HONEYPOTS, MONITORIZANDO A LOS ATACANTES

Se llama *honeypot* (en ingles, tarro de miel) a una herramienta usada en el ámbito de la seguridad informática para atraer y analizar el comportamiento de los atacantes en Internet. Parece una contradicción, puesto que la función habitual de las herramientas de seguridad es exactamente la contraria: mantener alejados a los atacantes o impedir sus ataques. Sin embargo, desde hace unos años, se utilizan los honeypots para atraer a atacantes hacia un entorno controlado, e intentar conocer más detalles sobre cómo estos realizan sus ataques, e incluso descubrir nuevas vulnerabilidades.

I Historia y orígenes

Lance Spitzner, consultor y analista informático experto en seguridad, construyó a comienzos del año 2000 una red de seis ordenadores en su propia casa. Esta red la diseñó para estudiar el comportamiento y formas de actuación de los atacantes. Fue de los primeros investigadores en adoptar la idea, y hoy es uno de los mayores expertos en honeypots, precursor del proyecto honeynet (www.honeynet.org), en marcha desde 1999, y autor del libro "Honeypots: Tracking Hackers".

Su sistema estuvo durante casi un año de prueba, desde abril del 2000 a febrero de 2001, guardando toda la información que se generaba. Los resultados hablaban por sí solos: en los momentos de mayor intensidad de los ataques, comprobaba que las vías de acceso más comunes a los equipos de su casa eran escaneadas, desde el exterior de su red, hasta 14 veces al día, utilizando herramientas de ataque automatizadas.

Desde entonces, se ha creado toda una comunidad de desarrolladores aglutinados alrededor de honeynet.org que ofrecen todo tipo de herramientas y consejos para utilizar estas herramientas.

II Clasificación

Un honeypot puede ser tan simple como un ordenador que ejecuta un programa, que analiza el tráfico que entra y sale de un ordenador hacia Internet, "*escuchando*" en cualquier número de puertos. El procedimiento consiste en mantener una debilidad o vulnerabilidad en un programa, en el sistema operativo, en el protocolo, o en cualquier otro elemento del equipo susceptible de ser atacado, que motive al atacante a usarlo, de manera que se muestre dispuesto a emplear todas sus habilidades para explotar dicha debilidad y obtener acceso al sistema.

Por otro lado, un honeypot puede ser tan complejo como una completa red de ordenadores completamente funcionales, funcionando bajo distintos sistemas operativos y ofreciendo gran cantidad de servicios. Cuando algún sistema que está incluido en dicha red sea atacado de alguna forma, se advierte al administrador.

Otra opción muy utilizada es crear honeypots completamente virtuales: programas específicamente diseñados para simular una red, engañar al atacante con direcciones falsas, IP fingidas y ordenadores inexistentes, con el único fin de confundirlo o alimentar el ataque para analizar nuevos métodos. Si algo tienen en común los honeypots es que no guardan ninguna información relevante, y si lo parece, si se muestran contraseñas o datos de usuario, son completamente ficticios.

Ilustración 1: Página principal del proyecto Honeynet

Fuente: INTECO

Los honeypots son clasificados según diferentes categorías:

III Honeypots de alta interacción

Suelen ser usados por las compañías en sus redes internas. Estos honeypots están construidos con máquinas reales, o consisten en una sola máquina real con un sistema operativo "normal", como el que podría utilizar cualquier usuario. Se colocan en la red interna en producción. Si están bien configurados, cualquier intento de acceso a ellos debe suponer una alerta a tener en cuenta. Puesto que no tienen ninguna utilidad más que la de ser atacados, el hecho de que de alguna forma se intente acceder a ese recurso significa por definición que algo no va bien.

