Honeynets como herramienta de prevención e investigación de ciberataques

Casanovas, Eduardo Esteban
Instituto Universitario Aeronáutico

Tapia, Carlos Ignacio
Instituto Universitario Aeronáutico

Abstract
Uno de los mayores retos de la Seguridad Informática como disciplina es la captura y análisis de logs de auditoría útiles para detectar ciberataques, y a la vez, poder interpretarlos y elaborar estrategias de mitigación. Con los ambientes de IT cada vez más complejos y con una enorme cantidad de eventos de auditoría siendo generados por los distintos dispositivos, es necesario encontrar una herramienta que permita discriminar rápida y fácilmente aquellos recursos bajo ataque y garantizar el aislamiento para observar en tiempo real los ataques o posibilitar el análisis forense posterior. Las herramientas a las que el presente paper hace foco son las que conforman las honeynets, más específicamente, honeynets virtuales diseñadas para atraer, engañar y capturar los ataques ya sea de las cada vez más crecientes botnets como así también ataques dirigidos por parte de intrusos que busquen acceder a información confidencial de una entidad. Este proyecto pretende demostrar las funcionalidades básicas de una honeynet y sus bondades, proponiendo un ciclo de vida para la misma y una arquitectura inicial para efectuar pruebas, la que servirá de base para que la honeynet crezca en cantidad de nodos y complejidad, registrando cada vez información más valiosa respecto a los ciberataques que se ejecuten contra dicha red ficticia especialmente preparada para tal fin.

Palabras Clave
Honeynet. Honeypot. IDS. Detección de intrusos. Computación forense. Mitigación de botnets.

Introducción

La Seguridad es ahora no un simple tema más a tratar dentro de las áreas de IT de las organizaciones si no que ha pasado a ser un requisito obligatorio para el cumplimiento de los objetivos. Con una conectividad cada vez más accesible, de mayor calidad y rapidez, la proliferación de amenazas tiende a crecer y a ser cada vez más compleja, como así también el nivel técnico de los atacantes se incrementa. Es vital poder seguir el ritmo a dichas amenazas y poder analizar el comportamiento de los atacantes para contar con información de primera mano respecto de las acciones que realizan o qué herramientas están utilizando y las estrategias de ataque puestas en juego.
Una herramienta, o más bien, una serie de herramientas que las organizaciones tienen a su disposición para contrarrestar y a la vez estudiar a los ataques son las honeynets.

Al hablar de honeynets, debemos definir antes honeypot: un honeypot es un software cuya intención es atraer a atacantes, simulando ser un sistema vulnerable o débil a los ataques. Es una herramienta de Seguridad Informática utilizada para recoger información sobre los atacantes y sus técnicas. La esencia de los honeypots es la contrainteligencia, teniendo en cuenta que se busca engañar a los intrusos invitándolos a sistemas supuestamente vulnerables, para beneficio del administrador del honeypot.

Los honeypots pueden distraer a los atacantes de las máquinas “reales” más importantes de la red, y advertir rápidamente al administrador del sistema de un ataque, además de permitir un examen en profundidad del atacante, durante y después del ataque al honeypot. Este concepto está muy ligado al análisis forense, teniendo en cuenta que el honeypot, luego de un ataque, contará con información valiosa respecto del atacante, sus estrategias e intenciones, pudiendo transformarse esta información en evidencia ante un potencial litigio. Aún si no se tuviera como objetivo la utilización de la

información contenida en el honeypot como evidencia legal, desde luego tiene un alto valor académico y de investigación para comprender las técnicas utilizadas por los intrusos, permitiendo mejorar las herramientas detectivas y preventivas de la red.

Un modelo conceptual que representa a un honeypot se compone de los siguientes elementos [1]:

A) ***Sistema señuelo***: si bien lo parece, no es un sistema productivo de la organización. Es un target atractivo para los intrusos, los cuales deben pensar que allí se almacenarán datos confidenciales, contraseñas, detalles de transacciones, o cualquier otra información que pudiera ser interesante. Mientras más atractivo, realista y complejo sea este componente, más posibilidades de éxito tendrá el honeypot.

B) ***Firewall***: en el modelo, el firewall provee logs de auditoría acerca de los intentos del intruso por acceder al honeypot. El firewall se configura para registrar todos los paquetes yendo al sistema señuelo, teniendo en consideración que ningún tráfico válido se dirigiría a ese host. Se hace hincapié en este concepto: el sistema señuelo se creó exclusivamente para ser atacado, con lo que, por definición, todo tráfico destinado a dicho sistema debe ser considerado hostil y sujeto a mayor análisis.

