Mecanismos para la detección de ataques e intrusiones

Sistemas de detección

Índice

Introducción

Las redes de ordenadores se encuentran expuestas a ataques informáticos con tanta frecuencia que es necesario imponer una gran cantidad de requisitos de seguridad para la protección de sus recursos.

Aunque las deficiencias de estos sistemas se pueden comprobar mediante herramientas convencionales, no siempre son corregidas. En general, estas debilidades pueden provocar un agujero en la seguridad de la red y facilitar entradas ilegales en el sistema.

La mayoría de las organizaciones disponen actualmente de mecanismos de prevención y de mecanismos de protección de los datos integrados en sus redes. Sin embargo, aunque estos mecanismos se deben considerar imprescindibles, hay que estudiar cómo continuar aumentando la seguridad asumida por la organización.

Así, un nivel de seguridad únicamente perimetral (basado tan solo en la integración en la red de sistemas cortafuegos y otros mecanismos de prevención) no debería ser suficiente. Debemos pensar que no todos los accesos a la red pasan por el cortafuegos, y que no todas las amenazas son originadas en la zona externa del cortafuegos. Por otra parte, los sistemas cortafuegos, como el resto de elementos de la red, pueden ser objeto de ataques e intrusiones.

Una buena forma de mejorar la seguridad de la red pasa por la instalación de mecanismos de detección, capaces de avisar al administrador de la red en el momento en que se produzcan estos ataques a la seguridad de la red.

Una analogía que ayuda a entender la necesidad de incorporar estos elementos podría ser la comparación entre la seguridad de una red informática y la seguridad de un edificio: las puertas de entrada ejercen un primer nivel de control de acceso, pero normalmente no nos quedamos aquí; instalaremos detectores de movimiento o cámaras de vigilancia en puntos claves del edificio para detectar la existencia de personas no autorizadas, o que hacen un mal uso de los recursos, poniendo en peligro la seguridad. Además, existirán vigilantes de seguridad, libros de registro en los que se apuntará a todo el personal que accede a un determinado departamento que consideramos crítico, etc. Toda esta información se procesa desde una oficina de control de seguridad donde se supervisa el registro de las cámaras y se llevan los libros de registro.

Todos estos elementos, proyectados en el mundo digital, configuran lo que se conoce en el ámbito de la seguridad de redes informáticas como mecanismos de detección.

Objetivos

En este módulo didáctico se fijan los siguientes objetivos:

1) Entender la necesidad de utilizar mecanismos adicionales para garantizar la seguridad de una red ya protegida con mecanismos de seguridad tradicionales.

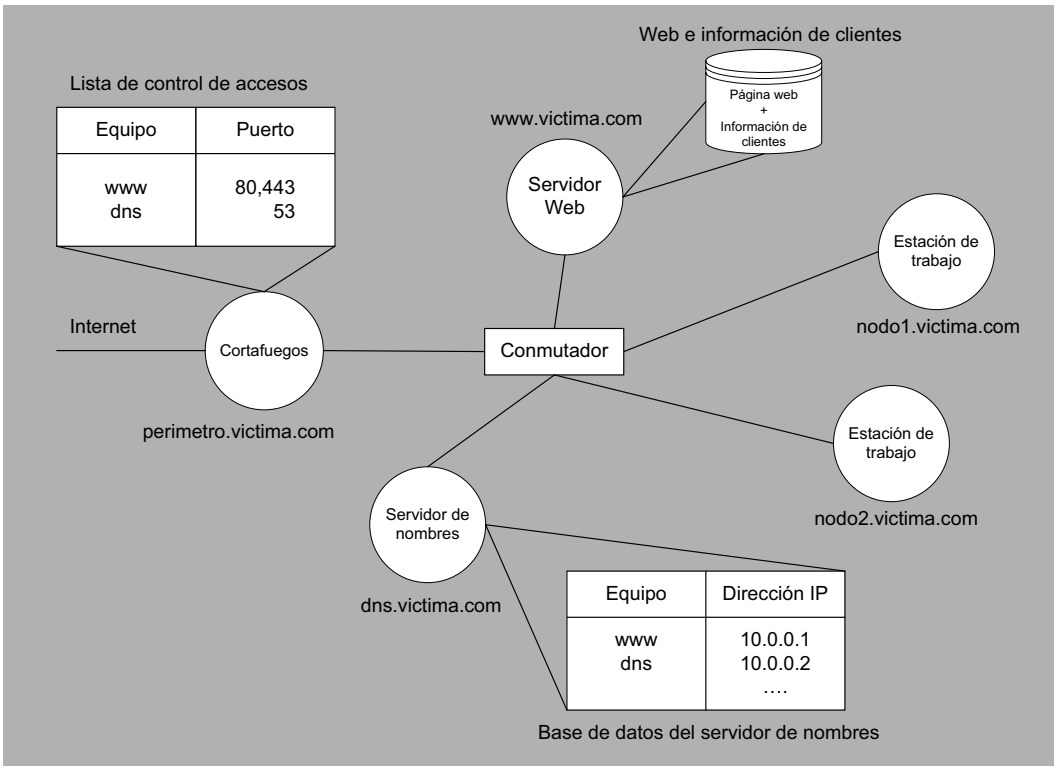
2) Comprender el origen de los primeros sistemas de detección y ver la arquitectura general de estos sistemas.

3) Ver otras tecnologías complementarias a los sistemas de detección tradicionales.

5.1. Necesidad de mecanismos adicionales en la prevención y protección

El escenario que presentaremos a continuación describe las posibles acciones de un atacante e ilustra la necesidad de una política de seguridad adicional que soporte y aumente las estrategias de seguridad presentadas hasta este momento.

Supongamos que un atacante está preparando introducirse en la red de una pequeña empresa para obtener los datos de sus clientes:



La empresa se dedica a la venta de artículos por internet y por ello tiene en marcha la web `www.victima.com`, que le permite la venta en línea de sus artículos.

Preocupados por la seguridad de su red (cuyo diagrama se muestra en la figura anterior) y, en concreto, por la seguridad de los datos de sus clientes, la empresa protege la red con un sistema cortafuegos, que permite únicamente la entrada de peticiones HTTP, HTTPS y consultas de DNS, aceptando también la transmisión de las respuestas a las peticiones HTTP, HTTPS y DNS.

El protocolo HTTPS se utiliza como un mecanismo de protección de los datos de los clientes a la hora de realizar transferencias seguras al servidor de HTTP, utilizando técnicas criptográficas para proteger la información sensible que el usuario transmite al servidor (número de targeta de crédito, datos personales, ...).

La intrusión que el atacante intentará llevar a cabo pasará por las siguientes cuatro fases:

- **Fase de vigilancia.** Durante la fase de vigilancia, el atacante intentará aprender todo lo que pueda sobre la red que quiere atacar. En especial, tratará de descubrir servicios vulnerables y errores de configuración.

- **Fase de explotación de servicio.** Este segundo paso describe la actividad que permitirá al atacante hacerse con privilegios de administrador (escala de privilegios) abusando de alguna de las deficiencias encontradas durante la etapa anterior.

- **Fase de ocultación de huellas.** Durante esta fase de ocultación se realizará toda aquella actividad ejecutada por el atacante (una vez ya producida la intrusión) para pasar desapercibido en el sistema.

  Dentro de esta tercera etapa se contemplan actividades tales como la eliminación de entradas sospechosas en ficheros de registro, la instalación y modificación de comandos de administración para ocultar la entrada en los sistemas de la red, o la actualización de los servicios vulnerables que ha utilizado para la intrusión (para evitar que terceras partes se introduzcan de nuevo en el sistema), etc.

- **Fase de extracción de información.** En esta última fase, el atacante con privilegios de administrador tendrá acceso a los datos de los clientes mediante la base de datos de clientes.

El intruso comenzará su ataque obteniendo el rango de direcciones IP donde se encuentra alojado el servidor de la web `www.victima.com`. Para ello, será suficiente realizar una serie de consultas al servidor de DNS de la compañía.

A continuación, realizará una exploración de puertos en cada una de las direcciones IP encontradas en el paso anterior. El objetivo de esta exploración de puertos es la búsqueda de servicios en ejecución en cada una de las máquinas del sistema, mediante alguna de las técnicas vistas en los módulos anteriores.

Gracias a los mecanismos de prevención instalados en la red de nuestro ejemplo (el sistema cortafuegos y las listas de control mostradas en la figura), la mayor parte de las conexiones serán eliminadas. De esta forma, el atacante sólo descubrirá dos de las máquinas de la red (el servidor de DNS y el servidor web).

El atacante decide atacar el servidor de HTTP. Para ello, tratará de descubrir qué tipo de servidor está funcionando en este equipo (le interesa el nombre y la versión del servidor en cuestión), ya que es muy probable que existan deficiencias de programación en la aplicación que está ofreciendo dicho servicio.

Por otra parte, el atacante también intentará descubrir el sistema operativo y la arquitectura *hardware* en la que se ejecuta el servidor. Esta información será importante a la hora de buscar los *exploits* que finalmente utilizará para realizar el ataque de intrusión.

Para obtener toda esta información, el atacante tiene suficiente con las entradas de DNS que la propia compañía le está ofreciendo (a través de los campos HINFO de las peticiones).

De esta forma, el atacante descubre que el servidor web está funcionando bajo una arquitectura concreta y que en este servidor hay instalado un determinado sistema operativo.

Otra fuente de información para descubrir el sistema operativo y la aplicación que ofrece el servicio web, y contrastar así la información ya obtenida, podrían ser las cabeceras de las respuestas HTTP que el servidor envía a cada petición de HTTP o HTTPS).

El atacante, que colecciona un amplio repertorio de aplicaciones para abusar de este producto, acabará obteniendo un acceso con privilegios de administrador. Supongamos, por ejemplo, que dicha intrusión la realiza gracias a la existencia de un *buffer* mal utilizado que existente en la aplicación en cuestión*.

* Ved la sección correspondiente a deficiencias de programación del primer módulo didáctico de este material para más información.

La primera observación que podemos indicar de todo el proceso que acabamos de describir es que los mecanismos de prevención de la red permiten la realización de este abuso contra el servidor de HTTP, ya que la forma de realizar el desbordamiento de *buffer* se realizará mediante peticiones HTTP legítimas (aceptadas en las listas de control del sistema cortafuegos).

Así pues, sin necesidad de violar ninguna de las políticas de control de acceso de la red, el atacante puede acabar haciéndose con el control de uno de los recursos conectados a la red de la compañía.

Una vez comprometido el servidor de HTTP, el intruso entrará en la fase de ocultación y comenzará a eliminar rápidamente todas aquellas marcas que pudieran delatar su entrada en el sistema. Además, se encargará de instalar en el equipo atacado un conjunto de *rootkits**. Una *rootkit* es una recopilación de herramientas de sistema, la mayoría de ellas fraudulentas, que se encargarán de dejar puertas abiertas en el sistema atacado, para garantizar así futuras conexiones con la misma escalada de privilegios, así como ofrecer la posibilidad de realizar nuevos ataques al sistema o a otros equipos de la red (denegaciones de servicio, escuchas en la red, ataques contra contraseñas del sistema, etc).

Las *rootkits*...

... son un conjunto de herramientas para garantizar, entre otras, la fase de ocultación de huellas durante el ataque de intrusión en un sistema.

Las *rootkits* suelen contener versiones modificadas de las herramientas básicas de administración, con la finalidad de esconder las acciones ilegítimas de un atacante y hacer pasar inadvertida la intrusión. Además, tratarán de garantizar futuras entradas en el equipo sin que el administrador del sistema las detecte.

Una vez finalizada la fase de ocultación de huellas, el atacante dispone de un equipo dentro de la red que le podrá servir de trampolín para realizar nuevos ataques e intrusiones en el resto de equipos de la compañía. Además, operando desde una máquina interna de la red, el atacante ya no estará sujeto a las restricciones impuestas por los sistemas de prevención.

Finalmente, una vez llegados a este punto el atacante dispondrá sin ningún problema de los datos que los clientes tienen almacenados en la base de datos.

Este ejemplo nos muestra cómo la existencia de un sistema cortafuegos (u otros mecanismos de prevención) y la utilización de comunicaciones cifradas (como un mecanismo de protección de datos) no es suficiente a la hora de defender nuestros sistemas de red.

5.2. Sistemas de detección de intrusos

La detección de ataques e intrusiones parte de la idea que un atacante es capaz de violar nuestra política de seguridad, atacando parcial o totalmente los recursos de una red, con el objetivo final de obtener un acceso con privilegios de administrador.

Los mecanismos para la detección de ataques e intrusiones tratan de encontrar y reportar la actividad maliciosa en la red, pudiendo llegar a reaccionar adecuadamente ante un ataque.

En la mayoría de los casos es deseable poder identificar el ataque exacto que se está produciendo, de forma que sea posible detener el ataque y recuperarse del mismo. En otras situaciones, sólo será posible detectar e informar de la actividad sospechosa que se ha encontrado, ante la imposibilidad de conocer lo que ha sucedido realmente.

Generalmente, la detección de ataques trabajará con la premisa de que nos encontramos en la peor de las situaciones, suponiendo que el atacante ha obtenido un acceso al sistema y que es capaz de utilizar o modificar sus recursos.

Los elementos más destacables dentro de la categoría de mecanismos para la detección de ataques e intrusiones son los sistemas de detección de intrusos*.

* En inglés, *Intrusion Detection System* (IDS).