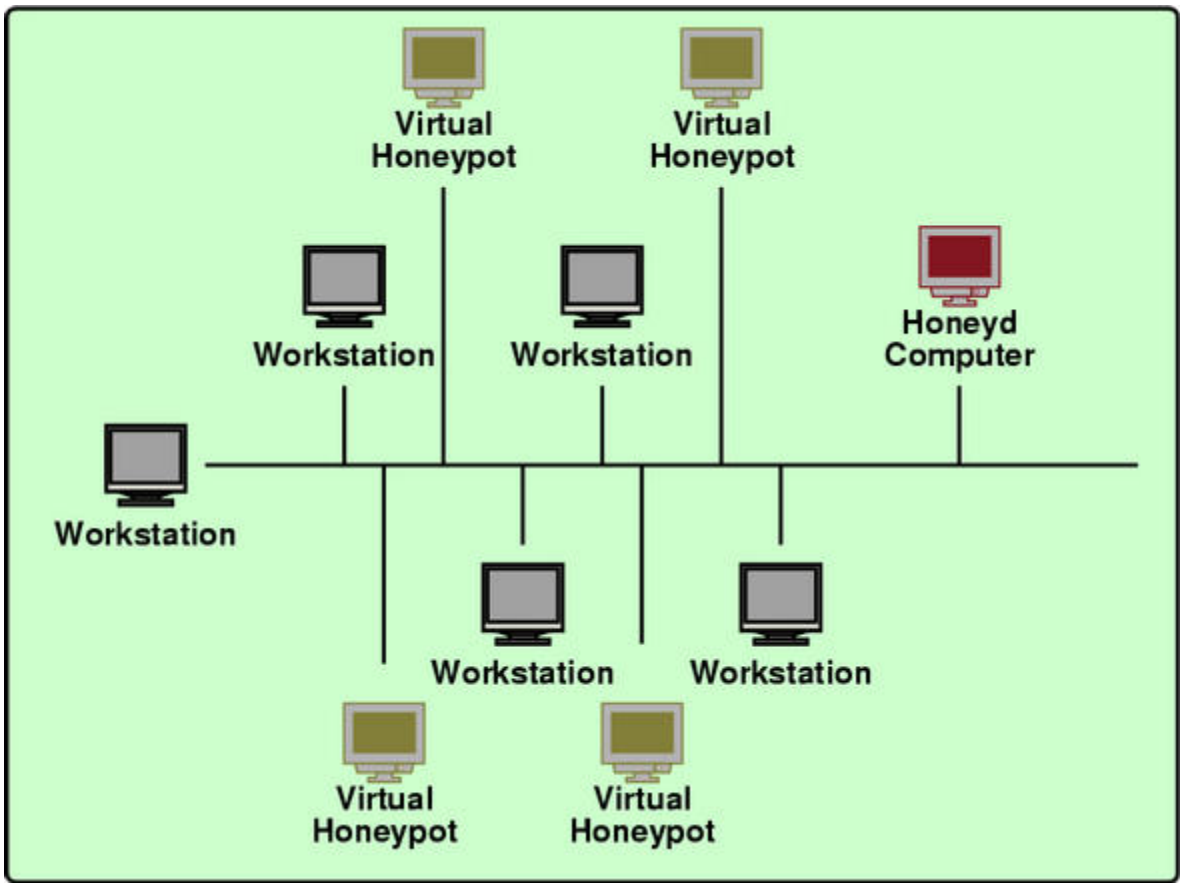
Cada interacción con ese honeypot se considera sospechosa por definición. Todo este tráfico debe ser convenientemente monitorizado y almacenado en una zona segura de la red, y a la que un potencial atacante no tenga acceso. Esto es así porque, si se tratase de un ataque real, el intruso podría a su vez borrar todo el tráfico generado por él mismo, las señales que ha ido dejando, con lo que el ataque pasaría desapercibido y el honeypot no tendría utilidad real.

Las ventajas que ofrecen los honeypots de alta interacción es que pueden prevenir ataques de todo tipo. Tanto los conocidos como los desconocidos. Al tratarse de un sistema real, contiene todos los fallos de software conocidos y desconocidos que pueda albergar cualquier otro sistema. Si un atacante intenta aprovechar un fallo desconocido hasta el momento (llamados en el argot "*0 day*"), será la propia interacción con la máquina, para intentar explotar el fallo, lo que alerte del problema y ayude a descubrir ese nuevo fallo. En contraposición, por ejemplo un detector de intrusos (IDS) basado en firmas, podría alertar en la red de solo intentos de aprovechar fallos o ataques ya conocidos, para los que tiene firmas que le permiten reconocerlos. La ventaja del honeypot es que, sea el ataque nuevo o no, el intento de ataque alertará al administrador, y esto le permitirá estar alerta cuanto antes del potencial peligro.

