Honeynets: Aprendiendo del Atacante

Eduardo Gallego, Jorge E. López de Vergara
{egallego, jlopez}@dit.upm.es
Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid
E.T.S.I. de Telecomunicación, Av. Complutense, s/n, 28040 Madrid, España
Tel. +34 91 549 57 00, Fax +34 91 336 73 33

RESUMEN: Una *honeynet* es una herramienta de seguridad diseñada para ser sondeada, atacada y comprometida por un hipotético intruso. Se trata de una red completa, compuesta por un conjunto de sistemas dispuestos a recibir estos ataques y una serie de mecanismos encargados de la monitorización, el registro y el control de estas acciones. Es una especie de pecera, en cuyo interior se puede observar al atacante en su hábitat natural.

Mediante el estudio del comportamiento de los intrusos durante el sondeo, el ataque y el compromiso de los sistemas de la *honeynet* y el análisis de su posterior actividad en el interior de los sistemas comprometidos, es posible aprender sobre las tácticas y los motivos de la comunidad de atacantes que puebla Internet. Esta ponencia presenta a las *honeynets* como el *honeypot* más potente, analiza las distintas propuestas existentes y hace énfasis en la virtualización de este tipo de herramienta.

1. Introducción

Durante los últimos años las intrusiones y los ataques informáticos a través de Internet se han incrementado notablemente. Este incremento en el número de incidentes ha venido acompañado por una clara evolución de las herramientas y técnicas utilizadas por los atacantes.

Parte de los responsables de estas acciones son usuarios avanzados que desarrollan sus propias aplicaciones y son capaces de crear y utilizar sofisticadas puertas traseras para introducirse en otros sistemas. Estos individuos son los más cercanos a la figura del *hacker* que tiene la opinión pública: expertos en informática, seguridad y redes de ordenadores, capaces de entrar en el ordenador más protegido de la compañía más importante, aunque para ello tengan que saltarse las medidas de seguridad más complejas.

Esta idea generalizada y en muchos casos mitificada sobre el perfil de estos intrusos hace pensar que sólo serán objeto de ataque aquellos equipos que contengan información trascendente. Sin embargo, esto es un grave error. Estos ataques selectivos dirigidos por expertos suponen un porcentaje muy pequeño de los que a diario se producen a través de La Red. La práctica totalidad de los incidentes que acontecen en Internet no van dirigidos contra equipos ni compañías específicas, sino que tienen como objetivo la víctima fácil. El blanco seleccionado puede ser cualquier equipo conectado a La Red que posea una debilidad específica que el atacante busca y es capaz de aprovechar para conseguir el acceso a la máquina.

Por otro lado, hoy ya no se necesita poseer unos conocimientos desbordantes sobre el funcionamiento de un sistema para poder atacarlo. De hecho, la mayoría de los intrusos se limita a utilizar herramientas creadas por otros, herramientas que se pueden encontrar fácilmente en Internet, que son cada vez más sencillas de manejar y que no exigen que el atacante conozca su modo interno de funcionamiento. Basta con que ejecute un simple comando o introduzca una serie de instrucciones que en muchas ocasiones se detallan al inicio del propio código de los programas o se incluyen en ficheros de texto que acompañan a las aplicaciones.

En los últimos años, la frecuencia de aparición de estos ataques indiscriminados se ha disparado, y este hecho, unido al creciente número de vulnerabilidades descubiertas en todo tipo de sistemas operativos y aplicaciones, convierte a cualquier sistema conectado a Internet en una víctima potencial. Este panorama plantea la necesidad de disponer de instrumentos que permitan descubrir y analizar tanto los agujeros de seguridad que pueda presentar un sistema como las técnicas y herramientas utilizadas por la comunidad de atacantes que puebla La Red.

Esta ponencia presenta un tipo de herramienta pensada para la detección y análisis de ataques: introduce el concepto de *honeypot* y expone en qué consiste una *honeynet*, analizando las arquitecturas existentes y las posibilidades que, para su desarrollo, ofrece la virtualización.

2. Honeypots

Un *honeypot* [1], o sistema señuelo, es una herramienta de seguridad diseñada para ser sondeada, atacada y comprometida, que tiene la capacidad de detectar y registrar estas acciones. Su funcionamiento está basado en tres simples conceptos:

- Un *honeypot* no es un sistema de producción y, por tanto, nadie debería tratar de comuni-

carse con él. No habrá falsos positivos.

- Cualquier tráfico que tenga por destino el *honeypot* será sospechoso de ser un sondeo o un ataque.
- Cualquier tráfico que tenga por origen el *honeypot* significará que el sistema ha sido comprometido.

Este tipo de herramienta puede servir de ayuda en la prevención de ataques, la detección de intrusiones y la respuesta a este tipo de incidentes, siendo estas dos últimas las dos tareas para las que un *honeypot* ofrece las mayores ventajas:

- Como herramienta de prevención, un *honeypot* puede utilizarse para desalentar al atacante haciendo que pierda su tiempo tratando de entrar en un sistema en el que no va a encontrar información que le resulte realmente de provecho. Este tiempo puede utilizarse para obtener información del intruso, aprender sus técnicas y proteger los sistemas reales de producción. Además, si el atacante sabe que la compañía utiliza señuelos, será consciente de que corre el riesgo de que sus acciones queden registradas. Este temor a caer en una trampa puede suponer una barrera psicológica que haga que un intruso se lo piense dos veces antes de atacar los sistemas de una organización.
- En lo que se refiere a su capacidad de detección de intrusiones, estas herramientas están diseñadas para detectar y recopilar información detallada sobre los ataques que vayan dirigidos contra los servicios que ofrecen.
- Por último, este tipo de herramientas pueden utilizarse para la captura y el análisis de intrusiones con la finalidad de aprender las distintas técnicas y herramientas que utilizan los atacantes para llevar a cabo sus acciones, y para descubrir las vulnerabilidades buscadas por los intrusos y los motivos que les lleva a tratar de atacar el sistema.