C) ***Unidad de monitoreo***: es un componente de evaluación de amenazas que supervisa las actividades maliciosas o violaciones de políticas sobre la red y/o sistemas, produciendo reportes que luego podrán ser consultados por el administrador de la honeynet. Con medidas como la revisión de las secuencias, las timestamps y los tipos de paquetes utilizados por el intruso para acceder al honeypot, y también como la presión de teclas, accesos a los sistemas, archivos cambiados, etc., se busca identificar las herramientas y metodologías utilizadas por los atacantes y sus intenciones. Normalmente, en el marco de un honeypot, esta tarea es delegada a un IDS (Intrusion Detection System).

D) ***Unidad de alertas***: los honeypots deben ser capaces de generar y enviar avisos a través de diferentes medios a su administrador para permitir la revisión de las actividades de los intrusos mientras están ocurriendo.

E) ***Unidad de registro***: este componente provee eficiencia en el almacenamiento tanto para los logs del firewall como del honeypot en sí, y todo el tráfico que circula entre el firewall y el honeypot.

Teniendo en cuenta los 5 componentes esenciales que conforman un honeypot funcional, en forma ideal su disposición en una red sería la que se grafica a continuación:

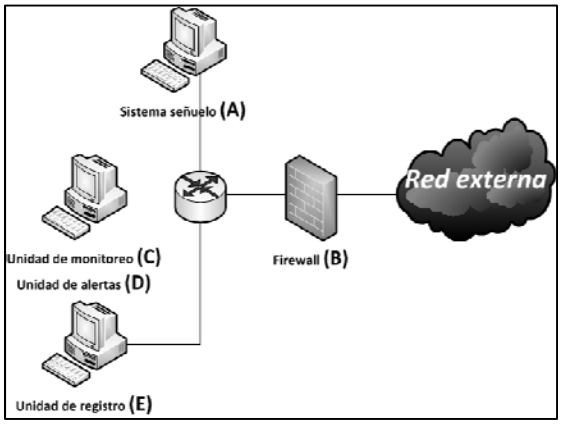

Figura 01: Componentes genéricos de un honeypot

Habiendo analizado el concepto anterior, una honeynet puede ser definida como un conjunto o red de honeypots que simula una

red productiva, creada para atraer atacantes, posibilitando disminuir ataques a los recursos informáticos reales y el estudio de dichos ataques para generar contramedidas.

Con una honeynet, no sólo se disponen honeypots aislados, si no que se genera una infraestructura de red que interconecta dichos honeypots, capturando en todo momento la actividad de los intrusos y garantizando la separación lógica de la red productiva de la organización. Esto último es clave, porque si no se pudiera garantizar dicha separación, los recursos informáticos productivos de la organización se encontrarían en peligro. Es vital que el software que genera la honeynet tenga configuraciones tales que impidan el lanzamiento de ataques desde los honeypots a la red real de la organización. Es perfectamente esperable que los honeypots sean comprometidos, pero desde luego, se busca que el accionar de los intrusos quede circunscripto sólo a la red ficticia creada con la finalidad de monitoreo e investigación.

Si bien, habitualmente están dirigidas a ambientes de investigación y generalmente son implementadas en universidades o en empresas dedicadas a la investigación de ataques informáticos, la aplicación práctica en cualquier tipo de organización es posible. Desde luego, es esencial que la implementación de esta tecnología sea lo más sencilla posible, como así también su operación y el análisis de los resultados, para evitar que se invierta tiempo, esfuerzo y dinero en una iniciativa que al poco tiempo pierda empuje porque no fue correctamente utilizada o mantenida.

No es poco común observar que muchos proyectos iniciados sobre honeynets fueron simples iniciativas académicas producto de alguna asignatura o tesis universitaria, que al cabo de un tiempo, deja de generar estadísticas y cesa su utilización, quedando obsoleta.

Cabe destacar en cuanto a proyectos relacionados con el tema al Honeynet Project, el cual tiene actualmente más de 40 capítulos en distintos países [2], siendo la información obtenida de ataques compartida entre dichos nodos pudiendo determinar tendencias de ataques y efectuar pronósticos para preparar, en la medida en que así sea posible, estrategias de mitigación conjuntas.