En este sentido, los honeypots se usan para mitigar los riesgos de las compañías, en el sentido tradicional de uso de las conocidas herramientas defensivas. Lo que la diferencia de los tradicionales cortafuegos o detectores de intrusos es su naturaleza "activa" en vez de pasiva. De modo figurado un honeypot se muestra como un anzuelo, no como un muro de contención para evitar ataques, muy al contrario, busca dichos ataques y se encarga de "entretenerlos". Muchas compañías lo usan como un valor añadido más a sus elementos de seguridad, como complemento a sus herramientas típicas. Se obtiene así una fácil detección y reconocimiento de los ataques, de forma que pueden elaborar con esos datos estadísticas que ayudan a configurar de manera más efectiva sus herramientas pasivas. Conociendo cuanto antes los problemas de seguridad a los que más se atacan o los nuevos objetivos, más eficazmente podrá defenderse una compañía concreta contra ellos.

Como toda herramienta destinada a mejorar la seguridad, los honeypots tienen sus ventajas e inconvenientes. Su mayor utilidad radica en su simpleza. Al ser un mecanismo cuyo único fin consiste en que intenten aprovechar sus debilidades, no realiza ningún servicio real, y el tráfico que transita a través de él va a ser muy pequeño. Si se detecta tráfico que va o viene hacia el sistema, casi con toda probabilidad va a ser una prueba, escaneo o ataque. El tráfico registrado en un sistema de este tipo es sospechoso por naturaleza, por lo que su gestión y estudio se simplifica en gran medida. Aunque, por supuesto, ocurran "falsos positivos", expresión que, en este caso, invierte su significado. Si un falso positivo se produce normalmente cuando una actividad sospechosa tomada como ataque no resulta serlo, en el ambiente de los honeypots, el falso positivo sería el tráfico gestionado por la máquina que no representa una amenaza. En esta simpleza de uso de tráfico y recursos, radica su mayor ventaja. En resumen, poca información, pero muy valiosa.

Ilustración 2: Esquema de situación de un honeypot en una red interna



Fuente: http://www.honeyd.org/

Entre los problemas que se pueden producir por el trabajo con honeypots, destaca la posibilidad de que se vuelva en contra del administrador. Si no se diseña de una manera absolutamente estudiada, si no se ata cada cabo, si no se aísla convenientemente, el atacante puede acabar comprometiendo un sistema real y llegar a datos valiosos conectados al honeypot.

IV Honeypots de baja interacción

Suelen ser creados y gestionados por organizaciones dedicadas a la investigación del fraude en Internet, o cualquier tipo de organización que necesite investigar sobre las nuevas amenazas en la red. Son mucho más complejos de administrar y mantener, y la información que reciben debe ser lo más extensa posible, ésta debe ser organizada y analizada para que sea de utilidad.

Se suelen tratar de sistemas específicos que emulan servicios, redes, pilas TCP o cualquier otro aspecto de un sistema real, pero sin serlo. Existe un “meta-sistema” detrás, invisible para el atacante, que está simulando ser cualquier cosa para la que esté programado ser. No tienen que implementar un comportamiento completo de un sistema

o servicio. Normalmente simulan ser un servicio, y ofrecen respuesta a un subconjunto de respuestas simple. Por ejemplo, un honeypot que simule ser un servidor de correo, puede simular aceptar conexiones y permitir que se escriba en ellas un correo, aunque nunca llegará a enviarlo realmente.

Normalmente este tipo de honeypots no está destinado a "atrapar" atacantes reales, sino herramientas automatizadas. Un ser humano podría detectar rápidamente si se trata de un servidor real o no, bien por su experiencia o por otras características que le hagan sospechar que no se encuentra en un entorno real. Sin embargo, sistemas automatizados como programas de explotación automática, gusanos, virus, etc., programados específicamente para realizar una acción sobre un servicio, no detectarán nada extraño. Harán su trabajo intentando explotar alguna vulnerabilidad, el honeypot simulará ser explotado, y el administrador del honeypot obtendrá la información que desea.

Este tipo de honeypots, tienen el problema de que en ellos resulta más complejo descubrir nuevos tipos de ataques. Están preparados para simular ciertos servicios que se saben atacados, y a responder de cierta manera para que el ataque crea que ha conseguido su objetivo. Pero en ningún caso puede comportarse de formas para las que no está programado, por ejemplo para simular la explotación de nuevos tipos de amenazas.

