CAPITULO 4

Honeypots y Honeynets

En los capítulos anteriores hemos examinado las distintas técnicas de ataques de denegación de servicio DOS/DDOS así como los sistemas de detección de intrusos (IDS). En este capítulo plantearemos el cambio de escenario que están sufriendo las comunicaciones por Internet debido al rápido avance tecnológico y cómo estas han ido modificando los ataques a las redes de ordenadores.

Nuevos tipos de ataques requieren nuevos modelos que permitan ampliar el espectro de seguridad cubierto anteriormente. Describiremos en profundidad las características, tipos y ubicaciones de los Honeypots. Estos son un intento de conocer al enemigo "desde dentro", conocer sus técnicas y motivaciones proporcionándole un entorno controlado pero interactivo en el que pueda "jugar" mientras es espiado.

Comentaremos los distintos tipos, sus arquitecturas así como las posibles ubicaciones que permiten y las implicaciones legales que entraña su uso.

Posteriormente nos centraremos en las Honeynets, que aparecen como desarrollo del concepto de Honeypot. Se explicarán sus distintas características y configuraciones (generaciones) y veremos como permiten la implementación de completos sistemas para el estudio de atacantes.

Finalmente se describirán las Honeynets virtuales que permiten mediante un uso mínimo de recursos una implementación total de una o varias Honeynets.

4.1 Nuevos escenarios de ataques

En los capítulos anteriores hemos analizado los escenarios típicos generadores de amenazas para cualquier sistema conectado a Internet. Con el aumento de los anchos de banda disponibles y el abaratamiento de los accesos a la red hemos podido observar una evolución de las técnicas de ataques existentes en la actualidad.

Anteriormente los ataques se basaban en razonamientos simples, dada una máquina/dominio/servicio conocido (por ejemplo el de correo personal gratuito de Hotmail → www.hotmail.com), bastaba con obtener su dirección IP (vía consulta DNS por ejemplo) y proceder con cualquiera de los ataques anteriormente descritos.

Este procedimiento que consideraremos "standard" circunscribía los destinatarios de ataques informáticos a grandes empresas, organismos oficiales y universidades. Sin embargo, la mejora de la conectividad nos permite ahora realizar combinaciones de ataques que hasta hace unos pocos años eran imposibles.

La realización de un análisis completo (*scan o probe*) de toda una clase A, B o C (ver figura 4-1) deja de ser una utopía para convertirse en una simple cuestión de días o incluso horas.

CLASE	RANGO		ORDENADORES
A	0.0.0.0	127.255.255.255	128 redes de 2^24 direcciones
B	128.0.0.0	191.255.255.255	2^14 redes de 2^16 direcciones
C	192.0.0.0	223.255.255.255	2^21 redes de 2^8 direcciones

FIG. 4-1: Redes y direcciones IP en Internet.

La proliferación y difusión de herramientas específicas para este tipo de comportamientos (podemos encontrar cientos de servidores WWW con unas simple búsqueda en Google), ha ido pareja al aumento de jóvenes con ganas de emular las películas de piratas informáticos.

Precisamente esta gran cantidad de público ha llevado a la creación de herramientas con interfaces (*front-ends*) muy sencillos y amigables que permiten prácticamente a cualquier persona sin muchos conocimientos (*script-kiddies*) rellenar un par de campos de parámetros (por ejemplo la dirección IP de inicio y final) y lanzar ataques indiscriminados.

Muchas herramientas simplemente comprueban si el ordenador atacado presenta una vulnerabilidad o versión antigua de software, cosa que antes o después les llevará a conseguir acceso a algún sistema conectado a Internet.

La proliferación de este tipo de herramientas junto a pensamientos erróneos del tipo "nadie me conoce" o "quien va a querer atacarme, no merece la pena" lleva a que el número de ordenadores comprometidos sea muy difícil de aproximar, y únicamente cuando es demasiado tarde (el ordenador ya ha sido utilizado para un ataque o han borrado información) las prisas y necesidades de seguridad en la empresas pasan a tomarse en serio.

Podemos observar como en el último trimestre de 2002 [Gre03] mas del 73% de los ataques observados en Internet pertenecen a la categoría de "**actividad sospechosa en la red**", indicando una bajada importante de los ataques directos y un gran aumento de comportamientos poco claros o sospechosos en la red.

Estos barridos "**ciegos**" por Internet tienen por objetivo la creación de listas de ordenadores conectados a la red (*shopping lists*). Posteriormente, estas listas que contienen las direcciones IP son utilizas por programas mas sofisticados que comprueban los servicios existentes en las máquinas buscando alguna vulnerabilidad que pueda ser aprovechada para obtener el control del ordenador.

Una vez conseguido el acceso se suelen instalar programas de puerta trasera (*back door*) para poder acceder de forma invisible a la máquina. Esta queda en estado de "espera" por tiempo indefinido hasta que el atacante (*hacker*) desee hacer uso de ella.

Cabe señalar también que muchas veces el acceso a un ordenador es utilizado como trampolín para efectuar nuevos barridos de forma que el verdadero origen del barrido no queda comprometido. Además, al igual que en los ataques DDOS, mas ordenadores comprometidos conectados a Internet significa mas potencia de búsqueda de posibles nuevas víctimas.

4.2 Historia de los Honeypots

Una vez definido el nuevo escenario hacia el que Internet se encamina, podemos observar como muchas de las premisas clásicas de seguridad ("si nadie conoce mi red nadie puede encontrarla" o "cuando menos documentada esté mi estructura más difícil será para un atacante el poder entrar") dejan de tener sentido.

Así mismo, las herramientas típicas de seguridad por excelencia (firewalls, IDS) dejan un hueco cada vez más importante sin cubrir.

Históricamente las primeras referencias a un sistema de monitorización de intrusos aparecen ya en la bibliografía sobre los años 90 de la mano de Clifford Stoll [Sto90], sin embargo los líderes en la investigación y desarrollo del concepto de Honeypot se agrupan en el **HoneyNet Project** [WWW119][WWW149].

Este proyecto se inició informalmente en la lista de correo "*Wargames*" en abril de 1999 [Kue01] gracias a los correos cruzados entre varios expertos en seguridad de redes que culminaron con el desarrollo formal del proyecto antes de finalizar el año.

En junio de 2000 y por espacio de tres semanas, el Honeypot del proyecto fue atacado y comprometido por un famoso grupo de hackers, lo que permitió el estudio del comportamiento de este grupo en "real" así como demostrar la viabilidad y utilidad de esta nueva herramienta de seguridad.

Este conocido incidente catapultó mediáticamente el concepto de Honeypot como la última tendencia en seguridad de redes convirtiendo su libro en un best-seller de lectura obligatoria para todos los profesionales de la seguridad [Hon01].

A inicios de 2001 se convirtió en una organización si ánimo de lucro [WWW123] dedicada al estudio de los hackers (*blackhats*) que actualmente está compuesta por más de 30 miembros permanentes.

4.3 Honeypots

El concepto de *Honeypot* no fue extraído o inventado de la nada, sino que es fruto de la realización de varios estudios en el campo de la seguridad de redes de ordenadores [Sch99][Hon01][Mcm01][Kue02][Ran02][VP02][Lev03].

Definiremos **Honeypot** (tarro de miel textualmente) como "*un recurso de red destinado a ser atacado o comprometido. De esta forma, un Honeypot será examinado, atacado y probablemente comprometido por cualquier atacante. Los Honeypot no tienen en ningún caso la finalidad de resolver o arreglar fallos de seguridad en nuestra red. Son los encargados de proporcionarnos información valiosa sobre los posibles atacantes en potencia a nuestra red antes de que comprometan sistemas reales.*" [Spit02].

Esta nueva aproximación a la seguridad de redes de ordenadores rompe muchos tabúes clásicos que se daban como axiomas en la seguridad informática "clásica":

- Por un lado, este nuevo elemento no sirve para eliminar o corregir fallos de seguridad existentes en nuestra red. Si nuestra red es vulnerable, añadir un Honeypot no solventará este fallo.

- Por otro lado, en lugar de evitar a cualquier precio que un atacante fije su interés en nuestra red, le invitamos o incitamos (para ser exactos deberíamos decir le permitimos) a que entre y ataque nuestra red.

Este interés en dejar una puerta abierta hacia Internet puede considerarse temeraria o incluso suicida. Si ya existen cientos de miles de ataques a sistemas "seguros" ¿porqué dejar un pote de miel en medio del camino del "oso"? ¿Por qué incitar a que ataquen nuestro sistema cuando nuestro objetivo es conseguir exactamente lo contrario?

Por muy eficientes que seamos en nuestro trabajo como administradores de red, es imposible mantener todos nuestros sistemas al día (*up to date*). Cada día se descubren al menos una docena de nuevos fallos (*bugs*) en el software existente [WWW21] (sistemas operativos, servidores de aplicaciones, servidores WWW...), consecuentemente de una forma regular salen nuevos parches y actualizaciones para todo tipo de software (ver figura 4-2).

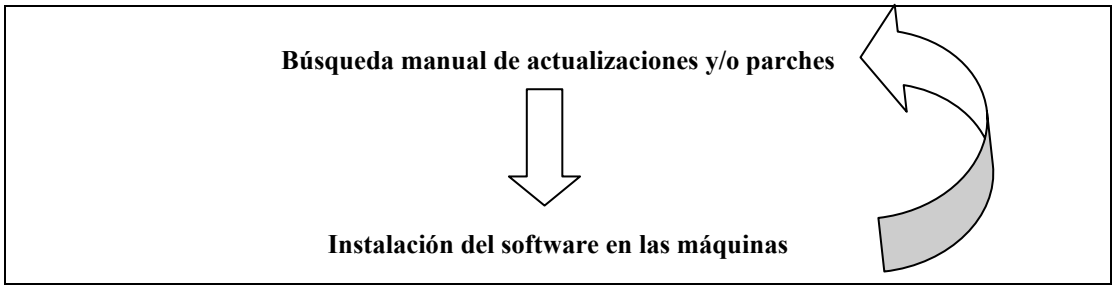

FIG. 4-2: Ciclo de actualización del software.

