subred de los ataques de los mismos. Es una parte esencial de una honeynet y cuenta con varios mecanismos para controlar las acciones de los bots capturados. El equipo que hace este papel debe ser invisible para los honeypots con el fin de estudiar en profundidad y sin interferir qué hacen y cómo.

2.- Política por defecto DENEGAR:

Todo está denegado, y solo se permitirá pasar por el firewall aquellos que se permitan explícitamente. (Izura) (Ortega)

Como podemos darnos cuenta la primera política es bastante permisiva por lo que facilita mucho la gestión del firewall, simplemente tenemos que preocuparnos por proteger aquellos puertos o direcciones que sabemos son de interés, al resto simplemente se lo deja pasar. Por ejemplo, si queremos proteger una máquina linux, podemos hacer un **netstat –ln**, el inconveniente que se nos podría presentar es que no se pueda controlar lo que está abierto, que en un momento dado se instale un software nuevo que abra un puerto determinado, o que no sepamos que algunos paquetes ICMP[11] son peligrosos.

Mientras la segunda política es más restrictiva, aquí a no ser que lo permitamos explícitamente, el firewall se convierte en un auténtico *muro* infranqueable. El problema es que es mucho más difícil preparar un firewall así, además debemos tener muy claro cómo funciona el sistema y que es lo que se tiene que admitir sin caer en el error de establecer reglas demasiado permisivas. Esta configuración de firewall es la recomendada, aunque no es aconsejable usarla si no se domina mínimamente el sistema.

5.2.1 PROCEDIMIENTO PARA ESTABLECER POLITICAS

Antes de establecer las políticas debemos proceder con varios pasos previos a la elaboración del script que ejecute las políticas del firewall.

[11] **ICMP**: Protocolo de Mensajes de Control de Internet

1. Puertos que va utilizar el firewall

Puerto # / Capa	Nombre	Descripción
	Ping	Es una utilidad diagnóstica en redes de computadoras que comprueba el estado de la conexión del host local con uno o varios equipos remotos por medio del envío de paquetes ICMP de solicitud y de respuesta. Mediante esta utilidad puede diagnosticarse el estado, velocidad y calidad de una red determinada.
22	SSH	Servicio de shell seguro (SSH)
25	SMTP	Protocolo simple de transferencia de correo (SMTP)
53	DNS	Servicios de nombres de dominio (tales como BIND)
67	DHCP	Protocolo de configuración dinámica de host (DHCP).
80	HTTP	Protocolo de transferencia de hipertexto (HTTP) para los servicios del World Wide Web (WWW)
110	POP3	Protocolo Post Office versión 3
995	POP3S	Protocolo de oficina de correos versión 3 sobre Capa de enchufe segura (POP3S)
143	IMAP	Protocolo de acceso a mensajes de Internet (IMAP)

(Linux) - (Wikipedia, 2011)

2. **Establecer el grupo de clientes en donde se incluye el propio firewall, estos son:**

- ✓ Servidor web – mail “SWM”
- ✓ Honeynet (incluido honeyserver) “HON”
- ✓ Firewall “FW”
- ✓ Internet (usuario o atacante) “INT”

Aquí establecemos relaciones entre las zonas, con esto determinamos si una zona es cliente o servidor de otra, por consecuencia si hay tráfico y en qué sentido.

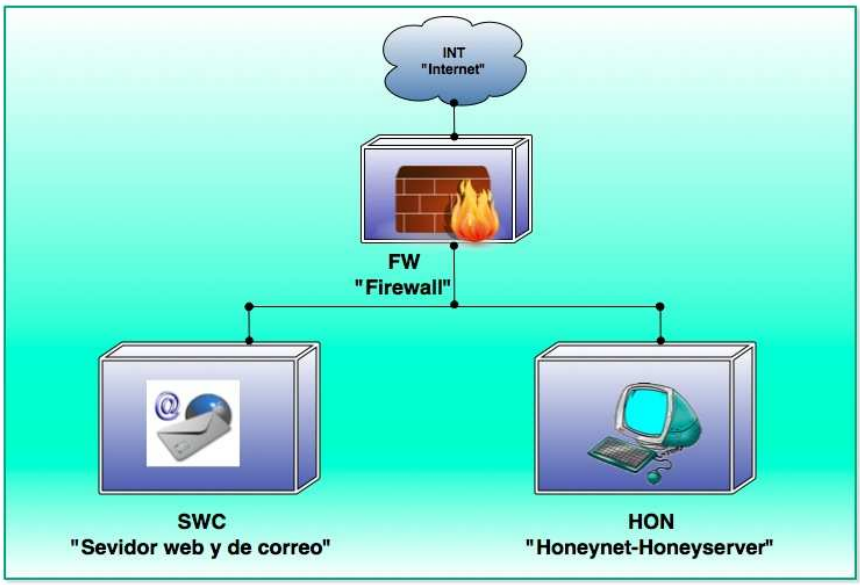


Ilustración 5-2 Zonas para el Firewall

Fuente: http://bibdigital.epn.edu.ec/bitstream/15000/1523/1/CD-2231.pdf

Verificamos la existencia de tráfico y lo anotamos en la siguiente tabla:

Cliente	Servidor	Tráfico
INT	FW	✓
INT	SWM	✓
INT	HON	✓
FW	INT	✓
FW	SWM	✓
FW	HON	✓
SWM	FW	✓
SWM	INT	✓
SWM	HON	✓
HON	FW	✓
HON	INT	✓
HON	SWM	✓

3. Implementación de reglas con la herramienta Iptables

IPtables es un sistema de firewall que está integrado con el kernel, es parte del sistema operativo, un firewall de iptables no es como un servidor que lo iniciamos o detenemos o que se pueda caer por un error de programación. Realmente lo que se hace es aplicar reglas, para ello se ejecuta el comando iptables, con el que añadimos, borramos o creamos reglas. Por ello un firewall de iptables no es sino un simple script de shell en el que se van ejecutando las reglas de firewall. (Xavier Pello)

Iptables está basado en el uso de **Tablas** dentro de las tablas, **Cadenas** formadas por agrupación de **Reglas**, parámetros que relativizan las reglas y

finalmente una **Acción**, que es la encargada de decir qué destino tiene el paquete.

- ✓ **CADENAS**

Es un conjunto secuencial de reglas, donde algunos paquetes pueden entrar en varias cadenas, es decir los paquetes entran en una cadena y atraviesan secuencialmente sus reglas, si el paquete cumple con las condiciones de una regla sigue un destino que puede ser aceptado, rechazado, o reenviado a otra cadena.

Cuando termina el paso por la cadena, y el paquete no cumple ninguna condición de la regla, existe una acción por defecto que es "RECHAZAR" o "ACEPTAR"

Existe un conjunto predeterminado de cadenas, pero el usuario puede crear nuevas cadenas.

- ✓ **REGLAS**

Es un conjunto de condiciones basadas en puertos, direcciones IP, etc. Sean estos de origen o de destino. Por ejemplo:

Si src=192.168.240.45 entonces RECHAZAR

- ✓ **TABLAS**

Son grupos de cadenas que tiene un objetivo desde el punto de vista de la operatividad del firewall. Existen tres tipos de tablas ya definidas, las cuales contienen cadenas predeterminadas y son:

1.- Nat table *"Tabla de traducción de direcciones de red"*

Contiene las cadenas de reescritura de direcciones o de puertos de los paquetes, el primer paquete en cualquier conexión pasa a través de esta tabla;

los veredictos determinan como van a reescribirse todos los paquetes de esa conexión:

PREROUTING chain

Cadena de preruteo: Los paquetes entrantes pasan a través de esta cadena antes de que sea consultada la tabla de ruteo.

POSTROUTING chain

Cadena de postruteo: Los paquetes salientes pasan a través de esta cadena antes de que sea consultada la tabla de ruteo.

OUTPUT chain

Cadena de salida: Filtra los paquetes salientes.

Se debe tener en cuenta que sólo el primer paquete de un flujo alcanzará esta cadena. Después, al resto de paquetes del mismo flujo de datos se les aplicará la misma acción que al primero.

Esta cadena deberá ser usada si el honeyserver está conectado al internet, en el caso de una demostración real. (El block de Alex4)

2.- Mangle table *"Tabla de destrozo"*

Esta tabla contiene todas las cadenas predefinidas y todos los paquetes pasan por ella. Esta ajusta algunas opciones de los paquetes.

PREROUTING chain

Cadena de preruteo: Los paquetes entrantes pasan a través de esta cadena antes de que sea consultada la tabla de ruteo.

INPUT chain

Cadena de entrada: Filtra todos los paquetes entrantes.

FORWARD chain

Cadena de redirección: Filtra los paquetes que atraviesan.

OUTPUT chain

Cadena de salida: Filtra los paquetes salientes.

POSTROUTING chain

Cadena de postruteo: Los paquetes salientes pasan a través de esta cadena antes de que sea consultada la tabla de ruteo.

3.- Filter table *"Tabla de filtros"*

Contiene las cadenas que pueden tener reglas de filtrado de los paquetes como

INPUT chain

Cadena de entrada: Filtra todos los paquetes entrantes.

OUTPUT chain

Cadena de salida: Filtra los paquetes salientes.

FORWARD chain

Cadena de redirección: Filtra los paquetes que atraviesan.

Según el contenido de los paquetes tomamos la determinación de desecharlos "DROP" o aceptarlos "ACCEPT". (Coletti, 2003)- (Cabrera, 2008)- (Manipulación de filtros)

Nosotros usaremos **FILTER TABLE** para pasar los paquetes desde la máquina intrusa al sistema, la cual en condiciones normales será un cliente web/mail y dirigirá las peticiones al servidor web/mail Centos5 con la IP **192.168.0.109**, para la simulación será un atacante en busca de vulnerabilidades, se dirigirá al honeyserver con la dirección **192.168.0.30**, en donde la puerta para el ingreso es la interfaz virtual **192.168.0.20**, el cual tiene el objetivo de ser poseído.

Para esta simulación usaremos el programa Nessus, el cual posee una variedad de plug-in y una serie de herramientas para auditar sistemas, muchas

de las cuales utilizan los hackers para evaluar sistemas remotos.

Está implementado en Windows XP, pero se lo puede hacer en otra plataforma como el mismo Linux.

Primero implementamos un script llamado honeywall.sh, en donde se efectúan todas las reglas y cadenas para las tablas.