El objetivo del presente trabajo es demostrar que esta tecnología puede ser implementada con herramientas simples, open source, con una operación y mantenimientos relativamente sencillos.

Elementos del Trabajo y metodología

Para la implementación de la honeynet contemplada en el presente trabajo se generarán honeypots de alta interacción [1], es decir que en ellos se monta un sistema operativo “real” con servicios “reales” para ser atacados. Si bien son sumamente atractivos para los atacantes, el éxito de un ataque puede implicar el apoderamiento del honeypot por parte de los intrusos o su salida de línea. Teniendo en cuenta que se utiliza software válido y que luce exactamente como productivo, se recopila con este tipo de honeypot una gran cantidad de información acerca de los ataques, del atacante y sus tácticas, permitiendo un análisis posterior de profundidad. El costo de estas bondades es que son relativamente difíciles de montar y más complicados de mantener. Si bien habitualmente se los despliega con fines de investigación de malware, exploits y vulnerabilidades, con un ambiente controlado es perfectamente posible implementar honeypots de alta interacción en producción.

Asimismo, los honeypots a utilizar y la propia honeynet que los contendrá serán virtualizados [1], entendiéndose por esto que toda la infraestructura de red, como así también todo el software de los honeypots se ejecutarán en máquinas virtuales. Esta

decisión se toma para lograr el máximo aprovechamiento, tanto de CPU como de memoria RAM del equipo físico a utilizar para el montaje de la honeynet. Desde luego que, la mencionada decisión de implementación no elimina la posibilidad de que, dado un crecimiento en la honeynet, ya sea en tráfico, necesidades de procesamiento o aumento de la cantidad de honeypots y/o su complejidad y realismo, la honeynet virtual se interconecte a otras honeynets virtuales o inclusive a otras redes/equipos físicos. El modelo de por sí es escalable, pudiéndose agregar n honeypots según el diseño de la red "trampa" que se quiera desarrollar y el tipo de ataques que se pretenda recibir.

Teniendo en cuenta la arquitectura de la honeynet y los avances tecnológicos en cuanto a monitoreo de intrusos, se habla distintas "generaciones" de honeynets. La generación I, si bien es exitosa a la hora de capturar ataques sofisticados o de principiantes, son muy limitadas para controlar y contener a los atacantes. Asimismo, en esta clase de honeynets es relativamente sencillo descubrir que los sistemas atacados no son reales. Estos problemas encuentran su solución con la aparición de las honeynets de segunda generación. La primera diferencia es que se utiliza un único sensor de honeynet. El sensor de la honeynet combina la funcionalidad de un IDS y también de un firewall, entonces, en lugar de tener que desplegar 2 componentes, ahora se instala 1 sólo. La diferencia pasa por la utilización de un firewall de capa 2, en donde convergen los servicios de IDS y firewall tradicionales.

La segunda diferencia relevante está en el sensor de la honeynet en sí. El sensor es de capa 2, de comportamiento similar a un puente (hub de 2 bocas). Esto hace que el dispositivo sea virtualmente invisible para los atacantes. En oposición al firewall de capa 3 de las honeynets de primera generación, con este sensor de honeynet no hay ruteo de paquetes, no se da el decremento de TTL, ni hay una dirección MAC que el intruso pueda detectar. Este dispositivo es bastante difícil de detectar ya que es transparente para los atacantes. Más allá de esto, como en las honeynets Gen I, todos los paquetes entrando o saliendo la honeynet deben pasar por el sensor.

Teniendo en cuenta que el sensor es de capa 2, las honeynets Gen II pueden conectarse directamente a redes productivas en lugar de propiciar mayor aislamiento como sí se debía hacer con las honeynets Gen I. La separación está sucediendo a nivel de capa 2, en lugar de capa 3 o capa de Red.

