Temas para la Educación
revista digital para profesionales de la enseñanza
Nº 5 - Noviembre 2009 | Federación de Enseñanza de CC.OO. de Andalucía | ISSN: 1989-4023 | Dep.Leg.: GR 2786-2008

HoneyNets, una desconocida en la seguridad informática

Introducción

La idea de Honeypot es desarrollada con el término Honeynet (Red Trampa). Esta expresión fue adoptada por The Money Project; una organización no lucrativa, fundada por Lance Spitzner. Este grupo está compuesto por expertos en seguridad, cuyo objetivo es aprender las herramientas, tácticas y motivos de los atacantes.

La seguridad de la información es un tema prioritario. Para nadie es un secreto que la tarea de mantener un servidor o un equipo cualquiera, conectado a una red, representa uno de los mayores riesgos posibles. En cualquier momento el sistema puede ser comprometido por un tercero y junto con él, la información.

Una Honeynet es una herramienta de investigación. Es un tipo de Honeypot que consiste en una red diseñada para ser comprometida por intrusos. Sirve para estudiar las técnicas utilizadas por los intrusos que han comprometido la seguridad de la red. El objetivo principal es conocer al enemigo, aprender de él, en definitiva es una herramienta diseñada con propósitos académicos.

Una Honeynet no es lo mismo que un sistema trampa tradicional, a continuación se describen las diferencias más significativas:

Una Honeynet no es un sistema solitario, sino una red. Esta red puede estar compuesta por distintos sistemas trampa, tales como Linux, Windows, Solaris, routers, conmutadores, etc. El hecho de proporcionar un entorno de red aporta un ambiente más creíble, más real desde el punto de vista del intruso, del atacante de la red. Un entorno de sistemas heterogéneos permite además, captar la atención de más intrusos, algunos de los cuales están especializados en atacar determinados sistemas operativos o servicios. Por otra parte, permite aprender un mayor y variado número de tácticas de ataque.

Los sistemas utilizados en una Honeynet son sistemas de producción, es decir, son sistemas reales, aunque no se utilicen con otro propósito que el de monitorizar su actividad. Ningún sistema o servicio es emulado. No se hace intento de alguno de disminuir la seguridad. Normalmente se instalan sistemas trampa conocidos, con la configuración que traen por defecto, como Linux Red Hat, servidores Windows o servidores Solaris.

Los Honeynets son herramientas de seguridad con un punto de vista diferente al tradicional, que es un comportamiento defensivo, tradicionalmente se intenta defender de ataques una red, mediante cortafuegos, medios de cifrado o sistemas de detección de intrusos (IDS). Los Honeynets son herramientas diseñadas básicamente para aprender y adquirir experiencia en el área de seguridad.

1.1. ¿Cómo trabaja una HoneyNet?

Desde el punto de vista técnico, una honeynet es una red de sistemas de producción o servidores, diseñada para ser comprometida y conformada por varias Honeypots.

El punto crucial que asegura el éxito de la honeynet es la creación de un ambiente que permita monitorizar todos y cada uno de los movimientos que el intruso llegue a realizar dentro de ella. Estos movimientos que realiza el intruso son monitorizados mediante la información que van dejando al utilizar herramientas, mediante las cuales siguen unas tácticas para conseguir algún objetivo que se imponen, hay que saber cuales son los objetivos y que motivos les llevan a ponerse esos objetivos.

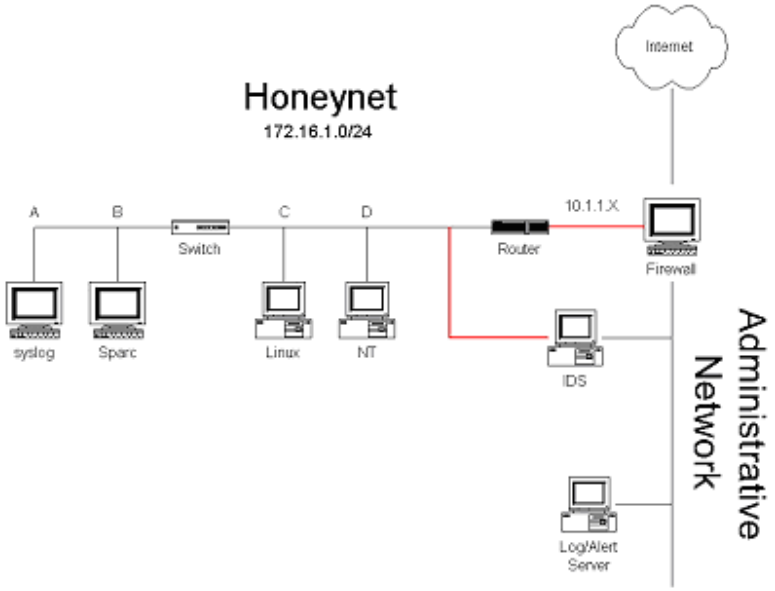

Figura 1.1: Ejemplo de Honeynet Project - project.honeynet.org

Identificar y localizar las actividades de los atacantes es el problema más serio, en la medida en que se debe hacer dentro del tráfico producido en la red, realizando un sniffer del tráfico y estudiando los distintos paquetes que utiliza el atacante con el fin de observarlo.

1.2. ¿Cómo recolectar información en una HoneyNet?

Una Honeynet debe capturar la mayor cantidad de información para analizar y encontrar nuevas estrategias, herramientas y ataques, sin que el atacante perciba que está siendo observado. Por ello, la captura de datos debe ser muy cuidosa. Lo ideal es hacerlo por capas. Es decir, tener varios elementos que recolecten datos en varios lugares dentro y fuera de la honeynet. Uno de éstos

es el firewall, el cual normalmente se coloca a la entrada, para poder capturar la información y seleccionar que tipo de trafico se quiere examinar, además de alertar en el momento de un ataque.