El script se lo puede editar con cualquier editor de texto ya sea en la plataforma Linux o Windows, el script es el siguiente, las líneas que comienzan con el carácter **"#"** no se ejecutan ya que son comentarios:

```
#SCRIPT HONEYWALL

iptables -F
iptables -X

#Establecemos a cero los paquetes y contadores de bytes de todas
las
#cadenas.

iptables -Z


#Configuración de Iptables
#Declarar una variable para el ejecutable de Iptables dentro

IPTABLES="/usr/sbin/iptables"

#Carga de módulos
#depmod hace una carga inicial de módulos en el archivo
modules.dep
#de kernel útiles para las aplicaciones

/sbin/depmod -a

#Módulos requeridos
#modprobe usa la lista generada por depmod en modules.dep y
#selecciona los módul os útiles

/sbin/modprobe ip_tables
/sbin/modprobe ip_conntrack
/sbin/modprobe iptable_filter
/sbin/modprobe ipt_LOG
/sbin/modprobe ipt_limit
/sbin/modprobe ipt_state

#Configuración de /proc
#Establecer a 1 la directiva ip_forward necesaria para permitir
```

```
el ruteo
#y el envío de paquetes hacia otros host.

echo "1" > /proc/sys/net/ipv4/ip_forward

iptables -P INPUT ACCEPT
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

#Creación de cadenas de usuario con la opción -N
#Creación de cadenas separadas para los paquetes ICMP, TCP y
UDP.

iptables -N allowed
iptables -N icmp_packets


iptables -A allowed -p TCP --syn -j ACCEPT
iptables -A allowed -p TCP -m state --state ESTABLISHED,RELATED
-j ACCEPT
iptables -A allowed -p TCP -j DROP

#Reglas ICMP
#Permitimos hacer ping hacia este host

iptables -A icmp_packets -p ICMP -s 0/0 --icmp-type 8 -j ACCEPT
iptables -A icmp_packets -p ICMP -s 0/0 --icmp-type 11 -j ACCEPT

#Determinar los paquetes que pueden ingresar a este equipo
#mediante la Cadena INPUT
#Paquetes desde Internet hacia este equipo
#Primer ping

iptables -A INPUT -p ICMP -i eth0 -j icmp_packets

#El servidor DNS para la honeyserver está en la interfaz
virtual, por
#que se permite conexiones internas al puerto 53.

iptables -A INPUT -i eth0 -p tcp -s 192.168.0.0/24 -d
192.168.0.20 --dport 53 -j allowed
iptables -A INPUT -i eth0 -p udp -s 192.168.0.0/24 -d
192.168.0.20 --dport 53 -j ACCEPT
iptables -A INPUT -i eth0 -p udp -s 192.168.0.0/24 -d
192.168.0.125 --dport 67 -j ACCEPT

#Cadena FORWARD
#Servidor DNS

iptables -A FORWARD -i eth0 -o eth0 -p udp -s 192.168.0.0/24 -d
192.168.0.111 --dport 53 -j ACCEPT
iptables -A FORWARD -i eth0 -o eth0 -p udp -s 192.168.0.111 -d
192.168.0.0/24 --sport 53 -j ACCEPT
iptables -A FORWARD -i eth0 -o eth0 -p tcp -s 192.168.0.0/24 -d
192.168.0.111 --dport 53 -j allowed
iptables -A FORWARD -i eth0 -o eth0 -p tcp -s 192.168.0.111 -d
```

```
192.168.0.0/24 --sport 53 -j ACCEPT

#Servidor HTTP

iptables -A FORWARD -i eth0 -o eth0 -s 192.168.0.0/24 -d
192.168.0.111 -p tcp --dport 80 -j allowed
iptables -A FORWARD -i eth0 -o eth0 -s 192.168.0.111 -d
192.168.0.0/24 -p tcp --sport 80 -j ACCEPT

#Cadena OUTPUT
#Decidimos que direcciones IP están permitidas. Deben tener
#coherencia con las solicitudes de entrada en la cadena INPUT al
host
#local.

iptables -A OUTPUT -o eth0 -p tcp -s 192.168.0.20 --sport 53 -d
192.168.0.0/24 -j ACCEPT
iptables -A OUTPUT -o eth0 -p udp -s 192.168.0.20 --sport 53 -d
192.168.0.0/24 -j ACCEPT

/sbin/service iptables save
iptables -L –v
```

Ilustración 5-3 Script honeywall.sh

Se ha implementado un script de firewall que se ejecuta en el servidor Centos5 para su propia protección, se basa en el script honeywall.sh con la diferencia de que solo el servicio DNS, WEB y MAIL está permitido, las demás conexiones están negadas. Las peticiones se procesan a través de la cadena INPUT y las respuestas se ejecutan a través de la cadena OUTPUT. Las políticas, la apertura o cierre de puertos, filtros y análisis de paquetes, dependen de en qué red o sistema va a ser implementado el firewall. (Wikipedia, Netstat, 2010) - (wikilearning, 2007)- (Bulma, 2006) - (Iptables)

5.3 SNORT COMO HERRAMIENTA PARA CAPTURA DE DATOS

5.3.1 DEFINICION

Snort es un sniffer de paquetes y un detector de intrusos basado en red, es un software muy flexible que ofrece capacidades de almacenamiento ya sea en archivos de texto como bases de datos, e implementa un motor de detección y ataques de barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía, funciona en diferentes unices ya sea en un simple PC con Linux, Solaris o cualquier BSD gratuito.

Snort tiene una base datos de ataques que se actualiza constantemente ya sea en forma manual o a través de internet. (Wikipedia, 2011) - (Introducción a Snort).

Snort puede trabajar aislado de otros sistemas de seguridad, pero dependiendo del administrador, o de las aplicaciones conjuntas, nos llenará el disco duro de información inútil, y puede colapsar el tráfico de red.

En el caso de los honeypot es utilizado como un apoyo para la captura de datos, y toda la información capturada se la considera 100% útil, ya que los honeypot son creados para sufrir ataques, y en base a esto aprender las estrategias de los atacantes, con el fin de mejorar la seguridad en los sistemas.

Snort incorpora un sistema bastante sencillo donde podemos escribir nuestras reglas, pudiendo así adaptarlo a nuestros requerimientos.

Cualquier tráfico que curse por un honeyserver es considerado un atentado bajo la siguiente situación:

- ✓ **Un usuario común**

Este usuario ignora como recibe el servicio web y de correo electrónico, él simplemente enciende su computador y con una aplicación web se dirige a un portal, este usuario no configura manualmente nada, simplemente el sistema le ayuda a establecer la conexión y él ejecuta la aplicación.

Para la simulación el usuario normal se conecta al firewall, éste le asigna una dirección IP y establece el servidor Centos5 como IP **192.168.0.109** como DNS y mediante DHCP ya puede utilizar el servicio web y de correo electrónico.

- ✓ **Un usuario con conocimientos**

Para un hacker que busca violar la seguridad en redes, lo primero que hace es buscar vulnerabilidades tales como puertos abiertos, IPs que estén disponibles, etc.

En la simulación se encontrará con dos caminos posibles, el uno que es el servidor Centos5, y el otro que es la honeynet donde tenemos el honeyserver, al momento de intentar acceder a la dirección IP **192.168.0.30**, el administrador deduce que se trata de un ataque ya que esta interfaz no es la del servidor real Centos 5, adicionalmente el intruso verá más aplicaciones y puertos abiertos en comparación con el servidor real, esto llamará su atención logrando así el objetivo primordial del honeypot.

5.3.2 MODOS DE OPERACIÓN

Snort tiene tres modos de operación:

- ✓ **Modo sniffer**

En el que se motoriza por pantalla en tiempo real toda la actividad en la red en que Snort es configurado.

- ✓ **Modo log**

"Registro de paquetes", en el que se almacena en un sistema de log toda la actividad de la red en que se ha configurado Snort para un posterior análisis.

- ✓ **Modo NIDS "Sistema de detección de intrusiones en red"**

En el que se motoriza por pantalla o en un sistema basado en log, toda la actividad de la red a través de un fichero de configuración en el que se especifican las reglas y patrones a filtrar para estudiar los posibles ataques.

Analiza el tráfico de red, cuando un paquete coincide con las reglas preestablecidas por el usuario, toma decisiones basadas sobre las políticas del administrador y así mismo ejecuta acciones.

Para nuestro honeyserver web y de correo electrónico utilizaremos este último modo, aquí el archivo de configuración es **snort.conf**. En el sistema operativo Centos5 no viene instalado por defecto, razón por la cual debemos instalarlo.
La instalación y configuración de Snort se lo realiza en el host anfitrión de UML, bajo el sistema Centos 5. Desde aquí podemos monitorear todo lo que entra y sale por la interfaz virtual tap0. (Pena, 2008) - (Fernandez, 2002)

5.3.3 ARCHIVO DE CONFIGURACIÓN SNORT.CONF

Snort está conformado por cinco componentes, los cuales interactúan entre si y forman un IDS en su totalidad.

1.- Definición de Variables

Aquí se definen las variables de red, por ejemplo el rango de direcciones IP de la red que vamos a monitorear. Esto permitirá que Snort monitoree ataques a los servicios que estén habilitados, con lo que el filtrado será más rápido y seguro.

Adicionalmente también definimos la ruta del directorio donde se encuentran las reglas que rigen el desempeño de snort, de acuerdo a los servicios, direcciones, puertos, previamente establecidos. Por default está en **/etc/snort/snort.conf**.

Para empezar definimos la red a ser monitoreada, esta es la que pertenece a la honeynet virtual.

var HOME_NET 192.168.0.0/24

Establecemos la red que va a ser considerada como internet, es decir la externa.

var EXTERNAL_NET 192.168.240.0/24

Creamos la lista de servidores activos, esta lista debe tener coherencia con lo establecido como red local.

#Lista de servidores DNS en la red
var DNS_SERVERS 192.168.0.30

#Lista de servidores SMTP en la red
var SMTP_SERVERS 192.168.0.32

#Lista de servidores Web en la red
var HTTP_SERVERS 192.168.0.32

En Snort ya están definidas las reglas que se aplicarán a cada servicio que hemos listado, dentro de snort.conf se debe indicar en qué directorio se encuentran estas reglas y su correspondiente preprocesador.

var PREPROC_RULE_PATH /etc/snort/preproc_rules
var RULE_PATH /etc/snort/rules

2.- Configuración de librerías dinámicas

Las librerías dinámicas son ficheros independientes que pueden ser invocados desde el ejecutable, aquí cargaremos los módulos dinámicos utilizados por los preprocesadores con la finalidad de interactuar con Snort. Esto también nos ayudará con el manejo de las alarmas.

dynamicpreprocesador directory/usr/local/lib/snort_dynamicpreprocesador/

Para la detección de paquetes:

dynamicengine /usr/local/lib/snort_dynamicengine/libsf_engine.so
dynamicdetection directory /usr/local/lib/snort_dynamicrule/

3.- Configuración de preprocesadores

Los preprocesadores son plug-ins que definen una manera de rastrear los paquetes y detectar un mal funcionamiento o un tipo de ataque.

La arquitectura de preprocesadores de Snort consiste en pequeños programas C que toman decisiones sobre qué hacer con el paquete. Estos pequeños programas C se compilan junto a Snort en forma de librería. Estos preprocesadores son llamados justo después que Snort realice la decodificación, y posteriormente se llama al Motor de Detección. Si el número de preprocesadores es muy alto el rendimiento de Snort puede caer considerablemente. Las configuraciones predeterminadas para estos subsistemas son muy generales, a medida que experimentemos con Snort podremos ajustarlas para obtener un mejor rendimiento y resultados. Para configurar el preprocesador escribimos:

preprocessor <name_of_processor>: <configuration_options>

Ahora configuramos el preprocesador del servicio DNS:

preprocesador dns:
ports {53}
enable_rdata_overflow

Con la opción **ports** definimos el puerto donde se escuchan las peticiones y con la opción **enable_rdata_overflow** verificamos si existe sobrecarga en las peticiones del cliente.

4.- Módulos de Salida

Snort se puede configurar para que tenga varios modos de salida. Estos modos

de salida pueden ser: salida por pantalla, vía sistema de logs, en una base de datos, con syslog (servidor de eventos del sistema) y con varios formatos como por ejemplo en formato binario (para exportar datos a otros programas). También puede ser configurado más de un módulo de salida, si se tiene varios del mismo tipo estos son evocados en secuencia cuando un evento ocurre.
Determinamos que la salida sea vía sistema de logs, su sintaxis es:

ouput <name>: <options>
output alert_syslog: LOG_AUTH LOG_ALERT

Establecemos la ruta donde descansa el archivo de clasificación de los tipos de ataques:
include /etc/snort/classification.config

5.- Configuraciones Adicionales
Limitamos el número máximo de flujo de bits que pueden ser utilizados dentro de la regla.

config flowbits_size: 256

6.- Personalizar el conjunto de reglas a ser utilizadas en el monitoreo
Esto depende de los servicios activos, por ejemplo web, DNS, etc. Lo hacemos dentro del archivo snort.conf, las reglas que no van a ser utilizadas se las comenta con el carácter "#".

include $RULE_PATH/web-cgi.rules
include $RULE_PATH/web-coldfusion.rules
include $RULE_PATH/web-iis.rules

include $RULE_PATH/web-frontpage.rules

#include $RULE_PATH/web-misc.rules

include $RULE_PATH/web-client.rules

include $RULE_PATH/web-php.rules

include $RULE_PATH/sql.rules

#include $RULE_PATH/x11.rules

#include $RULE_PATH/misc.rules

Establecemos la ruta de los archivos donde se encuentran las reglas que ejecutan los preprocesadores.

include $PREPROC_RULE_PATH/preprocessor.rules

include $PREPROC_RULE_PATH/decoder.rules

Ejecutamos snort:

```
[root@localhost ~]# snort -d -h 192.168.0.0/24 -l /var/log -c /etc/snort/snort.conf
```

Ilustración 5-4 Sistema Anfitrión "Centos5"

Donde la opción:

-d: muestra los datos de las aplicaciones que son monitoreadas.