El inconveniente que presentan los *honeypots* como herramientas de detección, captura y análisis de intrusiones es que adolecen de cierta miopía: sólo son capaces de reaccionar ante los ataques dirigidos contra el propio *honeypot* o, en casos puntuales, contra los equipos con los que comparta el segmento de red.

Existe una gran variedad de *honeypots*. Así, sí lo único que se pretende es detectar un sondeo de puertos o un intento de acceso a una vulnerabilidad conocida, hay soluciones que simulan algunos servicios y, además, son muy sencillas de instalar y configurar, apenas precisan un pequeño esfuerzo de mantenimiento y suponen un riesgo bajo para la organización. Como inconveniente, la información que son capaces de recopilar es muy limitada. BackOfficer Friendly[1], Specter[2] o incluso Honeyd[3] son ejemplos de este tipo de herramientas.

Si por el contrario lo que se quiere es poder ver en acción a un intruso, hay que utilizar un *honeypot* más complejo que ponga a disposición del atacante sistemas operativos completos, sin servicios simulados. Esto incrementa la dificultad de instalación, configuración y mantenimiento, así como el riesgo asociado a la utilización del *honeypot* comprometido para lanzar nuevos ataques. A cambio se consigue una herramienta que puede ofrecer información sobre el intruso más allá de su mera procedencia, como puede ser sus conocimientos, su experiencia o su psicología. Un ejemplo de este tipo de aplicación es Symantec Decoy Server[4].

3. Honeynets

Una *honeynet* es el *honeypot* más complejo, el que ofrece un nivel más alto de interacción con el intruso y el que permite recopilar mayor cantidad de información relativa a un ataque. Sin embargo, lejos de ser una herramienta empaquetada y lista para ser instalada, una *honeynet* es una red completa que contiene un conjunto de sistemas dispuestos para ser atacados [2].

Una *honeynet* puede contener cualquier componente de red imaginable, incluyendo *routers* y *switches*, lo que le permite replicar la red de cualquier organización. Este hecho, unido a que los equipos que contiene son sistemas reales con servicios y configuraciones habituales, hace que los riesgos y las vulnerabilidades que permite descubrir sean exactamente las mismas que se pueden encontrar en cualquier organización que cuente con sistemas similares a los expuestos.

Este tipo de redes ha ido evolucionando alrededor del Honeynet Project[5], organización creada oficialmente en junio del año 2000 con el objetivo de "Estudiar las técnicas, tácticas y motivos de la comunidad de atacantes y compartir las lecciones aprendidas". Este proyecto agrupa a miembros con perfiles muy distintos: expertos en los distintos sistemas operativos, desarrolladores de herramientas de seguridad, psicólogos, etc.

Aparte de los sistemas destinados a recibir los ataques, una *honeynet* cuenta con un conjunto de dispositivos adicionales que le permiten detectar, filtrar y registrar tanto el tráfico que entra y sale de la red como las acciones de un intruso en el interior de un sistema de la red de señuelos tras

[1] http://www.nfr.net/products/bof
[2] http://www.specter.com
[3] http://niels.xtdnet.nl/honeyd
[4] http://www.symantec.com
[5] http://www.honeynet.org

su compromiso. Todo esto se realiza de forma pasiva, para que el intruso no note ningún comportamiento extraño que le induzca a pensar que está siendo vigilado. Las funciones de estos elementos son, de manera ampliada:

- Control del intruso. Cuando un *honeypot* del interior de la *honeynet* sea comprometido por un intruso, será necesario tener la garantía de que no pueda ser utilizado para atacar otros sistemas que no pertenezcan a la *honeynet*. Con este fin, es preciso controlar todas y cada una de las conexiones que el atacante trate de realizar desde el sistema comprometido, filtrando aquellas que puedan ser nocivas.
- Captura de datos. La clave del éxito de una *honeynet* radica en su capacidad para capturar la mayor cantidad de información que sea posible, pues serán estos datos los que van a permitir estudiar las utilidades, tácticas y motivos de la comunidad de atacantes. Es fundamental capturar todo el tráfico que entre o salga de la *honeynet*, así como cualquier actividad del intruso en el interior de un sistema comprometido.
- Centralización de información. Para conseguir un mejor rendimiento en escenarios con distintas *honeynets* dispersas por Internet es recomendable que la información capturada se envíe de forma segura a un servidor centralizado para su almacenamiento y análisis. De este modo se puede tener un mayor control sobre los datos recogidos, se pueden reaprovechar experiencias y se puede obtener una imagen más clara de la evolución de los diferentes ataques presentes en La Red.

3.1. Arquitecturas

Hasta ahora, no existe un modelo cerrado de arquitectura de *honeynet*. Para su desarrollo hay una absoluta libertad a la hora de seleccionar tanto su topología como las herramientas a utilizar para realizar las tareas de control, registro y análisis de las acciones del intruso en su interior. A pesar de esto, si bien es cierto que no hay una estandarización clara, las distintas propuestas del Honeynet Project han venido marcando el modelo a seguir desde la aparición de esta herramienta de seguridad. Esta organización distingue dos arquitecturas distintas, que denomina *Honeynets* de Primera y Segunda Generación.

3.1.1. Honeynets de Primera Generación

Este modelo [3], utilizado por el Honeynet Project desde sus inicios hasta finales del año 2001, destaca por la sencillez con la que soluciona los problemas del control del intruso y la captura de información. Su arquitectura, como muestra la Figura 1, consta de una red de sistemas señuelo dispuestos a ser atacados o *honeypots*, un cortafuegos, un *router*, un detector de intrusiones basado en red o NIDS y un servidor centralizado de *logs* y alarmas.

La tarea de control del intruso se realiza de forma conjunta entre el cortafuegos y el *router*:

El cortafuegos es un filtro de paquetes a nivel de red que constituye el nexo de unión entre el interior de la *honeynet* e Internet y divide la propia *honeynet* en dos segmentos de red: uno de *honeypots* y otro administrativo que contiene el servidor remoto de *logs* y el sistema detector de intrusiones. Está configurado para permitir cualquier conexión desde el exterior de la *honeynet* a la red de *honeypots*, proteger los equipos de la subred administrativa y controlar las conexiones que se traten de establecer desde los *honeypots* hacia el exterior.