De esta forma, podemos observar como en la realidad es totalmente imposible mantener "controlada" una red con cientos de ordenadores y usuarios.

Si nosotros mismos examinamos periódicamente nuestra red como si fuéramos un intruso, es decir, examinando todos los ordenadores de la red en búsqueda de los servicios disponibles, podemos modificar el ciclo de actualizaciones de software (ver figura 4-3).

Así mismo, podemos por un lado corregir los problemas de seguridad en nuestras maquinas de producción (aplicación de parches de seguridad y actualizaciones de los productos comerciales) y por otro mantener "**aparentemente**" las vulnerabilidades corregidas de forma que nos permitan descubrir a potenciales atacantes.

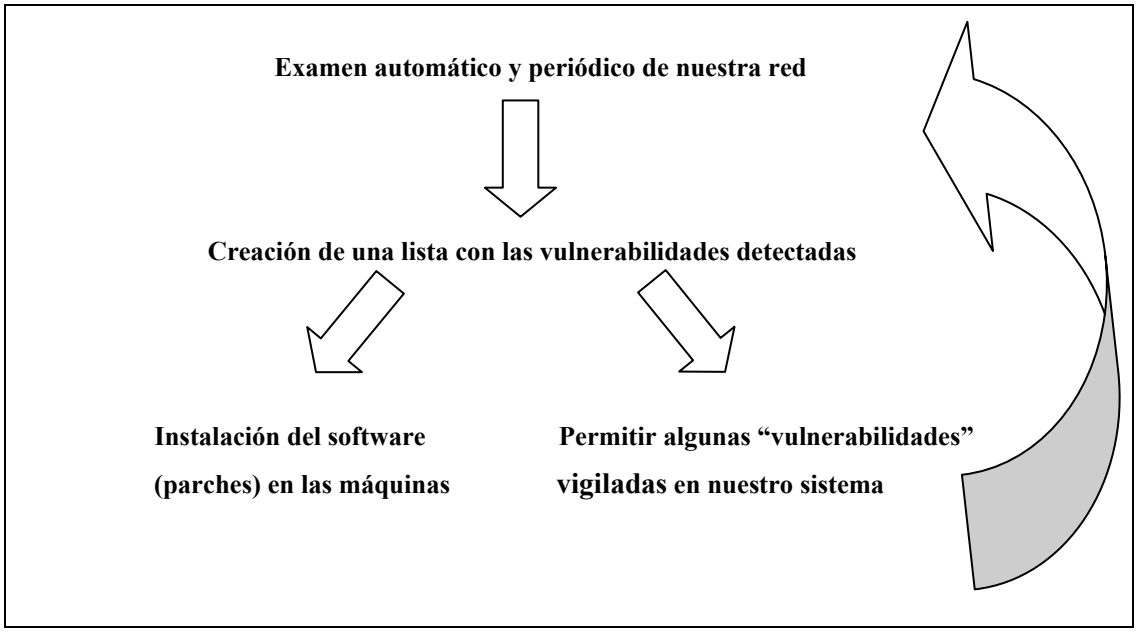

FIG. 4-3: Ciclo de actualización del software "modificado".

Estas vulnerabilidades son "mantenidas" expresamente en máquinas controladas (*honeypots*) por nuestros sistemas de seguridad (firewalls, IDS…).

De esta forma, desde el punto de vista de un atacante tiene ante sí un sistema candidato a ser explotado, por lo que invertirá más tiempo y herramientas menos sofisticadas (ya que el sistema es vulnerable por construcción) [Spi02][Spi03-2][Hon01][WWW122] que facilitarán su detección por nuestros sistemas.

Por otro lado, la inclusión de técnicas de engaño [Sch99] como el famoso "*Deception Toolkit*" [WWW134] evita que señalemos a los atacantes como y de que forma exacta pueden hacernos daño.

Si nuestra política es únicamente la de mantener nuestro sistema parcheado y al día (figura 4-2), cuando un atacante detecte una vulnerabilidad en nuestro sistema estamos perdidos, puesto que la desconocemos y esta existe realmente!.

Si mantenemos distintas vulnerabilidades de forma "controlada", estamos ofreciendo información confusa/engañosa al atacante, cosa que dificultará su trabajo. Así mismo, la detección de una vulnerabilidad en nuestro sistema no tiene porque implicar una caída del mismo, al contrario, nos permitirá descubrir a un atacante en potencia y tomar las medidas oportunas antes de que entre realmente en nuestro sistema.

Este enfoque nos permitirá obtener más información sobre el atacante que podremos utilizar en su contra. Además, este tipo de paradigma es válido tanto para atacantes internos como externos, que al intentar extender sus ataques a más ordenadores de la red, comprobarán que:

a) El resto de los ordenadores de nuestra red NO son vulnerables a los métodos que utiliza, obligándole a un cambio de estrategia que suele ser un abandono en búsqueda de otro sistema más sencillo de atacar.

b) El sistema informático de nuestra red es proactivo y reacciona a los accesos del atacante limitando su alcance.

4.3.1 Valor añadido

Una vez expuesto el nuevo ámbito de ataques en Internet y como este nos conducía hacia el concepto de Honeypot, pasamos a enumerar las principales características y ventajas que nos ofrecen los sistemas basados en Honeypots [SR01]:

- **Genera un volumen pequeño de datos:** Al contrario que los sistemas clásicos de seguridad (Firewalls, IDS…) que generan cientos de megas de ficheros de logs con todo tipo de información "poco útil", los Honeypots generan muy pocos datos y de altísimo valor.

  Los Honeypots son ordenadores "expresos" por lo que ningún usuario/sistema "normal" debe acceder a ellos. De esta forma, cualquier acceso nos revelará un atacante o una configuración errónea de un sistema. **No existen los falsos positivos.**

- **Necesita unos recursos mínimos:** A diferencia de otros sistemas de seguridad, las necesidades de un Honeypot son mínimas. No consume ni ancho de banda ni memoria o CPU extra. No necesita complejas arquitecturas o varios ordenadores centralizados, cualquier ordenador conectado a la red puede realizar este trabajo.

- **Universalidad:** Este tipo de sistemas sirven tanto para posibles atacantes internos como externos. De esta forma, obviamente, se ha de evitar poner a las máquinas nombres como "honeypot" o "attack-me" (muchas veces ni tan siquiera están dadas de altas en los DNS). Su objetivo es pasar desapercibidas en una red como una máquina más.

Por otro lado, y como siempre suele ocurrir, todo sistema tiene también unas contrapartidas o desventajas asociadas. En el caso de los Honeypots los principales inconvenientes son:

- Son elementos totalmente pasivos. De esta forma, si no reciben ningún ataque no sirven de nada.

- Son fuentes potenciales de riesgo para nuestra red. Debido a la atracción que ejercen sobre posibles atacantes, si no calibramos perfectamente el alcance del Honeypot y lo convertimos en un entorno controlado y cerrado (*jailed environment*) , puede ser utilizado como fuente de ataques a otras redes o incluso a la nuestra propia

- Consumen una dirección IP como mínimo. De todas formas, este inconveniente es mínimo, ya que lo ideal es asignar direcciones IP del rango de direcciones libres.

Del análisis de todas las características principales de los Honeypots y aplicando el desglose del concepto de seguridad [Sch00] en prevención, detección y reacción (ver figura 4-4) obtenemos el siguiente análisis:

1. Los Honeypots tienen un limitado carácter **preventivo**. No evitarán o disuadirán a ningún posible atacante.

2. Tienen un alto grado de **detección**. Si bien son elementos pasivos, los atacantes rara vez se centran en una simple máquina, sino que buscan por toda la red posibles víctimas, lo que hace que antes o después se encuentre con el Honeypot.

3. La **reacción** es otro de los valores que añade el uso de Honeypots. En los de Honeypots de producción se puede de forma automática generar los comandos necesarios para evitar el acceso del atacante al resto del sistema. En los de investigación, además nos permiten a posteriori la ejecución de técnicas forenses (*computer forense*) para examinar el comportamiento del atacante y descubrir sus comportamientos.

FIG. 4-4: Concepto de seguridad: prevención, detección y reacción.

4.3.2 Clasificación

La taxonomía de los diferentes tipos de Honeypots depende de la bibliografía consultada puesto que como todo nuevo concepto, su estandarización es compleja y aún no ha sido universalmente aceptada.

En este capítulo citaremos las dos clasificaciones mas aceptadas por la comunidad, ya que en el momento de la realización de este trabajo no se ha observado una imposición de una sobre la otra.

La primera clasificación presentada se basa en la funcionalidad que se desea asignar al Honeypot [Cho01][SR01][Ran02][Spi03] y [WWW130]:

- **Honeypot de producción** (*Production Honeypot System*): Su principal objetivo es el de mitigar el riesgo de un ataque informático a la red de una institución o empresa. De esta forma, un Honeypot de producción simula diferentes servicios con el único objetivo de ser atacado.

  Una vez descubierto el atacante/s, se "avisa" al resto del sistema para que tome las medidas oportunas (denegar cualquier acceso con un origen determinado, limitar las capacidades de un servicio, paralizar varios servicios momentáneamente…).

- **Honeypot de investigación** (*Research Honeypot System*): Su principal objetivo es el de recoger información sobre los distintos atacantes así como de sus comportamientos y técnicas asociados. También han sido diseñadas para ser comprometidas al igual que los de producción, sin embargo no añaden ninguna capacidad extra de seguridad o mitigación de los ataques.

  Suelen ofrecer servicios reales (no los simula) e incluso pueden llegar a permitir que el atacante tome el control total de la máquina (acceso *root*).

Otra posible clasificación de los Honeypots hace referencia al grado de compromiso o riesgo que esta introduzca en nuestra red [BP02][Spi02]:

- **Compromiso bajo** (*Low involvement*): El sistema Honeypot simplemente simula la existencia de que existe algún servicio común (WWW, FTP, TELNET...) escucha y almacena todas las peticiones recibidas en ficheros de logs. De esta forma, tenemos un sistema totalmente pasivo que simplemente registra peticiones de acceso ya que NO respondemos a ninguna de ellas o interaccionamos con el atacante.