Otra de las capas que pueden constituir la captura de datos dentro de la honeynet son los sistemas de detección de intrusos, IDS.

1.3. Centralización de la 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 que se encuentran presentes en la red.

Es ideal almacenar la información remotamente. Si se practica en la máquina que está siendo atacada dentro de la Honeynet, es posible que el atacante lo descubra y ponga en riesgo la honeynet completa. Los datos guardados pueden ser también perdidos o destruidos. No sólo tenemos que capturar cada movimiento del blackhat sin su conocimiento, sino que tendremos que guardar la información de forma remota. La clave está en capturar los datos por capas. No puede depender sólo de la información de una capa. Debe recoger datos de varios recursos. Así, de forma combinada, estas capas le permitirán pintar el gran cuadro.

1.4. Cuidados en una HoneyNet

Como se trata de una red normal, es necesario tener un cuidado especial. Es decir, cada vez que recibe un ataque hay que detectarlo y ponerle solución. De no hacerlo la red puede quedar por fuera de servicio durante largo tiempo y deja de cumplir su función principal. Igualmente, puede ser utilizada como trampolín para otros ataques.

Las Honeynets no son soluciones para enchufar y olvidar. son un complejo tipo de honeypot que requiere mantenimiento constante, administración y vigilancia. Para la máxima efectividad, necesita detectar y reaccionar a los incidentes tan pronto como sea

posible. Observando las actividades de los blackhat en tiempo real, puede maximizar las capacidades de captura de datos y análisis. Además, para detectar los desconocidos, se requiere una revisión constante de las actividades sospechosas. Esto requiere mucho tiempo y capacidad de análisis. Por ejemplo, en sólo 30 minutos un blackhat puede hacer el suficiente daño a un honeypot comprometido que requiera 30-40 horas para entender completamente qué ha ocurrido.

El mantenimiento constante es requerido para mantener la operabilidad de la Honeynet. Si algo va mal (y siempre hay algo) esto puede causar un fallo dentro de la honeynet. Sus procesos de alerta terminarán, los discos pueden llenarse, las firmas IDS pueden caducar, los ficheros de configuración corromperse, los registros del sistema necesitarán ser revisados, los cortafuegos necesitarán ser actualizados y parcheados. Esto representa algunos de los cuidados constantes e introducción de datos que se requiere para un correcto funcionamiento de la Honeynet. Su trabajo no hace más que empezar cuando construye una Honeynet.

Además, hay riesgos implícitos en la construcción e implementación de una Honeynet. Tenemos blackhats atacando y comprometiendo nuestros sistemas. Estableciendo una red para ser comprometida, nos exponemos nosotros mismos, y a otros a un cierto riesgo. Usted asume la responsabilidad de asegurarse de que la Honeynet, una vez comprometida, no puede ser usada para atacar o dañar otros sistemas.

No obstante, con un entorno como este, siempre hay un riesgo potencial de que algo vaya mal. Hemos implementado diversas medidas para reducir este riesgo. Sin embargo, es posible que un blackhat desarrolle un método o herramienta que le permita saltarse nuestros métodos de control de acceso. Además, se necesita actualizar y comprobar constantemente el entorno para asegurarnos de que las medidas de control funcionan correctamente. Nunca subestime el poder creativo de la comunidad blackhat. El uso de un cortafuegos, routers y otras técnicas ayudan a reducir el riesgo del uso de la Honeynet para dañar otros sistemas. Aún así, todavía hay riesgos.

Por último, las honeynets no solucionarán sus problemas de seguridad. Recomendamos encarecidamente que las organizaciones se centren primero en mejores prácticas como la autenticación fuerte, uso de protocolos cifrados, revisión de registros del sistema, y versiones seguras del sistema. Mediante la prioridad en políticas y procedimientos adecuados, las organizaciones pueden reducir considerablemente los riesgos.

1.5. Arquitectura de las HoneyNets

Hasta ahora, no existe un modelo cerrado de arquitectura de Honeynet. Para su desarrollo hay 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 ha definido dos tipos de arquitecturas básicas para sus Honeynets denominadas GEN I y GEN II.

1.5.1.GEN I

Esta arquitectura simple fue la primera en desarrollarse, en 1999. Una red es situada detrás de un dispositivo de control de acceso, generalmente un cortafuegos, como se muestra en la siguiente figura.

Las tecnologías Gen I implementan el Control de Datos y la Captura de Datos son medidas simples pero eficaces.

Primero trataremos como controlaremos a los agresores, y después como capturar sus actividades.

En el diagrama, puede ver un cortafuegos de capa tres separando la Honeynet en tres redes diferentes. Concretamente, la Honeynet, Internet y la red Administrativa. Cualquier paquete entrando o saliendo de la Honeynet tiene que pasar a través del cortafuegos y del router. El cortafuegos es nuestra principal herramienta para controlar las conexiones entrantes y salientes. El router se usa como suplemento a este filtrado. Nuestro cortafuegos está diseñado para permitir cualquier conexión entrante, pero controla las conexiones salientes.