Para el control del intruso se van a contar los intentos de conexión desde cada *honeypot* hacia cualquier equipo del exterior de la honeynet de modo que, a partir del instante en que se supere cierto umbral, se bloqueará cualquier nuevo intento. Para la selección de este umbral hay que tener en cuenta el tipo de ataque y atacante que se pretende estudiar: si el objetivo es capturar ataques automatizados, como los conducidos por gusanos o *auto-rooters*, no es necesario permitir ninguna conexión. Si lo que se desea es analizar intrusiones más avanzadas o se quiere obtener más información sobre el comportamiento del

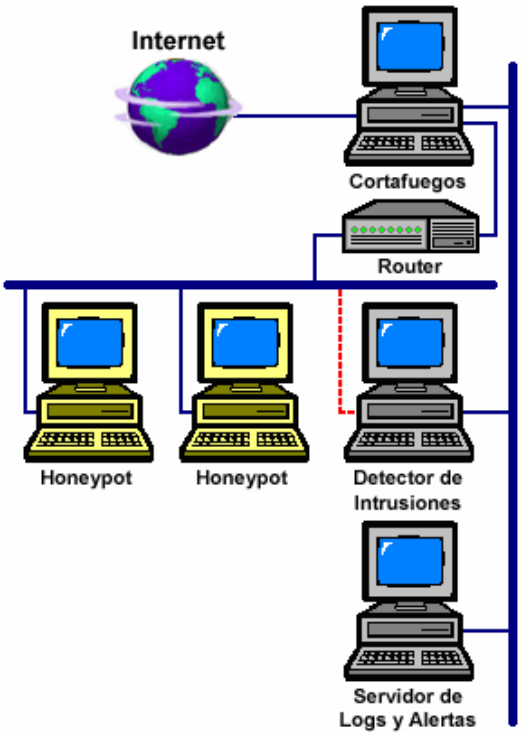

Figura 1. Honeynet de Primera Generación.

atacante, será necesario consentirle que realice algún número de conexiones desde el sistema comprometido.

A medida que se incrementa este número de conexiones permitidas se incrementa el margen de actuación del intruso. Este mayor margen permite recopilar más información sobre el atacante, pero también implica un incremento en el riesgo de que el sistema comprometido sea utilizado para atacar otros equipos. El Honeynet Project, basado en su experiencia, recomienda permitir entre cinco y diez conexiones por día. Con esto se permite que el atacante descargue herramientas o se comunique con otra gente, pero se evita que el *honeypot* sea utilizado para escanear otros equipos o para realizar ataques de denegación de servicio o de fuerza bruta.

El *router* se instala con la intención de ocultar la presencia del cortafuegos a ojos de los sistemas de la red de *honeypots*, de modo que cuando un intruso investigue el *gateway* de un sistema comprometido descubrirá un *router* y no un cortafuegos.

Para la captura de datos, colaboran el cortafuegos, el NIDS y los propios *honeypots* comprometidos:

El cortafuegos se encarga de registrar el tráfico que le llega, el tráfico que trata de atravesarlo, en cualquier dirección y sentido, y el tráfico que descarta, con los motivos del descarte.

El detector de intrusiones captura y almacena todos y cada uno de los paquetes que circulan por alguna de las subredes a las que tiene acceso. Puesto que la *honeynet* no contiene sistemas de producción, es de suponer que el volumen de información que habrá de registrar no será muy elevado, pero todo este tráfico será sospechoso. Esta captura será utilizada posteriormente para estudiar los ataques sufridos por sus sistemas y la actividad posterior de los atacantes, y será en este punto en el que se va a aprovechar su capacidad de análisis del tráfico capturado en busca de patrones conocidos de ataque.

Los *honeypots* recogen información tanto sobre la actividad del sistema como sobre las acciones del atacante en su interior. Sin embargo, toda la información almacenada localmente en un sistema comprometido corre el riesgo de ser modificada o eliminada. Por esto, toda la información recopilada por cada *honeypot* va a ser enviada a un servidor remoto de *logs*.

La posibilidad de enviar los registros a un servidor remoto es típica de los sistemas UNIX y está disponible para algún dispositivo de red. Por el contrario, la capacidad de registro de los sistemas Windows es más limitada y se precisará de herramientas auxiliares que se encarguen de realizar el envío al servidor remoto.

Este servidor remoto va a recolectar también los *logs* del cortafuegos y el detector de intrusiones y, en función de esta información, generará mensajes de aviso para el administrador: si detecta un intento de conexión desde el exterior puede significar un sondeo o un ataque. Si detecta un intento de conexión desde un *honeypot* significa que el sistema ha sido comprometido.

Para la captura de la actividad de un intruso en el interior de un sistema comprometido, en ocasiones los atacantes no utilizan protocolos seguros que codifican la información en tránsito. En este caso las instrucciones ejecutadas por el intruso podrán recuperarse a partir del tráfico capturado por el sistema detector de intrusiones.

Sin embargo, cada vez es más habitual que el atacante utilice protocolos seguros para sus comunicaciones y, cuando esto sucede, el único punto desde el que se podrá registrar la actividad del intruso será desde el interior del propio *honeypot* comprometido. En estos sistemas se instalarán versiones modificadas de la *shell*, en máquinas UNIX, o del command.com o cmd.exe, en sistemas Windows, que se encargarán de la captura de esta información.

El principal inconveniente que presentan las *Honeynets* de Primera Generación tiene que ver con sus limitaciones en el control del atacante: si le permite un cierto umbral de conexiones, en el peor de los casos, sería posible que todas y cada una de estas conexiones sea un ataque exitoso y las medidas de contención habrían fracasado.

3.1.2. Honeynets de Segunda Generación

Esta segunda familia es la que el Honeynet Project ha venido utilizando desde principios del año 2002 [4]. Introduce una serie de modificaciones, con respecto a las *Honeynets* de Primera Generación, con las que se pretende conseguir un entorno más difícil de identificar por parte del atacante. Además permiten al intruso una mayor libertad a la hora de realizar nuevas conexiones con el exterior, controlando de forma más estrecha sus acciones en el sistema comprometido.