  En el caso de un servidor TELNET, podríamos contestar a las peticiones TCP del puerto 23 permitiendo un establecimiento de conexión, pero cerrándola inmediatamente después del tercer paso (ver capítulo 1) evitando cualquier posibilidad de que el usuario pueda conectarse al sistema (envío y recepción de login y password).

  El riesgo que introduce esta variante es mínimo puesto que el atacante nunca podrá acceder a la máquina, lo que nos hace perder la posibilidad de investigar y analizar sus técnicas (ver figura 4-5).

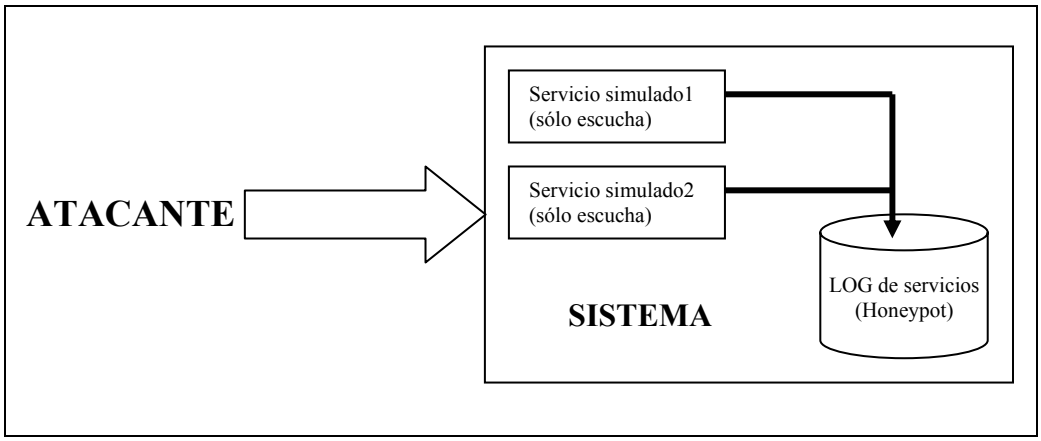

FIG. 4-5: Honeypots de compromiso bajo.

- **Compromiso medio** (*Medium involvement*): En este grupo los sistemas simulan la existencia de uno o varios servicios de forma más sofisticada. Con este tipo de Honeypots se pretende captar la atención del atacante y permitir un grado mayor de interacción que nos permitirá analizar minimamente el comportamiento del atacante.

  El grado de riesgo aumenta moderadamente, ya que por un lado el servicio sigue siendo una simulación, lo que permite tener acotado/enjaulado la interacción entre el atacante y el servicio. Por otro lado, si existe un fallo en la implementación del servicio simulado, el atacante puede aprovecharlo para atacar el sistema real[ℜ] (ver figura 4-6).

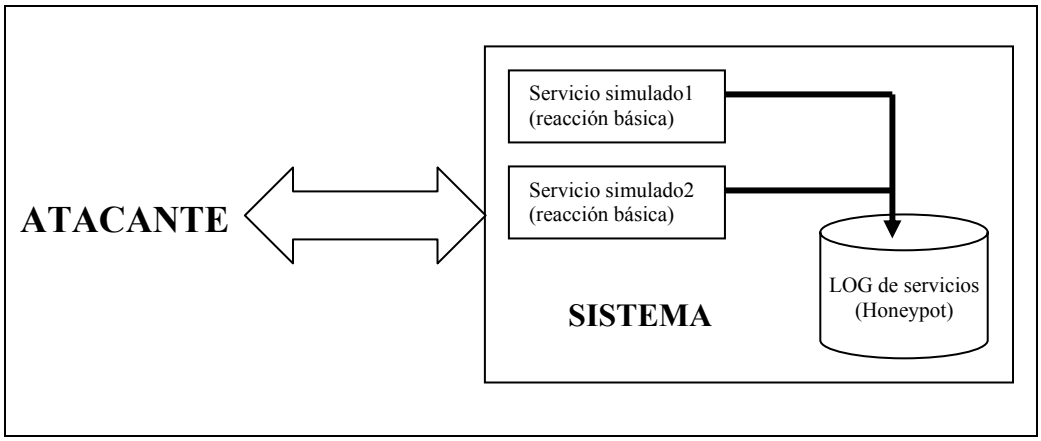

FIG. 4-6: Honeypots de compromiso medio.

- **Compromiso alto** (*High involvement*): En este grupo se encuentran aquellos sistemas que no simulan diferentes servicios, sino que utilizan un entorno real con servicios de verdad (*in the wild*).

  Este tipo de Honeypots son muy atractivas para los atacantes y permiten un estudio completo de su comportamiento. Deben estar constantemente monitorizadas ya que su peligro consiste en que si un atacante logra acceso a ella puede disponer de todo el sistema como le plazca.

[ℜ] Generalmente, estos programas son más seguros que lo servicios que simulan.

Esto significa que ya no podemos considerarlo como un lugar con logs "fiables" y puede ser utilizado para atacar otros sistemas de nuestra red o incluso de otras conectadas a Internet (ver figura 4-7).

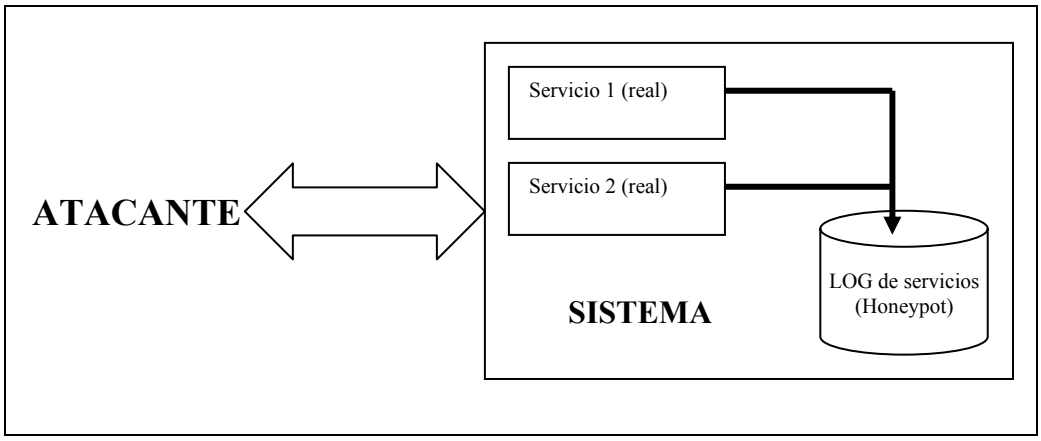

FIG. 4-7: Honeypots de compromiso alto.

En Internet podemos encontrar varias herramientas que implementan los distintos tipos de Honeypots comentados antes [WWW122][WWW123][WWW124][WWW134] y [Mar01][BP02]. Estas herramientas son muy flexibles y permiten la total configuración de servicios e incluso las vulnerabilidades a simular. Además, algunos incluso permiten escoger el tipo de sistema operativo y versión que se desea simular (Solaris, Linux, Windows…).

4.3.3 Ubicación

La ubicación de los Honeypots es esencial para maximizar su efectividad, ya que debido a su carácter intrínsecamente pasivo una ubicación de difícil acceso eliminará gran parte de su atractivo para potenciales atacantes. Por otro lado, si su ubicación es demasiado artificial u obvia cualquier experimentado atacante la descubrirá y evitará todo contacto con ella.

Por otro lado debemos tener en cuenta que debe integrarse con el resto del sistema que tenemos ya implantado (servidores WWW, servidores de ficheros, DNS…) y asegurarnos que no interfiere con las otras medidas de seguridad que puedan ya existir en nuestra red (Firewalls, IDS…).

Teniendo en cuenta que los Honeypots pueden servir tanto para la detección de atacantes internos como externos, debemos tener siempre en cuenta la posibilidad de establecer Honeypots internos para la detección de atacantes o sistemas comprometidos en nuestra red (por ejemplo sistemas infectados con un gusano o virus).

La bibliografía consultada [Eve00][BP02][Lev03-2] establece tres puntos básicos candidatos a albergar los Honeypots según nuestras necesidades:

1. **Antes del firewall** (*Front of firewall*): Esta localización permite evitar el incremento del riesgo inherente a la instalación del Honeypot. Como este se encuentra fuera de la zona protegida por el firewall, puede ser atacado sin ningún tipo de peligro para el resto de nuestra red (ver figura 4-7).

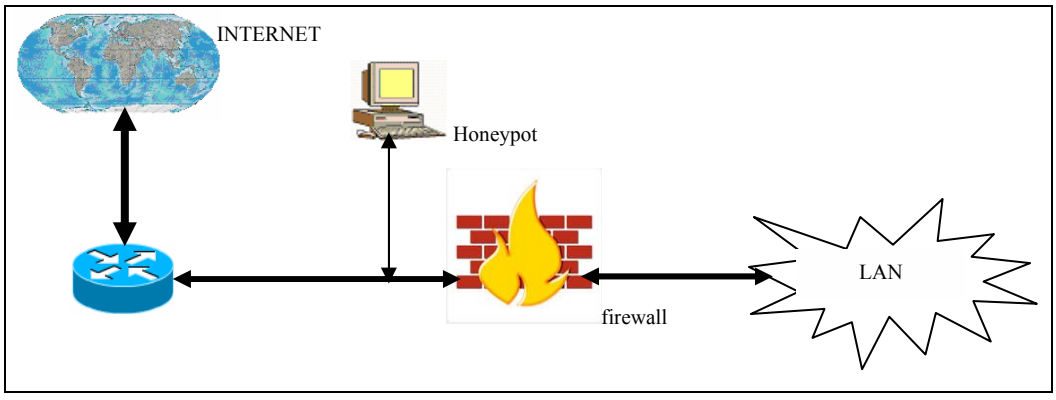

FIG 4-7: Ubicación del Honeypot antes del firewall.

Esta ubicación nos permite tener un acceso directo a los atacantes, puesto que el firewall ya se encarga de filtrar una parte del tráfico peligroso o no deseado, obteniendo trazas reales de su comportamiento y estadísticas muy fiables sobre la cantidad y calidad de ataques que puede recibir nuestra red.