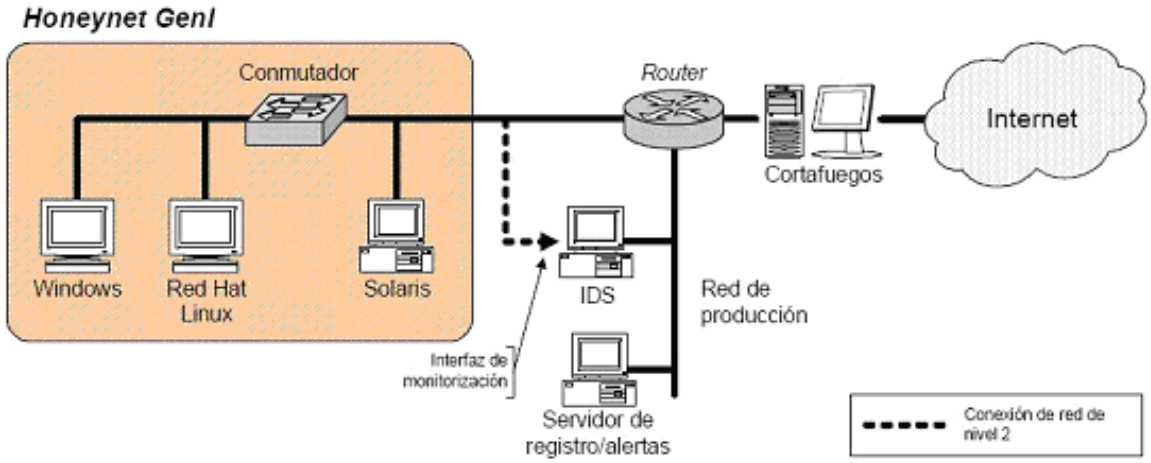

Figura 1.2: Esquema de HoneyNet de Primera Generación

El cortafuegos mantiene una traza de cuantas conexiones se realizan desde un honeypot saliendo a Internet. Una vez que un honeypot ha alcanzado un determinado límite de conexiones salientes, el cortafuegos bloqueará cualquier nuevo intento. Esto proporciona al blackhat la flexibilidad para ejecutar aquello que necesite, mientras que nos ofrece una protección automática contra el abuso. No hay un número correcto o incorrecto de conexiones a las que permitir. Esto depende de la

funcionalidad que se busque.

Si quiere capturar ataques automatizados, como los autorooters o los gusanos, entonces probablemente no necesitará conexiones salientes. Los ataques automatizados simplemente comprometen uno de los honeypots y entonces el cortafuegos bloquea todas las conexiones salientes, deteniendo así la reproducción de los ataques automáticos. De esta forma puede atrapar a las herramientas utilizadas para los ataques sin poner en riesgo a otros sistemas. Sin embargo, si desea descubrir actividad manual de los blackhat, o qué ocurre después de que un sistema sea comprometido, entonces probablemente necesitará permitir algún tipo de conexiones hacia el exterior. De nuevo, cuanta más actividad saliente permita, más podrá aprender pero mayores serán los riesgos.

Creemos que entre cinco y diez conexiones salientes por hora puede ser un buen número para mantener la felicidad del blackhat, mientras protegemos a otros de sus ataques. Esto protege a la Honeynet de ser usada como plataforma para escanear, sondear o atacar a otros sistemas. Algunas organizaciones no requerirán esta funcionalidad. Si puede permitirse el tener a alguien monitorizando la Honeynet 24 horas al día, entonces puede permitirse el permitir conexiones salientes ilimitadas. Si se identifica un ataque de Denegación de Servicio, la persona que monitoriza la Honeynet puede simplemente deshabilitar el ataque. Sin embargo, recomendamos la implementación de medios automatizados para hacer esto, puesto que muchas organizaciones no pueden permitirse una vigilancia de 24 horas al día.

Adicionalmente, un router se ha colocado entre el cortafuegos y la Honeynet. Esto se hace por dos razones:

- Primera, el router oculta al cortafuegos. Cuando una honeypot es comprometida, los blackhats encontrarán un router productivo entre ellos y las redes exteriores. Esto crea un entorno más realista y esconde al cortafuegos para no ser descubierto.
- La segunda razón es actuar como una segunda unidad de control de acceso. El router puede complementar al cortafuegos, asegurándose de que los honeypots comprometidos no se usan para atacar sistemas fuera de la Honeynet.

El router actúa como un control de acceso en la capa dos, ninguna Honeynet debe depender de una sola fuente para el Control de Datos. En principio utilizamos esto para protegernos del spoofing, DoS, o ataques basados en ICMP. El router sólo permite ser atravesado por los paquetes que tengan la dirección IP de origen de la Honeynet. Esto protege contra muchos ataques basados en el spoof, como el SYN Flooding o ataques SMURF. Además bloquearemos el tráfico ICMP saliente. Esto se lleva a cabo porque algunos de los cortafuegos tienen problemas con los estados de seguimiento de tráfico ICMP. Otras organizaciones no tienen que limitar ICMP de tal forma. Limitando el ICMP protegemos contra ataques como SMURF, reconocimiento de la topología de la red, o el Ping de la Muerte.

La combinación del cortafuegos y el router crea una técnica efectiva para controlar

el tráfico saliente, en la que se da a los blackhats la flexibilidad para ejecutar la mayor parte de lo que necesitan realizar, mientras que limita ataques que pueden lanzarse contra otros sistemas. En la captura de datos nos encontramos con una serie de capas que tienen funciones determinadas y que realizan un serie de tareas.

Estas capas son:

1. Primera capa de las actividades de captura: El cortafuegos. Antes, hemos tratado como podemos usar el cortafuegos para controlar los datos. Este mismo cortafuegos puede ser usado para capturar datos. El cortafuegos registra todas las conexiones iniciadas hacia y desde la Honeynet. Esta información es crítica, porque todas las conexiones son sospechosas. Hemos diseñado nuestro cortafuegos no sólo para que registre todas las conexiones, sino que además nos alerte cuando se intente realizar una conexión.