La arquitectura, como aparece en la Figura 2, es más sencilla que la que se mostró en el apartado anterior ya que tanto las tareas de control como las de captura y recolección de datos se realizan en un único sistema que el Honeynet Project denomina *honeywall*. Esta centralización simplificará también los procesos de desarrollo y administración de la *honeynet*.

El *honeywall* dispone de tres interfaces de red. La que aparece conectada al *router* se utiliza exclusivamente para la administración remota del sistema. Con respecto a las otras dos interfaces el sistema se va a comportar como un *bridge*: dichas interfaces van a carecer de direcciones IP

y MAC asociadas y el sistema ni hará encaminamiento de tráfico ni decrementará el TTL de los paquetes que lo atraviesen. Este comportamiento del *honeywall* hace posible que la *honeynet* se pueda integrar en la red de la organización pues, como muestra la Figura 2, puede incluso compartir VLAN con otros sistemas de producción de la compañía. Esto va a permitir el estudio de las amenazas, tanto externas como internas, que afectan a la organización.

Para controlar las acciones del intruso en el *honeywall* a la acción del cortafuegos se va a sumar la de un sistema de prevención de intrusiones o NIPS. Este tipo de herramienta analiza en tiempo real el tráfico que le llega en busca de patrones conocidos de ataque pero, a diferencia de los detectores de intrusiones, tras descubrir un intento de ataque tiene la capacidad de impedir que el ataque tenga éxito, ya sea descartando o modificando el tráfico para hacerlo inofensivo.

Los NIPS pueden bloquear cualquier ataque que esté registrado en su fichero de configuración, pero no suponen ningún tipo de barrera a los ataques nuevos. Por esto, en el *honeywall* se va a seguir utilizando la cuenta de conexiones para bloquear aquellos ataques que no descubra el NIPS.

La diferencia fundamental con el método de control utilizado en las *Honeynets* de Primera

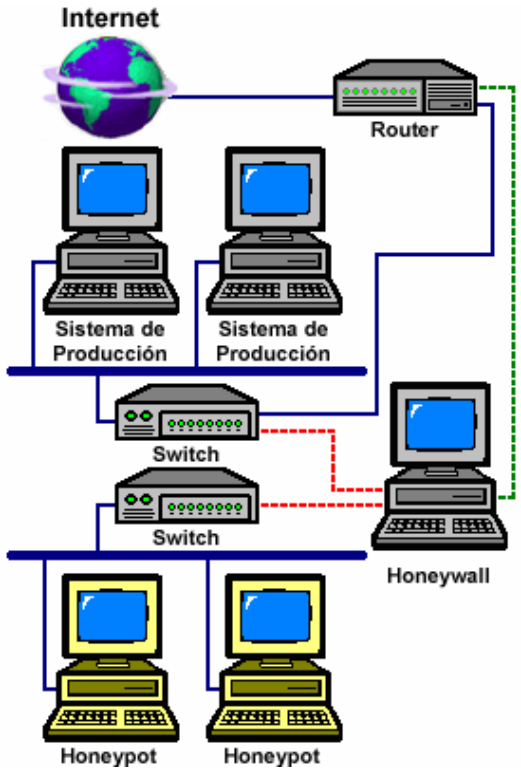

Figura 2. Honeynet de Segunda Generación.

Generación radica en que en este modelo se va a incrementar el umbral de conexiones permitidas. En las *Honeynets* de Segunda Generación el Honeynet Project permite un máximo de 15 conexiones TCP, 20 UDP, 50 ICMP y 15 de otros protocolos por cada *honeypot* al día. Hasta llegar a estos límites se permitirá al intruso realizar todas las conexiones que precise siempre que no constituyan un ataque.

La captura de datos se realiza de modo similar al empleado en las *honeynets* de la generación anterior con las única diferencia de que la recopilación de la información se va a realizar de forma centralizada desde el *honeywall* y que la captura de las sesiones de los intrusos en los *honeypots* UNIX, en lugar de utilizar *shells* modificadas, se va a realizar a través del *kernel* de estos sistemas mediante programas similares a algunos *rootkits*.

4. Honeynets Virtuales

Una *honeynet* virtual es una red de señuelos contenida completamente en un único ordenador mediante la utilización de *software* de virtualización. Estas herramientas permiten la ejecución simultánea de varios sistemas operativos en un único equipo físico de modo que, pese a compartir los recursos del sistema anfitrión, aparenten estar corriendo en máquinas distintas e independientes. Este *software* es capaz de albergar los sistemas operativos más extendidos, alcanzando un rendimiento cada vez más cercano al que se consigue sobre máquinas físicas.

Estas redes virtuales no constituyen una Tercera Generación de *honeynets* pues se basan en las mismas ideas e incluso pueden replicar las arquitecturas de Primera o Segunda Generación. Sin embargo, el empleo de estas herramientas de virtualización ofrece un conjunto de ventajas e implica una serie de limitaciones para el diseño de *honeynets* que incidirán en su arquitectura final y la apartarán de los modelos explicados anteriormente.

Aunque cada herramienta de virtualización presenta sus propias características y peculiaridades, hay una serie de cualidades que son comunes a todas ellas. El empleo de este *software* para el desarrollo de *honeynets* ofrece las siguientes ventajas frente a los modelos clásicos:

- Sólo se precisa de una máquina física, que funciona como anfitriona de la red virtual. Esto se traduce en una disminución significativa del coste del *hardware* y el espacio requerido por la *honeynet*.
- Permite administrar todos los sistemas de la *honeynet* de modo centralizado desde el equipo anfitrión.
- El hecho de que todo se ejecute en un único equipo convierte una *honeynet* en una solu-

ción "*plug-and-play*".

- Los discos duros de los distintos sistemas pueden ser virtuales, es decir, archivos en el sistema anfitrión. Si cada vez que se instala y configura correctamente un sistema se guarda una copia de seguridad, cuando se quiera recuperar una máquina atacada o se quiera sustituir un sistema por otro bastará con reemplazar los ficheros del sistema a sustituir por la copia de seguridad almacenada.