-h: determina la red que va ser monitoreada

-l: establece el directorio donde se guardan los log generados.

-c: determina la ruta donde se encuentra el archivo de configuración snort.conf. (WoWBerk) - (Adminso)- (Seguridad y redes)- (Zubeldia)

5.4 HACKERS, DEFINICIÓN, HERRAMIENTAS DEL HACKER

5.4.1 DEFINICION

Un hacker es aquella persona experta en alguna rama de la tecnología, a menudo informática, que se dedica a intervenir y/o realizar alteraciones técnicas con buenas o malas intenciones sobre un producto o dispositivo.

La palabra hacker es tanto un neologismo como un anglicismo. Proviene del inglés y tiene que ver con el verbo “hack” que significa recortar, alterar. A menudo los hackers se reconocen como tales y llaman a sus obras “hackeo” o “hackear”. El término es reconocido mayormente por su influencia sobre la informática y la Web, pero un hacker puede existir en relación con diversos contextos de la tecnología, como los teléfonos celulares o los artefactos de reproducción audiovisual. En cualquier caso, un hacker es un experto y un apasionado de determinada área temática técnica y su propósito es aprovechar esos conocimientos con fines benignos o malignos.

Existen distintos tipos de hackers. Aquellos que utilizan su sabiduría a los efectos de corregir errores o desperfectos de una tecnología, poner a disposición del público su saber, crear nuevos sistemas y herramientas, éstos son conocidos como “**white hats**” o “**hackers blancos**”. Se especializan en buscar “bugs” o errores en sistemas informáticos, dándolos a conocer a las compañías desarrolladoras o contribuyendo a su perfeccionamiento. A menudo se reúnen en comunidades online para intercambiar ideas, datos y herramientas. En cambio, los “**black hats**” o “**hackers negros**” son aquellos que también intervienen en los sistemas pero de una manera maliciosa, en general buscando la satisfacción económica o incluso personal. Sus acciones

con frecuencia consisten en ingresar violenta o ilegalmente a sistemas privados, robar información, destruir datos y/o herramientas y colapsar o apropiarse de sistemas. Y eso no es todo, dentro de la comunidad de hackers existen también otros personajes, como los **"lammer",** aquellos que pretenden hacer "hacking" sin tener el debido conocimiento para ello, o los "**luser**", el término con el cual los hackers se refieren al usuario común que no tiene saber sobre la tecnología, o los "**samurai**", los que llevan a cabo acciones maliciosas por encargo, sin conciencia de comunidad ni de intercambio. Otra categoría la configuran los **"piratas informáticos"** que, lejos de considerarse expertos en tecnología, su interés está dado por la copia y distribución ilegal de información, productos y conocimiento. (Victoria por la Tecnología, 2009)

Mencionaremos algunos hackers que han dejado huella en la historia de la informática:

- ✓ **Grace Hooper**

La graduada en Matemáticas y Física en el Vassar College, Grace Hooper, se asimiló en la Marina de Guerra de los Estados Unidos, llegando a ascender al grado de Almirante.

Se dedicó a la investigación e experimentación en el mundo de la programación y computación, Grace Hooper creó el lenguaje Flowmatic, con el cual desarrolló muchas aplicaciones y en 1951 produjo el primer compilador, denominado A-0 (Math Matic). En 1960 presentó su primera versión del lenguaje COBOL (Common Business-Oriented Language).

- ✓ **Kevin Mitnick**

Conocido como "El Cóndor", fue capaz de crear números telefónicos imposibles de facturar, de apropiarse de 20.000 números de tarjetas de crédito de habitantes de California y de burlarse del FBI por más de dos años con sólo un teléfono celular y un ordenador portátil.

- ✓ **Vladimir Levin**

Un graduado en Matemáticas de la Universidad Tecnológica de San Petesburgo, Rusia, acusado de ser la mente maestra de una serie de fraudes tecnológicos que le permitieron a él y la banda que conformaba, substraer más de 10 millones de dólares, de cuentas corporativas del Citibank.

- ✓ **Ian Murphy**

Conocido como el capitán Zap, ingresó de manera ilegal en los computadores de AT&T y cambió la configuración de los relojes internos encargados de medir los tiempos y tarifas a cobrar.

Así, hubo miles de personas que se sorprendieron al recibir la cuenta y comprobar que tenían grandes descuentos por llamadas realizadas en horario nocturno cuando en realidad lo habían hecho en pleno día.

- ✓ **Robert Tappan Morris**

Conocido como el “gusano Morris”, diseñó un gusano que fue capaz de botar 1/10 de la Internet de entonces (lo que significa que inhabilitó cerca de 6 mil computadores).

- ✓ **Los escuadrones Mod Y Lod**

Tipos de gran fama por tener numerosas formas de evitar el pago de llamadas telefónicas de larga distancia, además podían escuchar conversaciones privadas e incluso crear enormes líneas multiconferencias que compartían con sus amigos.

Hackearon muchas bases de datos, incluyendo la de la National Security Agency, AT&T y la del Bank of America. También pudieron acceder a los registros de la Credit Record Reporting Agency y de esta manera "ojear" los registros de los ricos y famosos.

- ✓ **David Smith**

Fue el creador del virus Melissa que se propagó rápidamente en centenares de millones de ordenadores de todo el mundo.

- ✓ **Onel e Irene De Guzmán**

Esta pareja de hermanos filipinos reconoció durante una rueda de prensa realizada en su país que había difundido el virus “I Love You” de manera accidental. (Seguridad PC)

5.4.2 HERRAMIENTAS UTILIZADAS POR LOS HACKERS

Existe un sin número de herramientas utilizadas por los hackers, esto dependiendo de sus propósitos. Entre estas tenemos:

- ✓ **Caín y Abel "Sniffer"**

Es una herramienta enfocada al monitoreo de la red que permite la recuperación de contraseñas para el sistema operativo Windows. Es capaz de romper contraseñas cifradas usando ataques del diccionario, de fuerza bruta y de criptoanálisis. Este programa aprovecha ciertos agujeros de seguridad de los protocolos.

- ✓ **John the Ripper**

Es una poderosa herramienta que utiliza ataques de diccionario y fuerza bruta para descifrar contraseñas. Es muy rápida y es capaz de romper algoritmos de cifrado como DES, SHA-1 y otros.

- ✓ **Tcpdump**

Es una poderosa herramienta empleada para el monitoreo y la adquisición de datos en redes. Permite el volcado del tráfico que se presenta en una red a un archivo, la pantalla, etc. Se puede usar para detectar problemas en la red, ping attacks o para monitorizar las actividades de una red

- ✓ **Netcat**

Se trata de una utilidad disponible para Unix y Windows que lee y escribe información a través de conexiones de red TCP o UDP. Permite a través del intérprete de comando y con una sintaxis muy sencilla abrir puertos TCP/UDP en un host, quedando netcat a la escucha, asociar a una Shell a un

puerto en concreto y forzar conexiones UDP/TCP, útiles como por ejemplo para realizar rastreos de puertos o realizar transferencias de archivos BIT a BIT entre dos equipos.

- ✓ **Nmap "Escaneador de redes"**

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 permite hacer el inventario y el mantenimiento del inventario de computadores de una red. Se puede usar entonces para auditar la seguridad de una red, mediante la identificación de todo nuevo servidor que se conecte. Nmap es a menudo confundido con herramientas para verificación de vulnerabilidades como Nessus. Nmap es difícilmente detectable, ha sido creado para evadir los Sistema de detección de intrusos (IDS) e interfiere lo menos posible con las operaciones normales de las redes y de las computadoras que son analizadas.

- ✓ **SuperScan " Escaner de Puertos TCP, pinger and host"**

Súper Scan es un software que no requiere instalación por lo que es 100% portable. Permite escanear redes y desplegar todos los equipos conectados a una red y lista los puertos abiertos en cada equipo. Se puede pedir que escanee algún equipo fuera de la red o redes remotas completas, es una herramienta muy útil para el diagnóstico de redes.

- ✓ **Hping**

Permite generar paquetes especiales de ICMP/UDP/TCP y buscar respuestas de pings. Hping es una herramienta muy versátil, que permite la manipulación de paquetes TCP/IP desde línea de comandos.

Es factible modificar la información contenida en las cabeceras de los paquetes ya sean TCP, UDP e ICMP, en función de los parámetros con que ejecutemos Hping.

- ✓ **Lcp**

Audita y restaura Pass de Windows NT / 2000 / XP / 2003.

Es un recuperador de contraseñas para Windows. Se puede usar cuando se ha olvidado una contraseña o por ejemplo para comprobar si una clave es válida. Dispone de tres métodos de recuperación. En primer lugar utilizando diccionarios, mediante fuerza bruta y una tercera que combina las dos anteriores.

- ✓ **Keylogger**

Es un tipo de software que se encarga de registrar las pulsaciones que se realizan en el teclado, para memorizarlas en un fichero y/o enviarlas a través de internet. Suele usarse como malware del tipo Daemon, permitiendo que otros usuarios tengan acceso a contraseñas importantes, como los números de una tarjeta de crédito, u otro tipo de información privada que se quiera obtener. (Wikipedia - Keylooger, 2011) - (Redes y seguridad, 2009) - (Lanrouter, 2005) - (Bujarra)

5.5 NESSUS, SOFTWARE PARA SIMULAR INTRUSIONES

Existe una variedad de herramientas comerciales para simular ataques al honeypot, las más conocidas son BSHAcker, Scan9 y Nessus, éste último es el que utilizaremos ya que es un potente software que permite simular intrusiones y además auditar sistemas. Se lo puede descargar desde su sitio oficial www.nessus.org.

5.5.1 DEFINICION

Nessus es un programa de escaneo de vulnerabilidades en diversos sistemas operativos. Consiste en Nessusd, el daemon Nessus, que realiza el escaneo en el sistema objetivo, y Nessus, el cliente "basado en consola o gráfico" que muestra el avance y reporte de los escaneos. Desde la consola Nessus puede ser configurado para hacer escaneos programados con cron.

En operación normal, Nessus comienza escaneando los puertos con nmap o con su propio escaneador de puertos para buscar puertos abiertos y después intentar varios exploits para atacarlo. Las pruebas de vulnerabilidad, disponibles como una larga lista de plugins, son escritos en **NASL** "Nessus Attack Scripting Language, Lenguaje de Scripting de Ataque Nessus por sus siglas en inglés", un lenguaje scripting optimizado para interacciones personalizadas en redes.

Opcionalmente, los resultados del escaneo pueden ser exportados en reportes en varios formatos, como texto plano, XML, HTML, y LaTeX. Los resultados también pueden ser guardados en una base de conocimiento para referencia en futuros escaneos de vulnerabilidades.