   Por ejemplo, si alguien intenta conectarse mediante telnet a un sistema en la Honeynet, el cortafuegos lo registrará y nos alertará del evento. Esto es muy útil para rastrear los escaneos.

   Otro uso es para las puertas traseras o los puertos propietarios. Muchos exploits (personas que sacan provecho) crean una shell o una puerta trasera en el sistema. Estas puertas traseras son fáciles de detectar cuando el cortafuegos alerta sobre una conexión a un sistema en algunos puertos altos y aleatorios. El cortafuegos también nos alerta cuando un honeypot de la Honeynet inicia una conexión saliente. El cortafuegos una vez más registra y nos alerta de esta actividad. Sin embargo, esta alerta es de mayor prioridad ya que indica que un sistema ha sido comprometido. Tal alarma sería enviada por correo electrónico, al móvil y/o al un busca, para que se pueda actuar de forma que se controle ó se actúe de una manera determinada.

2. Segunda capa: El sistema IDS.

   Tiene dos propósitos:

   El primero, y de lejos el más importante, es capturar toda la actividad en la red. Su principal tarea es capturar y registrar cada paquete y su carga que circula por nuestra red. Si volvemos a la figura de la arquitectura del honeynet podrá ver que el sensor IDS está en una zona físicamente compartida por todos los otros sistemas en la Honeynet. El sistema IDS reside en un 'puerto de monitorización', así que puede registrar toda la actividad de la red. Estos registros son utilizados para analizar las actividades del blackhat.

   La segunda función del sensor IDS es alertarnos de cualquier actividad sospechosa. Muchos IDS tienen una base de datos de marcas, así cuando un paquete en la red concuerda con una de las marcas, se genera una alerta. Esta función no es tan crítica para una Honeynet, ya que cualquier actividad es sospechosa por naturaleza. Sin embargo, los IDS pueden ofrecer información detallada sobre una conexión específica.

3. La tercera capa son los mismos sistemas, queremos capturar toda la actividad producida por parte del sistema y del usuario. El primer método para esto es tener todos los registros del sistema no sólo guardados localmente, sino que también en

un servidor de registros remoto.

De esta forma, información crítica del sistema como la actividad de los procesos, conexiones del sistema, y exploits intentados es copiada de forma segura a un sistema remoto. No queremos realizar ningún intento de ocultar el uso de un servidor 'syslog' remoto. Si el blackhat lo detecta, lo peor que puede hacer es deshabilitar 'syslogd' (lo que es un comportamiento estándar para muchos blackhats). Esto significa que no tendremos registros muy continuos, sin embargo tendremos al menos información de como consiguieron acceso y desde donde.

Blackhats más avanzados intentarán comprometer el servidor 'syslog' remoto intentando cubrir sus huellas. Esto es exactamente lo que queremos que ocurra. El servidor 'syslog' normalmente es un sistema bastante más seguro. Esto significa para el blackhat que para tomar el control de dicho sistema deberá utilizar técnicas más avanzadas, las cuales capturaremos y aprenderemos de ellos. Si es servidor 'syslog' resulta comprometido, no hemos perdido nada. Sí, el blackhat puede conseguir el control del sistema y limpiar los registros. Sin embargo, no lo olviden. nuestro IDS que está en la red ha capturado pasivamente y almacenado todos los registros de la actividad llevada a cabo en la red. En realidad, el sistema IDS actúa como un sistema secundario registro remoto, ya que de forma pasiva captura todos los datos de la red.

Este tipo de arquitectura es eficaz contra ataques automatizados, o contra atacantes de nivel básicos. Pero no son de gran utilidad contra ataques avanzados. El entorno proporcionado por las Honeynets de primera generación suele ser poco atractivo, consistiendo básicamente en instalaciones por defecto de sistemas operativos. Hay que destacar que este modelo apenas se implementa hoy en día, siendo más utilizado su sucesor, Honeynets de segunda generación que se comentan a continuación.

2.5.2. GEN II

Basada en la combinación de viejas y nuevas técnicas, la Honeynet GenII puede mejorar la flexibilidad, gestión, y seguridad de los despliegues de las honeynets. Esta arquitectura de Honeynets fue desarrollada en 2002, y fue pensada para solventar muchos de los problemas existentes en el modelo anterior. Con respecto a las tecnologías GenI, esta arquitectura es más fácil de implementar, más difícil de detectar y tiene un mantenimiento más seguro.

Como se puede ver en la figura, la primera diferencia con respecto a la arquitectura GEN I es que se utiliza un Honeynet Gateway, cuya traducción al español es Puerta de Enlace a la Red Trampa, que combina los elementos de IDS y cortafuegos que aparecían por separado en la primera generación.