Se utilizan entre muchas otras posibilidades, para generar estadísticas de explotación, detectar patrones de ataque y detectar nuevo tipo de malware. Este último punto resulta especialmente interesante. En la mayoría de las ocasiones, el malware aprovecha vulnerabilidades para descargar archivos (virus) desde un servidor. Para intentar evadir a los antivirus y pasar lo más desapercibido posible, este archivo descargado es muy variable, y pueden aparecer nuevas versiones cada pocas horas. Un honeypot puede simular el aprovechamiento de esa vulnerabilidad y permite que se descargue ese nuevo archivo. De esta forma un honeypot puede resultar un excelente recolector de nuevas versiones de virus y malware en general de forma sistemática y automatizada.

Al igual que los de alta interacción, estos sistemas deben estar muy bien protegidos para que no se vuelvan en contra del administrador del honeypot. Un atacante, ya sea de forma automática o manual,) podría llegar de alguna forma al "meta-sistema" que aloja el servicio simulado, y atacarlo.

Tabla 1: Comparación de principales características entre los honeypots

Alta interacción	Baja interacción
Servicios reales, sistemas operativos o aplicaciones	Emulan servicios, vulnerabilidades, etc.
El riesgo que corren es mayor	El riesgo que corren es menor
Capturan menos información, pero más valiosa	Capturan mucha información. Dependen de su sistema de clasificación y análisis para evaluarlo.

Fuente: Virtual Honeypots

V Honeymonkeys

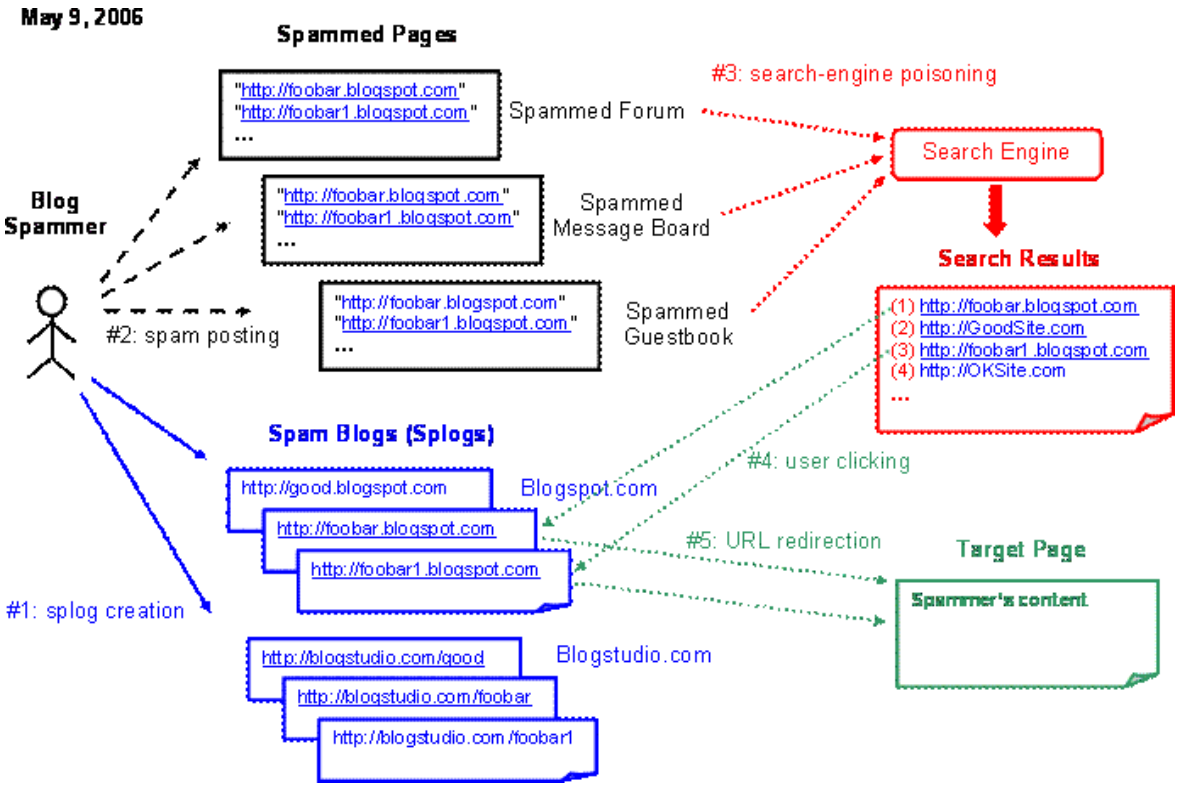
Un honeypot puede ser diseñado como un servidor en vez de cómo un equipo es decir, como un sistema (honeypot que hace de servidor) que espera ser contactado con un cliente (equipo). Desde el momento en el que uno de los objetivos de un honeypot es recavar información sobre ataques, surgió un concepto de un honeypot "cliente" que no espere a recibir ataques sino que los genere activamente. Se les llama honeymonkeys.