Sin embargo, no todo son ventajas. Este tipo de *honeynet* presenta también algunos inconvenientes frente a los modelos clásicos:

- La utilización de *software* de virtualización limita la variedad de sistemas operativos que pueden constituir una *honeynet* a los soportados por la herramienta. Asimismo, para la generación de la red virtual sólo se podrán utilizar aquellos componentes de red que la herramienta de virtualización sea capaz de simular, que generalmente serán *switches*.
- El hecho de que toda la *honeynet* se ejecute sobre un único equipo lo convierte en el talón de Aquiles de la arquitectura.
- El empleo de herramientas de virtualización introduce ciertas peculiaridades en las máquinas virtuales que pueden delatar la utilización de este tipo de *software*. El hecho de que un sistema sea virtual no implica que se trate de un *honeypot*, pero puede hacer que el intruso pierda su interés en la máquina.
- Si bien existen alternativas de código abierto que no requieren desembolso alguno, la licencia de utilización de las herramientas de virtualización puede ser costosa.

4.1. Herramientas de Virtualización

Algunas de las herramientas de virtualización que se pueden utilizar para el desarrollo de *honeynets* virtuales son: User Mode Linux[6], VMware[7] Workstation y GSX Server, o Microsoft Virtual PC[8]. Las características de la *honeynet* estarán íntimamente relacionadas con la solución *software* utilizada para su implementación.

User Mode Linux es un módulo del *kernel* que permite la ejecución simultánea de varios sistemas Linux como procesos de otra máquina Linux. Su utilización tiene una serie de ventajas:

- Se trata de una herramienta de código abierto, por lo que se podrá revisar, corregir y adaptar su código a las necesidades de la *honeynet*.
- Como *software* de libre distribución, se puede utilizar sin necesidad de pagar licencias.
- Permite capturar las sesiones del intruso de forma pasiva a través del *kernel* del sistema anfitrión.

Sin embargo, también presenta varios inconvenientes:

- Sólo alberga máquinas virtuales Linux.
- No ofrece interfaz gráfica y su utilización no resulta demasiado intuitiva. Tampoco existe una documentación clara y detallada sobre su modo de empleo, por lo que, en principio, no va es una herramienta sencilla de manejar.
- La instalación de máquinas virtuales es más complicada que en el resto de herramientas.
- Como herramienta de código abierto, carece de soporte técnico.

VMware Workstation es una herramienta de virtualización comercial diseñada para ejecutarse sobre equipos de sobremesa. Existen versiones tanto para Windows como para Linux, cuesta en torno a 300 $ y presenta las siguientes ventajas:

- Puede hospedar máquinas virtuales con sistemas operativos Windows, Linux, NetWare y FreeBSD, aunque potencialmente es capaz de albergar cualquier sistema que se ejecute sobre la plataforma X86 de Intel.
- Se maneja desde una intuitiva interfaz gráfica de usuario y ofrece una documentación detallada, lo que simplifica su utilización.
- El proceso de instalación de un sistema operativo en una máquina virtual es el mismo que se utiliza en los equipos físicos.

Como inconvenientes se puede mencionar la exigencia del pago de la licencia y que se trata de *software* propietario, por lo que ni se tiene acceso al código fuente ni se tiene derecho a realizar modificaciones en la aplicación.

VMware GSX Server es una herramienta de virtualización diseñada para su utilización en sistemas Windows o Linux que cuesta entre los 3025 $ y los 6050 $. Aparte de esto, todo lo que se ha expuesto de la versión Workstation es válido también para la versión GSX, a lo que se añaden las siguientes ventajas:

- Puede albergar máquinas virtuales más potentes y redes más complejas.
- Permite la administración remota de la herramienta a través de una interfaz *web* y una consola remota que permite el acceso a las máquinas virtuales instaladas.
- Incluye una API que permite el control de las máquinas virtuales.
- Se puede contratar soporte técnico de la herramienta.

Microsoft Virtual PC es una herramienta diseñada para funcionar sobre Windows, OS/2 y MAC OS que es comparable, tanto en rendimiento como en funcionalidades, a la versión Works-

[6] http://user-mode-linux.sf.net
[7] http://www.vmware.com
[8] http://www.microsoft.com/virtualpc

tation de VMware. El coste de su licencia está en torno a los 160 €.

4.2. Desarrollo de una Honeynet Virtual sobre VMware GSX Server

La Figura 3 muestra una propuesta de arquitectura de *honeynet* virtual [5]. Como se puede ver, se trata de una red compuesta por un sistema físico y siete virtuales, distribuidos en cinco segmentos, y un *router* a través del cual las distintas máquinas acceden y son accedidas desde Internet. Por motivos de seguridad, las direcciones IP que aparecen en la Figura son ficticias.

El sistema anfitrión es el equipo físico que albergará el *software* de virtualización sobre el que se va a ejecutar la red virtual. Para el desarrollo de la *honeynet* propuesta se ha utilizado un Pentium IV a 2 GHz con 1 GB de RAM, 40 GB de disco duro y 4 interfaces de red. Desde este sistema se van a administrar las distintas máquinas virtuales y, en su interior, se va a almacenar y analizar la información recogida por los distintos dispositivos de la *honeynet*.

De las tres conexiones que presenta el sistema anfitrión: a través de la conexión con el *honeywall* va a recibir los *logs* de este sistema; la conexión con el *router* se utilizará para la administración remota de la *honeynet*; y mediante la conexión con la máquina auxiliar, como se verá después, obtendrá copias de las particiones de las máquinas virtuales comprometidas.

El segmento inferior contiene cuatro máquinas virtuales que serán los *honeypots* que se van a exponer con la intención de recibir los ataques de la comunidad de intrusos. Independientemente de los sistemas operativos que muestran en la figura, estas máquinas podrán albergar cualquiera de los sistemas permitidos por la herramienta de virtualización. El número de *honeypots* puede incrementarse también añadiendo más memoria al sistema anfitrión.