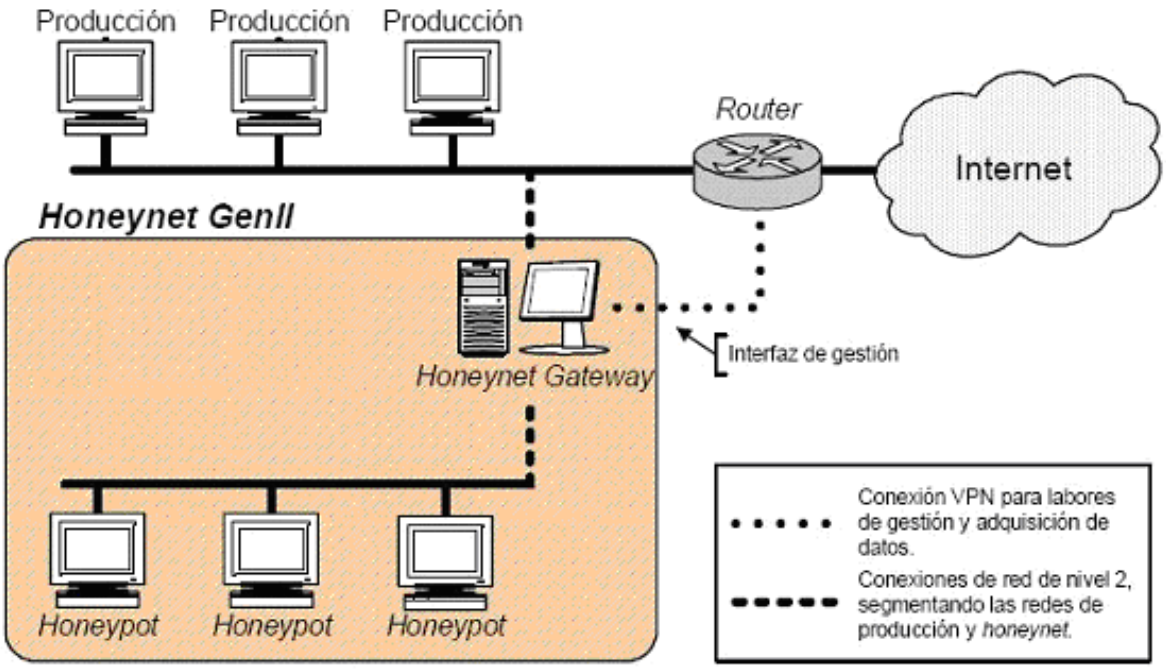

Figura 1.3: Esquema de HoneyNet de Segunda Generación

Esto se detallará más en el apartado dedicado a la arquitectura, ya que esta parte de Gen 2 ha sido divido en una serie de conceptos, a nuestro parecer los más importantes sobre la segunda generación.

Arquitectura

Una Honeynet no es un producto, no se instala simplemente un software desde un CDROM para ponerse luego en funcionamiento. Una Honeynet es una arquitectura, una red altamente controlada para contener y analizar atacantes en vivo.

El elemento clave de cualquier Honeynet es el gateway (puerta de enlace), que separa los sistemas Honeynet del resto del mundo. El gateway actúa como un muro, de hecho, podemos denominar al gateway como un Honeywall. Todo el tráfico entrante o saliente de la Honeynet debe pasar a través de este Honeywall. Este elemento es el centro de control de la Honeynet; toda la "magia.ocurre aquí.

En la figura, la puerta de enlace ó gateway es un bridge (puente) de nivel dos. Puede utilizarse un router de nivel tres, pero es preferible utilizar un puente, ya que es más difícil de detectar, debido a que no tiene dirección ip. En el diagrama, la Honeynet ha sido instalada en una red interna. En el pasado, la mayoría de las Honeynets se instalaban tradicionalmente en redes externas o perimetrales. Con el uso de un gateway de nivel dos, las Honeynets pueden integrarse en redes internas.

Esto nos permite registrar y aprender no sólo de las amenazas externas, sino también de las internas.

El Honeywall (el gateway puente) separa los sistemas de producción de la red Honeynet, alimentada por nuestros objetivos víctimas. La interfaz externa de nuestro gateway (eth0) está conectada a la red de sistemas de producción. El interfaz interno de nuestro gateway (eth1) está conectado a la red de sistemas Honeynet. Como es un puente, tanto el interfaz externo como el interno están en la misma red IP. También tenemos una tercera interfaz (eth2). El propósito de esta interfaz es para realizar tareas de administración remota del gateway, incluyendo el traslado de registros o datos capturados a algún punto centralizado. Las interfaces interna y externa están en modo puente, de forma que no tienen dirección IP asignada a las mismas. Sin embargo, la tercera interfaz (eth2) tiene una pila IP asignada. Esta es una red separada y segura, para labores de administración.

Las ventajas de esta arquitectura consisten en que el gateway es difícil de detectar, ya que no hay saltos de enrutamiento, no hay decrementos TTL (tiempo de vida), ni direcciones MAC asociadas al gateway. Además, simplificamos el desarrollo de la Honeynet combinando el Control de datos y la Captura de datos en un sólo gateway. El siguiente paso consiste en construir un gateway que soporte esta arquitectura.

Por desgracia, aunque muchas distribuciones soportan bridging, la mayoría no soportan IPTables en modo puente. IPTables es crítico, no sólo ayudan asegurando nuestro gateway, sino que pueden ser utilizados para el Control de datos. La configuración del gateway. Esto es realmente más sencillo de lo que parece. El Proyecto Honeynet ha desarrollado lo que denomina rc.firewall script. Este script (guión) implementa casi todos los elementos críticos de tu gateway. Iremos haciendo referencias a este guión a lo largo del artículo. Para nuestro gateway, el guión implementará las funciones de bridging, cortafuegos, configurará una interfaz de gestión, controlará quién podrá administrar el gateway, cómo y desde dónde, el registro de la actividad de red, e implementa el Control de datos. Como puedes ver, este script es importante, ya que configura tu gateway para que cumpla la mayoría de requisitos de la Honeynet. Para utilizar este script (y configurar tu Honeywall) tan sólo tienes que configurar las variables del mismo, y luego ejecutarlo. En vez de describir cada variable en detalle y cómo funciona (el script lo hace por ti).

Control de Datos

El propósito del Control de datos es el de evitar que los atacantes no utilicen la Honeynet para atacar o dañar otros sistemas que no pertenezcan a la Honeynet.

Con el Control de datos, una de las preguntas que debes hacer es ¿cuánta actividad saliente controlas? Cuanto más permitas hacer al atacante, más podrás aprender. Sin embargo, cuantas más cosas permitas hacer al atacante, más daño podrá potencialmente hacer. De modo que tienes que contener sus acciones lo suficiente como para evitar que perjudiquen a otros colegas, pero no puedes contenerlas demasiado o aprenderás poco. Cuánto permitas hacer al atacante depende en última instancia de los riesgos que estés dispuesto a asumir.