Desde hace varios años, el vector de ataque más utilizado en Internet es el navegador. Las medidas de seguridad han aumentado y cada vez resulta más difícil aprovechar vulnerabilidades en clientes –programas- de correo electrónico, que venía siendo el vector de ataque más usado. El uso popular de cortafuegos también hizo que cada vez fuese más complicado para atacantes aprovechar vulnerabilidades en el propio sistema operativo. Por tanto, con el traslado a la web de los servicios (foros, chats, etc.), el navegador se convirtió en el objetivo favorito de los atacantes. Con solo visitar una web, se intenta aprovechar todo tipo de vulnerabilidades en el navegador para ejecutar código en el cliente e infectarlo.

Tras este concepto surgen los honeymonkeys. Su función principal, al igual que la de los honeypots, es igualmente detectar nuevos tipos de ataques y fórmulas de infección e igual que los honeypots, están formados por un módulo de "exploración" y un módulo de recogida de datos. Sin embargo en el caso de los honeymonkeys, la exploración se hace activamente a través de navegadores. El honeymonkey funciona como un sistema automático de navegación que visita toda clase de páginas web con el fin de que alguna de ellas intente aprovechar vulnerabilidades en el navegador. Poseen una naturaleza mucho más activa que el honeypot, en el sentido en el que "patrullan" la red como si fueran un usuario visitando enlaces compulsivamente.

Fue Microsoft quien los bautizó. "Monkey" (mono, en inglés) hace alusión a los saltos y el dinamismo del tipo de acción que realizan. Con este método, al igual que los honeypots, se pueden encontrar nuevos exploits, gusanos, etc., siempre que se analice y procese convenientemente toda la información recogida.

Ilustración 3: Esquema del HoneyMonkey de Microsoft



Fuente: research.microsoft.com

VI Honeypots y honeynets

El propósito de las honeynet es, al igual que el honeypot, investigar el uso de las técnicas y herramientas que hacen los atacantes en Internet. Se diferencia básicamente de un honeypot en que no supone una sola máquina, sino múltiples sistemas y aplicaciones que emulan otras tantas, imitan vulnerabilidades o servicios conocidos o crean entornos "jaula" donde es posible una mejor observación y análisis de los ataques. Los requerimientos básicos e imprescindibles para construir una honeynet son dos, los llamados: Data Control (control de datos) y Data Capture (captura de datos).

Data Control

Suponen la contención controlada de la información y las conexiones. Lidiar con atacantes siempre supone un riesgo que hay que reducir al máximo, por lo que es preciso asegurarse que, una vez comprometido el honeypot, no se comprometerán sistemas legítimos. El reto consiste en mantener un absoluto control del flujo de datos sin que el atacante lo note. No se puede cerrar un sistema por completo para evitar el tráfico innecesario. Una vez comprometido el sistema, el atacante intentará realizar distintos tipos de conexiones para continuar su ataque, probablemente necesite bajar programas por FTP, correo o conexiones SSH. Si no se le permite esta flexibilidad de acciones, además de levantar sus sospechas, no se podrán estudiar otros pasos más importantes

que valdría la pena analizar. En los primeros intentos de los investigadores de poner en marcha proyectos de honeynet, no se permitieron ningún tipo de conexiones salientes para evitar ser plataforma de nuevos ataques. Pero sólo les llevaba a los atacantes unos minutos ver que algo andaba mal, y abandonar el intento de ataque. Los resultados así eran muy pobres. De esto se deduce una disyuntiva en la que reside el arte de construir una honeynet útil: es necesario encontrar el equilibrio entre la libertad de movimientos para el atacante, que supone un mayor riesgo, y la seguridad real del sistema, que puede derivar en resultados menos interesantes para el estudio.