A continuación introduciremos dos definiciones básicas en el campo de la detección de intrusos con el objetivo de clarificar términos comunes que se utilizarán más adelante.

Una **intrusión** es una secuencia de acciones realizadas por un usuario o proceso deshonesto, con el objetivo final de provocar un acceso no autorizado sobre un equipo o un sistema al completo.

La intrusión consistirá en la secuencia de pasos realizados por el atacante que viola una determinada política de seguridad. La existencia de una política de seguridad, en la que se contemplan una serie de acciones deshonestas que hay que prevenir, es un requisito clave para la intrusión. Es decir, la violación sólo se podrá detectar cuando las acciones observadas puedan ser comparadas con el conjunto de reglas definidas en la política de seguridad.

La **detección de intrusiones*** es el proceso de identificación y respuesta ante las actividades ilícitas observadas contra uno o varios recursos de una red.

* En inglés, *Intrusion Detection* (ID).

Esta última definición introduce la noción de proceso de detección de intrusos, que involucra toda una serie de tecnologías, usuarios y herramientas necesarias para llegar a buen término.

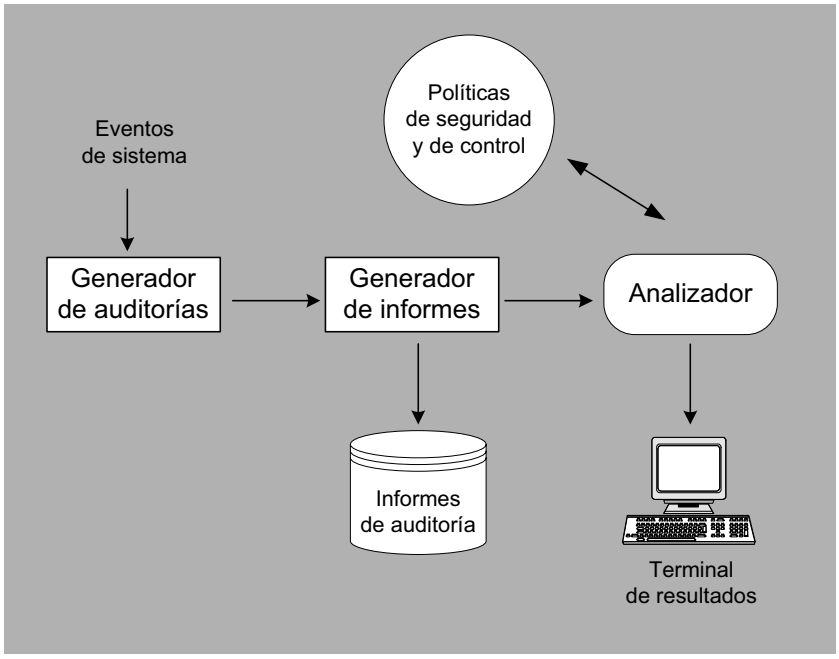
5.2.1. Antecedentes de los sistemas de detección de intrusos

Los sistemas de detección de intrusos son una evolución directa de los primeros sistemas de auditorías. Estos sistemas tenían como finalidad medir el tiempo que dedicaban los operadores a usar los sistemas. Con esta finalidad, se monitorizaban con una precisión de milésimas de segundo y servían, entre otras cosas, para poder facturar el servidor.

* En inglés, *Electronic Data Processsing* (EDP).

Los primeros sistemas aparecieron en la década de los cincuenta, cuando la empresa norteamericana *Bell Telephone System* creó un grupo de desarrollo con el objetivo de analizar el uso de los ordenadores en empresas de telefonía. Este equipo estableció la necesidad de utilizar auditorías mediante el procesamiento electrónico de los datos*, rompiendo con el anterior sistema basado en la realización de informes en papel. Este hecho provocó que a finales de los años 50 la *Bell Telephone System* se embarcara en el primer sistema a gran escala de facturación telefónica controlada por ordenadores.

La siguiente figura muestra un sencillo esquema del funcionamiento de un sistema de auditorías, en el cual los eventos de sistema son capturados por un generador de auditorías que llevará los datos hacia el elemento encargado de almacenarlos en un fichero de informe.



A partir de los años 70, el Departamento de Defensa de los EEUU empezó a invertir numerosos recursos en la investigación de políticas de seguridad, directrices y pautas de control. Estos esfuerzos culminaron con una iniciativa de seguridad en 1977 en la que se definía el concepto de *sistemas de confianza*.

Los ***sistemas de confianza*** son aquellos sistemas que emplean suficientes recursos *software* y *hardware* para permitir el procesamiento simultáneo de una variedad de información confidencial o clasificada. En estos sistemas se incluían distintos tipos de información repartida en niveles, que correspondían a su grado de confidencialidad.

Trusted Computer System Avaluation Criteria

Son una serie de documentos de la agencia nacional de seguridad (NSA) sobre sistemas de confianza, conocida también bajo el nombre de *Rainbow series* debido a los colores de sus portadas. El libro principal de esta serie es conocido como el *Libro naranja* (*Orange book*). Mirar la página web `www.fas.org/irp-/nsa/rainbow.htm` para más información.

A finales de la década de los setenta se incluyó en el *Trusted Computer System Avaluation Criteria* (TSCSEC) un apartado sobre los mecanismos de las auditorías como requisito para cualquier sistema de confianza con un nivel de seguridad elevado. En este documento, conocido bajo el nombre de *Libro marrón* (*Tan book*), se enumeran los objetivos principales de un mecanismo de auditoría que podemos resumir muy brevemente en los siguientes puntos:

- Permitir la revisión de patrones de acceso (por parte de un objeto o por parte de un usuario) y el uso de mecanismos de protección del sistema.
- Permitir el descubrimiento tanto de intentos internos como externos de burlar los mecanismos de protección.
- Permitir el descubrimiento de la transición de usuario cuando pasa de un nivel menor de privilegios a otro mayor (escalada de privilegios).
- Permitir el bloqueo de los intentos de los usuarios de saltarse los mecanismos de protección del sistema.
- Servir de garantía frente a los usuarios de que toda la información que se recoja sobre ataques e intrusiones será suficiente para controlar los posibles daños ocasionados en el sistema.

Primeros sistemas para la detección de ataques en tiempo real

El proyecto *Instrusion Detection Expert System* (IDES), desarrollado entre 1984 y 1986 por Dorothy Denning y Peter Neumann fue uno de los primeros sistemas de detección de intrusos en tiempo real. Este proyecto, financiado entre otros por la marina norteamericana, proponía una correspondencia entre actividad anómala y abuso o uso indebido (entendiendo por anómala aquella actividad extraña o inusual en un contexto estadístico).

IDES utilizaba perfiles para describir los sujetos del sistema (principalmente usuarios), y reglas de actividad para definir las acciones que tenían lugar (eventos de sistema o ciclos de CPU). Estos elementos permitían establecer mediante métodos estadísticos las pautas de comportamiento necesarias para detectar posibles anomalías.

Un segundo sistema de detección de ataques en tiempo real que hay que destacar fue *Discovery*, capaz de detectar e impedir problemas de seguridad en bases de datos. La novedad del sistema radicaba en la monitorización de aplicaciones en lugar de analizar un sistema operativo al completo. Mediante la utilización de métodos estadísticos desarrollados en COBOL, *Discovery* podía detectar posibles abusos.

Otros sistemas fueron desarrollados para ayudar a oficiales norteamericanos a encontrar marcas de ataques internos en los ordenadores principales de sus bases aéreas. Estos ordenadores eran principalmente servidores corporativos que trabajaban con información no clasificada pero muy confidencial.

Uno de los últimos sistemas de esta época a destacar fue MIDAS (*Multics Intrusion Detection and Alerting System*), creado por la NCSC (*National Computer Security Center*). Este sistema de detección fue implementado para monitorizar el *Dockmaster* de la NCSC, en el que se ejecutaba uno de los sistemas operativos más seguros de la época*. De la misma manera que IDES, MIDAS utilizaba un sistema híbrido en el que se combinaba tanto la estadística de anomalías como las reglas de seguridad de un sistema experto. MIDAS utilizaba un proceso de análisis progresivo compuesto por cuatro niveles de reglas. Además de estas reglas, también contaba con una base de datos que utilizaban para determinar signos de comportamiento atípico.

* Se trata del sistema operativo Multics, precursor de los sistemas Unix actuales.

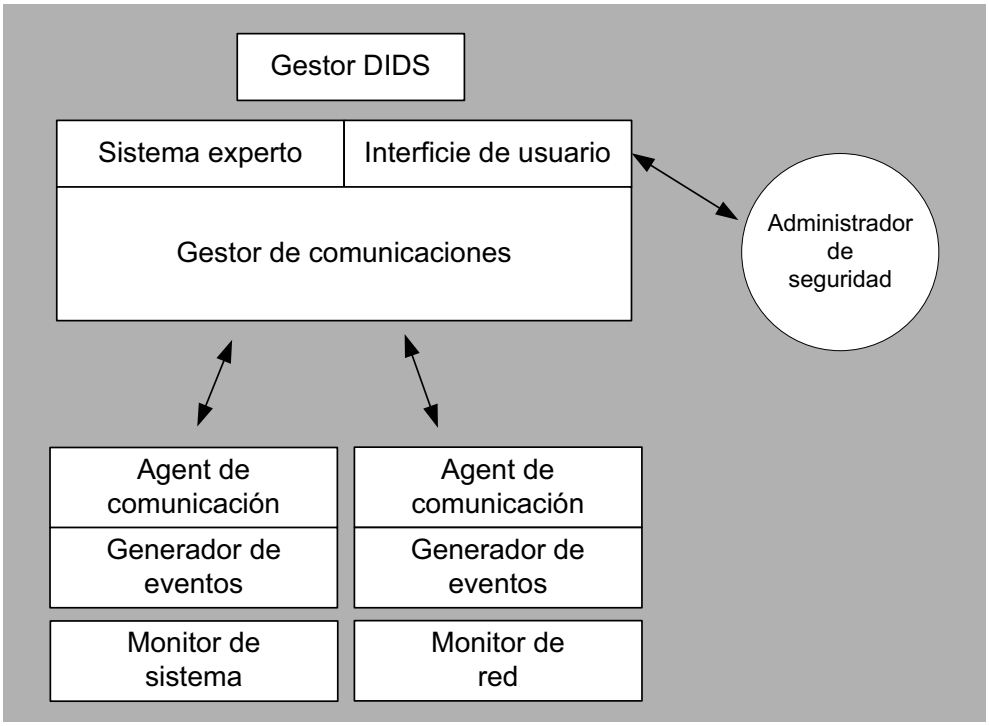
MIDAS fue uno de los primeros sistemas de detección de intrusiones conectados a internet. Fue publicado en la red en 1989 y monitorizó el mainframe *Dockmaster* en 1990, contribuyendo a fortalecer los mecanismos de autentificación de usuarios.

Sistemas de detección de intrusos actuales

A partir de los años 90, el rápido crecimiento de las redes de ordenadores provocó la aparición de nuevos modelos de detección de intrusiones. Por otro lado, los daños provocados por el famoso gusano de Robert Morris en el 1988 contribuyeron a aunar esfuerzos entre actividades comerciales y académicas en la búsqueda de soluciones de seguridad en este campo.

El primer paso fue la fusión de los sistemas de detección basados en la monitorización del sistema operativo (y aplicaciones del sistema operativo) junto con sistemas distribuidos de detección de redes, capaces de monitorizar en grupo ataques e intrusiones a través de redes conectadas a internet.

El objetivo inicial de este sistema distribuido era proporcionar medios que permitieran centralizar el control y la publicación de resultados en un analizador central. La siguiente figura muestra un diagrama de dicho sistema:



Por esta misma época comenzaron a aparecer los primeros programas de detección de intrusos de uso comercial. Algunas empresas los desarrollaban para ocupar una posición destacada en el ámbito de la seguridad, aunque otras lo hacían para mejorar los niveles de seguridad exigidos por la NCSC.

Actualmente, existe un gran número de sistemas de detección de intrusos disponibles para proteger redes informáticas. Aunque muchos de estos sistemas son comerciales o reservados para entornos militares y de investigación, existe hoy en día un gran número de soluciones libres que se pueden utilizar sin ningún tipo de restricción.

5.2.2. Arquitectura general de un sistema de detección de intrusiones

Como acabamos de ver, desde el comienzo de la década de los ochenta se han llevado a cabo multitud de estudios referentes a la construcción de sistemas para la detección de intrusos. En todos estos estudios se han realizado diferentes propuestas y diseños con el objetivo de cumplir los siguientes requisitos:

- *Precisión.* Un sistema de detección de intrusos no debe que confundir acciones legítimas con acciones deshonestas a la hora de realizar su detección.

  Cuando las acciones legítimas son detectadas como acciones maliciosas, el sistema de detección puede acabar provocando una denegación de servicio contra un usuario o un sistema legítimo. Este tipo de detecciones se conoce como *falsos positivos*. Cuanto menor sea el número de falsos positivos, mayor precisión tendrá el sistema de detección de intrusos.

- *Eficiencia.* El detector de intrusos debe minimizar la tasa de actividad maliciosa no detectada (conocida como *falsos negativos*). Cuanto menor sea la tasa de falsos negativos, mayor será la eficiencia del sistema de detección de intrusos. Éste es un requisito complicado, ya que en ocasiones puede llegar a ser imposible obtener todo el conocimiento necesario sobre ataques pasados, actuales y futuros.