Para complicar más las cosas, tenemos que limitar las acciones de los atacantes sin que se percaten de ello. Para lograr esto, implementaremos dos tecnologías, conteo de conexiones y NIPS.

El conteo de conexiones permite limitar el número de conexiones salientes que un honeypot (sistema trampa) puede iniciar.

El NIPS (Sistema de Prevención de Intrusiones de Red) puede bloquear (o deshabilitar) ataques conocidos.

Combinadas, estas dos tecnologías constituyen un potente y flexible mecanismo de control de datos. Implementaremos ambas tecnologías en nuestro gateway de nivel dos. Realizaremos el Control de datos aquí porque es por donde pasa todo el tráfico entrante y saliente; es el punto donde se concentra la actividad de los atacantes.

El siguiente paso consiste en implementar la limitación de conexiones. Podemos controlar el número de conexiones que un atacante puede iniciar desde un honeypot. El propósito aquí consiste en contar las conexiones salientes y, cuando se ha alcanzado un límite concreto, bloquear cualquier conexión adicional. Esto se hace principalmente para deducir el riesgo de escaneos masivos, o ataques de denegación de servicio (DoS); actividades que requieren gran número de conexiones salientes. Para esto utilizamos IPTables, configurado e implementado por el script rc.firewall nombrado antes. En el script, especificamos cuántas veces puede un atacante iniciar una conexión saliente TCP, UDP, ICMP, u OTHER.

El número de conexiones permitidas depende del riesgo que estás dispuesto a asumir. Limitar el número de conexiones salientes evita que los atacantes utilicen la Honeynet para escanear gran número de sistemas, o lanzar ataques de DoS. Es difícil de hacer mucho daño cuando te limitan el número de conexiones salientes que puedes iniciar.

Hay que tener en cuenta que esto puede indicar a un atacante que se encuentra dentro de una Honeynet y así ser capaz de detectarla simplemente iniciando conexiones salientes, y observando si son bloqueadas después de cierto número. Un ejemplo de limitación de conexiones puede ser el siguiente en el que la variable OTHER significa cualquier protocolo IP que NO SEA TCP, UDP, o ICMP (como IPsec, túneles IPv6, Protocolo de Voz de Red, etc.).

```
SCALE="day
"
TCPRATE="
15"
UDPRATE="
20"
ICMPRATE=
"50"
OTHERRAT
E="15"
```

Así es como IPTables implementa el límite de conexiones. Cuando un atacante entra en un honeypot puede iniciar conexiones salientes por varias razones (descargar toolkits, configurar bots automáticos, abrir sesiones IRC, enviar correos electrónicos, etc.). Cada vez que se inicia una de estas conexiones salientes, es contada por el firewall. Cuando se alcanza el límite, IPTables bloquea cualquier conexión posterior desde ese honeypot. Entonces, IPTables se reinicia a sí mismo, permitiendo tantas conexiones salientes por intervalo de tiempo permitidas. Para el ejemplo de la página anterior era una duración temporal de un día, pasado este tiempo se reinician las conexiones salientes.

Para ver un ejemplo real de este comportamiento, véase estos registros de IPTables de un honeypot Win2000 infectado con el gusano Code Red II, y sus intentos de escaneo externos. Una característica importante de IPTables, cuando se alcanza el límite TCP, es que no afecta en absoluto al tráfico UDP, ICMP u OTRO tráfico, hasta que sus límites también se alcanzan.

El siguiente paso consiste en implementar la funcionalidad NIPS. Como ya sabemos el objetivo de los NIPS es el de identificar y bloquear ataques conocidos. Esto lo hace investigando cada paquete que viaja a través de nuestro gateway. Si algún paquete concuerda con alguna de las reglas del IDS, no sólo se genera una alarma (como un NIDS tradicional), sino que el paquete puede ser eliminado (bloqueando el ataque) o modificado (deshabilitando el ataque). La ventaja es que
reducimos dramáticamente el riesgo de que un ataque saliente tenga éxito.

La desventaja es que esto sólo funciona con ataques conocidos. En el caso del valor límite, estamos permitiendo por defecto unas 15 conexiones TCP salientes por día ¿Qué ocurre si nuestra Honeynet es infectada con un gusano, y esas 15 primeras conexiones son intentos de infectar a otros sistemas? Aunque el valor límite ha reducido el número de sistemas que puede infectar, sigues teniendo ese riesgo. La idea de un NIPS es la de que puede bloquear o deshabilitar cualquier ataque identificado en esas primeras 15 conexiones.

Existen programas para esto como snort_inline que controlan los paquetes, permite eliminar o modifcar paquetes, pero la mayoría de estos programas no saben después como reenviarlos o dirigirlos porque no implementan la ip_forwarding, luengo necesitan algo más que realice esta función de redirección de paquetes.

Captura de Datos

El propósito de la Captura de datos es registrar toda la actividad de los atacantes. Este es el mayor objetivo de la Honeynet, recoger información. Sin la Captura de datos, nuestra Honeynet no tiene valor. La clave de la Captura de datos es la recopilación de información a tantos niveles como sea posible. Ningún nivel de forma aislada nos lo dice todo. Por ejemplo, mucha gente cree que todo lo que necesitas son las pulsaciones de teclado de los atacantes, sin embargo esto no es cierto ¿Qué pasa cuando el atacante lanza una herramienta, cómo sabrás qué hace la herramienta si no capturas la actividad

de la misma, o el tráfico de red? El Proyecto Honeynet ha identificado tres capas o niveles críticos de la Captura de datos: registros del cortafuegos, tráfico de red, y actividad de sistema.