Algunas de las pruebas de vulnerabilidades de Nessus pueden causar que los servicios o sistemas operativos se corrompan y caigan. El usuario puede evitar esto desactivando "**unsafe test"** (pruebas no seguras) antes de escanear.
Sus características principales son:

- ✓ Alertar acerca de configuraciones incorrectas en los firewalls, host y/o dispositivos de borde.
- ✓ Descubrir la aparición de nuevas vulnerabilidades como resultado de cambios en la configuración.
- ✓ Detectar la falta de parches y actualizaciones en los sistemas de la compañía.
- ✓ Localizar debilidades y vulnerabilidades conocidas antes de que un intruso lo haga. (Lacuesta) - (Wiki, 2007) - (HD Moore)

5.5.2 INSTALACIÓN DE NESSUS

Nessus está disponible para varias plataformas Linux tales como Debian y Fedora, también para Windows y Mac. Vamos a utilizar Nessus en Windows ya que si el objetivo es simular el ataque de un hacker, debemos tener en cuenta que éste lo hará bajo cualquier sistema operativo.
Nessus consta de dos partes: el servidor y el cliente. El servidor es el encargado de realizar los escaneos y las pruebas, el cliente es el encargado de mostrar una interfaz gráfica al usuario. Pueden estar instalados dentro del mismo host, o el cliente puede estar en un host remoto y conectarse a través de la red para solicitar el servicio.
En la página www.nessus.org seleccionar la opción download.

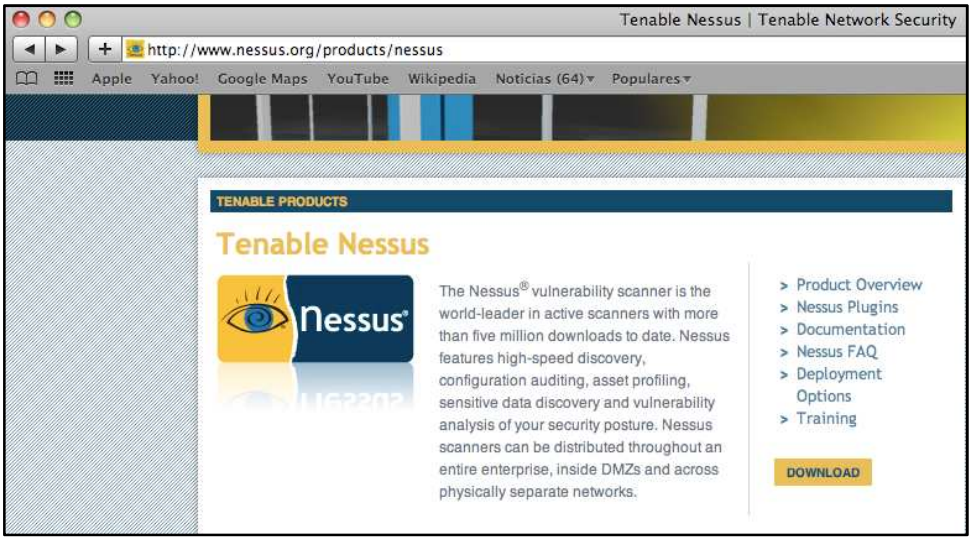


Ilustración 5-5 Nessus para Windows

Aceptar los términos

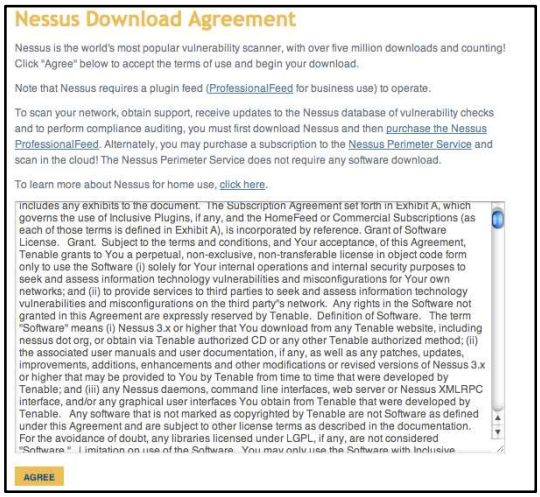


Ilustración 5-6 Términos para la descarga

Seleccionar la plataforma bajo la cual vamos a trabajar, en nuestro caso es Windows.

Ilustración 5-7 Selección de la plataforma bajo la cual trabajará Nessus

Descargamos la última versión de Nessus, ésta es la 4.4.1.

El instalador de Nessus para Windows incluye las dos partes cliente y servidor, mientras que para Linux se debe descargar las dos partes por separado.

Ejecutamos el instalador, se acepta todas las opciones por defecto y ya podemos iniciar la simulación.

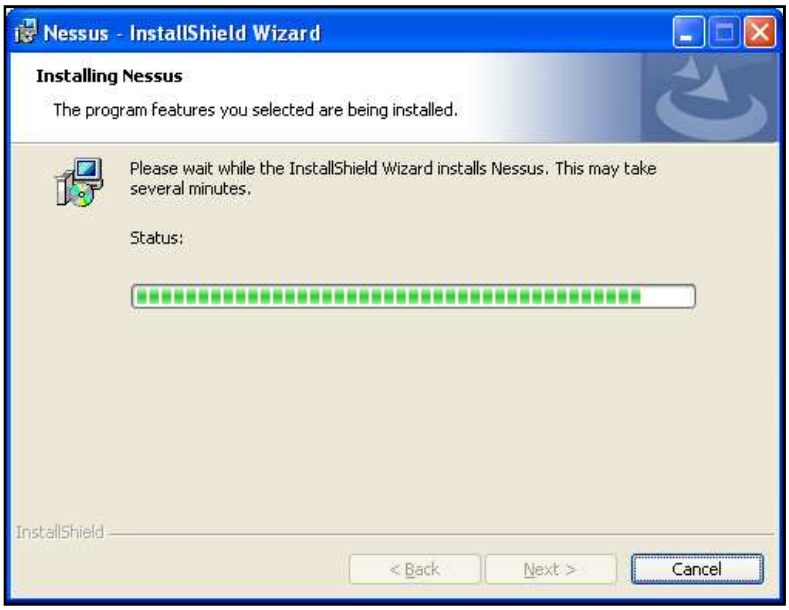


Ilustración 5-8 Proceso de instalación de Nessus

Ejecutar el icono “Nessus Server Manager” que tenemos en el escritorio y procedemos a activar el paquete.

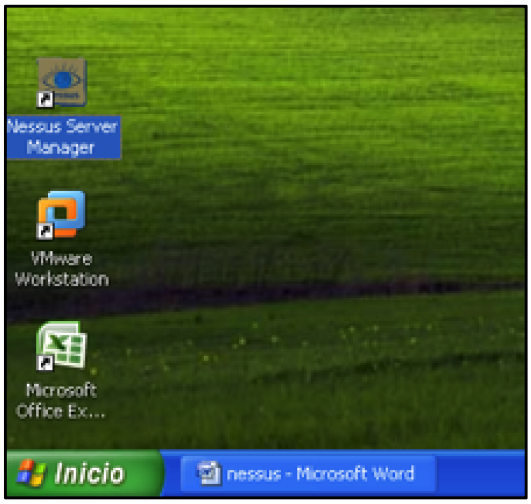


Ilustración 5-9 Nessus Server Manager

Clic en la opción “Obtain an activation code”

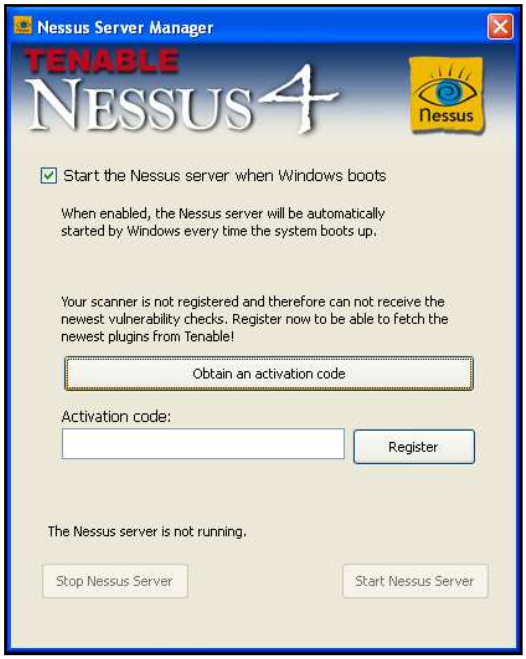


Ilustración 5-10 Obtención del código de registro

Ingresar los datos que se requieren. El código de registro será enviado a la dirección de correo electrónico ingresada.

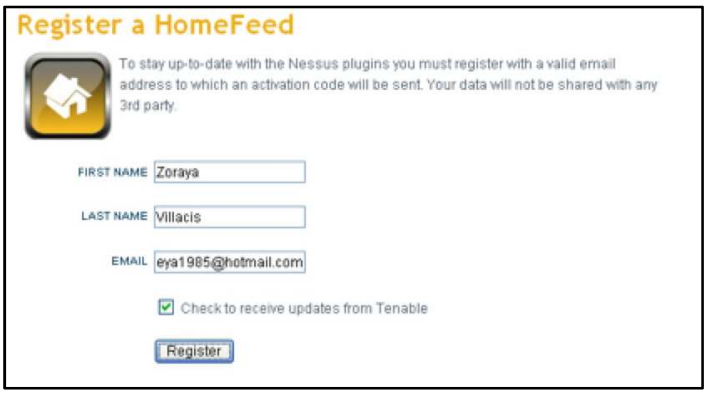


Ilustración 5-11 Registro de datos

En la casilla “Activation code” ingresar el código que nos fue enviado.

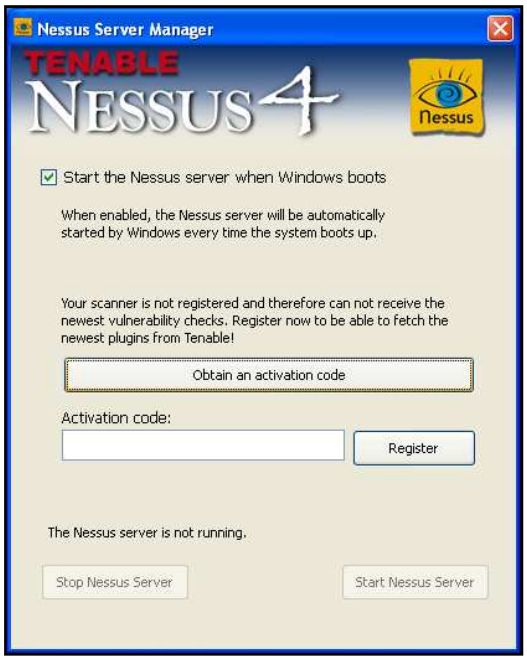


Ilustración 5-12 Código de activación

Inicia un proceso de descarga de varios plug-ins. Es recomendable primero descargar estos plug-ins antes de empezar a utilizar el servicio, esto tomará unos minutos.

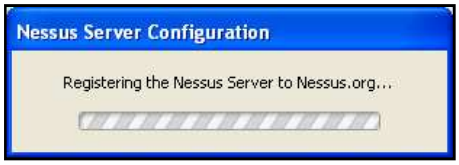


Ilustración 5-13 Descarga de plug-ins

Una vez finalizada la instalación, podremos arrancar el servicio cuando se inicie el sistema ya que así está establecido en la configuración por defecto.

5.5.3 CONFIGURACIÓN DE NESSUS

Antes de proceder a la configuración debemos comprobar que el servicio Nessus se esté ejecutando. Para verificarlo observamos si la casilla "**Start the Nessus server when Windows boots"** está marcada.