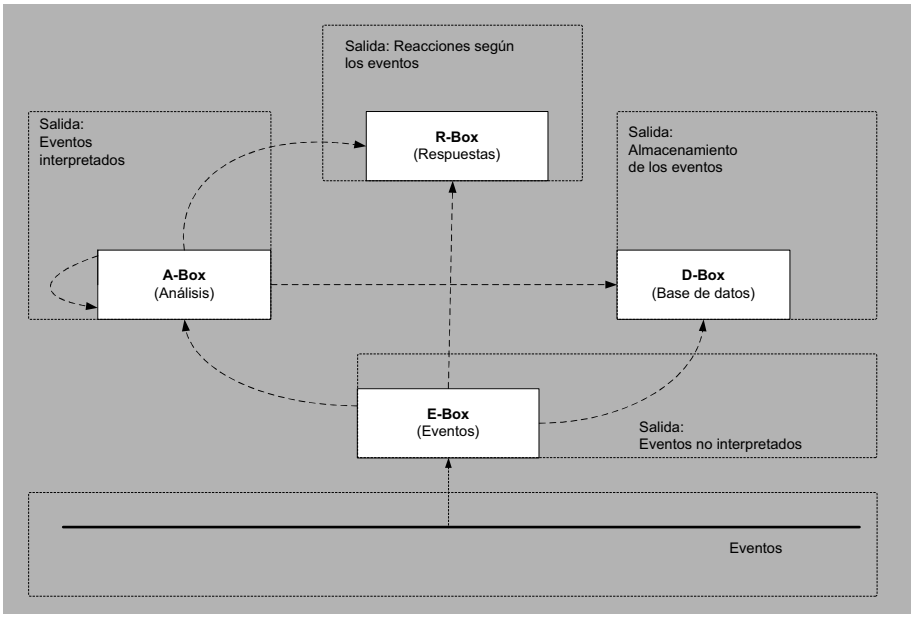
- *Rendimiento.* El rendimiento ofrecido por un sistema de detección de intrusos debe ser suficiente como para poder llegar a realizar una detección en tiempo real. La detección en tiempo real responde a la detección de la intrusión antes de que ésta llegue a provocar daños en el sistema. Según los expertos, este tiempo debería de ser inferior a un minuto.

- *Escalabilidad.* A medida que la red vaya creciendo (tanto en medida como en velocidad), también aumentará el número de eventos que deberá tratar el sistema. El detector tiene que ser capaz de soportar este aumento en el número de eventos, sin que se produzca pérdida de información. Este requisito es de gran relevancia en sistemas de detección de ataques distribuidos, donde los eventos son lanzados en diferentes equipos del sistema y deben ser puestos en correspondencia por el sistema de detección de intrusiones.

- *Tolerancia en fallos.* El sistema de detección de intrusiones debe ser capaz de continuar ofreciendo su servicio aunque sean atacados distintos elementos del sistema incluyendo la situación de que el propio sistema reciba un ataque o intrusión).

Con el objetivo de normalizar la situación, algunos miembros del IETF* presentaron a mediados de 1998 una arquitectura de propósito general para la construcción de sistemas de detección de intrusos, conocida como CIDF**. El esquema propuesto se corresponde con el diagrama mostrado en la siguiente figura:

- Internet Engineering Task Force.*
*-**Common Intrusion Detection Framework.*



Dos años más tarde se creó un nuevo grupo de trabajo en el IETF con la intención de estandarizar y mejorar la propuesta del CIDF. Este grupo de trabajo, conocido como IDWG*, replantea nuevamente los requisitos necesarios para la construcción de un marco de desarrollo genérico y se marca los siguientes objetivos:

**Intrusion Detection Working Group.*

- Definir la interacción entre el sistema de detección de intrusos frente a otros elementos de seguridad de la red, como pueden ser los sistemas de prevención (cortafuegos, listas de control de accesos, ...).

  Su primera propuesta se conoce con el nombre de *Tunnel Profile*. Se trata de la implementación de un mecanismo para la cooperación entre los distintos elementos de seguridad mediante el intercambio de mensajes. Este mecanismo garantiza una correcta comunicación entre los diferentes elementos, proporcionando privacidad, autenticidad e integridad de la información intercambiada (alertas, eventos, ...).

- Especificar el contenido de los mensajes intercambiados (eventos, alertas, ...) entre los distintos elementos del sistema. Por esto, proponen el formato *IDMEF*** y el protocolo de intercambio de mensajes *IDXP****.

-** Intrusion Detection Message Exchange Format
-****Intrusion Detection Exchange Protocol*

Observando las propuestas tanto del CIDF como las del IDWG podemos ver que los elementos necesarios para la construcción de un sistema para la detección de intrusos se pueden agrupar en las siguientes cuatro categorías que a continuación pasaremos a comentar con más detalle:

1) Recolectores de información.

2) Procesadores de eventos.

3) Unidades de respuesta.

4) Elementos de almacenamiento.

5.2.3. Recolectores de información

Un recolector de información, también conocido como **sensor**, es el responsable de la recogida de información de los equipos monitorizados por el sistema de detección.

La información recogida será transformada como una secuencia de tuplas de información (eventos) y será analizada posteriormente por los procesadores de información.

La información almacenada en estos eventos será la base de decisión para la detección del IDS. Por lo tanto, será importante garantizar su integridad frente a posibles ataques de modificación, a la hora de transmitir estos eventos entre el sensor que los generó y el componente de procesado que los tratará.

Existen diferentes formas de clasificar las posibles implementaciones de este componente. Detallamos a continuación tres de las propuestas más utilizadas.

El primer tipo, conocido como **sensores basados en equipo***, se encarga de analizar y recoger información de eventos a nivel de sistema operativo (como por ejemplo, intentos de conexión y llamadas al sistema).

En el segundo tipo encontramos sensores que recogen información de eventos sucedidos a nivel de tráfico de red (por ejemplo, analizando las cabeceras IP de todos los datagramas que pasan por la interfaz de red). Este tipo de componentes se conoce como **sensores basados en red****.

El tercer tipo, conocido como sensores **basados en aplicación***** , recibe la información de aplicaciones que se están ejecutando, y podrían ser considerados como un caso especial de los sensores basados en equipo.

-* En inglés, *host based sensors*.
-** En inglés, *network based sensors*.
-*** En inglés, *application based sensors*.

Elección de sensores

Durante los últimos años se ha debatido bastante cúal de los tres tipos de sensores ofrece mejores prestaciones. Actualmente, la mayoría de los sistemas de detección tratan de unificar las tres opciones, ofreciendo una solución de sensores híbrida.

- **Sensores basados en equipo y en aplicación.** Los sensores basados en equipo y en aplicación podrán recoger información de calidad, además de ser fácilmente configurables y de poder ofrecer información de gran precisión.

Además, estos datos pueden llegar a tener una gran densidad de información como, por ejemplo, la información reportada por los servidores de ficheros de registro del sistema. También pueden llegar a incluir gran cantidad de información de preprocesado, que facilitará el trabajo de los componentes de análisis de la información.

Por contra, estos sensores pueden repercutir notablemente en la eficiencia del sistema en el que se ejecuten.

- **Sensores basados en red.** La principal ventaja de los sensores basados en red, frente a las otras dos soluciones, es la posibilidad de trabajar de forma no intrusiva. Por lo tanto, la recogida de información no afecta a la forma de trabajar de los equipos o a la propia infraestructura. Al no residir forzosamente en los equipos que hay que analizar, son más resistentes a sufrir ataques.

  Por otra parte, la mayoría de los sensores basados en red son independientes del sistema operativo y pueden obtener información a nivel de red (como, por ejemplo, la existencia de fragmentación en datagramas IP) que no podría ser proporcionada por sensores basados en equipo.

  Algunos sensores basados en red son en realidad conmutadores con capacidad de análisis transparente frente al resto del sistema.

  Como desventaja principal de los sensores basados en red cabe destacar la escasa escalabilidad que esta solución ofrece. En el caso de redes con carga de tráfico muy elevada, es muy probable que estos sensores puedan perder paquetes, lo que supone una degradación en su capacidad de recogida de información.

  Estos sensores tendrán dificultades a la hora de trabajar en redes de alta velocidad como, por ejemplo, redes Gigabit Ethernet. Otro problema es el uso de comunicaciones cifradas, que hará que la información que se debe recoger sea incomprensible por el sensor, reduciendo de esta forma sus capacidades de detección.

Instalación de sensores

No es para nada trivial determinar el lugar exacto en el que se deben colocar estos componentes (desde dónde recoger la información). Los más sencillos de colocar son los sensores basados en aplicación, generalmente instalados en aquellas partes del programa donde se ofrecen servicios de depuración y generación de ficheros de registro. Pero la situación es mucho más difícil para las otras dos variantes.

Cuando consideramos la instalación de sensores basados en equipo, la gran variedad de sistemas operativos existentes y las distintas facilidades ofrecidas por cada uno de ellos, supone un serio problema. Además, no suele ser simple determinar qué parte de la información generada por el núcleo de un sistema operativo debería ser relevante a la hora de analizar.

En el caso de sistemas Unix, existe la propuesta del *Libro naranja* (ya comentado en este mismo módulo), en el que se muestran veintitrés puntos de interés donde debería analizarse información.

En el caso de sensores basados en red, la utilización de redes segmentadas mediante conmutadores de red supone un gran inconveniente en cuanto a escoger el lugar correcto en el que se deben colocar estos sensores.

Una topología en estrella consigue que los paquetes vayan encaminados únicamente entre las dos partes de una comunicación, por lo que sería necesario colocar el sensor en un punto en el que fuera capaz de analizar cualquier intercambio de información.

Una primera opción sería colocar el sensor sobre el enlace donde se unen todos los equipos de la red. Esta opción podría suponer la necesidad de analizar una cantidad de datos tan elevada que el sensor acabaría perdiendo información.

La otra opción sería la colocación del sensor entre el enlace de red que separa el interior y el exterior, como si se tratara de un sistema de prevención perimetral adicional.

Una variante de estas dos opciones sería la utilización del puerto de intervención (*tap port*) que ofrecen muchos conmutadores. Se trata de un puerto especial que refleja todo el tráfico que pasa a través del equipo. Desgraciadamente, este puerto podría fácilmente sobrecargar la capacidad de análisis de los sensores si la cantidad de tráfico es muy elevada. Además, el ancho de banda interno del dispositivo es suficiente para tratar con todos los puertos activos a la vez, pero si el tráfico analizado comienza a crecer, es posible que se supere la capacidad de intervención del puerto, con la correspondiente pérdida de paquetes que ello comportaría.

5.2.4. Procesadores de eventos

Los procesadores de eventos, también conocidos como **analizadores**, conforman el núcleo central del sistema de detección. Tienen la responsabilidad de operar sobre la información recogida por los sensores para poder inferir posibles intrusiones.

Para inferir intrusiones, los analizadores deberán implementar algun esquema de detección. Dos de los esquemas más utilizados para realizar la detección son el modelo de detección de usos indebidos y el modelo de detección de anomalías. A continuación pasaremos a comentar brevemente estos dos esquemas de detección.

Esquema de detección basado en usos indebidos

La detección de intrusiones basada en el modelo de usos indebidos cuenta con el conocimiento *a priori* de secuencias y actividades deshonestas. Los procesadores de eventos que implementan este esquema analizan los eventos en busca de patrones de ataque conocidos o actividad que ataque vulnerabilidades típicas de los equipos.

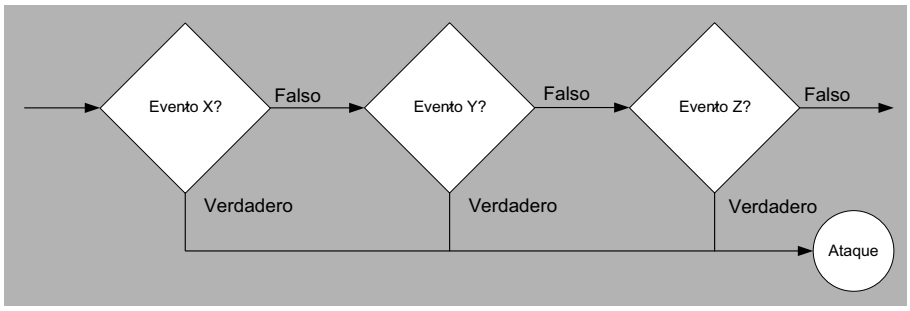
Estas secuencias o patrones se conocen bajo el nombre de *firmas de ataques* y podrían ser comparadas con las firmas víricas que utilizan los productos actuales de detección de virus.

Así pues, los componentes de detección basados en el modelo de usos indebidos compararán los eventos enviados por los sensores con las firmas de ataque que mantienen almacenadas en sus bases de conocimiento.

En el momento de detectar concordancia de algún acontecimiento o secuencia de eventos con alguna firma de ataque, el componente lanzará una alarma.

A la hora de implementar un esquema de detección basado en usos indebidos, dos de los modelos más utilizados son los analizadores basados en el reconocimiento de patrones y los analizadores basados en transiciones de estados.