Independientemente de los cambios de arquitectura, los mecanismos de control de datos en las honeynets Gen II son sustancialmente distintos en relación a las Gen I. En lugar de utilizar un firewall de capa 3 que aplique ACLs basándose en los encabezados IP, las honeynets Gen II usan una tecnología llamada "IDS gateway". Un gateway IDS es un dispositivo similar a un firewall en el sentido de que controla el acceso a los recursos de la red. Sin embargo, un gateway IDS no sólo bloquea conexiones basándose en servicios, sino también posee la "inteligencia" suficiente para distinguir entre un ataque y tráfico legítimo. El gateway IDS tiene incorporada una base de datos de firmas. Cuando un ataque conocido coincide con una firma de la base de datos, la conexión puede ser bloqueada. Esta tecnología presenta varias ventajas cuando es usada para control de datos:

a) La primera ventaja es que el IDS tiene "inteligencia" para analizar la actividad maliciosa. En lugar de observar la cantidad de conexiones entrantes o salientes, revisa específicamente la actividad que se está desarrollando. A pesar de lo dicho, los gateways IDS tienen ciertas limitaciones: sólo pueden detectar ataques conocidos. Es por esto que esta tecnología es combinada

habitualmente con conceptos de honeynets Gen I.

b) Luego, como segunda ventaja tenemos la manera en que las honeynets Gen II responden a la actividad no autorizada. En lugar de simplemente bloquear conexiones, se modifican o regulan las actividades de los atacantes. Estas respuestas serán mucho más difíciles de detectar por parte de los intrusos, aparentando el sistema un comportamiento normal como si fuera productivo. Esto se logra mediante la modificación de los paquetes a medida que atraviesan el gateway de capa 2. Por ejemplo, si un atacante compromete un honeypot, quizás intente lanzar un exploit contra sistemas existentes fuera de la honeynet. Teniendo en cuenta este ejemplo, en una honeynet Gen I, el control de datos es limitado en el sentido de que luego del décimo intento de conexión saliente, toda la actividad siguiente incluyendo el exploit es bloqueada. Sin embargo, teniendo el mismo ejemplo en una honeynet Gen II, el intento de exploit sería identificado y luego modificado para que el ataque sea inefectivo. El Gateway de capa 2 modificaría varios bytes del código del exploit, deshabilitándolo y malogrando el ataque. El atacante vería que el ataque se lanzó y los paquetes de regreso, pero no comprendería por qué no funcionó el exploit. Este método nos permite conocer las intenciones y técnicas del atacante, sin que éste lo sepa.

Respecto del software con el que se generará la honeynet, se contempla la utilización de Honeywall [3], el cual es un compendio de utilidades de licencia libre que permite, de una manera sencilla generar una honeynet de alta interacción conectando a ella la cantidad de honeypots que se requiera.

Al disponer una honeynet de alta interacción, todos los hosts, sistemas operativos y servicios existentes en la red creada a los efectos de detectar e investigar ataques son reales y efectivamente están corriendo, ya sea en hardware dedicado o en ambientes virtualizados (o un ambiente mixto). Dicho de otra manera, no se utilizarán meras emulaciones de servicios de red, si no servicios que efectivamente estén escuchando y atendiendo peticiones de clientes.

Honeywall es un producto desarrollado por el Honeynet Project [4], entidad de investigación de temas de Seguridad Informática sin fines de lucro, que, como se comentara en la Introducción, se dedica a investigar los últimos ataques y a desarrollar software de seguridad de código abierto con el fin de mejorar la seguridad de Internet.

El software de implementación de honeynets Honeywall Roo v1.4 fue lanzado en Abril de 2009, siendo para esa época, su Sistema Operativo vigente junto con su paquete de aplicaciones asociadas. Asimismo, es válido destacar que el software sobre el cual estaba sustentado Honeywall en su lanzamiento estaba, desde luego, preparado para funcionar en hardware de dicha época y limitado por su rendimiento.

Teniendo en cuenta lo dicho en el párrafo anterior e independientemente de los resultados positivos que Honeywall pueda entregar, es importante recalcar que, por ejemplo, Snort (software IDS) ya se encontraba desactualizado al momento de lanzamiento de Roo 1.4. De manera similar, el mismo Sistema Operativo CentOS 5 no tiene sus repositorios de actualización activados, con lo que una vez en funcionamiento, no recibe parches. Con estas dificultades de actualización del Sistema Operativo subyacente y del paquete de aplicaciones, la comunidad usuaria de Honeywall clama por una profunda reestructuración del ambiente.