Las máquinas virtuales se van a instalar sobre discos duros virtuales. Cada sistema va a tener su propio directorio en el sistema anfitrión, que contendrá varios ficheros: uno de configuración, otro de *log* y uno, o varios, con los discos duros de la máquina virtual. Una vez instalada y configurada correctamente una máquina virtual, se hará una copia de seguridad de su directorio, de modo que la máquina virtual podrá restaurarse a partir de esta copia sin tener que repetir los procesos de instalación y configuración.

Para desarrollar esta *honeynet* virtual, si bien es posible hacer el diseño sobre sistemas Windows o Macintosh empleando las herramientas expuestas anteriormente, se ha optado por emplear Linux como sistema operativo del equipo anfitrión. Este sistema ofrece las siguientes ventajas:

- Existe una gran variedad de herramientas de seguridad contrastadas diseñadas para funcionar sobre Linux.
- Permite montar sistemas de archivos utilizados por otros sistemas operativos, como FAT, HPFS o NTFS, lo que hace posible el análisis de las particiones de los sistemas comprometidos desde el sistema anfitrión.

Como herramienta de virtualización se ha op-

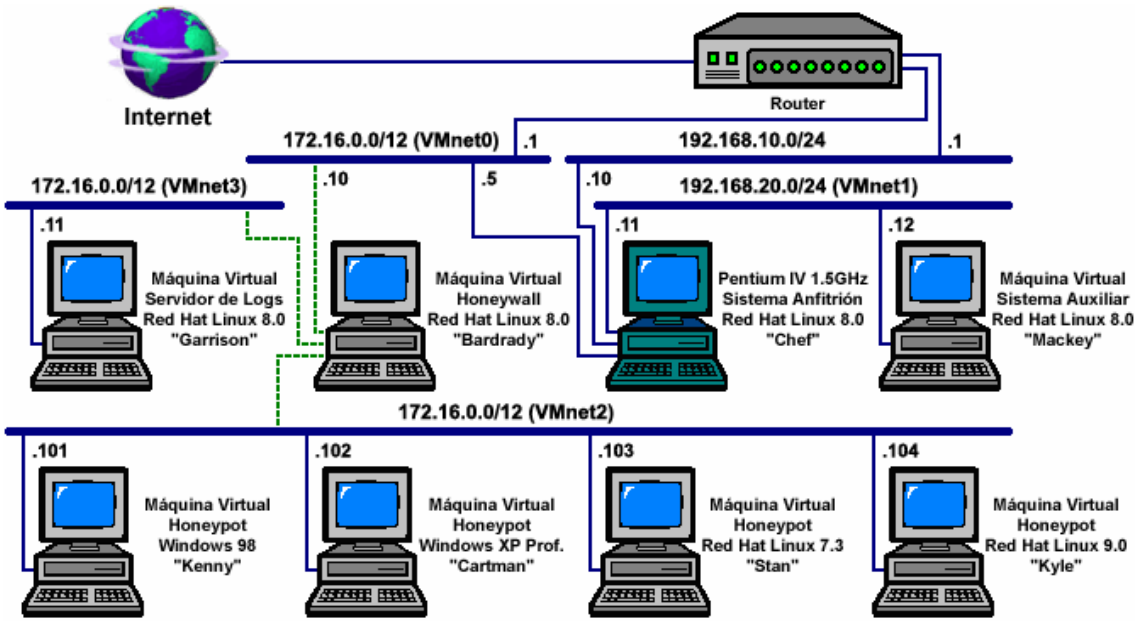

Figura 3. Modelo propuesto de Honeynet Virtual.

tado por VMware GSX Server, que a las ventajas de la versión Workstation suma las siguientes:

- La interfaz de administración *web* y la consola abren las puertas a la administración remota de la red virtual.
- La API que incluye permite crear un sistema de administración a medida de la *honeynet*.

Como el resto de herramientas de virtualización, VMware presenta una serie de peculiaridades que pueden delatar la utilización de este *software* de virtualización. Entre los detalles que conviene limar para ocultar la naturaleza virtual de los sistemas de la *honeynet* destacan las VMware Tools, cuya instalación habrá que evitar, y algunos parámetros del *hardware* virtual de las máquinas como las direcciones MAC de las tarjetas de red, que deberán ser modificadas.

La tarea de control de las acciones del intruso para evitar que lance ataques contra sistemas del exterior desde un sistema comprometido se realizará desde el *honeywall* del modo descrito en el análisis de las Honeynets de Segunda Generación. Para esto se configurará la máquina virtual en modo *bridge* y se utilizará la acción combinada de Iptables[9] y Snort Inline[10].

Iptables es un cortafuegos que funciona como un módulo de las versiones 2.4 del *kernel* de los sistemas Linux. Es capaz de filtrar tráfico en función del estado de una conexión y puede redirigir tráfico hacia otras herramientas instaladas en su sistema, como en este caso un NIPS.

Snort Inline es una versión modificada de Snort[11] que inspecciona el tráfico que le llega en busca de patrones conocidos de ataque. Utiliza la misma base de reglas que este detector de intrusiones, aunque retocada para descartar o modificar el tráfico nocivo para hacerlo inofensivo en lugar de limitarse a registrar y capturar tráfico y generar alertas como hace Snort.

Con todo esto, el comportamiento del *honeywall* será el siguiente:

- Registra y permite cualquier intento de conexión desde el exterior de la *honeynet* a un sistema de la red de *honeypots*.
- Registra y descarta cualquier intento de utilizar IP *Spoofing* desde la red de *honeypots*.
- Permite a los distintos sistemas el acceso a un servidor de DNS y uno de NTP del exterior, para resolver nombres y sincronizar los relojes respectivamente.
- Registra y permite el envío de *logs* desde los *honeypots* al servidor remoto y desde el propio *honeywall* al sistema anfitrión.
- Registra y filtra cualquier intento de acceso desde la red de *honeypots* al resto de los sistemas de la *honeynet*.
- Registra y filtra el tráfico generado en los *honeypots* dirigido a sistemas del exterior de la *honeynet* en función de los patrones de ataque conocidos por Snort Inline.
- Registra y descarta los intentos de conexión desde la red de *honeypots* hacia los sistemas del exterior de la *honeynet* a partir de ciertos umbrales: 15 TCP, 20 UDP, 50 ICMP y 15 del resto del tráfico.