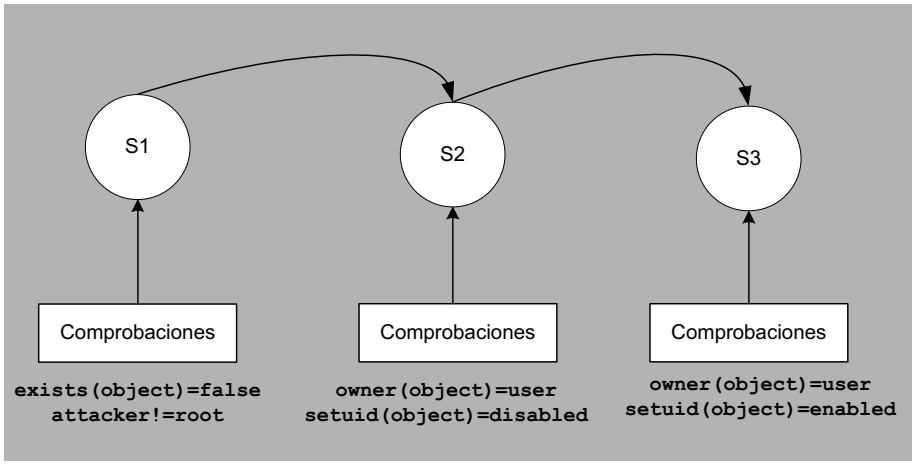
- **Analizadores basados en reconocimiento de patrones.** Mediante la utilización de reglas del tipo *if-then-else* para examinar los datos, estos analizadores procesan la información por medio de funciones internas en el sistema, de forma completamente transparente al usuario. La siguiente figura muestra el esquema de una regla *if-then-else*.



Aunque este modelo permite detectar una intrusión a partir de patrones conocidos *a priori*, su desventaja principal es que los patrones no definen un orden secuencial de acciones.

Detectar mediante este modelo ataques compuestos por una secuencia de eventos puede llegar a comportar grandes dificultades. Por otra parte, el mantenimiento y la actualización de la base de datos de patrones son otros puntos críticos de este modelo.

- **Analizadores basados en transiciones de estados.** Este modelo hace uso de autómatas finitos para representar los ataques, donde los nodos representan los estados, y las flechas (arcos), las transiciones.



La utilización de diagramas de transición facilita la asociación entre los estados y los distintos pasos que realiza un atacante desde que entra en un sistema, con privilegios limitados, hasta que se hace con el control del mismo.

Como principales ventajas de este modelo se puede destacar que los diagramas de transición permiten realizar una representación a alto nivel de escenarios de intrusión, ofreciendo una forma de identificar una serie de secuencias que conforman el ataque.

Por otra parte, estos diagramas definen de forma muy sencilla los ataques que se deben detectar. El motor de análisis podría llegar a utilizar diferentes variantes del mismo diagrama para identificar ataques similares.

Por contra, los diagramas de transición, y por lo tanto, los distintos pasos de la secuencia, se deben crear mediante lenguajes específicos que, en muchas ocasiones, suelen ser muy limitados e insuficientes para recrear ataques complejos.

Esta limitación provoca que este modelo no pueda detectar algunos de los ataques más comunes, siendo necesario el uso de motores de análisis adicionales como complemento del mismo.

Esquema de detección basado en anomalías

Los procesadores de eventos que basan su detección en un esquema de anomalías tratarán de identificar actividades sospechosas comparando el comportamiento de un usuario, proceso o servicio, con el comportamiento de perfil clasificado como normal.

Un perfil sirve como métrica (medida de un conjunto de variables) de comportamientos normales. Cualquier desviación que supere un cierto umbral respecto al perfil almacenado será tratado como una evidencia de ataque o intrusión.

Uno de los requisitos de este modelo es la necesidad de inicialización de un perfil por defecto que se irá adaptando progresivamente al comportamiento de un usuario, proceso o servicio no sospechoso. Es necesario, por lo tanto, el uso de heurísticas y descriptores estadísticos que ayuden a modelar correctamente cambios en el comportamiento tan pronto como suceda. Otras propuestas tratan de incorporar técnicas de inteligencia artificial para realizar estas tareas (como, por ejemplo, el uso de redes neuronales o de algoritmos genéticos).

La detección basada en anomalías ofrece claras ventajas respecto a la detección basada en usos indebidos. La ventaja más destacable es la posibilidad de detectar ataques desconocidos. Esto es posible porque ,independientemente de cómo haya conseguido el atacante la intrusión en un sistema, tan pronto como sus actividades comiencen a desviarse del comportamiento de un usuario normal, el procesador de eventos lanzará una alarma avisando de una posible intrusión.

Aun así, el esquema de detección basado en anomalías presenta bastantes inconvenientes*. El primero que debemos destacar es la falta de garantía en el proceso de detección: un intruso podría realizar sus acciones lentamente para ir provocando cambios en el perfil de usuario del procesador de eventos, con la finalidad que su presencia en el sistema pasara desapercibida.

* Estos inconvenientes provocan que la mayoría de los sistemas de detección comerciales disponibles en la actualidad implementen sus analizadores mediante el esquema de detección de usos indebidos.

Como segundo incoveniente podemos destacar la dificultad que aparece a la hora de clasificar y describir con precisión los ataques detectados mediante analizadores basados en anomalías. Generalmente, un analizador no sólo tiene que lanzar una alarma sino que deberá especificar de dónde procede el ataque, qué cambios ha sufrido el sistema, ...

Además, la tasa de falsos positivos y negativos que puede darse utilizando este esquema de detección es un gran inconveniente, ya que no siempre una desviación respecto al perfil esperado coincidirá con un ataque o intento de intrusión. En el caso de procesadores cuyos eventos procedan de sensores basados en red, es posible que el número de alarmas lanzadas (en una red de tamaño medio) supere fácilmente el centenar. Esto provoca que, con frecuencia, los administradores de la red acaben ignorando las alarmas lanzadas por el sistema de detección, o incluso desactivando el sistema al completo.

5.2.5. Unidades de respuesta

Las unidades de respuesta de un sistema de detección se encargaran de iniciar acciones de respuesta en el momento en que se detecte un ataque o intrusión. Estas acciones de respuesta pueden ser automáticas (**respuesta activa**) o requerir interacción humana (**respuesta passiva**).

Las respuestas activas tienen como objetivo actuar contra el ataque, intentando su neutralización, en el momento en el que es detectado (o mientras una intrusión todavía continúa en curso). Un ejemplo de respuesta activa puede ser la cancelación de la conexión en red que originó el ataque o el propio seguimiento del ataque que permitiría más adelante el análisis correspondiente. Por contra, las respuestas pasivas se limitan a lanzar una alarma para informar y describir el ataque detectado en el administrador del sistema. La mayoría de los componentes de respuesta pasiva ofrecen distintas formas de hacer llegar esta información al administrador como, por ejemplo, mediante un correo electrónico, mediante la utilización de mensajes SMS, etc.

El problema de las respuestas activas es que pueden acabar en una denegación de servicio contra usuarios o sistemas legítimos. Es muy probable que algunas de las alarmas que los procesadores hacen saltar sean incorrectas. Por ejemplo, si la unidad de respuesta cortara inmediatamente con la conexión que originó esta alarma, o con aquellos procesos considerados sospechosos, ello podría suponer la pérdida de trabajo de un usuario o servicio inocente.

En la mayoría de los sistemas (por ejemplo, servidores de comercio electrónico) este tipo de errores puede suponer la pérdida de clientes, la cual cosa es inadmisible. Por este motivo, la mayoría de empresas del sector del comercio electrónico se decantan por la contratación de especialistas que, manualmente, analicen los informes generados por el sistema de detección para determinar si es necesaria una respuesta activa ante tal aviso.

Al igual que los sensores, las unidades de respuesta se podrían clasificar en distintas categorías según el punto de actuación. Las dos categorías más generales son las unidades de respuesta basadas en equipo y las unidades de respuesta basadas en red.

- **Unidades de respuesta basadas en equipo.** Se encargan de actuar a nivel de sistema operativo (como, por ejemplo, bloqueo de usuarios, finalización de procesos, etc).

- **Unidades de respuesta basadas basadas en red.** Actúan a nivel de red cortando intentos de conexión, filtrando direcciones sospechosas, etc.

5.2.6. Elementos de almacenamiento

En algunas situaciones, el volumen de información recogida por los sensores del sistema de detección llega a ser tan elevado que se hace necesario, previo análisis, un proceso de almacenamiento. Supongamos, por ejemplo, el caso de que todos los paquetes de una red de alta velocidad deban ser inspeccionados por los analizadores del sistema de detección. En este caso, será necesario plantearse una jerarquía de almacenamiento que reduzca el volumen de información sin penalizar las posibilidades de análisis.

Una posibilidad es la clasificación de la información en términos de análisis a corto y largo plazo.

En el caso de análisis a corto plazo, la información será almacenada directamente en los propios sensores (en *buffers* internos) de forma que después de realizar un procesado previo de los datos, y su transformación a un formato de evento, éstos sean transmitidos a los elementos de análisis.

En el caso de información a medio plazo, los datos preprocesados serán almacenados en dispositivos secundarios (con el formato apropiado) en lugar de ser transmitidos a los analizadores del sistema.

El tiempo de almacenamiento de una información a medio plazo puede ser del orden de dos o tres días, con el objetivo de que pueda ser consultada por los analizadores del sistema en el caso de que el proceso de análisis así lo requiera.

Eventualmente, y después de un proceso de compresión (para reducir el tamaño), parte de la información a medio plazo podrá continuar almacenada durante largos períodos de tiempo (del orden de meses o incluso años) a la espera de que pueda ser consultada por procesos de detección a largo plazo.

5.3. Escáners de vulnerabilidades

Los escáners de vulnerabilidades son un conjunto de aplicaciones que nos permitirán realizar pruebas o tests de ataque para determinar si una red o un equipo tiene deficiencias de seguridad que pueden ser explotadas por un posible atacante o comunidad de atacantes.

Aun no siendo formalmente un elemento de detección tradicional, los escáners de vulnerabilidades poseen una estrecha relación con las herramientas de detección utilizadas en los sistemas de detección de intrusos. En realidad, en muchos ámbitos se les considera un caso especial de estas herramientas y, generalmente, son utilizados para realizar un análisis de intrusiones.

Esto es así porque dentro de los mecanismos de detección de ataques podemos distinguir entre elementos de detección de tipo dinámico (sería el caso de las herramientas de detección utilizadas en un sistema de detección de intrusos) y elementos de detección de tipo estático (los escáners de vulnerabilidades). En los primeros se trabaja de forma continua (como lo haría una videocámara de vigilancia) mientras que los segundos se concentran en intervalos de tiempos determinados (sería el caso de una cámara fotográfica).

A causa de este aspecto estático, los escáners de vulnerabilidades únicamente podrán detectar aquellas vulnerabilidades contenidas en su base de conocimiento. Además, sólo son capaces de identificar fallos de seguridad en los intervalos en que se ejecutan. No obstante, son unas herramientas de gran utilidad y un buen complemento de los sistemas de detección instalados en una red.

El funcionamiento general de un escáner de vulnerabilidades se podría dividir en tres etapas:

- Durante la primera etapa se realiza una extracción de muestras del conjunto de atributos del sistema, para poder almacenarlas posteriormente en un contenedor de datos seguro.

- En la segunda etapa, estos resultandos son organizados y comparados con, al menos, un conjunto de referencia de datos. Este conjunto de referencia podría ser una plantilla con la configuración ideal generada manualmente, o bien ser una imagen del estado del sistema realizada con anterioridad.

- Finalmente, se generará un informe con las diferencias entre ambos conjuntos de datos.

Las tres etapas anteriores se podrían mejorar mediante la utilización de motores de comparación en paralelo o incluso mediante la utilización de métodos criptográficos para detectar cambios en los objetos monitorizados.

A la hora de clasificar este tipo de herramientas encontramos básicamente dos categorías principales, según la localización desde la que se obtienen datos: escáners basados en máquina o escáners basados en red.

5.3.1. Escáners basados en máquina

Este tipo de herramientas fue el primero en utilizarse para la evaluación de vulnerabilidades. Se basa en la utilización de información de un sistema para la detección de vulnerabilidades como, por ejemplo, errores en permisos de ficheros, cuentas de usuario abiertas por defecto, entradas de usuario duplicadas o sospechosas, etc.

Esta información se puede obtener mediante consultas al sistema, o a través de la revisión de distintos atributos del mismo.

Un simple guión de sistema como el siguiente se encargaría de avisar mediante correo electrónico al administrador del sistema en caso de encontrar entradas anómalas en el fichero de contraseñas del sistema:

Las vulnerabilidades que se suelen encontrar mediante la evaluación basada en máquina acostumbran a estar relacionadas con ataques de escalada de privilegios.

```
#!/usr/bin/perl
$count==0;
open(MAIL, "| /usr/lib/sendmail mikal");
print MAIL "To: Administration\n";
print MAIL "Subject: Password Report\n";
open(PASSWORDS, "cat /etc/passwd |");

while(<PASSWORDS>) {
      $linenumber=$.;
      @fields=split(/:/, $_);
      if($fields[1] eq "") {
            $count++;
            print MAIL "\n***WARNING***\n";
            print MAIL "Line $linenumber has a blank password.\n";
            print MAIL "Here's the record: @fields\n";
      }
}

close(PASSWORDS);
if($count < 1) print MAIL "No blank password found\n";
print MAIL ".\n";
close(MAIL);
```

Los motores de análisis de vulnerabilidades basados en máquina están muy relacionados con el sistema operativo que evalúan, lo cual provoca que su mantenimiento sea un tanto costoso y complica su administración en entornos heterogéneos.