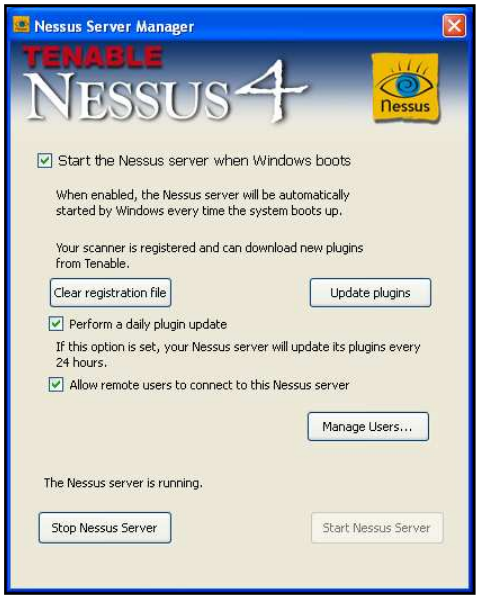


Ilustración 5-14 Comprobación de la ejecución de Nessus

Si el servidor está en el mismo host del cliente no se cambia la configuración por defecto, la IP es la local, y el puerto que utiliza Nessus para los escaneos es el **1241**. Es importante tener en cuenta que el host en el cual se va a ejecutar tenga el firewall desactivado para que no cause conflictos en la prueba.

Agregamos un usuario que va a utilizar Nessus Client, para esto elegimos "Añadir nuevo usuario" en la opción "Manage Users", le asignamos una contraseña y le damos los permisos de administrador.

Ilustración 5-15 Registro de Usuarios

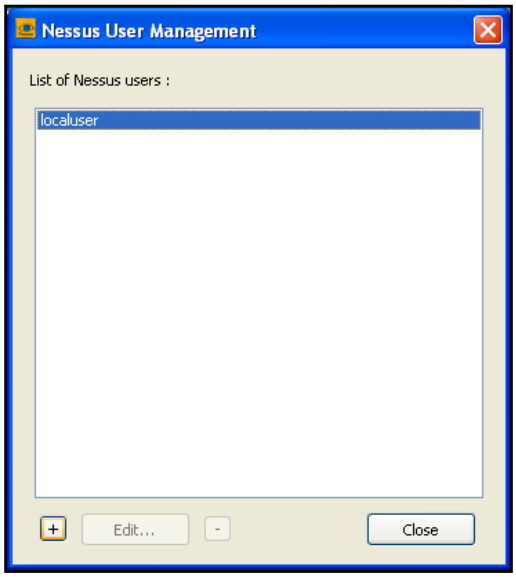


Ilustración 5-16 Lista de usuarios

El usuario se llamará **hacker**, guardamos los cambios y de esta manera el usuario ya queda autenticado.

Ilustración 5-17 Creación de un nuevo usuario de Nessus

5.5.3.1 CONFIGURACIÓN DE LAS POLITICAS DE NESSUS

Damos doble clic en el icono **"Nessus Client"** que está en el escritorio, aparecerá una alerta de seguridad al momento de abrir por lo que damos clic en la opción **"Si"** para continuar.

Ingresar con el nombre de usuario y su respectivo password que fueron autenticados anteriormente.

Ilustración 5-18 Autenticación del usuario

Existen cuatro políticas que ya están establecidas, para agregar una nueva política damos clic en la pestaña "**Policies"**.

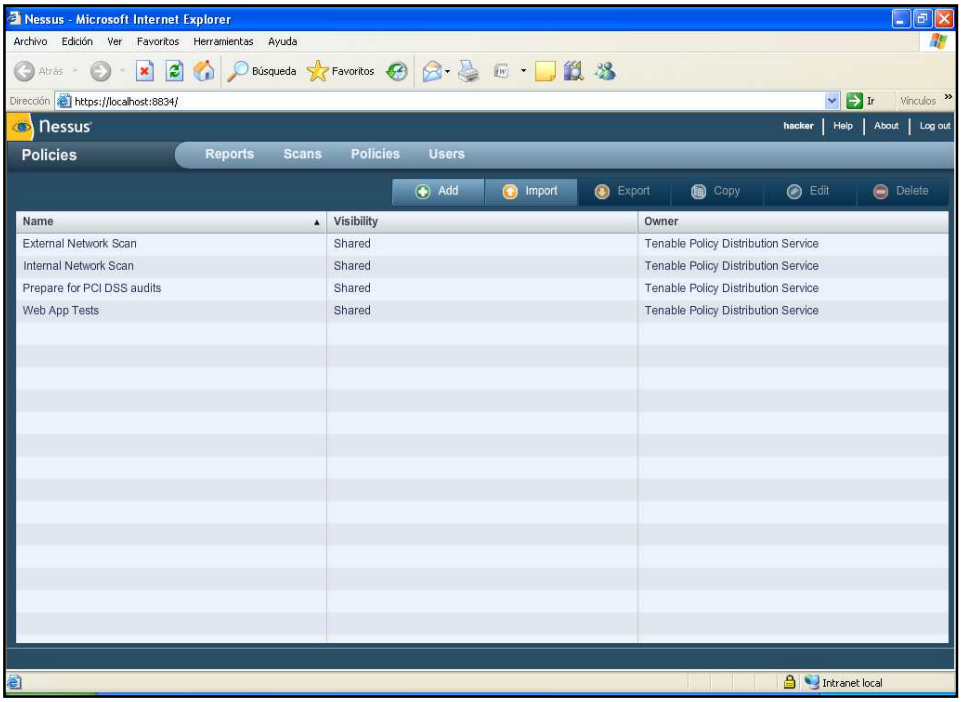


Ilustración 5-19 Políticas de Nessus

Al dar clic en la pestaña **Add** (añadir) se desplegará un menú con cuatro opciones que nos permitirán definir cada uno de los parámetros de la política.

- ✓ **Opción "General":**

Aquí se ingresa el nombre de la política, el tipo de visibilidad, el rango de puertos, el tipo de escaneo, reducción de conexiones paralelas para la congestión de la red, en sí aquí se establecen un conjunto de parámetros globales para que los plug-ins sean ejecutados por Nessus.

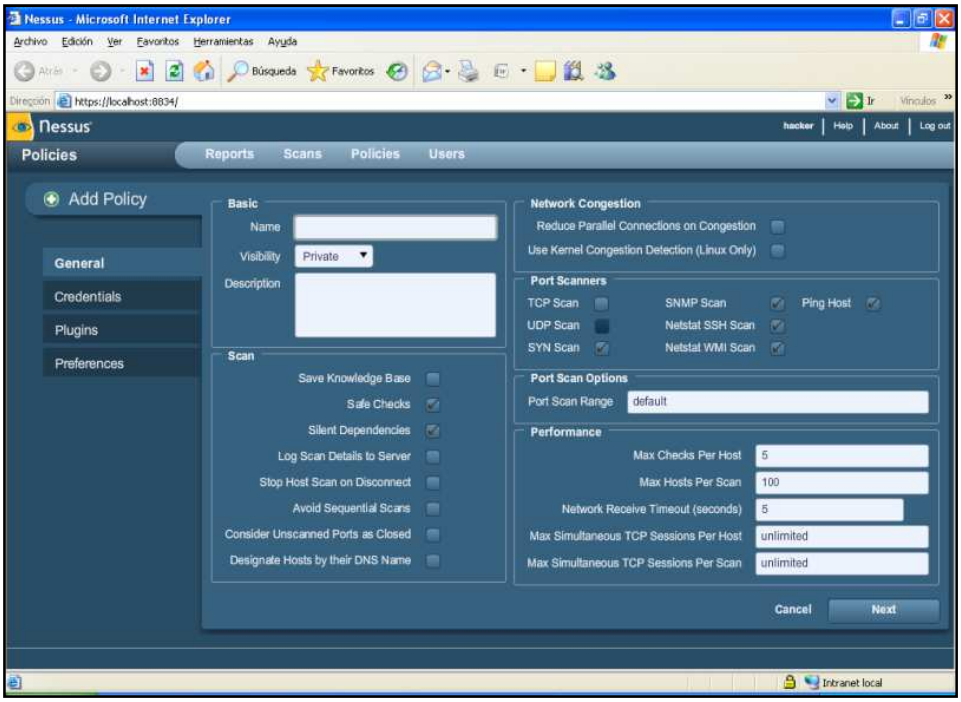


Ilustración 5-20 Configuración de nueva política. Ficha General

- ✓ **Opción "Credentials":**

Esta opción nos permite configurar y autenticar usuarios que generalmente tienen un acceso remoto.

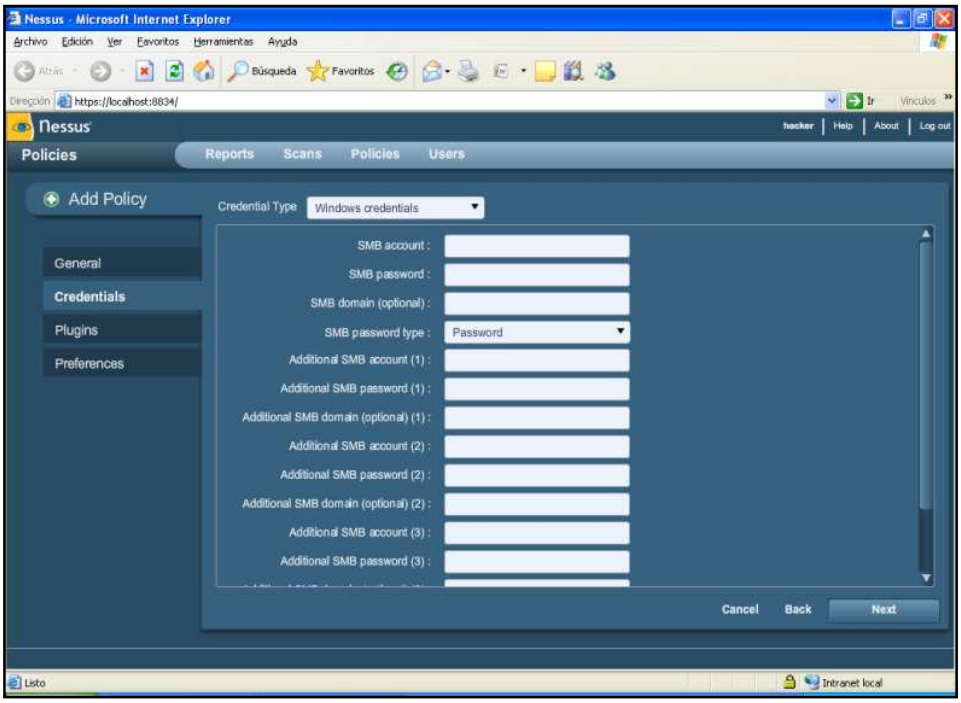


Ilustración 5-21 Configuración de nueva política. Ficha Credencial

✓ **Opción "Plugins":**

Permite al usuario elegir que plugins utilizar para la intrusión. Esta opción es la más importante ya que aquí se pueden definir y analizar a un cierto tipo de vulnerabilidades. Según la selección, a mayor cantidad de opciones, mayor será el tiempo que se demore la intrusión.