La captura de datos se va a realizar desde distintos dispositivos de modo que, en caso de que uno falle, no se perderá toda la información en juego. En esta tarea van a participar el *honeywall*, el sistema anfitrión y cada uno de los *honeypots* de la red, y toda la información recogida será almacenada finalmente en el sistema anfitrión para su posterior análisis.

El *honeywall* se va a encargar de registrar los intentos de conexión que tengan como origen o destino alguno de los *honeypots* de la *honeynet*. Estos *logs* los enviará al sistema anfitrión mediante Syslog-ng[12], donde serán analizados en tiempo real mediante un demonio de Swatch[13] que utilizará esta información para generar mensajes de correo para alertar al administrador.

Para entender el modo en que se va a capturar el tráfico de red hay que explicar que VMware simula *switches* para interconectar las máquinas virtuales. Este hecho implica que desde el *honeywall* no se va a poder capturar un ataque dirigido de un *honeypot* a otro, y por este motivo esta captura se realizará desde el propio sistema anfitrión. Los *switches* virtuales pueden configurarse de dos modos:

- Los *switches host-only* permiten interconectar máquinas virtuales entre sí, así como con el sistema anfitrión, en cuyo interior se genera una interfaz de red virtual. Este tipo de *switch* es el que se utiliza para conectar el sistema auxiliar con el anfitrión y, puesto que no se va a habilitar el reenvío de paquetes en el sistema anfitrión, la máquina auxiliar estará aislada del resto de la red.
- Cada *switch* en modo *bridge* se asocia a una interfaz física de red del sistema anfitrión. A través de esta tarjeta las máquinas virtuales conectadas al *switch* pueden acceder, con una dirección IP propia, al mismo segmento de red al que esté conectada la interfaz física del sistema anfitrión, como si fueran unos equipos más de los conectados al segmento.

Utilizando *switches* configurados en modo

[9] http://www.netfilter.org
[10] http://snort-inline.sf.net
[11] http://www.snort.org
[12] http://www.balabit.com
[13] http://swatch.sf.net

bridge las máquinas virtuales hacen uso de la interfaz de red asociada a cada *switch* para enviar y recibir paquetes. Si se pone esta tarjeta en modo promiscuo permitirá la captura de todo el tráfico que entre o salga de cada una de las máquinas conectadas al *switch* virtual. Esto se puede realizar incluso en el caso de que la interfaz de red física del sistema anfitrión asociada al *switch* no esté conectada a ningún sitio o carezca de dirección IP, lo que permite capturar el tráfico de forma totalmente pasiva.

Los segmentos VMnet0, VMnet2 y VMnet3 de la Figura 3 son *switches* virtuales configurados en modo *bridge*, y forman la VLAN que va a constituir el verdadero cuerpo de la *honeynet*. Desde el sistema anfitrión, utilizando Snort, se va a capturar a través de las interfaces asociadas a cada uno de estos *switches* todo el tráfico que atraviese estos segmentos. Un inconveniente que tiene este modo de captura es que desde el sistema anfitrión se capturarán dos paquetes seguidos por cada paquete generado en una máquina virtual. Esto se corrige fácilmente editando la captura mediante Ethereal[14] y realizando un filtrado en función de la similitud y la diferencia de tiempo entre la captura de dos paquetes consecutivos.

El último elemento de captura serán los propios *honeypots* de la red. En ellos se recopilará información tanto sobre lo que ocurra en el sistema operativo, las aplicaciones y los servicios que se ejecuten en su interior como sobre las acciones del intruso en su interior.

En lo referente a la información recogida por el entorno, serán el propio sistema operativo y las propias aplicaciones y servicios los encargados de registrar los incidentes que detecten. Una copia de estos registros se enviará al servidor remoto de *logs* mediante Syslog. Para hacer esto en las máquinas UNIX basta con configurar el demonio que se encarga de hacer el registro. En los sistemas Windows NT, 2000, XP y 2003 se va a utilizar una aplicación denominada Eventlog to Syslog[15] que, como su nombre indica, envía una copia de los registros del Eventlog del sistema a un servidor remoto de Syslog.

Puesto que los paquetes enviados al servidor remoto de *logs* no van cifrados, la información que contienen puede recuperarse a partir del tráfico capturado por el sistema anfitrión. De este modo, no se pierde nada si un atacante borra la información contenida en el servidor de *logs*, por lo que este sistema se puede considerar un *honeypot* más de la red. Eso sí, al ser una máquina más protegida exigirá que el atacante que pretenda comprometerlo aplique técnicas más sofisticadas que las que pudieran haberle valido para acceder a los sistemas de la red de *honeypots*.

Para la captura de sesiones del intruso se va a utilizar Sebek[16] en los sistemas Linux y Solaris y la combinación de Comlog[17] y SpyBuddy[18] en las máquinas Windows:

Sebek es un módulo del *kernel* que captura las teclas pulsadas por el intruso y los ficheros descargados por este en el sistema comprometido. Este *software* se oculta en el sistema para evitar ser detectado o desinstalado y envía la información recopilada a un servidor remoto donde es recuperada por un *sniffer*. Para la generación de este tráfico no utiliza la pila TCP/IP del sistema, sino que se comunica directamente con el dispositivo de red. De este modo los usuarios de estos *honeypots* no podrán ni descubrir ni bloquear este tráfico.

La gran mayoría de los ataques contra sistemas Windows utilizan el *command prompt* del sistema. En este contexto, ComLog es un troyano del cmd.exe, la *shell* de los sistemas Windows NT, 2000, XP y 2003, que registra las instrucciones que recibe y redirige la llamada hacia el ejecutable original del sistema.