Uno de los primeros escáners de vulnerabilidades en sistemas Unix fue COPS, una herramienta que se encargaba de analizar el sistema a la búsqueda de problemas de configuración típicos como, por ejemplo, permisos erróneos de ficheros, directorios y servicios, contraseñas de usuario débiles, bits de suplantación impropios, etc. Éste sería un ejemplo de informe reportado por COPS:

```
ATTENTION:

Security Report for Sun Abr 20 20:57:09 CET 2003 from host vm3

Warning! NFS filesystem exported with no restrictions!
Warning! NFS filesystem exported with no restrictions!
Warning! NFS filesystem exported with no restrictions!
Warning! NFS filesystem exported with no restrictions!
Warning! /dev/fd0 is_World_writable!
Warning! /dev/fd0 is_World_readable!
Warning! /var/spool/mail is_World_writable!
Warning! /etc/security is_World_readable!
Warning! /usr/local/bin is_World_writable!
Warning! /root/adduser.log is_World_readable!
Warning! /root/bash.man is_World_readable!
Warning! /root/bin is_World_readable!
Warning! /root/control is_World_readable!
Warning! /root/cops_1_04.tar is_World_readable!
Warning! /root/cops.man is_World_readable!
Warning! /root/cops_man.html is_World_readable!
```

Otra herramienta similar es TIGER que, al igual que COPS, se compone de un conjunto de aplicaciones y guiones de sistema con el objetivo de realizar auditorías de seguridad en sistemas Unix. Su objetivo principal era el de informar de las distintas formas en las que puede comprometerse el sistema.

La siguiente imagen muestra un ejemplo de informe reportado por TIGER:

```
#hosts.equiv      This file describes the names of the
#                 hosts which are to be considered "equivalent",
#                 i.e. which are to be  trusted enought
#                 for allowing rsh (1) commands.
#
#hostname
#Checking accounts from /etc/passwd...
#Performing check of .netrcfiles...
#Checking accounts from /etc/passwd...
#Performing check of PATH components...
#Only checking user'root'

--WARN--[path002w]/usr/bin/amadmin in root's
         PATH from default is not owned  by root (owned by amanda).
--WARN--[path002w]/usr/bin/amcheckdb in root's
         PATH from default is not owned by root (owned by amanda).
--WARN--[path002w]/usr/bin/amcleanup in root's
         PATH from default is not owned by root (owned by amanda).
--WARN--[path002w]/usr/bin/amdump in root's
         PATH from default is not owned by root (owned by amanda).
```

5.3.2. Escáners basados en red

Los escáners de vulnerabilidades basados en red aparecieron posteriormente y se han ido haciendo cada vez más populares. Obtienen la información necesaria a través de las conexiones de red que establecen con el objetivo que hay que analizar.

Así pues, los escáners de vulnerabilidades basados en red realizan pruebas de ataque y registran las respuestas obtenidas. No se deben confundir estos analizadores de vulnerabilidades basados en red con los analizadores de sistemas de detección de intrusos. Aunque un escáner de estas características puede ser muy similar a una herramienta de detección de intrusiones, no representa una solución tan completa.

Dos de las técnicas más utilizadas para la evaluación de vulnerabilidades basadas en red son las siguientes:

- **Prueba por explotación.** Esta técnica consiste en lanzar ataques reales contra el objetivo. Estos ataques están programados normalmente mediante guiones de comandos. En lugar de aprovechar la vulnerabilidad para acceder al sistema, se devuelve un indicador que muestra si se ha tenido éxito o no. Obviamente, este tipo de técnica es bastante agresiva, sobre todo cuando se prueban ataques de denegación de servicio.

- **Métodos de inferencia.** El sistema no explota vulnerabilidades, sino que busca indicios que indiquen posibilidades de ataque, tratando de detectar posibles deficiencias de seguridad en el objetivo.

  Este método es menos agresivo que el anterior, aunque los resultados obtenidos son menos exactos.

  Ejemplos de técnicas de inferencia pueden ser la comprobación de versión de sistema para determinar si existe una vulnerabilidad, la comprobación del estado de determinados puertos para descubrir cuáles están abiertos, la comprobación de conformidad de protocolo mediante solicitudes de estado, etc.

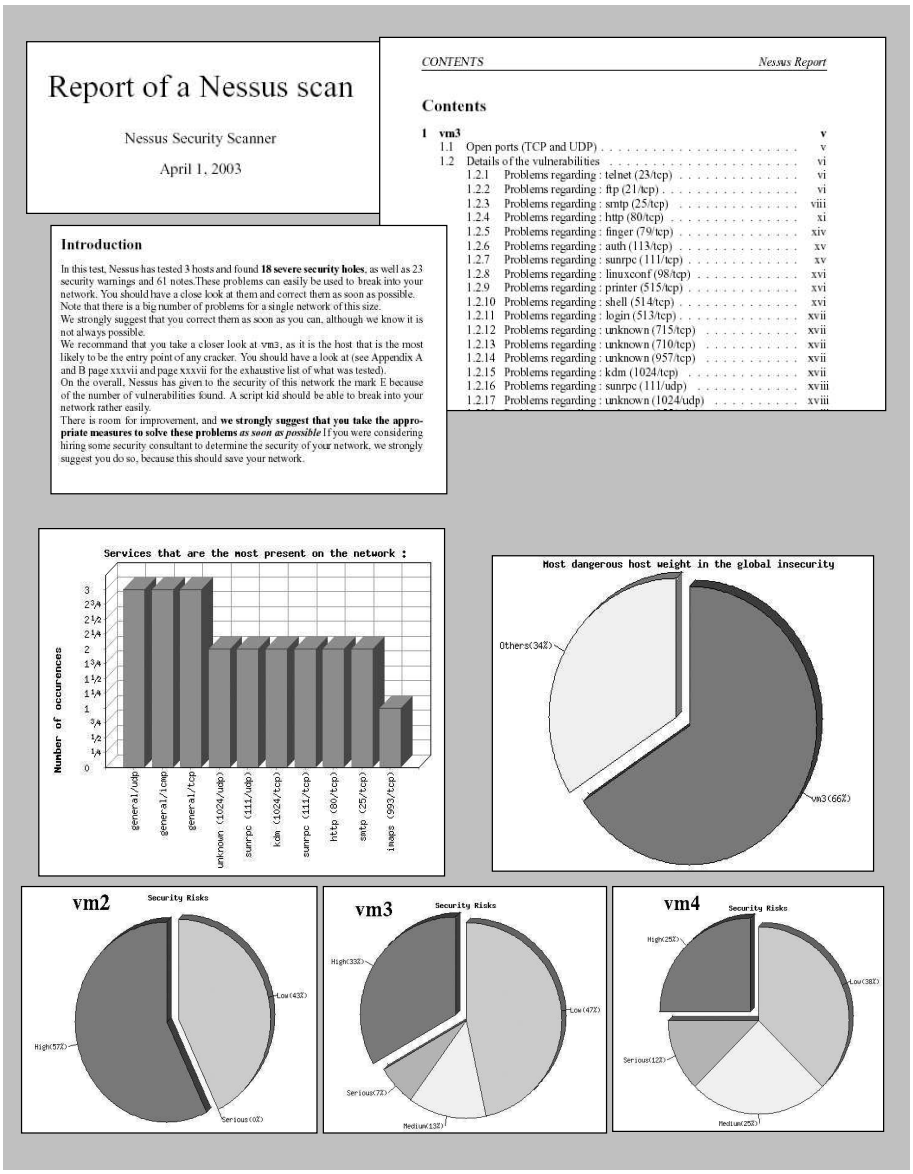
Uno de los productos más utilizados actualmente como escáner de vulnerabilidades basado en red es Nessus.

Nessus ...

... es un escáner de vulnerabilidades de red desarrollado bajo el paradigma de *software* libre, distribuído inicialmente bajo licencia GPL (*General Public License*) de GNU y actualmente bajo licencia LGPL (*Lesser General Public License*) de GNU. Fue desarrollado por Renaud Deraison en 1998. Su precursor fue SATAN, otro escáner de vulnerabilidades de red, desarrollado por Wietse Venema y Dan Farmer. Ved la página web `www.nessus.org` para más información.

Nessus es una herramienta basada en un modelo cliente-servidor que cuenta con su propio protocolo de comunicación. De forma similar a otros escáners de vulnerabilidades existentes, el trabajo correspondiente para explorar y probar ataques contra objetivos es realizado por el servidor de Nessus (`nessusd`), mientras que las tareas de control, generación de informes y presentación de los datos son gestionadas por el cliente (`nessus`).

La siguiente figura muestra un ejemplo de informe reportado con Nessus:



5.4. Sistemas de decepción

Hasta el momento, los mecanismos de seguridad que hemos visto buscan abordar el problema de la seguridad de una red desde un punto de vista defensivo. El inconveniente de este acercamiento es que es únicamente defensivo y sólo es el atacante quien toma la iniciativa.

Como novedad, los sistemas de decepción tratarán de cambiar las reglas del juego, ofreciendo al administrador de la red la posibilidad de tomar la iniciativa.

Los sistemas de decepción, en vez de neutralizar las acciones de los atacantes, utilizan técnicas de monitorización para registrar y analizar estas acciones, tratando de aprender de los atacantes.

Pese a que en algunos países no están claramente definidos los aspectos legales de estos sistemas, lo cierto es que cada vez son más utilizados.

A continuación trataremos de resumir distintas estrategias que se pueden emplear para la construcción de este tipo de sistemas.

5.4.1. Equipos de decepción

Los equipos de decepción, también conocidos como tarros de miel o ***honeypots***, son equipos informáticos conectados en que tratan de atraer el tráfico de uno o más atacantes. De esta forma, sus administradores podrán ver intentos de ataques que tratan de realizar una intrusión en el sistema y analizar cómo se comportan los elementos de seguridad implementados en la red.

Otro de los objetivos es la obtención de información sobre las herramientas y conocimientos necesarios para realizar una intrusión en entornos de red como los que pretendemos proteger. Toda esta información acabará sirviendo para detener futuros ataques a los equipos de la red de producción.

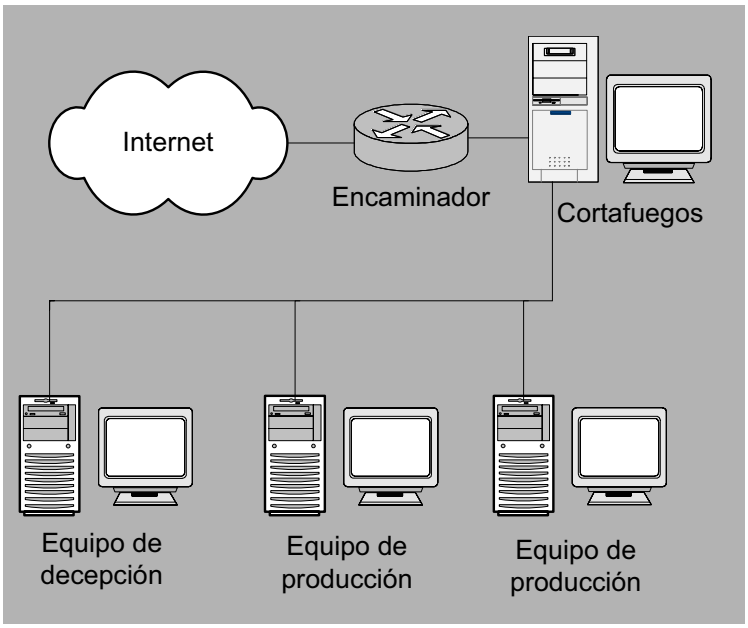
La idea conceptual de un equipo de decepción existe desde hace varias décadas. Como primera aproximación podríamos definirlo como un recurso de la red diseñado para que varios atacantes puedan introducirse en él de forma sencilla.

Estos equipos suelen estar diseñados para imitar el comportamiento de equipos de producción reales y conseguir así ser de interés para una comunidad de atacantes.

Suelen contar con mecanismos de prevención para que un atacante con éxito no pueda acceder a la totalidad de la red. Naturalmente, si un intruso consigue atacar el equipo, no debe percatarse de que está siendo monitorizado o engañado.

Así pues, estos equipos deberían estar instalados detrás de sistemas cortafuegos configurados para que se permitan conexiones de entrada al equipo de decepción, pero limitando las conexiones de salida (para evitar que el intruso pueda atacar sistemas de producción reales desde el equipo de decepción).

La siguiente figura muestra la ubicación de un posible equipo de decepción dentro de una red local:



Examinando la actividad reportada dentro del equipo de decepción, será posible identificar el problema y detectar cómo se ha conseguido la intrusión en el sistema, además de poder reportar la actividad desencadenada a partir de este momento.

5.4.2. Celdas de aislamiento

Las celdas de aislamiento tienen una metodología muy similar a los equipos de decepción que acabamos de ver. Mediante el uso de un dispositivo intermedio (con capacidades de detección y encaminamiento) todo el tráfico etiquetado como malicioso será dirigido hacia un equipo de decepción (conocido en este caso como celda de aislamiento).

* En inglés, *padded cell*.