Data Capture

Es el rastreo y almacenamiento de la información que se persigue, esto es, los logs (registros de datos) de sus actos, y que serán analizados a posteriori. Se debe capturar tanta información como sea posible aislada del tráfico legal, evitando la posibilidad de que el atacante sepa que se le están recogiendo sus acciones. Lo más importante para conseguir esto es evitar el almacenamiento de resultados localmente en el propio honeypot, puesto que pueden ser potencialmente detectados y borrados con la lógica intención de no dejar huellas del ataque. La información debe ser almacenada remotamente y en capas. No se puede limitar al registro de una simple capa de información, sino tomarla de la mayor variedad posible de recursos. Combinando todos los equipos y las capas de datos se formará el cuadro de información deseado.

Herramientas virtuales

Las herramientas para construir un honeypot o una honeynet son muy variadas, pero el método más frecuente es el uso de máquinas físicas o virtuales para construir el honeypot.

Dado el potencial peligro del uso de honeypots, y a su propia naturaleza, el uso de herramientas virtuales resulta muy conveniente y es ampliamente aceptado. Las ventajas de un sistema virtual sobre uno físico son evidentes:

- **Permiten ser restauradas en cuestión de minutos en caso de accidente**, desastre o compromiso: la mayoría de sistemas virtuales permiten almacenar un estado “ideal” y volver a él en cualquier momento de manera mucho más rápida que si hubiese que restaurar un sistema físico y devolverlo a un estado anterior.

- **Permiten ser portadas a diferentes máquinas físicas que la alojan**: los sistemas virtuales, por definición, se ejecutan por igual en cualquier máquina física, que emulan el entorno necesario a través de un programa para poder reproducir el sistema virtual.

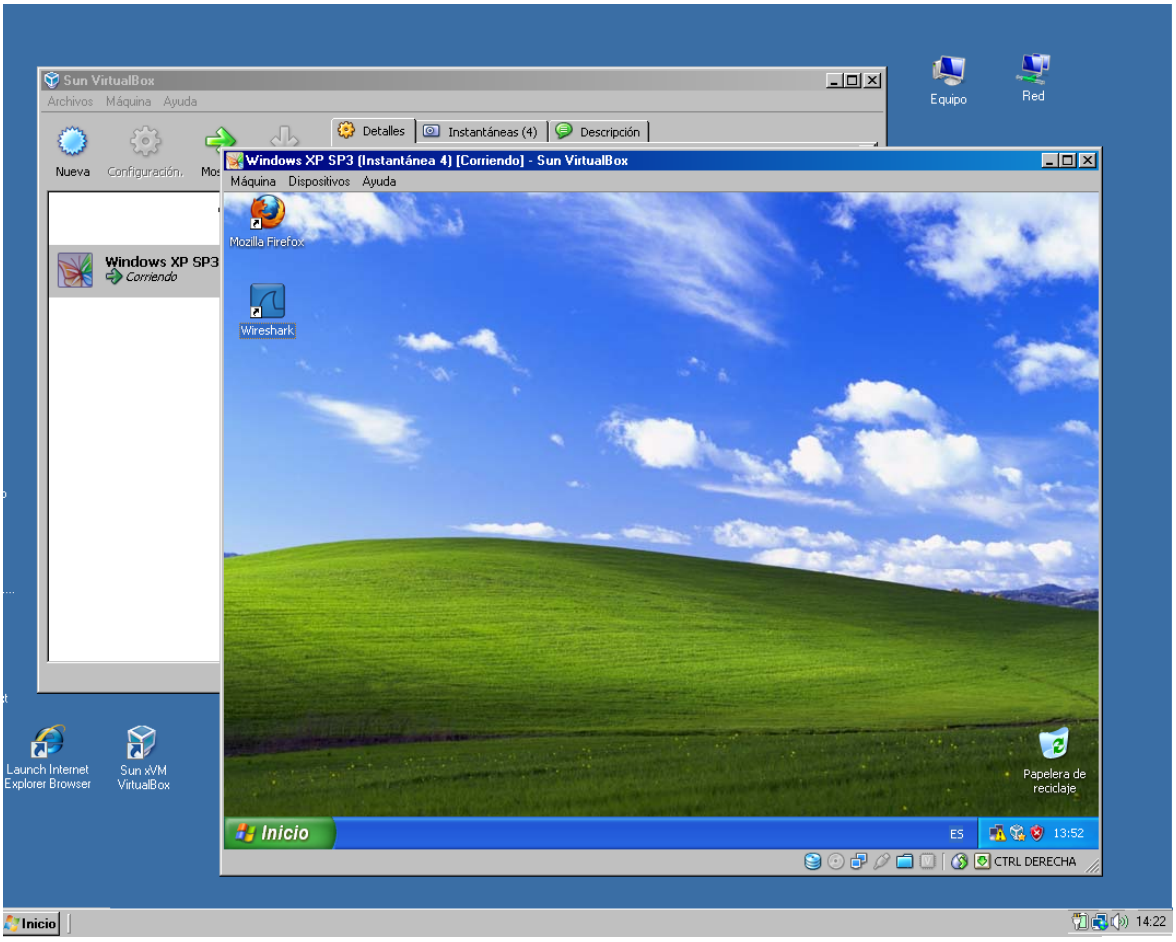
- **Permiten ahorrar costes**: una misma máquina física puede alojar un número indeterminado de máquinas virtuales, tantas como le permitan sus recursos, y con tantos sistemas operativos como se desee.