SpyBuddy es una herramienta comercial enfocada a monitorizar el comportamiento de los usuarios que acceden a un sistema Windows que, entre otras cosas, registra las teclas pulsadas por el usuario y monitoriza la creación, acceso, cambio de nombre y borrado de ficheros y directorios en el sistema.

Si bien es deseable que la información recuperada sea almacenada en un equipo remoto en cuya integridad se pueda confiar, la utilización de VMware hace que esto, aunque siga siendo recomendable, no sea imprescindible. Utilizando este *software* de virtualización, los discos duros virtuales pueden configurarse de los siguientes modos:

- Persistente. Las modificaciones se realizan en tiempo real en el disco duro virtual.
- No persistente. Las modificaciones realizadas se descartan al apagar la máquina virtual, volviéndose al estado inicial.
- Restaurable. Al apagar el sistema se ofrece al usuario la opción de fijar o descartar los cambios en el disco duro.

Utilizando discos duros en modo restaurable, las modificaciones se van almacenando en ficheros con extensión .REDO en el directorio de la máquina virtual. Xtail[19] es una herramienta simi-

[14] http://www.ethereal.com
[15] https://engineering.purdue.edu/ECN
[16] http://www.honeynet.org/tools/sebek
[17] http://iquebec.ifrance.com/securit
[18] http://www.exploreanywhere.com
[19] http://www.unicom.com/sw/xtail

lar al tail de las máquinas UNIX, con la diferencia de que permite monitorizar cambios en varios ficheros a la vez aunque estos no se produzcan al final del archivo monitorizado. Si, como se propone en [6], se utiliza esta herramienta para monitorizar las modificaciones en los ficheros .REDO de las máquinas virtuales, se filtra la respuesta en busca de patrones ASCII y se almacena el resultado en un fichero, se pueden rescatar todos los ficheros de texto creados o modificados en cada máquina virtual aunque hayan sido borrados posteriormente por el intruso.

Otra herramienta que se emplea para descubrir modificaciones en las máquinas virtuales es Tripwire[20]. Se trata de un detector de intrusiones basado en *host* que genera firmas de los ficheros y directorios de un sistema y que, en base a estas firmas, es capaz de detectar cualquier modificación realizada en los mismos.

Cuando un *honeypot* esté configurado y preparado para ser expuesto al ataque de los intrusos, además de realizar una copia de seguridad para su posterior restauración, se utilizará esta herramienta para obtener la firma del estado inicial de la máquina. Los ficheros que se modifican por el mero hecho de encender la máquina no se monitorizarán, pero cualquier otro cambio será detectado durante el análisis posterior mediante esta herramienta.

Finalmente, para el análisis post-mortem de las particiones de los sistemas comprometidos se puede utilizar The Sleuth Kit[21]. Para la generación de discos duros virtuales, VMware no utiliza un sistema de archivos estándar, por lo que las particiones del disco no se podrán montar directamente en el sistema anfitrión para su análisis. Es aquí donde aparece el sistema auxiliar. En este sistema se van a montar los discos duros de los sistemas comprometidos y mediante Netcat[22] y la instrucción dd se enviará una réplica exacta bit a bit de cada una de las particiones al sistema anfitrión, particiones que ya sí serán montables.

5. Conclusiones

El despliegue de *honeynets* permite capturar y analizar ataques a sistemas informáticos, con lo que es posible aprender acerca de las amenazas que afrontan los sistemas conectados a Internet. Estas redes de señuelos están compuestas por sistemas reales situados en un entorno controlado diseñado para vigilar las acciones de los intrusos.

Una *honeynet* es una herramienta intensiva en recursos *hardware* que puede exigir una inversión elevada en equipos informáticos y dispositivos de red, con las consiguientes necesidades de espacio físico. Las *honeynets* virtuales surgen como solución a estos problemas, al poder utilizar un único ordenador para alojar los señuelos y los distintos sistemas de control. En este artículo se ha mostrado cómo se puede desarrollar este tipo de herramienta, explicando el *software* utilizado y su función dentro de cada uno de los componentes de la *honeynet*.

Se trata de un tipo de herramienta de seguridad relativamente reciente y, como tal, aún está evolucionando. Existen varias líneas de trabajo abiertas para su mejora: para reducir la complejidad del proceso de despliegue de una *honeynet* sería interesante generar un DVD que se encargase de automatizar la instalación y configuración de la red de señuelos. Para simplificar la gestión de la herramienta convendría desarrollar una plataforma capaz de administrar de manera centralizada tanto los *honeypots* como los dispositivos de control, detección y registro.

También hay líneas abiertas orientadas a una posible explotación comercial de las *honeynets*: mediante las técnicas de *hot zoening* [7] se pretende distinguir el tráfico lícito del ilegal, redirigiendo el primero hacia sistemas de producción y el segundo hacia los sistemas de la *honeynet*. La generación de granjas de *honeynets* [8] hace posible la externalización de estas herramientas, permitiendo que una empresa ofrezca a otras organizaciones servicios de detección, captura y análisis de intrusiones.

6. Referencias

[1] Lanze Spitzner, *Honeypots, Tracking Hackers*, Addison Wesley, 2003.

[2] The Honeynet Project, *Know Your Enemy. Revealing the security tools, tactics, and motives of the blackhat community,* Addison Wesley, 2002.

[3] The Honeynet Project, *Know Your Enemy. Honeynets,* Noviembre de 2003.

[4] The Honeynet Project, *Know Your Enemy. GenII Honeynets,* Noviembre de 2003.

[5] Eduardo Gallego Revilla, *Desarrollo de una Red Virtual de Señuelos para la Detección y Análisis de Intrusiones en Sistemas Informáticos,* ETSI de Telecomunicación, UPM. Diciembre de 2003.

[6] Ryan C. Barnett, *Monitoring VMware honeypots,* Septiembre de 2002.

[7] The Honeynet Project, *Know Your Enemy. Hot Zoneing,* Febrero de 2003.

[8] Lanze Spitzner, *Honeypot Farms,* Agosto de 2003.

[20] http://tripwire.sf.net

[21] http://www.sleuthkit.org

[22] http://www.atstake.com