A continuación, analizamos los componentes de Honeywall, comenzando por el software estándar:

a) ***Argus***: típicamente, Argus observa el flujo de datos pasando a través de la interfaz de red y los flujos dirigidos hacia la placa de red desde, como máximo, 5 outputs [5]. Su propósito es generar flujos de datos que correlacionen tanto los paquetes de datos asociados como el total de tiempo de las comunicaciones, sirviendo como método de auditoría de datos de red. En Honeywall, los datos son recolectados de una manera tradicional y ruteados a través del proceso Hflowd donde se correlacionan, formatean y filtran con otra información coleccionada desde otro software (p0f y Snort) y solamente una representación reducida de estos registros de actividad de sistemas es almacenada en la base de datos Hflow.

b) ***Iptables***: ésta no es más que una de las herramientas de la suite de NetFilter para Linux [6]. La implementación de Iptables para Honeywall no tiene ninguna variante especial o detalle particular, solamente su inclusión tiene como objetivo proveer cierta apariencia de seguridad para la red ficticia. Teniendo en cuenta esto, se espera que Iptables actúe como firewall en la interfaz bridge de la red de Honeywall regulando los paquetes que ingresan y egresan la honeynet, siendo esto no tanto para prevenir que los atacantes entren a la red falsa como sí apunta a hacerlos pensar que están en una red productiva verdadera, surtiendo efecto la finalidad de la honeynet. El rol de Iptables respecto de la interfaz de administración sería el de un firewall tradicional permitiendo el acceso sólo a los usuarios y administradores de Honeywall. Esto se logra mediante la imposición de la restricción de las direcciones IP de hosts/redes específicos a los cuales se les permite acceso a la interfaz de administración mediante reglas de permiso sólo a cierto tipo de tráfico como lo es el del puerto UDP 53 (DNS), TCP 80 (HTTP) y TCP 443 (HTTPS).

c) ***p0f***: Es una utilidad para realizar fingerprinting de sistemas operativos que es muy versátil y rápida [7]. Es útil para el monitoreo del tráfico que ingresa a la honeynet ya que puede ayudar a identificar aquellas direcciones IP en las cuales el sistema operativo va cambiando y permite mejorar la exactitud de los eventos de IDS. La información de p0f se correlaciona con Argus y Snort para obtener una imagen más clara de los flujos de información conocidos y luego escritos en la base de datos Hflow2.

d) ***Snort***: es un famoso IDS de red [8], gratuito y de código libre, que actúa como gateway para toda la información colectada en Honeywall. Los datos recapturados a partir de las demás herramientas (Argus, p0f y Sebek) sólo son escritos en la base de datos Hflow2 si el primero de los paquetes de un flujo de datos coincidió con una regla de Snort (los paquetes restantes son descartados). Sin embargo, una copia “cruda” de los datos es guardada en un archivo .pcap para posibilitar la investigación más minuciosa de aquellos paquetes que hayan sido “salteados” por las reglas de Snort. En Honeywall, los datos que son almacenados en la base de datos Hflow2 están disponibles a través del método de acceso “fast path” mientras que la información “cruda” es accesible por la “slow path”. Un problema manifiesto se sucede en este sentido: Honeywall viene con un la versión 2.6 de Snort,

obsoleta ella, y al ya no ser una versión soportada, ésta no recibe más actualizaciones de reglas. Es relevante este comentario respecto de la versión de Snort, teniendo en cuenta interactúa con otro software para aumentar su performance.

A continuación, se expone y describe el software no estándar desarrollado exclusivamente para Honeywall:

a) ***Hflow2***: es un software de recopilación y correlación de los datos obtenidos a partir de Snort, p0f, Argus y Sebek, ensamblando dichos datos en un flujo unificado el que se almacena en una base de datos particular [9]. La base de datos Hflow2 funciona como método de acceso "fast path" a la información recolectada para conseguir alto nivel de comprensión con cierto nivel de degradación en el detalle. Funciona utilizando a las colecciones de datos de los demás utilitarios (Argus, Snort, p0f y Sebek) como módulos que transmiten datos dentro del proceso Hflowd. Las porciones de flujo que se distinga que pertenecen a un único flujo bidireccional de datos de red son etiquetadas e insertadas en la base de datos. Hflow2 también actúa como el controlador de los inicios para las dependencias entre los módulos, ayudando al aseguramiento de que el proceso no genere errores debido a que alguno de los módulos no esté disponible.