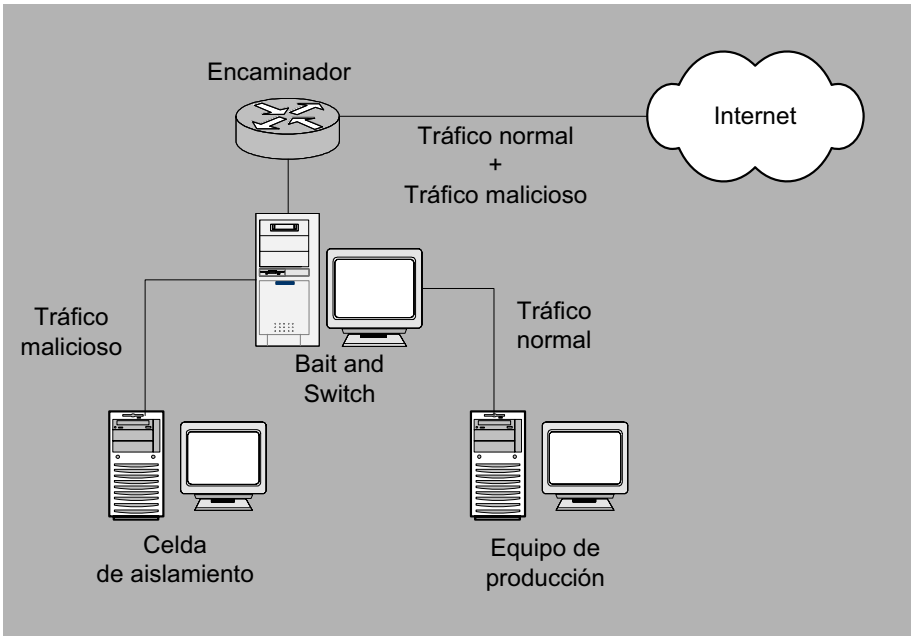
Al igual que en el caso anterior, una celda de aislamiento ofrece al atacante un entorno aparentemente idéntico a un equipo real o de producción. No obstante, la celda estará protegida de tal manera que no pueda poner en riesgo al resto de equipos de la red o del exterior. En la mayoría de situaciones, estas celdas de aislamiento son copias exactas de los sistemas de producción reales hacia los que va dirigido el tráfico malicioso, proporcionando de esta forma un escenario más creíble.

Al igual que los equipos de decepción, las celdas de aislamiento se pueden utilizar para comprender mejor los métodos utilizados por los intrusos.

Bait and Switch ...

es un ejemplo de herramienta que podría implementar la idea de celda de aislamiento. Se trata de una utilidad que se instalará en un dispositivo con tres interfaces de red y que encamina el tráfico hostil hacia la celda de aislamiento, basándose en la utilización de `snort`, `iproute2`, `netfilter` y código propio de la aplicación.

La siguiente figura muestra un esquema sencillo de una celda de aislamiento mediante el producto *Bait and Switch*:

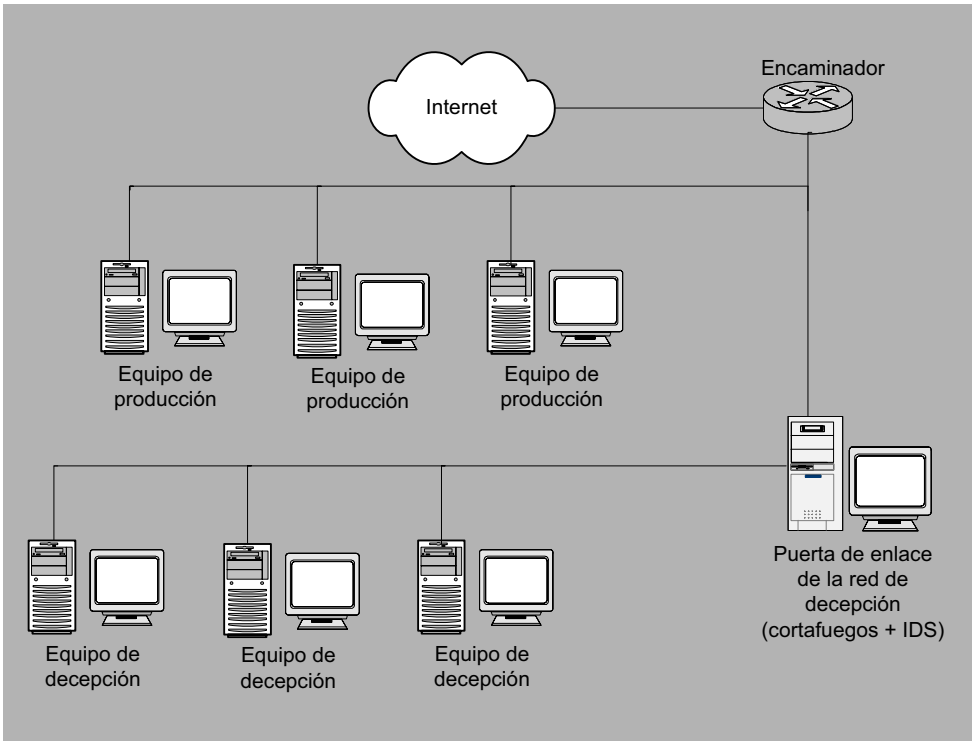


5.4.3. Redes de decepción

Un enfoque más avanzado que los anteriores consiste en la construcción de todo un segmento de red compuesto únicamente por equipos de decepción, preparados todos ellos para engañar a los intrusos (permitiendo su acceso sin demasiada dificultad).

Los equipos de este segmento ofrecerán servicios configurados de tal modo que puedan atraer la atención a toda una comunidad de intrusos con el objetivo de registrar todos sus movimientos mediante los equipos de la red de decepción.

La siguiente figura muestra un posible esquema para la construcción de este tipo de redes:



Como se puede ver en la figura anterior, una pasarela (que combina en su interior elementos de detección y de prevención) une la red de decepción con la red de producción. Esta pasarela funciona en modo puente, de forma que se podrá prescindir de dirección IP, reduciendo las posibilidades de detección por parte de los atacantes.

Todos los sistemas instalados dentro de la red de decepción tendrán deberán ser sistemas de decepción y ofrecerán sus servicios de la forma más realista posible. Para ello, deberían ofrecer servicios reales, como los que podríamos encontrar en cualquier equipo de producción.

Al no haber en estos equipos de decepción servicios simulados, todas las conclusiones extraídas durante el análisis de una intrusión se podrán extrapolar directamente en la red de producción real. Así, todas las deficiencias y debilidades que se descubran dentro de la red de decepción podrán servir para describir las existentes en la parte de producción.

El funcionamiento de la red de decepción se basa en un solo principio: todo el tráfico que entra en cualquiera de sus equipos se debe considerar sospechoso.

A través de los mecanismos de detección instalados en la pasarela se realizará el proceso de monitorización, detectando ataques basados en tendencias o estadísticas ya conocidas. Sin embargo, las posibilidades de investigar toda la actividad de una red de decepción debería ayudar a detectar ataques desconocidos.

Las redes de decepción se deben contemplar como herramientas de análisis para mejorar la seguridad de las redes de producción. Son una solución muy valiosa si una organización puede dedicarle el tiempo y los recursos necesarios.

5.5. Prevención de intrusos

Los **sistemas de prevención de intrusos*** son el resultado de unir las capacidad de bloqueo de los mecanismos de prevención (encaminadores con filtrado de paquetes y pasarelas) con las capacidades de análisis y monitoritación de los sistemas de detección de intrusos.

*En inglés, *Intrusion Prevention Systems* (IPS).

Como ya hemos visto en el primer apartado de este módulo didáctico, los sistemas de detección de intrusos pueden llegar a ser sistemas de seguridad proactivos. Pero generalmente los productos de detección más ampliamente implantados suelen ser únicamente reactivos, es decir, esperan a que tenga lugar un ataque para emitir una alarma. Por el contrario, los sistemas de prevención de intrusos son sistemas con capacidad de detener un ataque o intrusión antes de que éste pueda llegar a causar daños.

La mayor parte de especialistas consideran que estos sistemas de prevención son un caso especial de sistema de detección de intrusos, puesto que ambos sistemas comparten la misma metodología básica de detección. En realidad, la mayoría de expertos los considera una evolución directa de los sistemas de detección de intrusos e se llegan a considerar como la siguiente generación de estos sistemas**.

** Ved el artículo ïntrusion Prevention Systems: the Next Step in the Evolution of IDS"que encontraréis en la página web `http://www.security-focus.com/infocus-/1670` para más información.

Así pues, el comportamiento de un sistema de prevención de intrusos es similar al de un sistema de detección de intrusos de respuesta activa (los que disponen de unidad de respuesta capaz de responder ante los ataques detectados), de forma que se encargan de descartar o bloquear los paquetes sospechosos tan pronto como son identificados. Así, todos los paquetes que pertenezcan a una misma sesión sospechosa (detectada a partir de los sensores y los procesadores de eventos del sistema de prevención) serán eliminados de la misma forma.

Algunos sistemas de prevención de intrusos también contemplan la posibilidad de detectar anomalías en el uso de protocolos, como paquetes manipulados malintencionadamente.

-*** En inglés, *Host based Intrusion Prevention Systems* (HIPS).
-**** En inglés, *Network based Intrusion Prevention Systems* (NIPS).

Atendiendo a la fuente de datos que utilicen, los sistemas de prevención de intrusos se podrían clasificar en las dos categorías que la mayor parte de los elementos de detección que hemos visto hasta ahora: basados en máquina* y basados en red**.

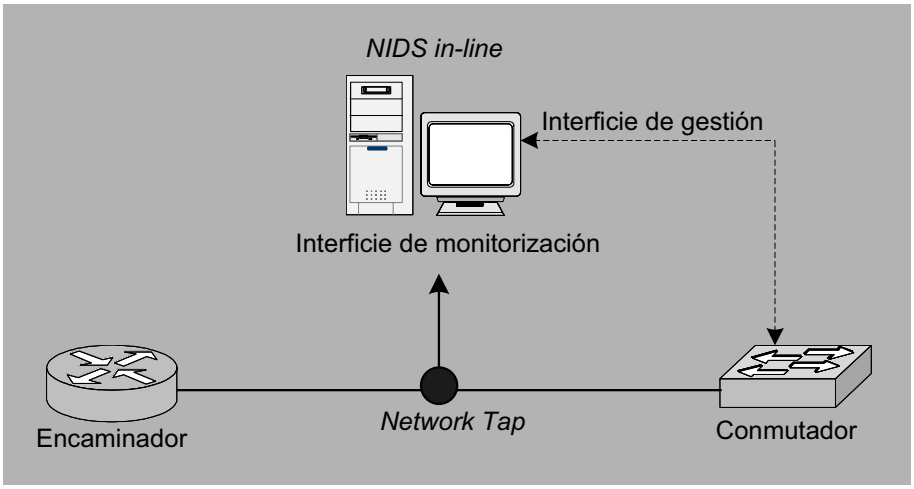
En el primer caso, los sistemas de prevención de intrusos basados en equipo (HIPS) suelen utilizar aplicaciones instaladas directamente en la máquina que hay que proteger. Estas aplicaciones suelen estar muy relacionadas con el sistema operativo del sistema y sus servicios. El segundo grupo, sistemas de prevención de intrusos basados en red, suelen ser dispositivos de red con al menos dos interfaces (una de monitorización interna y otra de monitorización externa), integrando en el mismo producto las capacidades de filtrado de paquetes y motor de detección.

A continuación haremos un breve repaso sobre los modelos existentes más relevantes para construir un sistema de prevención tal como acabamos de definir.

5.5.1. Sistemas de detección en línea

La mayor parte de los productos y dispositivos existentes para la monitorización y detección de ataques en red se basan en la utilización de dos dispositivos de red diferenciados. Por una parte, uno de los dispositivos se encarga de interceptar el tráfico de su segmento de red, mientras que el otro se utiliza para efectuar tareas de gestión y administración. En la siguiente figura vemos un ejemplo típico de dispositivo de detección en modo de escucha, conocido como sistemas de detección en línea:

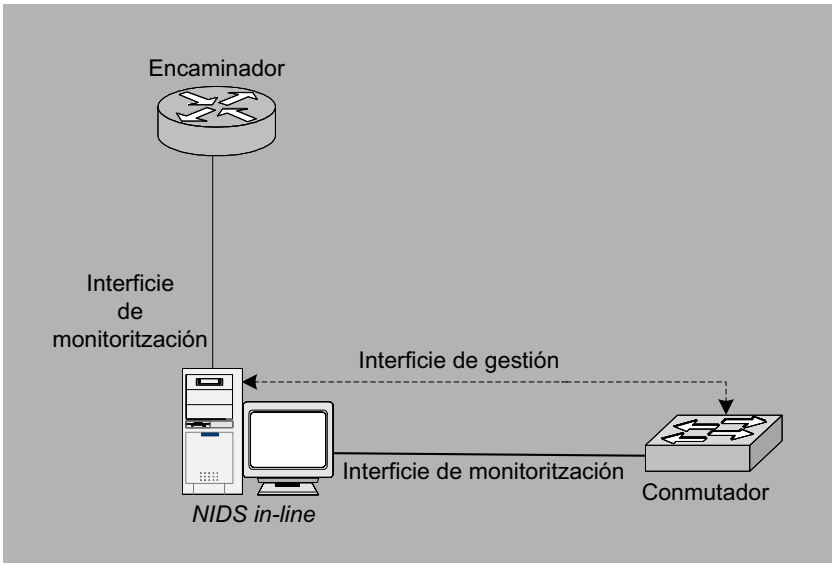
-* En inglés, *tap mode*.
-** En inglés, *network tap*.