Además con esta configuración evitaremos las alarmas de otros sistemas de seguridad de nuestra red (IDS) al recibir ataques en el Honeypot. Sin embargo, existe el peligro de generar mucho tráfico debido precisamente a la facilidad que ofrece el Honeypot para ser atacado.

Cualquier atacante externo será lo primero que encuentra y esto generará un gran consumo de ancho de banda y espacio en los ficheros de log. Por otro lado, esta ubicación nos evita la detección de atacantes internos.

2. **Detrás del firewall** (*Behind the firewall*): En esta posición, el Honeypot queda afectado por las reglas de filtrado del firewall. Por un lado tenemos que modificar las reglas para permitir algún tipo de acceso a nuestro Honeypot por posibles atacantes externos, y por el otro lado, al introducir un elemento potencialmente peligroso dentro de nuestra red podemos permitir a un atacante que gane acceso al Honeypot un paseo triunfal por nuestra red.

   La ubicación tras el firewall nos permite la detección de atacantes internos así como firewalls mal configurados, máquinas infectadas por gusanos o virus e incluso atacantes externos (ver figura 4-8).

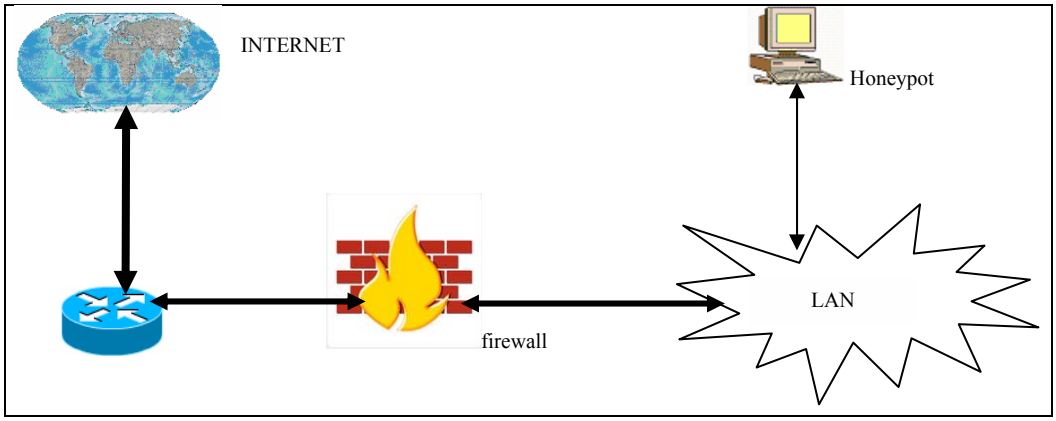

FIG 4-8: Ubicación del Honeypot tras el firewall.

Sin embargo las contrapartidas mas destacables son la gran cantidad de alertas de seguridad que nos generarán otros sistemas de seguridad de nuestra red

(Firewalls, IDS…) al recibir ataques el Honeypot y la necesidad de asegurar el resto de nuestra red contra el Honeypot mediante el uso de firewalls extras o sistemas de bloqueo de acceso, ya que si un atacante logra comprometer el sistema tendrá vía libre en su ataque a toda nuestra red.

Hay varias circunstancias que obligan a este tipo de arquitectura, como por ejemplo la detección de atacantes internos o la imposibilidad de utilizar una dirección IP externa para el Honeypot.

3. **En la zona desmilitarizada** (*into DMZ*[98]): La ubicación en la zona desmilitarizada permite por un lado juntar en el mismo segmento a nuestros servidores de producción con el Honeypot y por el otro controlar el peligro que añade su uso, ya que tiene un firewall que lo aísla de resto de nuestra red local (ver figura 4-9).

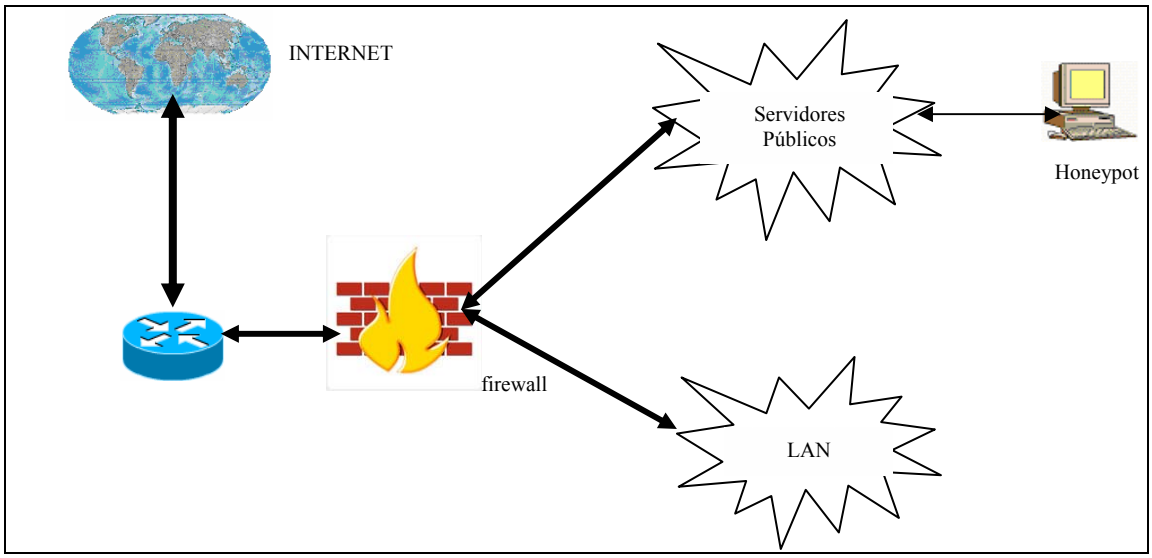

FIG 4-9: Ubicación del Honeypot en la zona desmilitarizada (DMZ).

Esta arquitectura nos permite tener la posibilidad de detectar ataques externos e internos con una simple reconfiguración de nuestro sistema de firewall puesto que se encuentra en la zona de acceso público.

[98] Para ver una descripción sobre DMZ ver [WWW137].

Además eliminamos las alarmas de nuestros sistemas internos de seguridad y el peligro que supone para nuestra red al no estar en contacto directo con esta.

La detección de atacantes internos se va algo debilitada, puesto que al no compartir el mismo segmento de red que nuestra LAN un atacante local no accederá al Honeypot. Sin embargo, desde la red local si que es posible acceder al Honeypot, con lo que un atacante interno que intente atacar nuestros servidores públicos u otros sistemas externos (un gusano por ejemplo) muy probablemente acabe siendo detectado.

4.3.4 HoneyTokens

Cabe señalar que aunque históricamente se ha asociado erróneamente un Honeypot con un ordenador, esto no solo no es cierto, sino que la misma definición se desmarca de esta mala interpretación definiéndolo como un recurso de red, no como un ordenador o recurso físico concreto.

De la misma forma, podemos definir Honeytoken como "*un recurso digital de cualquier tipo (un documento Word, Excel, un fichero de música, un número de tarjeta de crédito…) destinado únicamente a interaccionar con posibles atacantes*" [Spi03-3].

Este concepto de "información falsa" en sí mismo no es nuevo[ℜ] [Sto00], sin embargo la denominación de Honeytoken nace en 2003 de la mano de Augusto Paes de Barros en una serie de discusiones mantenidas en las listas de correo de Honeypots [Spi03-3] y [WWW119][WWW130].

Un Honeytoken es un recurso que siempre que sea utilizado u accedido lo será de forma ilícita, y por tanto lo utilizará únicamente un atacante.

Obviamente, estos recursos deben cumplir dos requisitos fundamentales:

[ℜ] Por ejemplo muchas empresas cartográficas introducían calles inexistentes en los callejeros para detectar la venta de guías plagiadas.

1. No deben tener valor real ni afectar de ninguna manera que se comprometan o divulguen.

2. Deben ser similares a uno real. El objetivo es que el atacante se lo "crea" totalmente y confíe en su autenticidad como si fuera de verdad. Verosimilitud.

Estas características intrínsecas debido a su construcción, les confieren las mismas ventajas y características que los Honeypots [Spi03-3][WWW130]. Otra característica que les hace interesantes es lo simple que resulta utilizarlos. No necesitamos ordenadores, ni licencias de ningún tipo, un simple fichero creado en 3 minutos o un número de tarjeta inventado nos bastan para tener un perfecto Honeytoken.

Un ejemplo de funcionamiento de esta técnica, consiste en introducir un número de tarjeta de crédito falso (por ejemplo el **4356974837584710**) en nuestra base de datos. El número es único y cumple los dos requisitos anteriores, es plausible y nadie de forma natural accederá nunca a él.

Simplemente necesitamos activar nuestros sensores para la búsqueda de esta información viajando por la red (por ejemplo IDS o Firewall que examine los paquetes IP) y tendremos un acceso no autorizado.

Un eventual atacante tratará de conseguir cualquier (o incluso toda) información posible, con lo que no podrá distinguir si es un Honeytoken o es un valor correcto. Para entornos de comunicación cifrados, simplemente se han de desarrollar herramientas a nivel del sistema operativo que interactúen con las llamadas del *kernel* de lectura (read()).

Cabe señalar que la detección de un Honeytoken circulando por nuestra red, no implica necesariamente que esta haya sido comprometida, sin embargo sí prueba la existencia de un comportamiento inadecuado e identifica a las partes implicadas. Una vez detectado, otros medios complementarios (IDS, ficheros de logs…) deben probar la culpabilidad real de las partes.

4.3.5 WI-FI Honeypots

Inicialmente las redes de comunicaciones fueron expandiéndose lentamente por centros militares, universitarios y centros de investigación hasta el estallido de Internet. A partir de este momento la obsesión por estar conectados ha llevado a la instalación de millones de metros de cables que cubren todo tipo de edificios y superficies.

Sin embargo, el ansia/dependencia de Internet continúa alimentando el deseo de estar conectados en cualquier sitio y a cualquier hora. Bares, librerías e incluso campus universitarios están iniciando un nuevo paso en las comunicaciones, las redes inalámbricas (*wireless networks*).