Los registros del cortafuegos son muy sencillos. Esta información es crítica, ya que es nuestra primera indicación de lo que está haciendo un atacante. También es nuestra primera alarma que nos dice cuándo se han iniciado los ataques salientes.

El segundo elemento consiste en capturar cada paquete y con su contenido completo (payload) que entra o sale de la Honeynet. Configuramos y ejecutamos un segundo proceso para capturar toda esta actividad. Este fichero de configuración captura todo el tráfico IP a un fichero de registro tcpdump para su futuro análisis. Mediante la monitorización del interfaz interno, sólo capturarás el tráfico entrante y saliente de la Honeynet, que es exactamente lo que quieres. Otra ventaja del script de inicio es que estandariza dónde se registra la actividad capturada, cosa de vital importancia si tienes múltiples Honeynets registrando a una localización central.

El tercer elemento es el que más desafíos supone; capturar toda la actividad del atacante ocurrida en el propio honeypot. Hace años esto era sencillo, ya que la mayoría de la interacción remota con sistemas se hacía sobre protocolos de texto en claro, como FTP, HTTP, o Telnet. Tan sólo había que monitorizar las conexiones para capturar las pulsaciones de teclado. Hoy se suelen utilizar canales SSH o 3DES para comunicarse con las máquinas comprometidas. Ya no podemos capturar las pulsaciones de teclado monitorizando el segmento de red, en vez de ello tenemos que obtenerlas directamente del sistema. Una ventaja de este sistema es que la mayoría de lo que se cifra se descifra en el sistema de destino, en nuestro caso el honeypot. Si podemos capturar los datos descifrados en el honeypot, podemos evitar las comunicaciones cifradas.

Alarmas

Que alguien entre en tu Honeynet puede ser una gran experiencia de aprendizaje, a menos que no sepas que alguien ha entrado en la misma. Asegurarte de que serás notificado de un ataque (y responder al mismo) es crítico para una buena Honeynet. Lo ideal sería contar con una monitorización periódica a cargo de administrador experimentado. No obstante, para las organizaciones que no cuentan con personal 24/7, una alternativa es la de las alertas automáticas.

Se necesita una herramienta de monitorización automática muy completa capaz de avisar a los administradores de posibles ataques que han tenido éxito en la Honeynet. Debe monitorizar ficheros de registro en busca de patrones indicados en un fichero de configuración. Cuando encuentra un patrón puede enviar alarmas de forma automática mediante correo electrónico, system bells, llamadas de teléfono, y puede ser ampliado para ejecutar otros comandos/programas.

Cuando un patrón coincide, se envía un correo electrónico al administrador del honeypot. En esta configuración, los correos de aviso son enviados a razón de no más de 10 por hora. Los patrones buscados y las acciones tomadas variarán según la implementación de cada honeynet.

El objetivo de las alertas automáticas es el de proporcionar a los administradores

tanta información como sea posible para responder ante un ataque con éxito.

Un ejemplo de este tipo es Swatch.

Incluso con las herramientas automáticas descritas en la sección de Control de datos, una Honeynet requiere supervisión constante. Si se configura adecuadamente este tipo de programas como Swatch pueden ser utilizados para notificar rápidamente a los administradores de los eventos de su red.

1.5.3. HoneyNet Virtual

La aparición de herramientas de emulación y soporte virtual han hecho posible este modelo de implementación de honeynets. Este enfoque consiste en crear una honeynet completa en un sólo equipo físico. Una Honeynet virtual no es una arquitectura, sino una forma de implementarlas; de esta manera, se puede utilizar para crear tanto arquitecturas tipo GenI como GenII.

Las ventajas son coste reducido y más fácil manejo, ya que todo está combinado en un único sistema. Sin embargo, esta simplicidad también nos cuesta. Primero, estás limitado a qué tipos de sistemas operativos puedes implantar debido al hardware y al programa virtual. Segundo, las Honeynets virtuales traen un riesgo, específicamente que un atacante puede salirse del programa virtual y tomar el sistema Honeynet, saltándose el mecanismos de Control de Datos y de Captura de Datos.

Una Honeynet Virtual coge la tecnología de una Honeynet y la combina en un único sistema. Esto las hace más baratas de construir, más fáciles de implantar, y más simples de mantener. Sin embargo, también tienen desventajas comunes, incluyendo un único punto de fallo y limitaciones con el hardware y el software virtual. Depende de ti decidir que solución es mejor para tu entorno.

Una Honeynet virtual puede ser Autocontenida o Híbrida, como se detallarán a continuación cada una de ellas.

HoneyNet Virtual Autocontenida

La honeynet virtual autocontenida engloba a una honeynet en un solo equipo. La red entera está virtualmente contenida en un único y físico sistema. Una red Honeynet típicamente consiste de un cortafuegos para Control de Datos y Captura de Datos, y los honeypots dentro de la Honeynet. Un esquema de esta clase de honeynet sería la que muestra la figura que aparece a continuación.