Las principales herramientas virtuales usadas en honeypots son:

- **VMware**: se trata del sistema de virtualización más usado y famoso. Puede simular máquinas que ejecutan cualquier sistema operativo y a su vez hacerlo sobre cualquier sistema operativo. Muchas de las utilidades de virtualización que proporciona se ofrecen de forma gratuita, tales como VMWare Player.

- **VirtualBox**: es un proyecto de Sun, gratuito y de código abierto. Al igual que VMWare, puede simular máquinas que ejecutan cualquier sistema operativo y a su vez hacerlo sobre cualquier sistema operativo.

Ilustración 4: Máquina virtual XP corriendo en Windows Vista con VirtualBox



Fuente: INTECO

- **Qemu**: proyecto de código abierto que puede usarse tanto como virtualizador como emulador. Disponible solo para entornos Linux. Resulta más complejo de usar que VMWare.

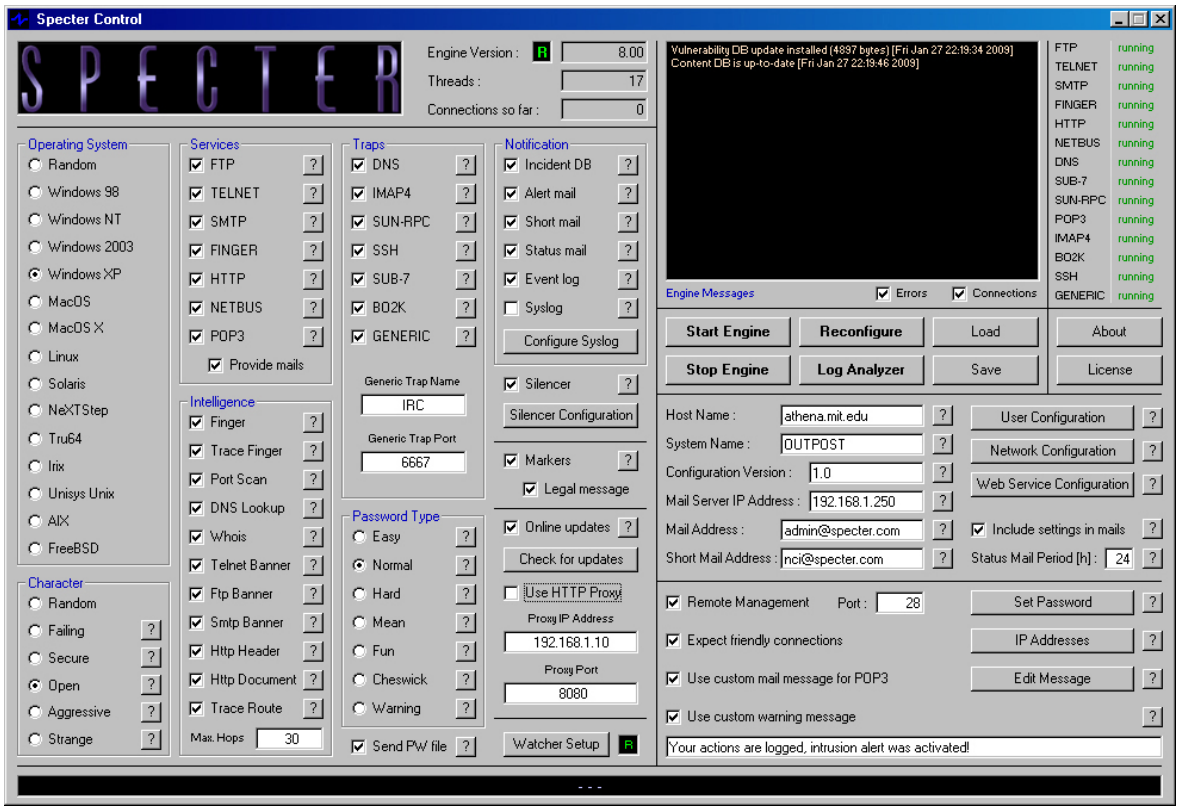
- **User-Mode Linux**: es una forma de simular un núcleo de Linux "virtual" como si se tratase de un proceso. Solo puede ser ejecutado desde un sistema Linux y sólo puede simular otro kernel, pero resulta muy útil para la puesta en marcha de honeypots.