Obviamente, no podemos descartar el papel de catalizador que la gran expansión de los teléfonos móviles ha tenido en el desarrollo de estas redes sin hilos. Actualmente ya existen varios estándares de IEEE que regulan estas comunicaciones como el 802.11b y el 802.11g [WWW141].

Como ocurre con cualquier red de comunicaciones, los accesos no autorizados o los ataques perpetrados por hackers están a la orden del día. Sin embargo, el peligro de este tipo de redes se dispara debido a tres factores diferenciales:

1. Los estándares y especificaciones para este tipo de redes son "nuevos" y poco probados. Como siempre que se realiza una cosa por primera vez, la existencia de indefiniciones, malas interpretaciones o fallos de conceptos es bastante probable. Estos aspectos difusos son los más utilizados por los atacantes para obtener acceso a este tipo de redes.

2. Cualquiera con un simple portátil (o incluso con un PDA) puede intentar acceder de forma fraudulenta a nuestra red inalámbrica. A diferencia de otros tipos de redes, no necesita estar físicamente conectado o estar confinado en un área concreta. Los ataques pueden venir de cualquier sitio.

3. La mayor ventaja de las redes sin hilos es la movilidad que proporcionan. Este aspecto también puede jugar en nuestra contra, ya que el atacante puede ir cambiando de ubicación mientras realiza su ataque.

La utilización de Honeypots en las redes inalámbricas [Pou02] es una de las últimas aplicaciones que se están probando con éxito en Estados Unidos. Actualmente el programa **WISE** (*Wireless Information Security Experiment*) [WWW143] lidera la investigación en este campo.

4.3.6 Repercusiones legales

Hasta ahora, en los diferentes capítulos de este trabajo se han presentado diferentes herramientas y/o sistemas que permiten de una forma u otra aumentar la seguridad de nuestra red.

En este punto se comentarán brevemente (puesto que va más allá de las pretensiones de este trabajo) las implicaciones legales que pueden acarrear el uso de Honeypots. Cabe destacar que las siguientes líneas hacen referencia al código de leyes norteamericano, y como tal NO es aplicable a España o la CEE. Sin embargo, debido al peso específico de los Estados Unidos debe de ser tenido en cuenta siempre[ℜ]…

La ley norteamericana (al igual que la europea) protege la inviolabilidad de las comunicaciones personales (*interception of communications*) de una forma muy estricta. El punto de discusión se basa en que dependiendo del tipo de Honeypot que utilicemos, podemos violar esta ley al recoger una serie de información sobre nuestro atacante que la ley protege explícitamente [Pou03].

Tal y como se ha explicado anteriormente, los Honeypots de producción tienen un objetivo mucho mas concreto (proteger nuestra red) que los Honeypots de investigación

[ℜ] Los EEUU suelen ser los primeros en regular todo lo referente a la seguridad informática, y en especial a cualquier cosa que afecte Internet. Muchas leyes estatales y/o europeas se basan en las americanas, cosa que incrementa su interés para cualquier profesional.

(cuyo interés se centra en conseguir tanta información de los atacantes como sea posible para comprender/estudiar su comportamiento y técnicas).

Dependiendo del grado de configuración de nuestro Honeypot, podemos "espiar" los movimientos y comunicaciones que realice nuestro atacante dentro de nuestra Honeypot (hacia nuestra red o hacia el propio Honeypot) y desde nuestra Honeypot hacia terceros.

Precisamente en este "espionaje" de las comunicaciones que realice el atacante desde nuestra red hacia terceros sitios viene el problema. Estamos espiando una comunicación entre terceros sin ningún tipo de autorización para ello.

El surrealismo aumenta cuando este supuesto permite al atacante denunciarnos por invadir su intimidad y ganar el caso [Pou03][9].

De todo esto, podemos observar que antes que cualquier otra cosa debemos responder a la siguiente pregunta [Pou03][Spi03-4]: **« ¿Cuál es el objetivo de mi Honeypot? »**

De que la respuesta sea únicamente "proteger mi red" a que sea "conocer la identidad de los atacantes" o "recoger información de posibles ataques o técnicas nuevas" puede depender ganar o perder un juicio.

Lance Spitzner [Spi03-4] desglosa las posibles responsabilidades legales derivadas del uso de Honeypots en tres cuestiones básicas (obviamente siguiendo el sistema de leyes norteamericano):

1. **Trampa** (*Entrapment*): Es el proceso realizado por los cuerpos policiales (*law enforcement*) de "inducir" a alguien a cometer un acto punible con el objetivo de iniciar la acción judicial pertinente [Gar99]. En este caso del Honeypot, aunque es un elemento pasivo creado por nosotros para ser atacado (sin que seamos parte de los cuerpos policiales) si no deseamos perseguir judicialmente esta intrusión en el Honeypot, no realizamos ninguna trampa. El objetivo del Honeypot es recibir los ataques, no recoger información para demandar a los atacantes del Honeypot.

[9] Obviamente en EEUU. Pero teniendo en cuenta que este país no duda en ampliar su jurisdicción a terceros países (hay varios precedentes como casos de patentes de marcas) debemos ser precavidos.

2. **Privacidad** (*Privacy*): Que el Honeypot recoge información es innegable. Sin embargo, la información recogida puede dividirse en información transaccional (*transactional*) e información de contenido (*content*).

   La información transaccional (meta-información) no hace referencia a la información en sí, sino a aspectos de esta como la dirección IP, la fecha y hora, valores de las cabeceras de los paquetes IP…

   La información de contenido es propiamente la comunicación que realiza en atacante con terceros. Precisamente este es el objetivo del debate (y también el de los Honeypots de investigación). La intercepción de una comunicación privada es la piedra angular que puede permitir a un atacante demandarnos ante un juzgado y probablemente ganar. En cualquier caso, todos los autores están de acuerdo que se deberían incluir mensajes de advertencia y renuncia (*disclaimer*) como el indicado a continuación en todos los Honeypots. Sin embargo esto no exime del problema, ya que el hecho de que pongamos un aviso no significa que un eventual atacante lo vea o lea…

```
############################################################
#                 READ BEFORE CONTINUING:                  #
#                                                          #
# This system is for the use of authorized users only.     #
# By using this computer you are consenting to having      #
# all of your activity on this system monitored and        #
# disclosed to others.                                     #
############################################################
```

3. **Responsabilidad** (*Liability*): Este aspecto hace referencia a las posibles demandas que podemos recibir en el caso de que un atacante utilice nuestro Honeypot como plataforma de lanzamiento de ataques.

   Las demandas se basarían en que nosotros no hemos realizado unos mínimos esfuerzos de seguridad en nuestra red, sino que al contrario, facilitamos el acceso a nuestros recursos para que sean utilizados en todo tipo de ataques. ¿Qué grado de culpabilidad tenemos cuando un atacante usas nuestros recursos para atacar a otros? ¿Es punible la incompetencia en materia de seguridad informática?

4.4 Honeynets

Históricamente, una vez definido el concepto de Honeypot y realizadas las primeras pruebas con éxito, se propuso la extensión de este concepto.

Podemos definir una Honeynet como "*un tipo concreto de Honeypot. Específicamente es un Honeypot altamente interactivo diseñado para la investigación y la obtención de información sobre atacantes. Una Honeynet es una arquitectura, no un producto concreto o un software determinado.*" [Hon03] [Pro03].

El nuevo enfoque consiste no en falsear datos o engañar a un posible atacante (como suelen hacer algunos Honeypot) sino que el objetivo principal es recoger información "real" de cómo actúan los atacantes en un entorno de verdad.

Para conseguir este entorno real (con sistemas reales, no con simples emulaciones de servicios) y altamente interactivo, se dispone una configuración de red típica con todos sus elementos (ver figura 4-10).

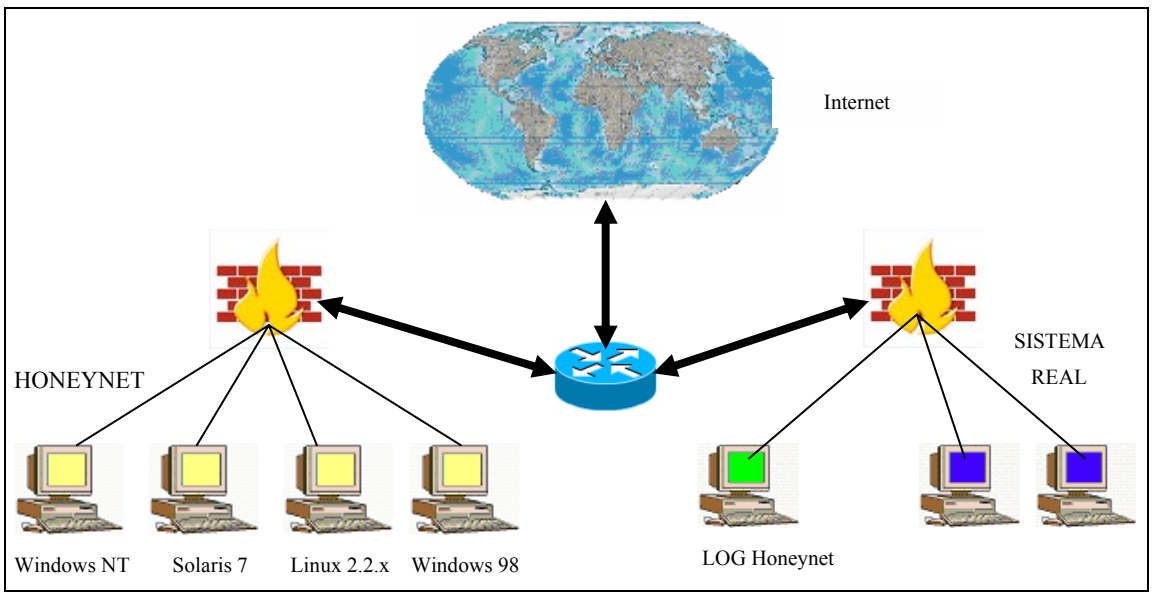

FIG 4-10: Arquitectura típica de una Honeynet.