b) ***Sebek***: Este utilitario [10] es un keylogger que también registra las llamadas API que son realizadas al kernel de la máquina host. Es un software que trabaja con la arquitectura cliente-servidor, en la cual el cliente se encuentra instalado en los honeypots de la honeynet y el componente servidor corre en el servidor Honeywall. Los datos recolectados de los clientes son transmitidos al servidor mediante formato cifrado con los llamados "paquetes sebek", los cuales son luego incorporados al proceso Hflowd para su recopilación con el flujo de datos de red antes de ser insertados finalmente en la base de datos Hflow2. Es válido recalcar que Sebek es soportado sólo por pocos y viejos Sistemas Operativos, no estando en los planes su desarrollo futuro. Actualmente, existe una iniciativa dentro del Honeynet Project para migrar de Sebek a Quebek (keylogger especializado que no se ha expandido mucho).

c) ***Walleye***: Esta es una aplicación desarrollada por el mismo proyecto Honeynet Project [11], inicialmente pensada para probar el método de acceso a datos "fast path" y sus capacidades de análisis. Se considera como la herramienta primaria para configurar y monitorear el ambiente de Honeywall. A través suyo se pueden definir la mayoría de los parámetros más importantes de manera rápida y ayudando a los nuevos administradores para que sea comprensible a los distintos niveles de capacidades técnicas. Las excepciones, es decir, los parámetros que no se pueden configurar a través de Walleye y que deben establecerse en cada utilitario o en el Sistema Operativo son:

- Zona horaria.
- Tripwire [12].
- Postfix [13].

También, Walleye es el medio a través del cual se examina la "fast path". Dentro de la aplicación, es posible analizar en detalle cada flujo de datos, permite filtrar por día y hora. Además, es posible aislar más aún los flujos de datos IP destino y origen, y/o también por números de puertos. De esta forma, Walleye

permite ver la comunicación relevante entre 2 hosts sin distraer con otra información capturada que no sea de interés para el análisis que se pretenda llevar adelante. Es importante tener en cuenta, además, que se puede acceder a los logs "crudos" del firewall y Snort, habilitando al administrador a una revisión a nivel de paquetes del flujo de datos.

Interpretando la implementación de la honeynet como un ciclo de vida, podríamos destacar las siguientes fases:

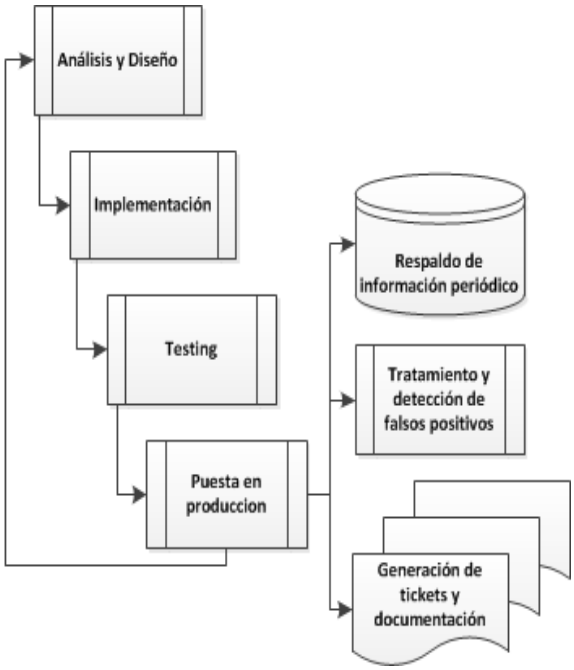

Figura 02: Ciclo de vida de la honeynet

- ***Análisis y Diseño*:** en esta etapa se estudiará el nivel de complejidad que se debe otorgar a la honeynet, la cantidad y tipo de honeypots que se dispondrán. También se documentarán todas las configuraciones más relevantes tanto de Honeywall como de los honeypots para utilizar en la siguiente fase.

- ***Implementación*:** en la presente fase, se dispone el software de virtualización, las máquinas virtuales con sus respectivos componentes de hardware virtual, se instala el software de base, se instalan y configuran los servicios que se expondrán a los atacantes de modo tal que el esquema quede operacional.

- ***Testing*:** se plantean casos de uso para corroborar que la arquitectura se desempeña como se espera, es decir, se realiza una serie de ataques a los honeypots y se deja asentada la reacción de los distintos componentes de la honeynet, lo que se puede visualizar en la interfaz gráfica de Honeywall (Walleye).

- ***Puesta en producción*:** la habilitación en ambiente productivo hace que la honeynet quede disponible a los atacantes para los cuales se pretende realizar el estudio. Como derivación de esta fase es vital que se efectúe periódicamente el respaldo de los logs generados por las distintas herramientas que componen Honeywall. Adicionalmente, también es crucial se detecten, discriminen y traten aquellos eventos "falsos positivos", de modo que se evite considerar ataques que no fueron tales. Finalmente, es necesario realizar la documentación oportuna de los eventos de ataques "reales" más relevantes que se vayan detectando en la honeynet, pudiendo utilizarse a tal fin un sistema de gestión de tickets (por ejemplo, Mantis [14]).