En el dispositivo de la figura, la interfaz de red utilizada para la monitorización está conectada a un dispositivo de escucha** que le permite analizar el tráfico del segmento de red. Además, esta interfaz no suele tener asignada ninguna dirección IP, disminuyendo de esta forma las posibilidades de ser detectada. Con ello, un sistema de detección en línea actúa en la capa de red del modelo TCP/IP, como si de un dispositivo puente se tratara.

Mediante una de las interfaces recibirá el tráfico del exterior (potencialmente hostil), mientras que por el otro podrá transmitir por la red que hay que proteger. Generalmente, estos sistemas tienen una tercera interfaz para las tareas de administración y gestión.

Esta situación le permitirá un control total sobre el tráfico que pasa por el segmento de red en el que está colocado. No sólo podrá analizar todo el tráfico que reciba, sino que podrá gestionar el ancho de banda.



Una de las aplicaciones que podríamos utilizar para desarrollar esta idea es la herramienta *hogwash*. Se trata de una utilidad de red que utiliza el procesador de eventos de *snort* para anular todo aquel tráfico malicioso encaminado contra la red que se quiere proteger.

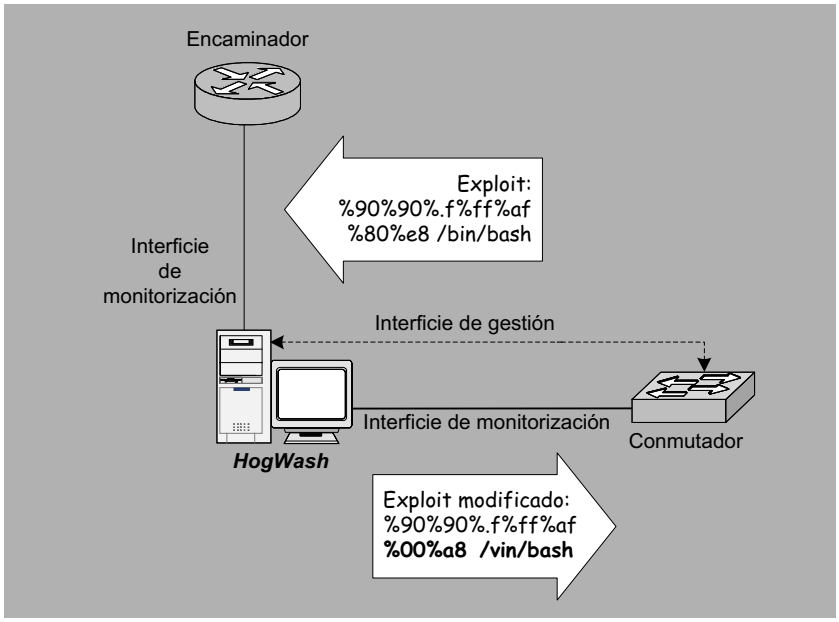
Hogwash

Ved la página web `http://hogwash.sourceforge.net` para más información.

Como herramienta de prevención, *hogwash* implementa las capacidades de detección y de bloqueo de tráfico. Adicionalmente, también ofrece la opción de reescribir el tráfico de red. Así, si un atacante envía una petición maliciosa, *hogwash* puede modificarla antes de encaminar este tráfico hacia el otro segmento de la red:

Snort ...

... es una de las herramientas de detección más utilizadas por la mayoría de los sistemas de detección actuales. Esta herramienta, desarrollada bajo el paradigma de *software* libre, es capaz de realizar análisis de tráfico de red en tiempo real. Ved la página web `www.snort.org` para más información.

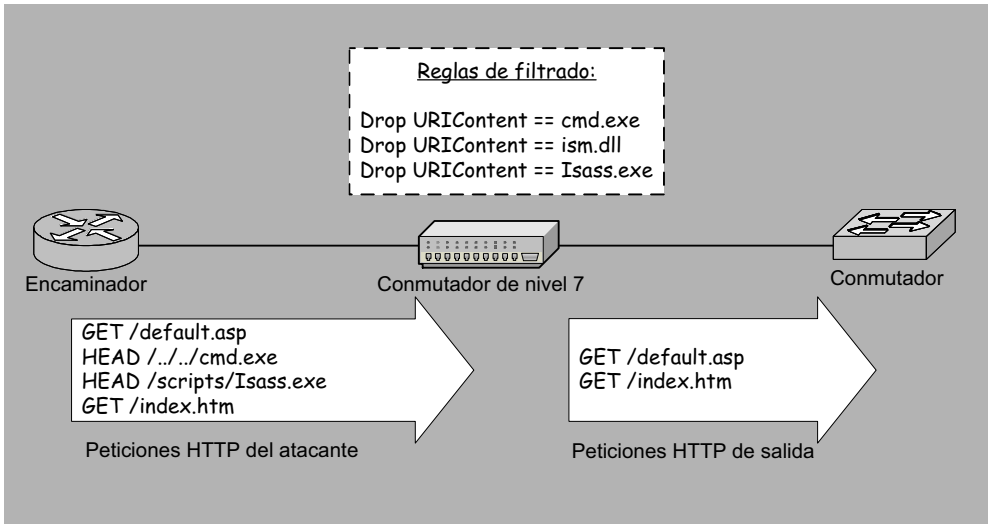


5.5.2. Conmutadores de nivel siete

Aunque los conmutadores han sido tradicionalmente dispositivos de nivel de red, la creciente necesidad de trabajar con grandes anchos de banda ha provocado que vayan ganando popularidad los conmutadores a nivel de aplicación (nivel siete del modelo OSI).

Estos dispositivos se suelen utilizar para realizar tareas de balanceo de carga de una aplicación entre varios servidores. Para ello, examinan la información a nivel de aplicación (por ejemplo HTTP, FTP, DNS, etc.) para tomar decisiones de encaminamiento. Adicionalmente, estos mismos dispositivos podrán proporcionar protección frente a ataques contra las redes que conmutan como, por ejemplo, descartar tráfico procedente de una denegación de servicio.

En la siguiente figura podemos ver el procedimiento general de funcionamiento de un conmutador de nivel siete:



Los ataques que mejor reconocen estos conmutadores de nivel siete son los ataques de denegación de servicio. El motor de detección utilizado suele basarse en la detección de usos indebidos, implementada en la mayoría de casos mediante el uso de patrones de ataque.

Una de las primeras ventajas de trabajar con estos dispositivos es la posibilidad de realizar detecciones de ataques en redes de alta velocidad conmutadas.

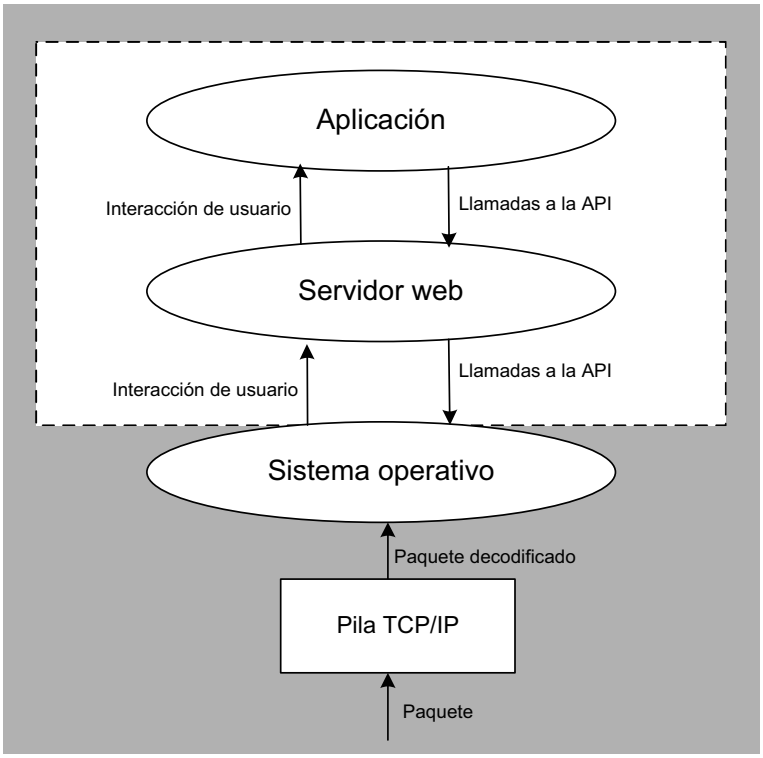
Otra ventaja, que no se encuentra en otros sistemas de prevención, es la posibilidad de redundancia. Esta redundancia se puede conseguir con la utilización de sistemas secundarios configurados para activarse en caso de fallo por parte de dispositivos primarios.

5.5.3. Sistemas cortafuegos a nivel de aplicación

Los sistemas cortafuegos a nivel de aplicación, al igual que los conmutadores de nivel siete que acabamos de ver, trabajan en el nivel de aplicación del modelo OSI. Se trata de unas herramientas de prevención que se puede instalar directamente sobre el sistema final que se quiere proteger.

Aparte de realizar un análisis sobre el tráfico de red*, estos dispositivos se pueden configurar para analizar eventos tales como la gestión de memoria, las llamadas al sistema o intentos de conexión del sistema donde han sido instalados.

* Ved el módulo didáctico *Mecanismos de prevención* de este mismo material para más información.



Para realizar este tipo de análisis, se basan en la utilización de perfiles estadísticos. Esta técnica se basa en una primera fase de inicialización de perfiles (fase de entrenamiento) y una segunda fase en la que las acciones son comparadas por el sistema contra estos perfiles.

Durante la fase de entrenamiento, se procede a registrar la actividad de las aplicaciones para elaborar un modelo de comportamiento que sirva para la detección de posibles intrusiones, junto con una serie de políticas de seguridad. Así, todas las acciones que no hayan sido definidas durante la creación de perfiles serán identificadas como maliciosas por el dispositivo y podrán ser bloqueadas.

De los distintos esquemas de prevención que hemos visto hasta ahora, éste es el único que monitoriza la actividad en las aplicaciones y la relación entre éstas y el sistema operativo. Además, podrá ser instalado en cada máquina física que hay que proteger, lo cual garantiza un alto nivel de personalización por parte de los administradores y usuarios finales.

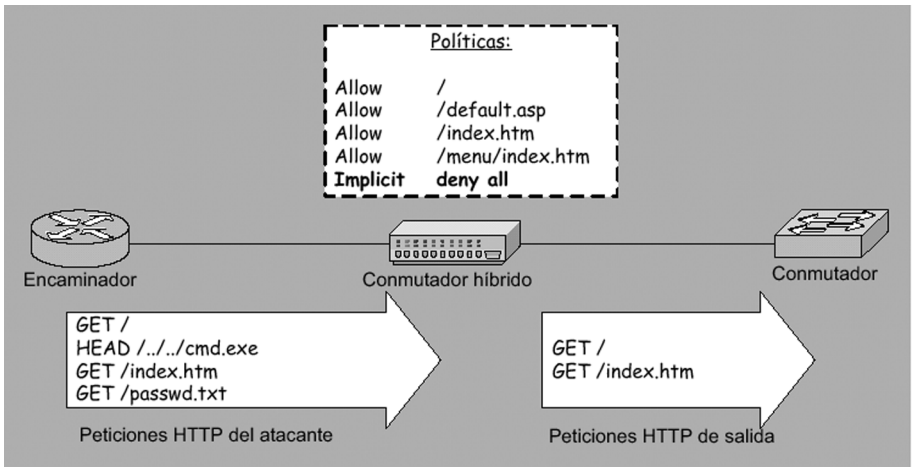
5.5.4. Conmutadores híbridos

El último modelo de prevención de intrusos que veremos es una combinación de los conmutadores de nivel siete y de los sistemas cortafuegos a nivel de aplicación que acabamos de presentar. Así, un conmutador híbrido será aquel dispositivo de red instalado como un conmutador de nivel siete, pero sin utilizar conjuntos de reglas.

Su método de detección está basado en políticas, como el de los sistemas cortafuegos a nivel de aplicación. Por lo tanto, estos conmutadores analizarán el tráfico de red para poder detectar información definida en las políticas que tienen configuradas.

La combinación de un sistema cortafuegos a nivel de aplicación junto con un conmutador de nivel siete permite reducir problemas de seguridad asociados a una programación deficiente*, así como la posibilidad de detectar ataques a nivel de aplicación.

* Ved el capítulo de *Deficiencias de programación* del primer módulo didáctico de este mismo material para más información



Como vemos en la figura anterior, el dispositivo tendrá conocimientos sobre el servidor que protege (servidor FTP, HTTP, SMTP, ...), como cualquier otro conmutador de nivel siete, pero también tendrá conocimiento sobre las aplicaciones que se sitúan por encima.

Los conmutadores híbridos pueden combinarse con otros conmutadores de nivel siete para reducir carga. Así, los conmutadores de nivel siete complementarios podrían redirigir únicamente peticiones consideradas como potencialmente maliciosas, para que el conmutador híbrido finalice la detección.