Obviamente, esta red ha sido diseñada para ser comprometida, por lo que debe estar separada de forma segura y controlada de la de producción.

Por otro lado, como nuestro objetivo es el de hacer creer al atacante que está ante una red "real", debemos añadir los distintos elementos que conforman una arquitectura "normal" en cualquier red (distintas máquinas, distintos sistemas operativos…)

Una Honeynet presenta dos requerimientos básicos para ser realmente útil y que nos permita la extracción de información valiosa [Hon03][Hon03-2]:

1. **Control del flujo de datos** (*Data control*): Siempre que interactuamos con un atacante, el peligro aumenta exponencialmente. Todo y que el objetivo de la Honeynet es el de ser comprometida y atacada, debemos mantener siempre un control del flujo de datos para evitar que el atacante la utilice contra terceros o contra nuestra propia red.

   Si bien es cierto que cuanta más interacción permitamos con el exterior más datos reales podremos obtener del atacante, debemos evaluar los riesgos que conlleva. Análogamente, una Honeynet que no permita ningún tipo de actividad con el exterior dejará de ser atractiva para un atacante y perderá toda su utilidad. Como siempre, la búsqueda de un equilibrio nos debe guiar en este aspecto.

2. **Captura de datos** *(Data capture*): La captura de todos los movimientos y acciones que realice el atacante en nuestra Honeynet nos revelará sus técnicas y motivaciones. Si bien es esencial que el nivel de vigilancia y captura sea alto, si este es excesivo o detectado por el atacante dejará de ser efectivo.

   Obviamente, la captura de datos debe hacerse sigilosamente y sin despertar ningún tipo de sospecha, por lo que debe planificarse cuidadosamente.

El lugar dónde se almacena esta información debería encontrarse fuera de la Honeynet, ya que si realmente compromete un sistema puede encontrarla o incluso peor, falsearla o borrarla. Esto eliminaría cualquier utilidad a la Honeynet.

4.4.1 Valor añadido

Tradicionalmente, la mayoría de los sistemas de seguridad han sido siempre de carácter defensivo. IDS, Firewalls y demás soluciones se basan en la defensa de nuestros sistemas, y cuando un ataque o vulnerabilidad es detectado mirar de arreglar el problema tan rápido como sea posible.

Estas aproximaciones siempre van siempre un paso por detrás de los atacantes, ya que nuestra reacción depende directamente de los ataques sufridos y detectados. Si no recibimos ningún ataque o no descubrimos alguna vulnerabilidad, nuestros sistemas de defensa permanecerán "iguales". No hay mejora intrínseca o proactividad propia de los sistemas de ningún tipo.

Las Honeynets [Hon03-2] miran de cambiar esta actitud mediante el estudio de los ataques y atacantes. Obtener nuevos patrones de comportamiento y nuevos métodos de ataque con el objetivo de prevenirlos en los sistemas reales.

Sin Honeynets, cada vez que se produzca un ataque "nuevo" y exitoso a un sistema real existente, este dejará de dar servicio y se verá comprometido. Con las Honeynets, un ataque exitoso o nuevo no tiene porqué afectar a ningún sistema real.

Además perderá el factor sorpresa, ya que habremos obtenidos datos precisos de su ataque en el estudio de los logs, cosa que permitirán contrarrestarlo de una manera más eficiente.

Al igual que los Honeypots, la cantidad y calidad de información producida es muy importante, ya que cualquier actividad existente es sospechosa.

4.4.2 GEN I

Tal y como sucede en cualquier programa (software) o especificación formal (RFC por ejemplo), con el tiempo se van perfilando nuevas mejoras y ampliaciones de funcionalidades. Las Honeynets no son ninguna excepción a esta regla, y en el momento de escribir este capítulo, existen dos generaciones diferenciadas de Honeynets [Hon03].

Las Honeynets de primera generación (**GEN I**) se caracterizan por implementar únicamente los mecanismos básicos de control de flujo de datos (*Data control*) y captura de datos *(Data capture*). La arquitectura utilizada presenta una subdivisión en tres subredes (la Honeynet, la red de producción e Internet) separadas perfectamente por un firewall (ver figura 4-11).

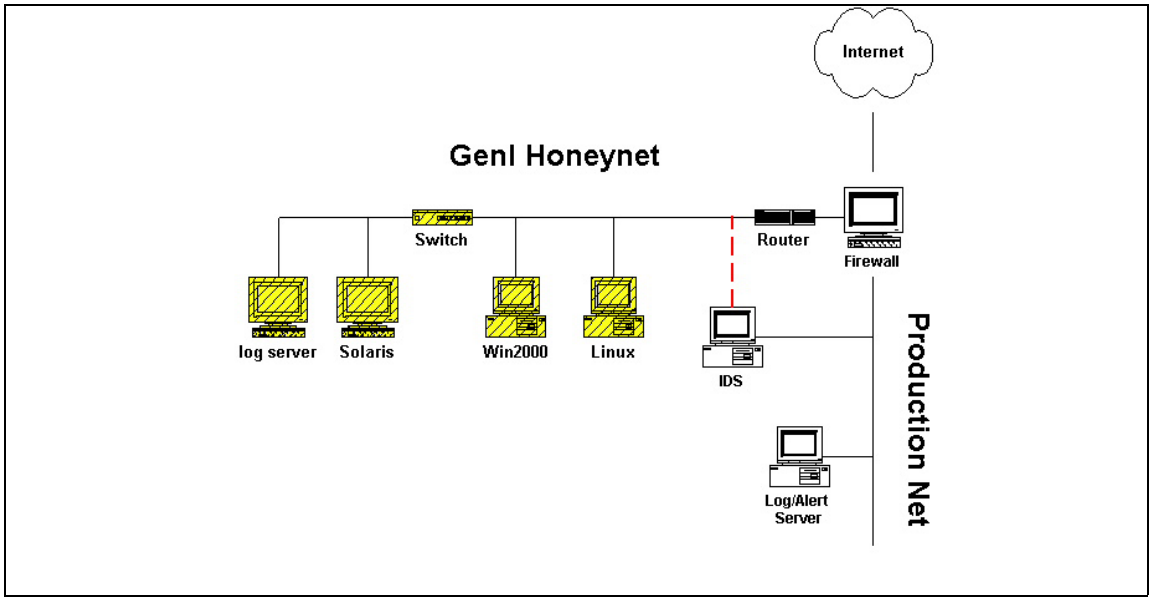

FIG 4-11: Arquitectura típica de una Honeynet de primera generación (GEN I).

En esta arquitectura, el control de todas las redes se realiza mediante un firewall, ya que cualquier paquete de entrada o salida debe pasa obligatoriamente por él. El router también se utiliza como medio de control y filtrado, pero únicamente como soporte al firewall.

El firewall está configurado para llevar la cuenta del número de conexiones que se establecen tanto hacia dentro como hacia fuera. De esta forma, cuando el número de conexiones desde nuestra Honeynet hacia el exterior supera un límite preestablecido, estas quedan anuladas. El límite de conexiones depende del lo atractiva que deseamos hacer nuestra red a un eventual atacante, se recomiendan valores entorno a las diez conexiones por hora [Hon03].

Una red que permita cualquier tipo y número de conexiones al exterior es un gran atractivo para cualquier atacante. Sin embargo, también es un riesgo potencial demasiado grande, ya que puede ser utilizada como trampolín de ataque a otros sistemas externos. Por otro lado, una que no permita conexiones al exterior carece de interés y por lo tanto eliminará la obtención de datos "reales" (*in the wild*) que es nuestro principal objetivo.

En esta configuración, el firewall se coloca expresamente por delante del router. Los motivos de esta configuración son por un lado esconder/disimular el firewall a los atacantes y por otro dotar de medidas extras de seguridad a nuestra Honeynet para que no sea utilizada para ataques a terceros, ya que el router también realizará parte del filtrado de las comunicaciones de entrada y salida. Nuestro control de flujo de datos (*Data control*) no debe depender únicamente de un solo punto de control[ℜ].

Cuando un atacante tome el control de nuestra Honeynet, encontrará que puede salir a Internet directamente, ya que pasa por un router real de producción, asumiendo que el filtrado de paquetes que exista (nuestro firewall) será un filtrado realizado por nuestro ISP y no por nosotros. De esta forma reforzamos su opinión de que somos una red "desvalida".

La captura de datos (*Data capture*) se inicia en el propio firewall, ya que es un dispositivo lo suficientemente potente y colocado estratégicamente para recibir todos los paquetes de entrada y salida. Como sabemos que cualquier tipo de actividad hacia o desde nuestra Honeynet es potencialmente peligroso, debemos capturar todo este tráfico en el firewall.

[ℜ] Esto nos permite aplicar técnicas de ingres/egress filtering así como filtros contra posibles ataques DOS/DDOS

La captura de puertas traseras (*back doors)* en nuestros sistemas es otra de las posibilidades que nos brinda este tipo de arquitectura, ya que una correlación de peticiones a puertos "no standards" revelará rápidamente su existencia.

La captura de todo el tráfico existente en nuestra Honeynet es realizado por el sistema de detección de intrusos (IDS). En la arquitectura de GEN I podemos observar como el IDS está conectado a nuestra Honeynet de forma que por un lado registra la actividad existente en la red y por otro actúa como un detector de firmas (*signatures*)

Finalmente, toda esta información se va copiando diariamente en un ordenador "seguro" para poder analizarla o consultarla como un histórico. Esto nos asegura que en caso de que el atacante encuentre la información de sus movimientos y decida destruirla no afecte excesivamente al sistema.

Obviamente, la existencia de un ordenador-almacén puede ser fácilmente descubierta por un atacante. Sin embargo esta arquitectura no trata de esconderlo, ya que lo más que puede hacer el atacante es desactivar el "*log daemon*" de la máquina atacada, comportamiento ya muy habitual en muchos atacantes.

En caso de que el atacante decidiera atacar directamente a nuestro sistema de almacenamiento "seguro", debería enfrentarse a un sistema mucho mas seguro. Incluso en el caso de que lograra entrar y eliminar todos los logs del sistema no sería catastrófico, puesto que el sistema IDS registra toda la actividad de la red y podríamos ver al sistema IDS como un segundo log del sistema.