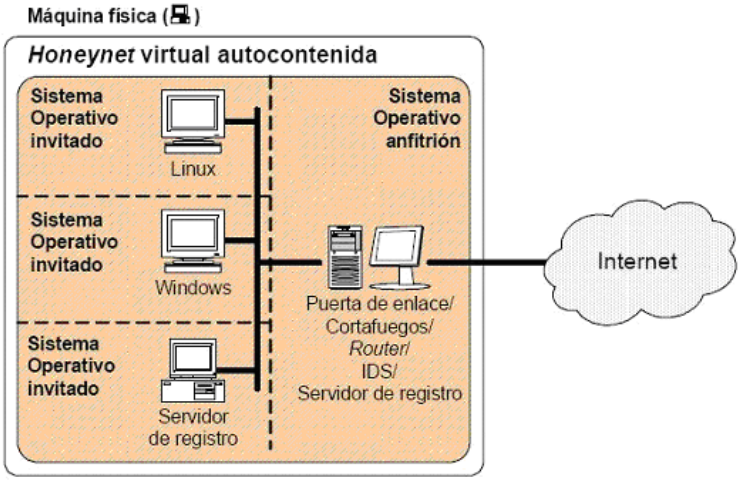

Figura 1.4: Esquema de HoneyNet Virtual Autocontenida

Las ventajas que presentan este tipo de honeynet virtuales son:

- Fácilmente transportable, especialmente si se instala en un portátil.
- Rápida puesta en funcionamiento. Una vez instalada, sólo hay que conectarla a la red y configurarla en pocos minutos.
- Es barata y ocupa poco espacio. Sólo nos hace falta un ordenador.

Las desventajas que presentan este tipo de honeynet virtuales son:

- Si falla el hardware, la Honeynet entera podría dejar de funcionar.
- Necesidad de un ordenador de altas prestaciones. Aunque sólo requiere un ordenador, tiene que tener suficiente memoria y capacidad de procesador.
- Seguridad. Como todos los sistemas comparte el mismo hardware, puede que un atacante acceda a otras partes del sistema. Tiene mucha dependencia del software virtual.
- Limitación por software. Como todo tiene que ejecutarse en una sola máquina, hay software que no se podrá utilizar por problemas de incompatibilidad. Por ejemplo un sistema operativo de cisco en una máquina con un procesador de Intel.

HoneyNet Virtual Híbrida

Una Honeynet Híbrida es una combinación de la clásica Honeynet y del software

virtual. Captura de Datos, como por ejemplo cortafuegos, y Control de Datos, es decir, los sensores de IDS y el almacenamiento de registros, están en un sistema separado y aislado, para reducir el riesgo de compromiso. Sin embargo, todos los honeypots son ejecutados en una única máquina.

Un esquema de esta clase de honeynet sería la que muestra la figura que aparece a continuación.

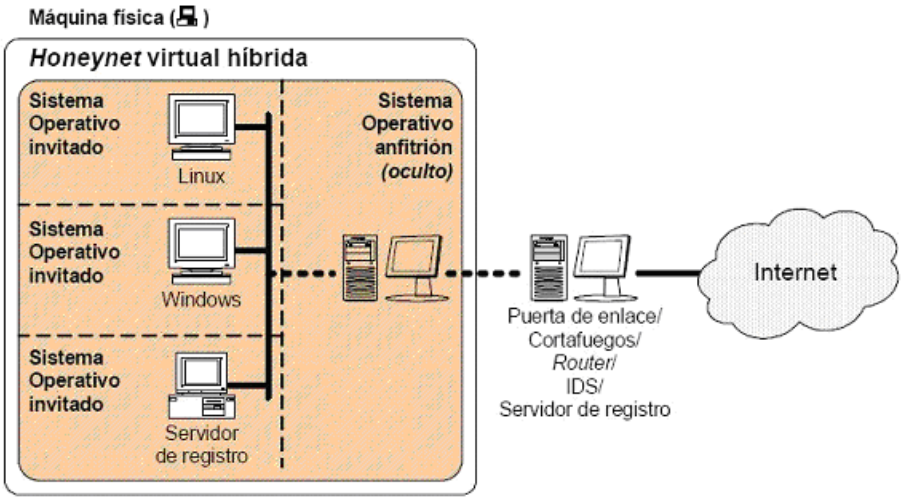

Figura 2.5: Esquema de HoneyNet Virtual Híbrida

Las ventajas que presentan este tipo de honeynet virtuales son:

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

Las desventajas que presentan este tipo de honeynet virtuales son:

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

Bibliografía

- Honeynet Project, 'Conoce a tu enemigo', http://his.sourceforge.net/honeynet/papers/enemy/
- Honeynet Project, 'Conoce a tu enemigo: Honeynets',

http://his.sourceforge.net/trad/honeynet/papers/honeynet/

- Honeynet Project, 'Conoce a tu enemigo: Redes trampa en universidades', http://his.sourceforge.net/honeynet/papers/edu/
- Honeynet Project, 'Conoce a tu enemigo: Definiendo honeynets virtuales', http://his.sourceforge.net/honeynet/papers/virtual/
- Rafael San Miguel Carrasco, 'SISTEMAS HONEYNESISTEMAS HONEYNET', http://www.fistconference.org/data/presentaciones/honeynets.pdf
- Eduardo Gallego, Jorge E. López de Vergara, 'Honeynets: Aprendiendo del atacante', http://jungla.dit.upm.es/ jlopez/publicaciones/mundointernet04.pdf
- Gabriel Verdejo Alvarez, 'SEGURIDAD EN REDES IP: Honeypots y Honeynets', http://tau.uab.es/ gaby/
- Fred Cohen and Associates, 'The Deception Toolkit Home Page and Mailing List', http://www.all.net/dtk/index.html
- Monkey.org Developments, 'Developments of the Honeyd Virtual Honeypot', http://www.honeyd.org/
- Honeynet Project, 'Sebek Homepage', http://www.honeynet.org/tools/sebek/
- Honeynet Project, 'Sebek Homepage', http://www.honeynet.org/tools/sebek/
- Jose Rosalén Trenco, 'Detección De Ataques Y AnáLisis Forense En Sistemas Honeypot', http://ircisco30.uv.es/documents/jorotren.pdf