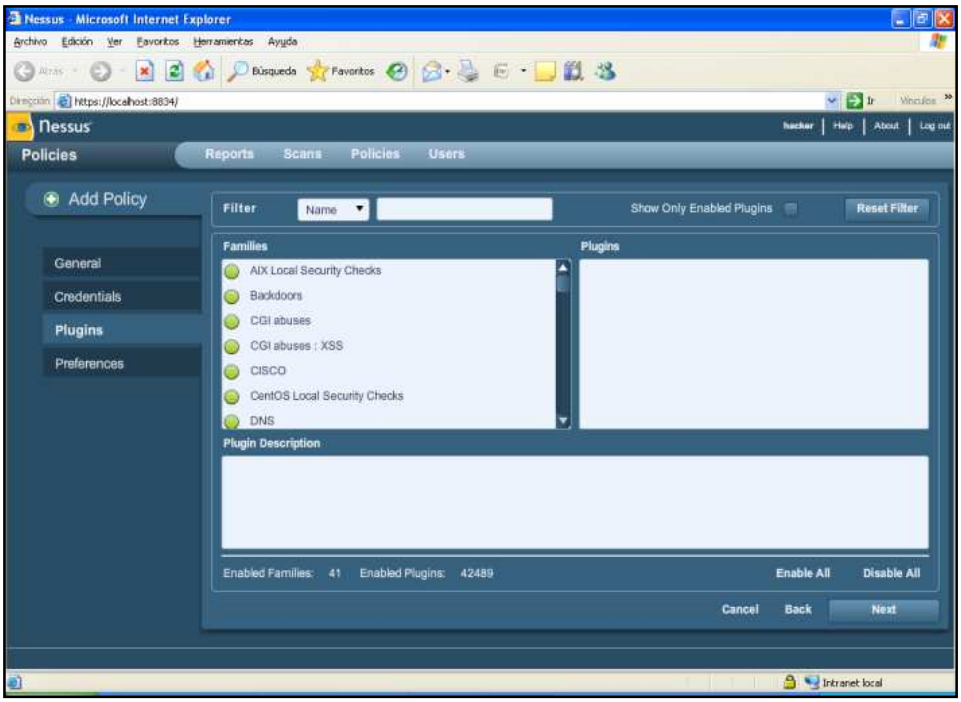


Ilustración 5-22 Configuración de nueva política. Ficha Plugins

✓ **Opción "Preferences":**

Esta opción incluye configuraciones especiales, nos permite controlar el escaneo ya que es una configuración dinámica e interviene directamente con la ejecución de plug-in. (Youtube - Descubre vulnerabilidades en tu sistema con Nessus)

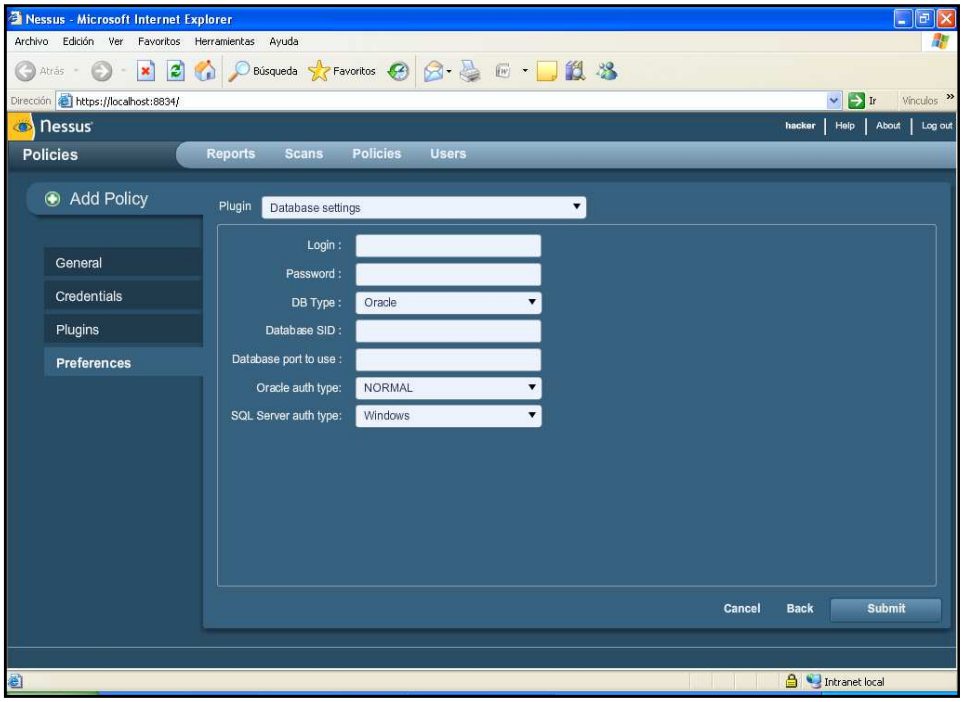


Ilustración 5-23 Configuración de nueva política. Ficha Preferences

5.6 PROTOCOLO DE PRUEBAS

Estrategias efectivas con relación a la seguridad de la información en las empresas no solo requieren de políticas y procedimientos adecuados, además se necesita la ejecución de un conjunto de acciones, un plan de acción para afrontar riesgos de seguridad, o un conjunto de reglas para el mantenimiento de cierto nivel de seguridad. Pueden cubrir cualquier cosa desde buenas prácticas para la seguridad de un solo ordenador, reglas de una empresa o edificio, hasta las directrices de seguridad de un país entero.

La **política de seguridad** debe ser enriquecida y compatibilizada con otras políticas dependientes de ésta, objetivos de seguridad. Debe estar fácilmente accesible de forma que los empleados estén al tanto de su existencia y entiendan su contenido. Puede ser también un documento único o inserto en

un manual de seguridad. Se debe designar un propietario que será el responsable de su mantenimiento y su actualización a cualquier cambio que se requiera. (Wikipedia, Políticas de Seguridad, 2011) - (Segu-Info, 2009)

Se deben tener en cuenta varias medidas de seguridad tales como la utilización de protocolos seguros, instalación y configuración de firewall, instalación y configuración de IDS, instalación y configuración de antivirus y filtrado de contenidos, entre otros.

Entre los protocolos seguros más utilizados tenemos:

- ✓ **SSH "Secure Shell"**

Usado exclusivamente en reemplazo de telnet.

- ✓ **SSL "Secure Sockets Layer"**

Es el más utilizado por su simplicidad, cuyo uso principal es cifrar el número de tarjetas al realizar cualquier tipo de transacción online. El protocolo SSL ofrece servicio de cifrado de datos, autenticación del servidor, integridad de mensajes y, en menor medida, la identificación del cliente para conexiones TCP/IP.

- ✓ **IPSec "Internet Protocol Security"**

Su función es asegurar las comunicaciones sobre el protocolo de Internet autenticando y/o cifrando cada paquete IP en un flujo de datos. (Wikipedia, 2011)

5.6.1 PROCEDIMIENTO PARA EL PROTOCOLO DE PRUEBAS

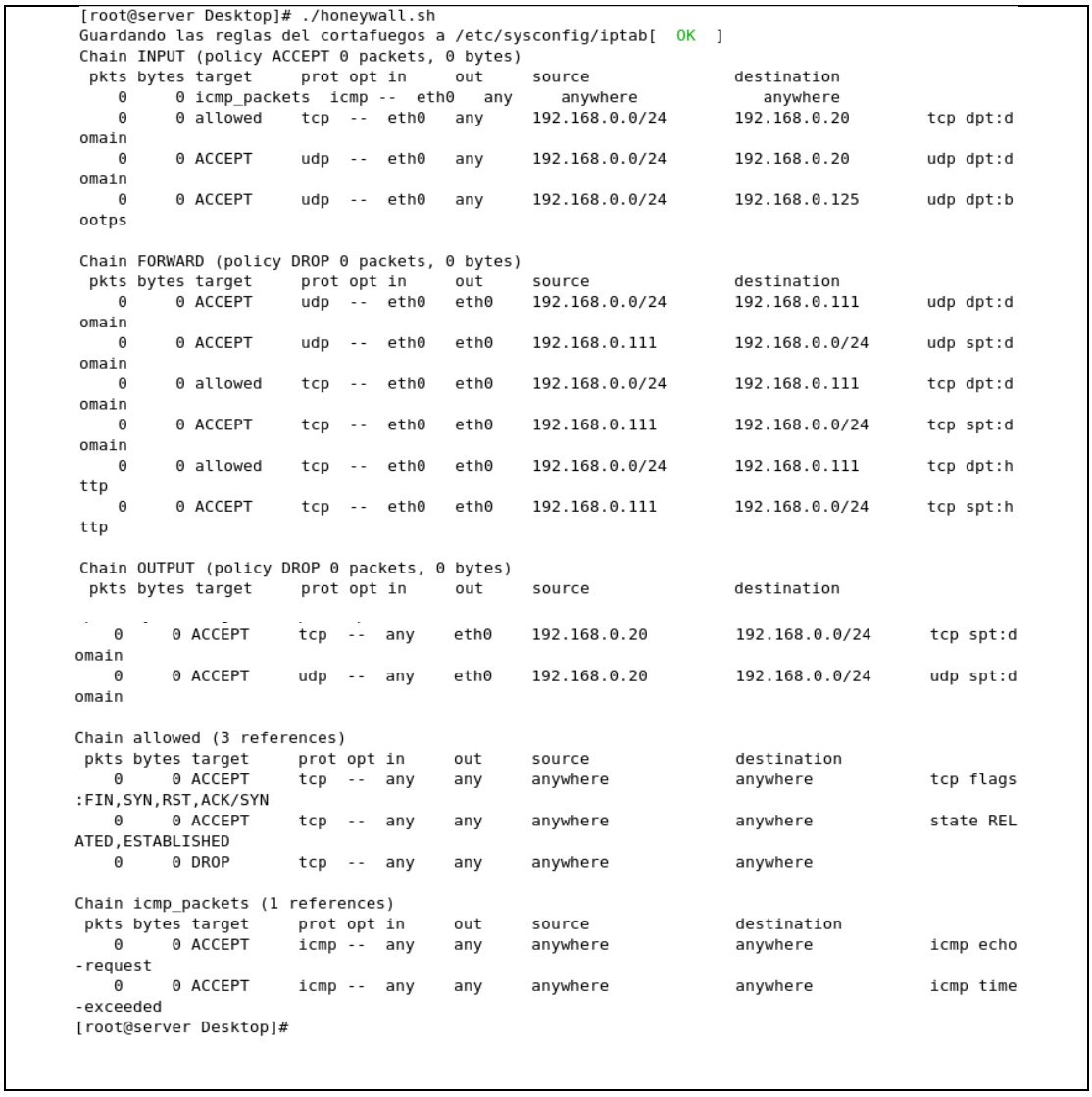
Los pasos para realizar el Protocolo de pruebas son:

1.- Conexión

El host anfitrión de UML posee dos interfaces de red:
La interfaz **eth1** cuya IP es **192.168.0.120** que es el gateway por defecto para el servidor real implementado en Centos5 cuya IP es **192.168.0.109**, los dos equipos deben estar conectados mediante cable UTP-5 cruzado.
La interfaz **eth0** cuya IP es **192.168.0.1** que será la puerta de acceso a internet y como ya dijimos anteriormente hará las veces de cliente se conecta a un switch mediante cable UTP-5 directo, desde aquí podrá conectarse a uno o varios clientes. A éste también se conecta el host con sistema operativo Windows donde se está ejecutando Nessus, de la misma manera con cable directo.