4.4.3 Gen II

Las Honeynets de segunda generación (**GEN II)** se caracterizan por combinar todos los requisitos de la primera generación en un solo dispositivo (ver figura 4-12). De esta forma tanto el control de flujo de datos (*Data control*) como la captura (*Data capture*) y la recolección de datos quedan agrupadas en una misma entidad [Hon03-3].

Esta entidad es un dispositivo de capa 2 (*Layer 2 gateway*) con capacidades de transmisión de paquetes tipo puente (*bridge*). Esta capacidad de puente hace referencia a la transparencia con que actúa este dispositivo de red, simplemente se encarga de enviar por un extremo lo mismo que recibe por el otro, lo que le convierte en un dispositivo "invisible".

El hecho de presentarse como un dispositivo de capa 2 hace que este dispositivo carezca de pila IP, por lo que no generará ningún tipo de tráfico que el atacante pueda observar.

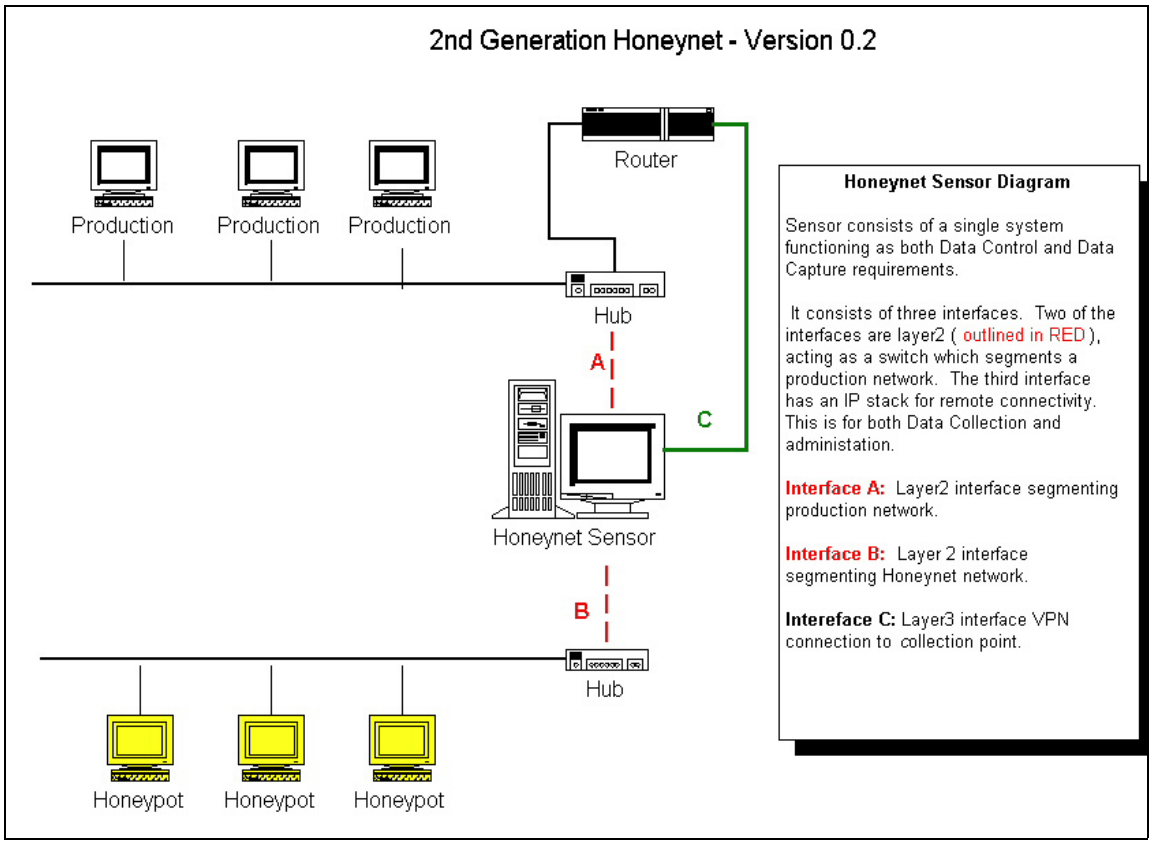

FIG 4-12: Arquitectura típica de una Honeynet de segunda generación (GEN II).

En alguna bibliografía [Hon03-3] tanto para GEN I como para GEN II, se denomina al dispositivo que controla todo el acceso de entrada y salida a nuestra Honeynet **Honeywall**. Esta nomenclatura viene dada porque conceptualmente este dispositivo es el muro que separa nuestra Honeynet del resto de las redes.

Esta unión de capacidades en una única entidad compacta facilita la instalación y administración de la Honeynet. Debido a su actuación como dispositivo transparente, la detección por parte de un atacante se dificulta enormemente.

Pese a que en el esquema parezca que hay dos redes (la de producción y la Honeynet) realmente sólo hay una. El dispositivo es de capa 2, y por lo tanto NO tiene dirección IP en los interfaces A y B, lo que "virtualmente" significa que a nivel de IP no existe. El dispositivo C si que tiene asignada una dirección IP y servirá como canal seguro (VPN) para el control, administración del Honeywall y para el movimiento de los ficheros de logs hacia un repositorio "seguro".

La primera mejora que introduce esta arquitectura hace referencia al control del flujo de datos. En la GEN I realizábamos un control del total de conexiones desde nuestra Honeynet hacia el exterior. De forma que si no se alcanzaba un umbral definido, las conexiones hacia el exterior eran "permitidas".

Con este paradigma de GEN II, obtenemos un control total de todas y cada una de las conexiones que pueda realizar, independientemente de si su número es mayor, menor o igual al límite existente en las Honeynet de GEN I.

La segunda mejora que obtenemos hace referencia directa a la forma en que podemos responder ante las actividades no autorizadas desde nuestra Honeynet. En lugar bloquear simplemente cualquier tipo de acceso no permitido, podemos modificar y dosificar la actividad del atacante. De esta forma, el ataque como tal sale de la Honeynet (cosa que puede ver y comprobar el atacante) sin embargo no es efectivo debido a las modificaciones que podemos realizar en tiempo real sobre el paquete de datos (cosa que muy difícilmente puede detectar el atacante).

Esta capacidad añade credibilidad ante el atacante, que verá como sus paquetes "salen" hacia sus nuevas víctimas y retornan sin ningún tipo de filtro o control. Sin embargo, estos paquetes son modificados en tiempo real de forma que si utiliza una vulnerabilidad del servicio FTP, sea detectada por nuestro sistema y modifique el paquete de forma que sea inocuo.

Los paquetes entran y salen "libremente", sin embargo su contenido no. Esta nueva característica añade las capacidades de un sistema NIPS (*Network Intrusion prevention System*) a las capacidades de bloqueo ya existentes en la GEN I.

Este sistema es tan flexible que no sólo permite modificar los paquetes que salen hacia fuera, sino que incluso nos permite modificar la respuesta que el atacante obtiene por parte del sistema externo que quiere atacar. Podemos por ejemplo devolverle paquetes TCP-RST (reset) simulando que el sistema atacado rechaza sus peticiones de control.

La captura de la actividad que realiza el atacante en cualquiera de los sistemas incluidos en nuestra Honeynet ha variado substancialmente en los últimos años. Las capturas directas del teclado o las capturas de paquetes de red utilizando *snifers* son inefectivas ante el uso de SSH o cualquier otro sistema de comunicación cifrado.

La herramienta SEBEK2 [WWW148] permite la captura de este tipo de información en espacio de kernel. No entraremos detalles puesto que su funcionamiento se sale de los propósitos de este trabajo. Para ampliar este punto consultar la bibliografía proporcionada.

4.4.4 Honeynets virtuales

Uno de los problemas mas graves a los que se enfrenta cualquier administrador de redes y por tanto el grupo de seguridad, es el de la disponibilidad de recursos. Los esquemas presentados (GEN I y II) cada vez demandan mas máquinas y recursos, lo que puede llevar a la paradoja de tener dos servidores de producción y cuatro ordenadores en una Honeynet.

Obviamente, este punto es algo exagerado, sin embargo si que puede pasar que mucha gente en pequeñas empresas descarte los Honeypots/Honeynets simplemente porque no tiene los recursos necesarios o porque estos son iguales o superiores a las máquinas de producción.

El concepto de **Honeynet virtual** se puede definir como "*la solución que permite implantar el esquema de Honeynet utilizando un único ordenador. El adjetivo virtual viene dado porque todos los sistemas operativos que existen en la Honeynet aparentemente tienen su propia máquina, aunque realmente se ejecutan en el mismo hardware*." [Hon03-4].

Las Honeypots virtuales pueden clasificarse en dos grupos:

- **Honeynets autocontenidas** (*Self-contained virtual Honeynet*): En este tipo de Honeypots virtuales todo el software se ejecuta en una misma máquina (ver figura 4-13). Usualmente una Honeynet se compone de un firewall o Honeywall encargado de controlar el acceso así como de registrar los diferentes logs y de uno o varios Honeypots.

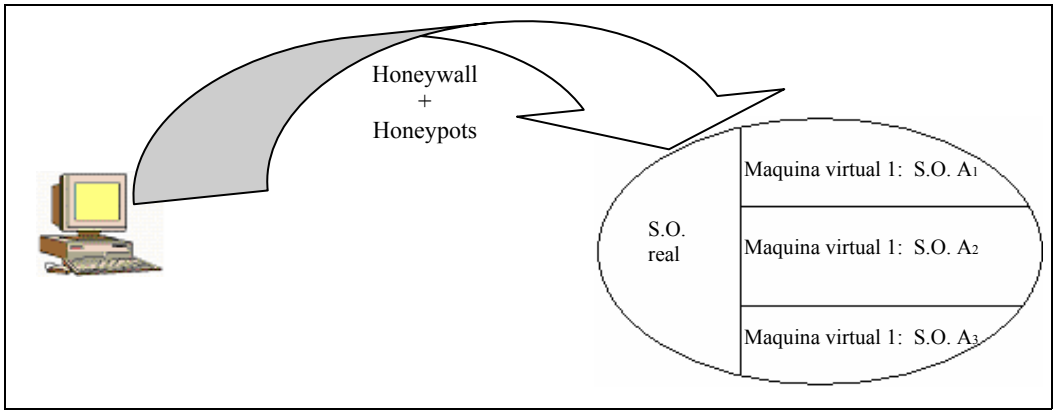

FIG 4-13: Honeynet autocontenida (self-contained).