En la página siguiente, se presenta la arquitectura de red de la honeynet inicial que, desde luego, podrá ir ganando (y es esperable que así sea) en complejidad y en cantidad de honeypots, dispositivos de networking, aplicaciones ficticias que parezcan ser sistemas productivos con información realista pero no verdadera.

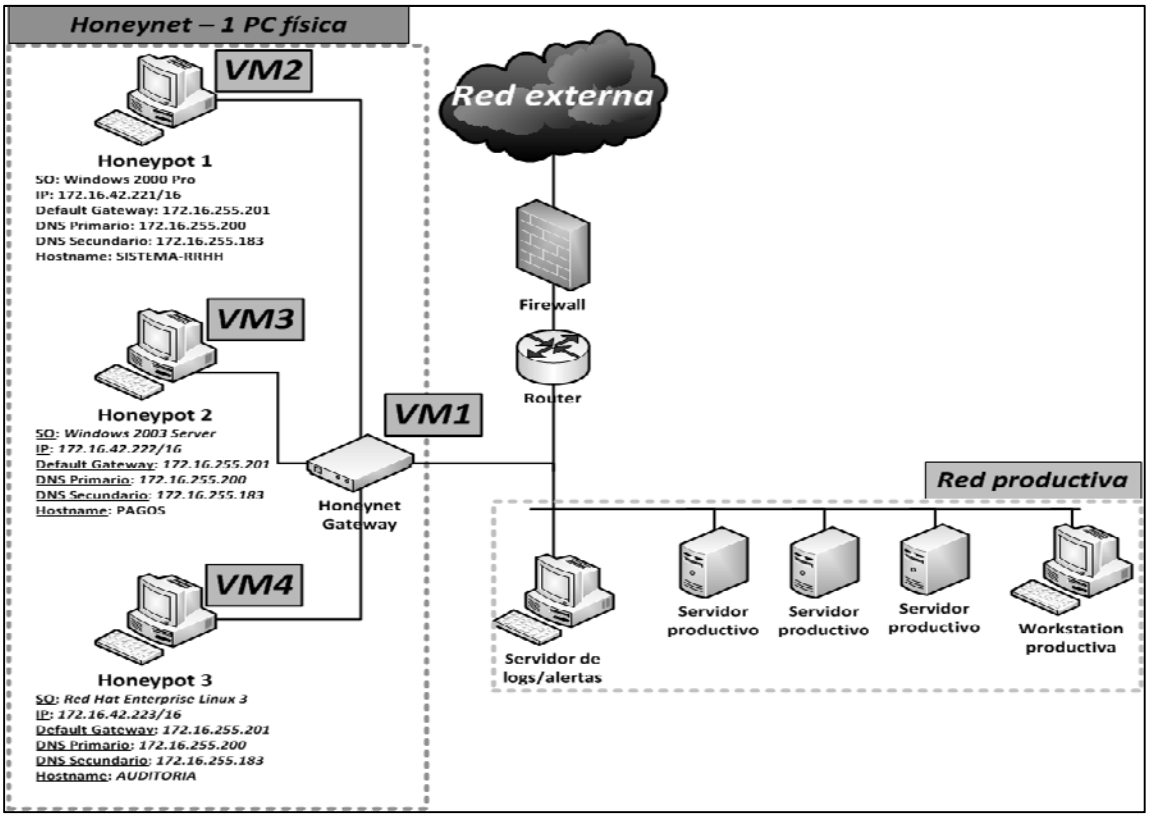

Figura 03: Implementación inicial de la honeynet

Resultados

La implementación inicial de la honeynet mediante el software Honeywall se comportó de la manera esperada, es decir, luego de su instalación y configuración se realizó un testing básico desde un host dispuesto como "intruso" en la red equipado con herramientas de pentesting esenciales, ejecutándose los siguientes ataques sobre los honeypots a modo de prueba:

01) Ping al honeypot 1.
02) Portscan con Nmap [15] al honeypot 3.
03) Telnet a puerto TCP 47001 del honeypot 2.
04) Brute Force con Bruter [16] a puerto TCP 8585.
05) Obtención de shell de Windows con Metasploit [17] al honeypot 1.