2.- Ejecución del script "honeywall.sh"

Este script se encuentra en el servidor real en el escritorio, dicho script que ya se lo describió anteriormente está basado en el script honeywall que es el encargado de permitir solo conexiones web y mail, además de buscar proteger al servidor ya sea de conexiones no deseadas o de intrusiones a causa de la simulación.



```
[root@server Desktop]# ./honeywall.sh
Guardando las reglas del cortafuegos a /etc/sysconfig/iptab[  OK  ]
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 icmp_packets  icmp --  eth0   any     anywhere             anywhere
    0     0 allowed    tcp  --  eth0   any     192.168.0.0/24       192.168.0.20        tcp dpt:d
omain
    0     0 ACCEPT     udp  --  eth0   any     192.168.0.0/24       192.168.0.20        udp dpt:d
omain
    0     0 ACCEPT     udp  --  eth0   any     192.168.0.0/24       192.168.0.125       udp dpt:b
ootps

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     udp  --  eth0   eth0    192.168.0.0/24       192.168.0.111       udp dpt:d
omain
    0     0 ACCEPT     udp  --  eth0   eth0    192.168.0.111        192.168.0.0/24      udp spt:d
omain
    0     0 allowed    tcp  --  eth0   eth0    192.168.0.0/24       192.168.0.111       tcp dpt:d
omain
    0     0 ACCEPT     tcp  --  eth0   eth0    192.168.0.111        192.168.0.0/24      tcp spt:d
omain
    0     0 allowed    tcp  --  eth0   eth0    192.168.0.0/24       192.168.0.111       tcp dpt:h
ttp
    0     0 ACCEPT     tcp  --  eth0   eth0    192.168.0.111        192.168.0.0/24      tcp spt:h
ttp

Chain OUTPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

    0     0 ACCEPT     tcp  --  any    eth0    192.168.0.20         192.168.0.0/24      tcp spt:d
omain
    0     0 ACCEPT     udp  --  any    eth0    192.168.0.20         192.168.0.0/24      udp spt:d
omain

Chain allowed (3 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere            tcp flags
:FIN,SYN,RST,ACK/SYN
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere            state REL
ATED,ESTABLISHED
    0     0 DROP       tcp  --  any    any     anywhere             anywhere

Chain icmp_packets (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     icmp --  any    any     anywhere             anywhere            icmp echo
-request
    0     0 ACCEPT     icmp --  any    any     anywhere             anywhere            icmp time
-exceeded
[root@server Desktop]#
```

Ilustración 5-24 Ejecución del script honeywall.sh

3.- Activación de la honeynet y el honeyserver

En el host anfitrión Centos5 dentro del directorio **/home/proyecto/build** se encuentra el kernel ejecutable de UML llamado **linux**, aquí también se encuentran las imágenes del sistema de archivos que vamos a utilizar. Trabajaremos con dos imágenes distintas, ya que ejecutaremos dos máquinas virtuales simultáneamente.

La primera imagen corresponde al honeyserver con sistema Fedora7, su hostname es **serverespe**, aquí se ejecuta un servidor web y de correo virtual, y un servicio ssh.

La segunda imagen es el Centos5, comprobando de esta manera que se puede ejecutar una máquina virtual Centos dentro de un sistema anfitrión Centos5, a este lo denominaremos **serverdns**.

A continuación un ejemplo de ejecución, que además ya lo hemos utilizado en secciones anteriores:

```
[root@localhost linux-2.6.38]# ./linux ubda=/home/proyecto/build/Fedora7-x86-roo
t_fs
```

Ilustración 5-25 Ejemplo de la ejecución de UML – Sistema Operativo Anfitrión Centos5

4.- Configuración de las interfaces virtuales

Para la honeynet se utilizará la subred **192.168.0.0/24,** cada UML tiene su respectiva dirección IP y crea una interfaz virtual denominada **tapX**, donde X=0, 1, 2, 3… dependiendo el orden de creación. Por lo tanto en el host anfitrión además de las interfaces físicas eth0 y eth1 se añadirán las interfaces virtuales, estas se conectan internamente gracias a las aplicaciones **uml_net** y **uml_switch** que ya fueron compiladas anteriormente en las herramientas de uml en **uml_utilities**.

Dentro de UML se asigna que interfaz será el gateway para comunicarse con el host anfitrión de la siguiente manera:

```
[root@serverespe ~]# eth0=tuntap,,,192.168.0.20
```

Ilustración 5-26 Asignación del gateway

Autenticamos UML en el host anfitrión Centos5 con un ID generado por el proceso, este se encuentra en el directorio del usuario que ejecutó la aplicación, en este caso root, en un directorio llamado .uml

```
[root@localhost ~]# cd /home/tesis/build/linux-2.6.38
[root@localhost linux-2.6.38]# uml_mconsole ZyqcJp eth0=tuntap,,,192.168.0.20
```

Ilustración 5-27 Autenticación de UML

Se activará la interfaz virtual **tap0**, esta pertenece al host anfitrión Centos5, que comunica el honeyserver con el host anfitrión, la interfaz **eth0** del honeyserver se la configura como cualquier sistema Fedora7, su dirección IP es **192.168.0.30**.

Comprobamos la conectividad desde el honeyserver hacia el host anfitrión Centos.

```
[root@serverespe network-scripts]# ping 192.168.0.120
PING 192.168.0.120 (192.168.0.120) 56(84) bytes of data.
64 bytes from 192.168.0.120: icmp_seq=1 ttl=64 time=117 ms
64 bytes from 192.168.0.120: icmp_seq=2 ttl=64 time=0.135 ms
64 bytes from 192.168.0.120: icmp_seq=3 ttl=64 time=0.122 ms
64 bytes from 192.168.0.120: icmp_seq=4 ttl=64 time=0.122 ms
64 bytes from 192.168.0.120: icmp_seq=5 ttl=64 time=0.135 ms
```

Ilustración 5-28 Sistema Operativo "Fedora7"

Luego comprobamos la conectividad desde el servidor web y mail Centos5 hacia el honeywall

```
[root@localhost linux-2.6.38]# ping 192.168.0.30
PING 192.168.0.30 (192.168.0.30) 56(84) bytes of data.
64 bytes from 192.168.0.30: icmp_seq=1 ttl=64 time=0.260 ms
64 bytes from 192.168.0.30: icmp_seq=2 ttl=64 time=0.124 ms
64 bytes from 192.168.0.30: icmp_seq=3 ttl=64 time=0.204 ms
64 bytes from 192.168.0.30: icmp_seq=4 ttl=64 time=0.128 ms
64 bytes from 192.168.0.30: icmp_seq=5 ttl=64 time=0.119 ms
64 bytes from 192.168.0.30: icmp_seq=6 ttl=64 time=0.244 ms
```

Ilustración 5-29 Servidor web y mail Centos5

Finalmente lo hacemos desde el host externo (internet) hacia el honeywall.

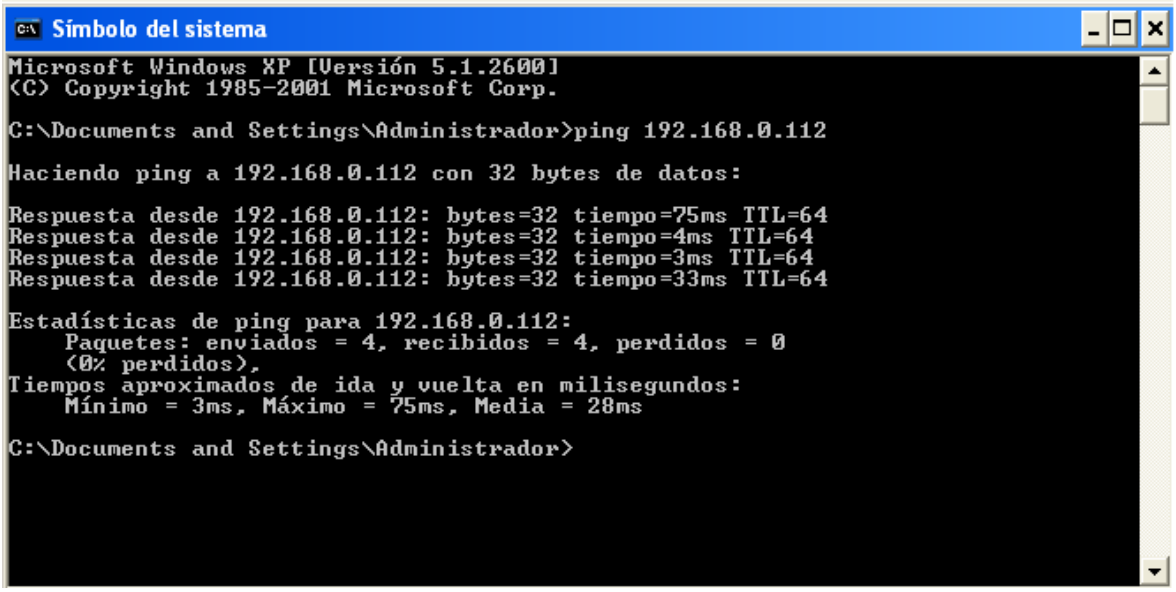


Ilustración 5-30 Host externo

5.- Ejecución de Nessus Client

Para simular la intrusión en el host de escaneo o host externo ejecutamos Nessus Client y determinamos los datos de la red a escanear, se debe tener mucho cuidado ya que estos datos deben ser coherentes con el rango asignado a la honeynet, es decir con **192.168.0.0/24**.

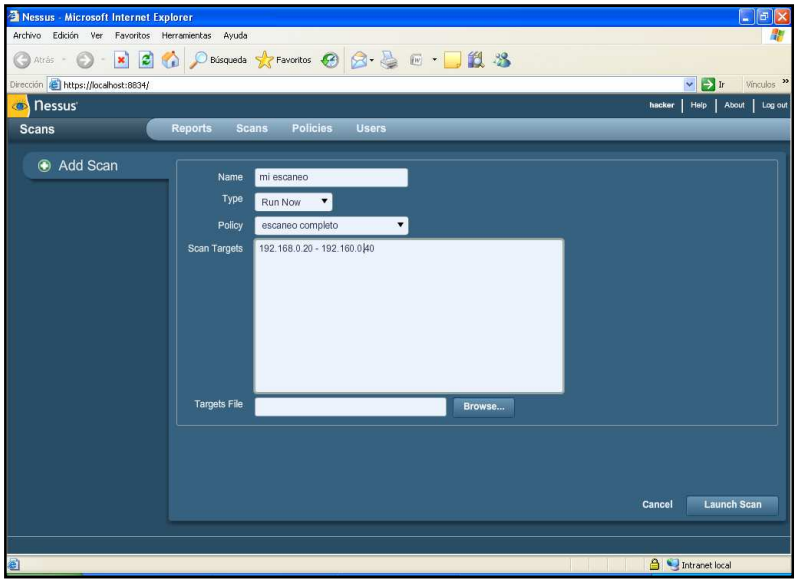


Ilustración 5-31 Ingreso de datos de la red a escanear

Inicia el escaneo a la red 192.168.0.0/24

Ilustración 5-32 Escaneo de la red

Ejecutamos Snort dentro del sistema anfitrión Centos5 para que monitoree la actividad en la red.

```
[root@localhost ~]# snort -d -h 192.168.0.0/24 -l /var/log -c /etc/snort/snort.conf
```

Ilustración 5-33 Inicio de Snort

5.6.2 IPTRAF, HERRAMIENTA PARA DIFERENCIAR EL TRÁFICO MALICIOSO

En el momento en que los servidores fallan, se comienzan a producir infracciones de seguridad y no se respetan las políticas de la empresa, es motivo suficiente para el cual un monitoreo de red constante sea la herramienta ideal para los administradores y encargados de gestionar las redes.

Anteriormente cuando no se disponía de estas herramientas de monitoreo, era necesario delegar o tercerizar este servicio a otras empresas especializadas en estos servicios y que además disponían de estas herramientas, pero el costo era muy elevado. En la actualidad podemos encontrar una gran cantidad de software para todas las plataformas y con características particulares.

Una de estas alternativas es la herramienta IPTraf, es útil para el intercambio de paquetes IP que se transmiten en la red desde y hacia las PC.