VII Ejemplos de Honeypots

Existen pocas herramientas comerciales que cubran este mercado de honeypots, sin embargo, en el mundo del código libre, se ofrecen muchas utilidades que pueden servir como honeypots, tanto a empresas como a particulares.

Uno de los Honeypots comerciales más conocidos es **Specter**.

Ilustración 5: Consola de Specter, honeypot comercial más famoso para Windows



Fuente: specter.com

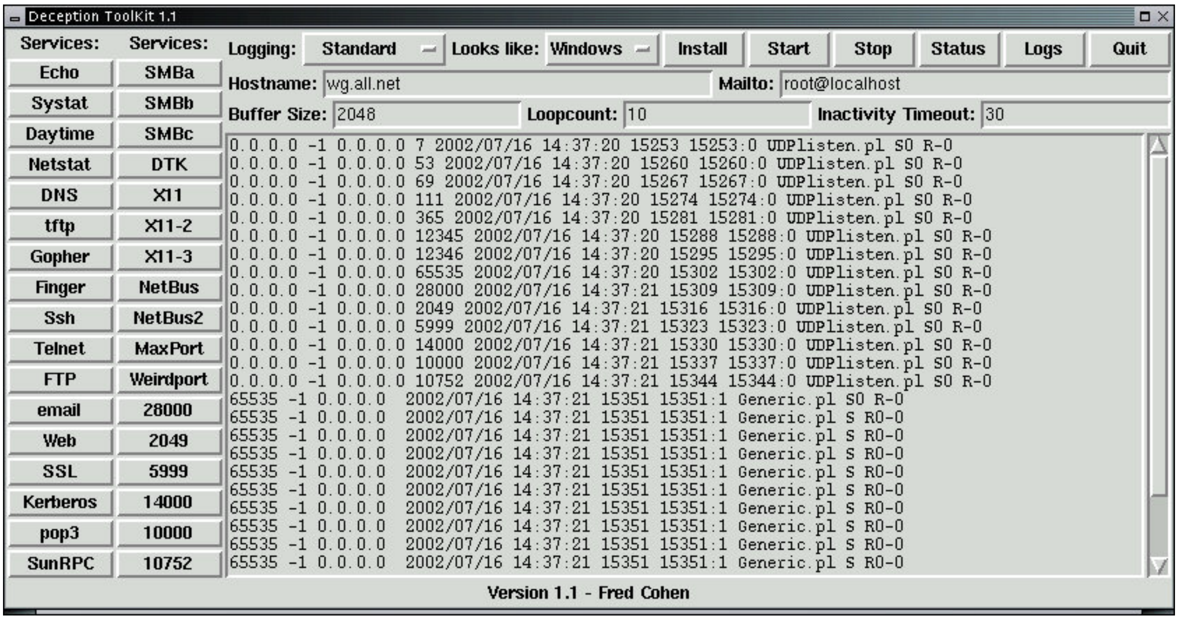
Es capaz de simular hasta 14 sistemas operativos diferentes, y funciona bajo Windows. Su principal atractivo es su facilidad de uso.

KFSensor también es un honeypot comercial que actúa como honeypot e IDS para sistemas Windows.

En el mundo del código abierto, se pueden encontrar muchos ejemplos de Honeypots que cubren todos los aspectos de estas herramientas:

Bubblegum Proxypot, Jackpot, BackOfficer Friendly, Bigeye. HoneyWeb, Deception Toolkit, LaBrea Tarpit, Honeyd, Sendmail SPAM Trap, etc.

Ilustración 6: Consola de Deception Toolkit



Fuente: Deception Toolkit