En todos los casos, Honeywall reportó las actividades sospechosas permitiendo la documentación de cada caso. Esta información, acumulándose en el tiempo, permite observar los comportamientos a modo de tendencia o conocer las estrategias de ataque y hasta las herramientas utilizadas por los intrusos.

Discusión

Con la implementación inicial de la honeynet se pudo observar que, más allá de ciertos inconvenientes como por ejemplo, la falta de documentación actualizada por parte del Honeynet Project en relación a Honeywall y sus herramientas, el objetivo de puesta en producción se cumple con relativa facilidad. Asimismo, la interfaz de Honeywall es suficientemente amigable como para recorrer los eventos y rescatar los más importantes para dar seguimiento y documentar.

Un tema de relevancia que deberá ser abordado como continuación del proyecto es el hecho de que el software de base

utilizado por Honeywall ya es obsoleto y la mayoría de sus herramientas no cuentan con actualizaciones. Respecto de este tópico, una alternativa a considerar es la de adaptar una distribución de Linux específica y acotada, con herramientas que permitan generar la honeynet que mantengan su actualización y soporte en el tiempo.

Ya sea con la implementación actual de honeynet mediante Honeywall o a través de otra infraestructura de software de base y aplicaciones destinadas a generar la red ficticia, se pretende como siguiente paso, exponer dicha red a la influencia de una botnet para observar su comportamiento y capacidad de contención.

Conclusión

Como resultado del presente trabajo se enumeran los objetivos logrados luego de efectuar el análisis de la tecnología de honeynets y realizar pruebas de funcionamiento sobre ella mediante el paquete de utilidades Honeywall:

- La herramienta es de instalación relativamente simple y rápida.
- Su operación es sencilla a través de interfaz gráfica.
- La arquitectura es escalable según las necesidades.
- El ciclo de vida expuesto es sustentable y adecuado para el correcto aprovechamiento de la honeynet.
- Fue posible contener y aislar los ataques.
- Honeywall mostró los ataques y fue posible documentar los mismos, permitiendo su estudio en mayor detalle.

Habiendo sido cumplidos exitosamente los objetivos definidos para el proyecto, la continuación del mismo deberá tener foco en la actualización y/o desarrollo de aquellas herramientas que han dejado de tener soporte, de modo que se logre una plataforma estable y sustentable a futuro, teniendo en consideración la alta velocidad con la que surgen nuevas amenazas en las redes actuales.

Referencias

[1] R.C. Joshi, Anjali Sardana. “Honeypots, a new paradigm to Information Security”. CRC Press. 2011.

[2] The Honeynet Project. Project Chapters. http://www.honeynet.org/og. 2013.

[3] Lance Spitzner. Honeywall CDROM. http://www.honeynet.org/project/HoneywallCDROM. 2008.

[4] THP. http://www.honeynet.org/. 2013.

[5] Carter Bullard. http://qosient.com/argus/. 2013.

[6] The Netfilter Core Team. Netfilter “iptables” project. http://www.netfilter.org/projects/iptables/index.html. 2013.

[7] Michal Zalewski and William Stearns. Passive OS Fingerprinting Tool. www.stearns.org/ p0f /README. 2004.

[8] Snort Team. Snort Official Documentation. http://www.snort.org/docs. 2013.

[9] Christian Seifert. HFlow2 - Data Analysis System. http://www.honeynet.org/project/HFlow2. 2011.

[10] Lance Spitzner. Sebek. http://www. honeynet.org /project /sebek. 2008.

[11] Edward Balas, Camilo Viecco. Towards a third generation data capture architecture for honeynets. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1495929. 2005.

[12] Allison Hrivnak. SANS Institute. Host Based Intrusion Detection: An Overview of Tripwire and Intruder Alert. 2002.

[13] Wietse Venema. Postfix Documentation. http://www.postfix.org/documentation.html. 2013.

[14] The MantisBT Team. MantisBT Documentation. http:// www. mantisbt. org /documentation .php. 2013.

[15] Gordon “Fyodor” Lynn. Chapter 8. Remote OS Detection. http://nmap.org/book/osdetect.html. 2009.

[16] Worawita. Bruter tool. http://sourceforge.net / projects / worawita/. 2013.

[17] Metasploit Help by Rapid7. http://help.metasploit .com. 2013.