IPTraf es una de esas utilidades basadas en lo que comúnmente se conoce como interfaces **curses** "Curses es una biblioteca de control de terminal para sistemas basados en Unix, posiblemente considerada como las primeras librerías para interfaces de usuarios" y lo que hace es interceptar los paquetes que se están transfiriendo en la red para luego brindarnos información sobre los mismos. (Taringa, 2008)

Iptraf monitoreará el tráfico en las interfaces del host (sistema anfitrión de la honeynet) conformada por las UML que a su vez es el honeywall. Las interfaces a monitorear son: **192.168.0.20, 192.168.0.26 y 192.168.0.24**, si existe algún tipo de tráfico en estas interfaces el administrador del sistema de inmediato debe activar Snort de modo que se pueda capturar la intrusión, pero si lo que ocurre es solo una incursión en la interfaz **192.168.0.1** ésta se debe permitir porque puede ser un usuario común que necesita usar el servidor

web. En el caso de tener una sobrecarga de peticiones o de autenticaciones en el servidor SMTP se deberá dar una alerta sobre esta interfaz.

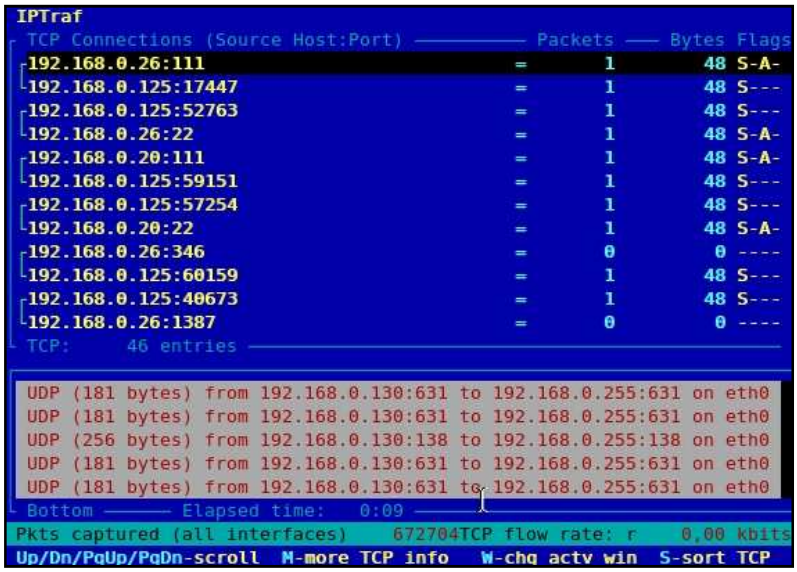


Ilustración 5-34 Tráfico cursado a la Honeynet

5.6.3 ANÁLISIS DE LOS LOGS PRODUCIDOS DESPUÉS DE LA EJECUCIÓN DE LA INTRUSIÓN.

Una vez comprobado con la herramienta Iptraf que existe una intrusión en la honeynet se deben analizar los archivos generados por las aplicaciones, estos se denominan logs del sistema. Esto nos permitirá observar que herramientas utilizó el atacante, además que servicios fueron afectados, y saber cómo se hizo la intrusión hacia la honeynet a través de la información obtenida.

Los logs principales son generados por el script **honeywall.sh** y el **sniffer snort** se encuentran dentro del directorio **/var/log**.

Los archivos generados por Snort que proporcionan mayor información son: Sfportscan y Auth.log

5.6.3.1 ANALISÍS DEL ARCHIVO LOG SFPORTSCAN

Este archivo es generado por el procesador sfportscan y alerta si se ha iniciado algún tipo de escaneo en los puertos de algún host virtual del honeyserver.

Ilustración 5-35 Archivo generado por el procesador Sfportscan

Esta información nos indica que el escaneo a los puertos se ha iniciado desde la máquina con la dirección IP **192.168.0.125** y con la utilización del software **NESSUS**.

5.6.4 RESULTADOS Y CONSECUENCIAS PRODUCIDAS POR LA INTRUSIÓN

Como resultado tenemos los logs que son generados por el escaneo de los puertos y establecemos que a consecuencia de la intrusión podemos tener un colapso parcial o total de uno o varios host del sistema.

Una vez finalizada la intrusión se observa que el host virtual UML con imagen Centos5 cuyo hostmane es **serverdns** y la IP es **192.168.0.32,** no ha sido afectada, en los log se constata que ha sido escaneado, pero no ha sido afectado.

Al analizar el honeyserver Fedora7 con una IP **192.168.0.30** de hostname **serverespe,** se verifica la serie de ataques vinculados sobre el servicio http en el **puerto 80**, los cuales han sido registrados por Snort en el archivo **auth.log**,

la serie de procesos que manejan el demonio httpd que administra el servicio web, se interrumpieron, si esto no se controla provocaría un colapso en el servicio http o del honeyserver e incluso del sistema anfitrión.
El sistema UML queda en espera sin responder con varios procesos httpd, razón por la cual corre el riesgo de que colapse.

5.7 APLICACIÓN DE POLÍTICAS PREVENTIVAS Y CORRECTIVAS EN LOS SERVIDORES

El objetivo de la honeynet es ser atacado, y no mostrar que se está dando una seguridad defensiva a una red o a un servidor, para de esta manera aprender de las intrusiones y según esto implementar políticas de seguridad para corregir las vulnerabilidades detectadas.

El principal problema que encontramos es en el honeyserver por la falta de memoria RAM así que la respectiva corrección es implementar un Swap propio para el honeyserver, esto ya lo implementamos anteriormente.

```
[root@localhost uml]# dd if=/dev/zero of=/var/local/uml/swap_fs=1M count=
1 seek=250
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0,000193 seconds, 2,7 MB/s
```

Ilustración 5-36 Sistema Operativo Anfitrión Centos5

Las políticas de seguridad son innumerables pero a pesar de esto podemos deducir que no existe la seguridad perfecta. Hemos brindado una opción que no define ni repara la seguridad pero hemos aprendido del enemigo y en base a esto se ha tomado medidas para que la seguridad de la empresa u organización no se vea tan vulnerable.

6 CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- ✓ El Honeypot además de ser una herramienta informática también es una herramienta de investigación ya que permite recoger información acerca de los atacantes y las técnicas que utilizan para ingresar a la red, esto nos ayuda a aprender las estrategias de los intrusos y de esta manera mejorar la seguridad en los sistemas.

- ✓ Snort nos ofrece capacidades de almacenamiento, puede trabajar aislado de otros sistemas de seguridad, pero dependiendo del administrador, o de las aplicaciones conjuntas, nos llenará el disco duro de información inútil, y puede colapsar el tráfico de red. En el caso de los Honeypot es utilizado como un apoyo para la captura de datos, y toda la información capturada se la considera 100% útil.

- ✓ Nessus es un software que no solamente puede ser utilizado para simular intrusiones sino que además alerta acerca de configuraciones incorrectas en los firewalls, detecta la falta de parches y actualizaciones en los sistemas de la Compañía.

6.2 RECOMENDACIONES

- ✓ Utilizar el Honeypot no solo como medida de seguridad sino como una herramienta para la investigación de los estudiantes ya que constituye un recurso educativo de naturaleza demostrativa cuyo objetivo se centra en aprender patrones de ataque y amenazas de todo tipo.

- ✓ Trabajar con otro IDS como por ejemplo Sebek, el cual está orientado a registrar las pulsaciones de teclado del atacante, permitiéndonos así conocer de mejor manera las técnicas utilizadas por los intrusos.

- ✓ Debido a las múltiples prestaciones que tiene la herramienta Nessus, usarla no solo para simular ataques sino también para localizar debilidades y vulnerabilidades antes de que un intruso lo haga.

REFERENCIAS BIBLIOGRÁFICAS

- ✓ (s.f.). Obtenido de Introducción a Snort: http://www.maestrosdelweb.com/editorial/snort/
- ✓ /clamav, I. o. (2005). *Index of /clamav*. Obtenido de Index of /clamav: http://packages.sw.be/clamav/
- ✓ Andersson, O. (2001). *Tutorial de IPtables 1.1.19es*. Obtenido de ie.etcr: http://www.ie.itcr.ac.cr/marin/telematica/wan/iptables-tutorial.es.pdf
- ✓ *Apache*. (2011). Obtenido de http://httpd.apache.org/docs/2.0/es/bind.html
- ✓ Balboa, M. A. (s.f.). *Instalacion y configuracion de un servidor DNS en linux*. Obtenido de http://www.inegi.gob.mx/inegi/contenidos/espanol/prensa/contenidos/articulos/tecnologia/dns03.pdf.
- ✓ Blanco Ramos, O., Alamillo i Domingo, I., Gutierrez Covarrubias, M. R., León, P., Ortega García, R., Rojas de la Escalera, D., y otros. *Honeypots* (Marzo 2008 ed.).
- ✓ Bulma. (05 de 11 de 2006). *Iptables*. Obtenido de listes.bulma: http://llistes.bulma.net/pipermail/bulmailing/Week-of-Mon-20061030/079072.html
- ✓ Cabrera, N. L.-M. (23 de 10 de 2008). *FIREWALL Configuración básica de iptables para VoIP*. Obtenido de gt voip: http://www.voip.unam.mx/mediawiki/index.php/FIREWALL_Configuración_básica_de_iptables_para_VoIP#.C2.BFQu.C3.A9_es_Iptables.3F
- ✓ Cano Nuñez, J. *Honeypost: Alta Interacción* (2006/06/05 ed.).
- ✓ *Clamav*. (2002). Obtenido de http://www.clamav.net/lang/en/

- ✓ Cócaro, F., García, M., Jose, M., & Rouiller. *Ubicación de los Honeypots* (Julio 2007 ed.).
- ✓ Coletti, D. E. (27 de 06 de 2003). *Tablas, cadenas y reglas*. Obtenido de Tablas, cadenas y reglas: http://www.danielcoletti.com.ar/Documentos/Tech/Iptables/iptables/node7.html
- ✓ EPN. (2009). Obtenido de http://bibdigital.epn.edu.ec/bitstream/15000/1523/1/CD-2231.pdf
- ✓ Gallego, E., & Lopez de Vergara, J. E. *Honeynet: Aprendiendo del atacante "Honeynet Virtual"*. España, Madrid.
- ✓ García, Jess. *Herramientas_Honeypots* (2006/12/24 ed.).
- ✓ *Honeynet Project*. (s.f.). Obtenido de http://his.sourceforge.net/honeynet/papers/uml/
- ✓ Izura, P. X. (s.f.). *Iptables - Manual Practico*. Obtenido de http://www.pello.info/filez/firewall/iptables.html
- ✓ Jara, C., Gaeta, M., & Villalón, N. *Honeypot: Honeynets Virtuales* (2008/11/20 ed.).
- ✓ *Manipulación de filtros*. (s.f.). Obtenido de http://www.bdat.com/documentos/cortafuegos/x195.html
- ✓ Portatiles, O. y. (s.f.). *Servidores web*. Obtenido de http://www.ordenadores-y-portatiles.com/servidor-web.html
- ✓ Quiroz, B. (s.f.). *Linux tambien como firewall*. Obtenido de http://jornadas.lugmen.org.ar/files/material/Quiroz-B_Firewall_20061005.pdf
- ✓ Segu-Info. (2009). *Políticas de Seguridad*. Obtenido de Segu-Info: http://www.segu-info.com.ar/politicas/
- ✓ Taringa. (2008). *Monitorizar Redes con IPtraf*. Obtenido de http://www.taringa.net/posts/linux/2151751/Monitorizar-Redes-con-IPTraf.html

- ✓ wikilearning. (2007). *IPtables*. Obtenido de wikilearning: http://www.wikilearning.com/monografia/iptables_guia_rapida-como_se_escribe_un_script_para_iptables/5838-5
- ✓ Wikipedia. (25 de 04 de 2011). *Políticas de Seguridad*. Obtenido de Wikipedia: http://es.wikipedia.org/wiki/Pol%C3%ADticas_de_seguridad
- ✓ Wikipedia. *User Mode Linux* (2009/08/04 ed.).
- ✓ *Youtube - Descubre vulnerabilidades en tu sistema con Nessus*. (s.f.). Obtenido de http://www.youtube.com/watch?v=BzIYec7mR7U