De esta forma primero instalamos el sistema operativo anfitrión (real) y después el software de emulación virtual[ℜ]. A continuación realizamos para cada software de emulación la instalación del sistema operativo correspondiente y finalmente realizamos la configuración de cada Honeynet.

Sus principales ventajas son el reducido número de recursos necesarios y la facilidad de instalación.

[ℜ] Existe mucho software de emulación disponible, los más populares actualmente son el VMWARE y el WINE.

Por el contrario, tenemos un único punto de servicio (si la máquina se estropea o el atacante se hace con el control, quedamos sin ningún tipo de protección).

- **Honeypots híbridas** (Hybrid virtual Honeynet): En este caso se produce una división entre el Honeywall (punto de entrada, control y recolección de información de la Honeynet) y los Honeypots existentes en la Honeynet (ver figura 4-14).

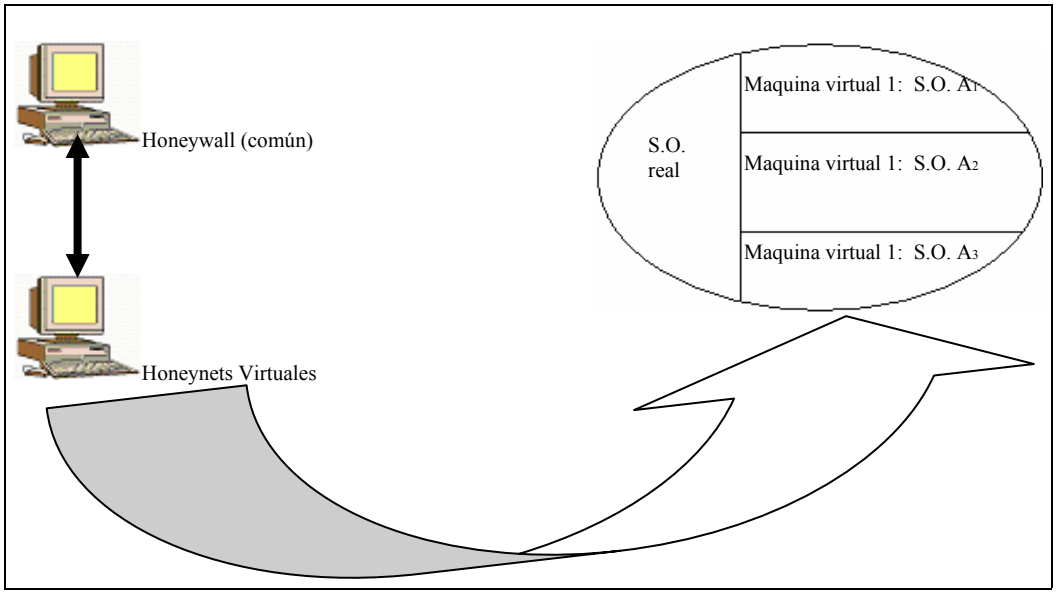

FIG 4-14: Honeynet híbridas (Hybrid).

En este esquema el peligro queda reducido al acceso del atacante al resto de Honeynets. Necesitamos un ordenador más y la configuración es algo más compleja. Por el contrario, ganamos en flexibilidad y seguridad ya que tenemos el control de acceso y los logs fuera de la Honeynet virtual.

Podemos ver que ambos esquemas tienen ventajas e inconvenientes, sin embargo las contrapartidas comunes al uso de esta implantación de las Honeynets virtuales son principalmente cuatro:

1. El hardware de que disponemos 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.

2. La potencia de la máquina empleada marca la cantidad de Honeynets que podemos realizar. Si tenemos un PC 486 con 8MB de RAM no tiene sentido intentar emular varias Honeynets con decenas de ordenadores distintos.

3. Si el atacante toma el control de la máquina, podría obtener el control de todos los sistemas virtuales

4. Las técnicas de *fingerprinting* (obtención del tipo y versión del sistema operativo mediante el envío de paquetes IP específicamente construidos) podrían revelar la naturaleza real de nuestro sistema operativo "base". Incluso aunque en principio esta técnica no funcionara, si el atacante toma el control de una de las Honeynets probablemente pudiera averiguar que se encuentra en un sistema simulado, lo que falsearía su comportamiento.

4.4.5 Honeynets distribuidas

El siguiente paso que se está realizando con las Honeynets es la aplicación del viejo principio de "**la unión hace la fuerza**". El escenario con el que nos encontramos puede ser el de una gran organización internacional con múltiples redes en múltiples países o varios equipos de expertos en seguridad diseminados por todo el planeta (ver figura 4-15) que desean compartir la información generada por sus Honeynets [Pro03].

Obviamente, la posibilidad de centralizar (o al menos comunicar) las distintas Honeynets para la recolección de información es básico puesto que nos permitirá la correlación de resultados así como la comunicación de nuevos descubrimientos de forma más rápida y eficiente.

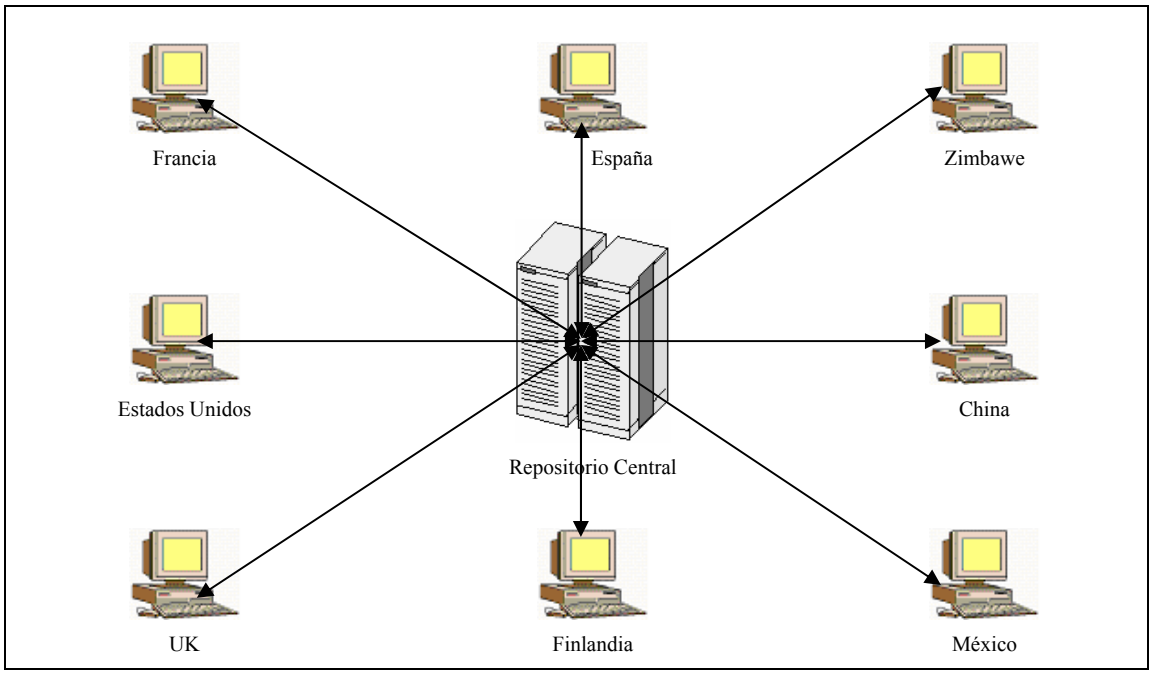

FIG 4-15: Arquitectura distribuida de Honeynet.

Una vez que ya tenemos definidas unas arquitecturas estables (GEN II) y lo suficientemente flexibles como para permitir la interconexión de los sistemas, el siguiente paso es plantearse cómo hacerlo.

El modelo escogido es mediante la creación de túneles privados virtuales cifrados con **IPSec** (*IPSec VPN*). De esta forma, una vez al día los diferentes Honeywalls se conectarán por el interface C (ver figura 4-12) de forma segura al repositorio central para depositar los ficheros de log del día anterior.

4.5 RESUMEN

En este capítulo hemos examinado el cambio de escenario de los ataques informáticos que ha producido el avance tecnológico de las redes de ordenadores. Se ha introducido la necesidad de nuevos modelos de seguridad para cubrir este nuevo espacio concluyendo en la presentación del concepto de Honeypot.

Los **Honeypots** (tarro de miel) son elementos pasivos diseñados para ser atacados que nos permiten examinar el comportamiento de los atacantes en un medio interactivo.

Se han comentado sus tipos (de producción o investigación), ubicaciones posibles (antes del firewall, tras el firewall o en la DMZ), puntos fuertes (uso mínimo de recursos, facilidad de integración en las redes, detección, análisis) y débiles (introducen un riesgo potencial, no disuaden a los atacantes y no arregla fallos de seguridad ya existentes en nuestra red) así como las repercusiones legales de su uso según la legislación de Estados Unidos.

También se ha introducido el concepto de **Honeytoken** (recurso digital diseñado para ser comprometido) así como sus características principales (económicas y fáciles de generar) y valor añadido (detección de actividad ilícita).

Definimos **Honeynet** como un conjunto de Honeypots destinada a recoger información sobre potenciales atacantes. Describimos los componentes de su arquitectura interna (control de flujo de datos y captura de datos) así como las características de las dos generaciones (GEN I y GEN II) existentes en la actualidad.

Finalmente hemos introducido el concepto de **Honeynet virtual** (solución que permite implantar el esquema Honeynet con un único ordenador) como respuesta a las necesidades de economía de recursos así como sus diferentes tipos (autocontenidas e híbridas) y características.