5.6. Detección de ataques distribuidos

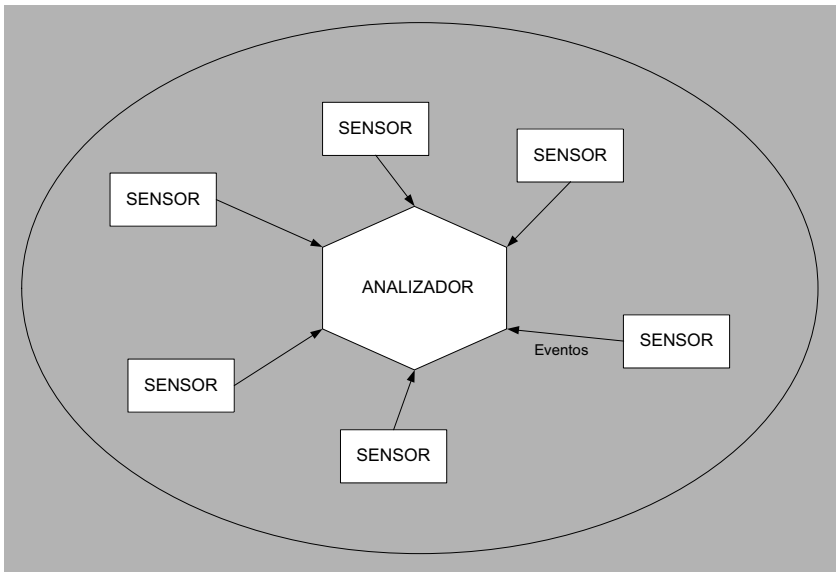
Un caso de interés especial dentro de los mecanismos de detección es el de la identificación de ataques distribuidos o coordinados. Un ejemplo de este tipo de ataques son las denegaciones de servicio basadas en modelos *master-slave* que hemos descrito en el primer módulo de estos materiales. Este tipo de ataques, que no pueden ser indentificados buscando patrones de forma aislada, deben ser detectados a partir de la combinación de múltiples indicios encontrados en distintos equipos de una red monitorizada.

A continuación veremos, de forma muy resumida, las distintas propuestas que existen para poder poner en correspondencia los eventos recogidos en distintos equipos de la red, a fin de implementar una detección de ataques e intrusiones distribuida.

5.6.1. Esquemas tradicionales

Las primeras propuestas para extender la detección de ataques desde un equipo aislado hacia un conjunto de equipos tratan de unificar la recogida de información utilizando esquemas y modelos centralizados. Así, estas propuestas plantean la instalación de sensores en cada uno de los equipos que se desea proteger, configurados para poder retransmitir toda la información hacia un punto central de análisis.

Desde este punto central, toda la información recibida será analizada utilizando distintos métodos de detección (detección basada en usos indebidos, detección basada en anomalías, ...), como vemos en la siguiente figura:



Este diseño presenta un claro problema de sobrecarga sobre el punto central de análisis, debido a la gran cantidad de información que éste podría llegar a recibir.

La mayor parte de las soluciones existentes basadas en este esquema utilizan esquemas de reducción de información (realizando procesos de prefiltrado y compresión) para minimizar este inconveniente.

Un **prefiltrado masivo** en los propios sensores reduce el flujo de información que hay que transmitir hacia al componente central de procesado. Pero esta solución no siempre es posible, puesto que existen situaciones en las que se hace imposible decidir de forma local qué tipo de información es relevante para la detección.

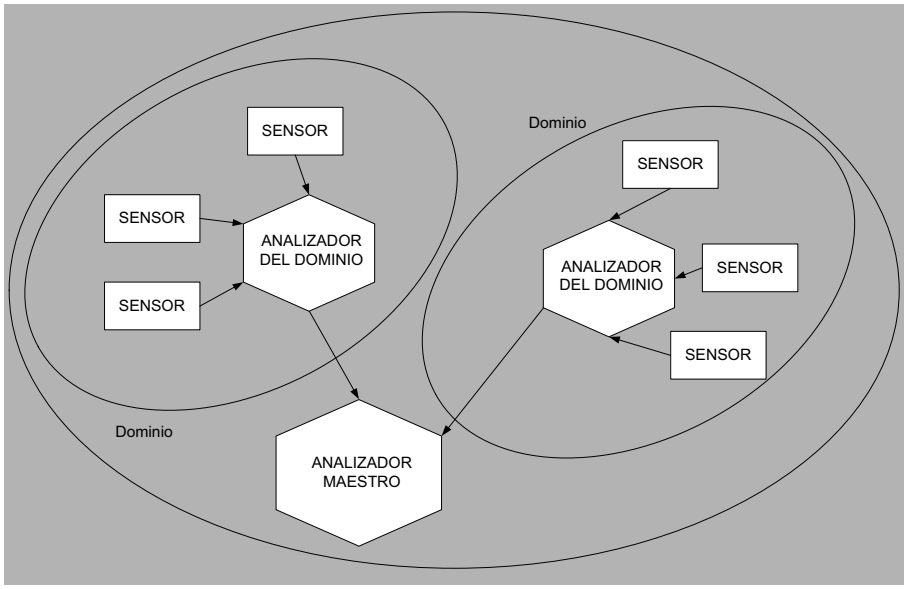
Los sistemas que siguen esta propuesta de prefiltrado masivo a nivel de sensor corren el riesgo de registrar altas tasas de falsos negativos, debido a la gran probabilidad de haber descartado información necesaria en el proceso de filtrado.

Para solucionar los cuellos de botella observados en los esquemas de correlación de eventos centralizada en redes de gran tamaño, es necesario plantearse nuevas propuestas. Pero encontrar un esquema de reducción de información eficiente, capaz de aislar únicamente información de relevancia en cualquier tipo de escenarios, es verdaderamente difícil.

Una primera forma de solucionar parcialmente este inconveniente consiste en la realización de una división del punto central de recogida de información en distintos puntos de recogida, organizados de forma jerárquica. De esta forma, tanto la carga en la red, al enviar todos los eventos a un único punto central, como la carga computacional, a causa de la existencia de un único punto de análisis, es distribuida sobre un conjunto intermedio de analizadores.

Así pues, esta segunda propuesta se basa en la utilización de nodos intermedios, dedicados a observar toda la información relevante de un área de detección pequeña y manejable. Únicamente aquella información considerada como relevante para la detección global será transmitida al nodo raíz.

Como vemos en la figura siguiente, los analizadores intermedios examinarán eventos en distintos dominios del conjunto global de la red, y enviarán sus resultados parciales a un nodo raíz, que tratará de realizar las inferencias necesarias.



Aunque esta solución mueve las decisiones de prefiltrado a un nivel superior, padece la misma problemática que la propuesta centralizada. Mientras que cada una de las áreas es monitorizada completamente, la correlación global de sus eventos puede producir una sobrecarga o una pérdida de información.

5.6.2. Análisis descentralizado

La recogida de eventos de forma distribuida crea una cantidad masiva de información que debe ser analizada, en la mayoría de las situaciones, bajo durísimas restricciones de tiempo real.

Dado que los diferentes fragmentos de información que podrían delatar un ataque distribuido se pueden encontrar en cualquier equipo de la red, es realmente complicado poder llegar a paralelizar este procesado de información.

Las dos posibles soluciones que hemos visto en el apartado anterior tratan de solventar esta dificultad de paralelización mediante el prefiltrado de información en los sensores del sistema y mediante la utilización de nodos intermedios.

Las soluciones tradicionales son vulnerables a errores o a ataques deliberados contra la infraestructura de procesado de eventos.

En el momento en que uno de los nodos centrales de procesamiento presente problemas, el sistema de detección se quedará ciego.

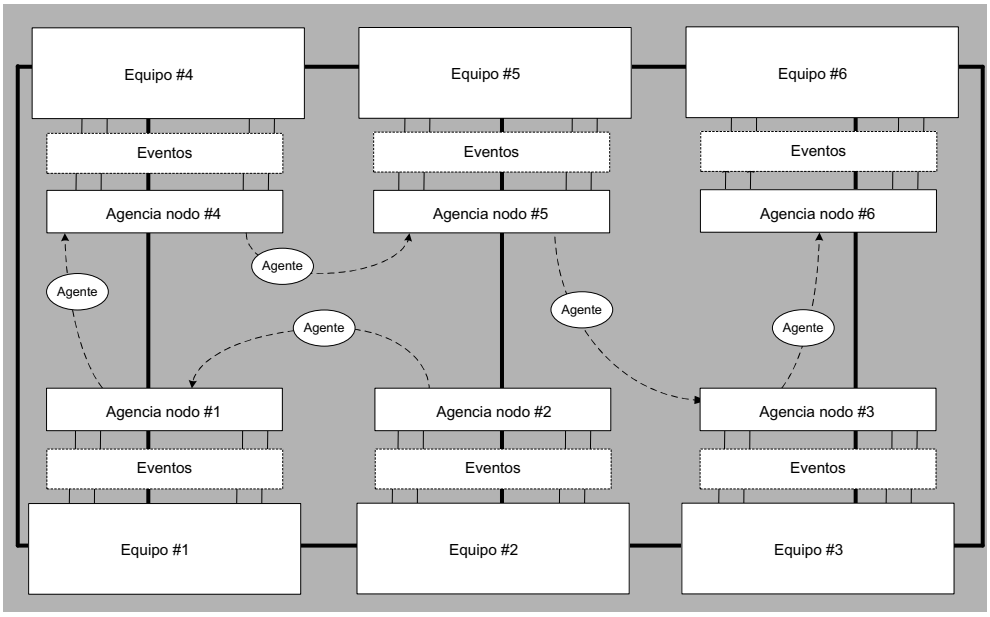
Con el objetivo de solucionar las dificultades inherentes a la recogida centralizada por parte de nodos de procesado dedicados, han aparecido a lo largo de los últimos años nuevas propuestas basadas en la realización de un análisis descentralizado de la información.

Aunque es realmente complicado identificar y analizar en paralelo distintos fragmentos de información, un algoritmo de detección descentralizado sería realmente efectivo para solventar la problemática planteada.

Dos de las propuestas existentes para implementar procesos descentralizados de análisis de información son, por un lado, la utilización de código móvil, y la utilización de nodos cooperativos que realizan un proceso descentralizado de análisis mediante mecanismos de paso de mensajes, por otro.

Análisis descentralizado mediante código móvil

Las propuestas basadas en código móvil para realizar una detección de ataques distribuidos utilizan el paradigma de agentes *software* para mover los motores de detección por la red que hay que vigilar (en forma de agente móvil). A medida que estos detectores móviles vayan recogiendo la información que les ofrezcan los sensores, los agentes irán realizando un proceso de análisis descentralizado.



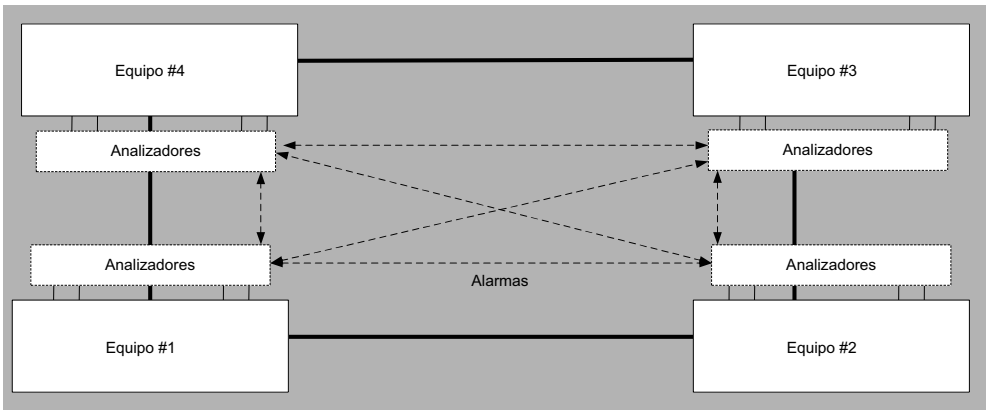
Los elementos de recogida de información (sensores) serán herramientas estáticas, es decir, se ejecutarán en los equipos en los que se produzca la recogida de información y se encargarán de hacerla llegar más tarde a los agentes de análisis de información que se mueven por la red.

Mediante el análisis descentralizado realizado por parte de los agentes *software* será posible realizar el proceso de correlación de eventos y la creación de estadísticas sin necesidad de elementos centrales.

Por otra parte, y a diferencia de los sistemas tradicionales que acabamos de ver, los agentes podrán moverse dinámicamente por el sistema, consiguiendo un mejor balance de la carga y la evasión de ataques potenciales contra sí mismos. Cuando las anomalías detectadas por los agentes móviles cubran un nivel de sospecha determinado, se podrán desplazar una serie de agentes reactivos que serán enviados hacia los equipos involucrados para neutralizar el ataque detectado.

Análisis descentralizado mediante paso de mensajes

Al igual que la propuesta anterior, este nuevo esquema trata de eliminar la necesidad de nodos centrales o intermediarios ofreciendo, en lugar de una o más estaciones de monitorización dedicadas (encargadas de recibir toda la información recogida), una serie de elementos de control encargados de realizar operaciones similares de forma descentralizada. Pero a diferencia del esquema basado en código móvil, estos nuevos elementos son estáticos y tan sólo necesitan una infraestructura común de paso de mensajes para realizar su proceso de detección descentralizado.



Tan pronto como una acción que puede desencadenar en ataque es detectada por uno de los elementos de control, ésta será comunicada al resto de elementos involucrados. Así, la información recogida por los sensores no será transmitida mediante difusión a todos los elementos de control, sino sólo a los elementos afectados o con información relacionada.

Resumen

El objetivo de este último módulo didáctico ha sido el de presentar toda una serie de elementos complementarios a los mecanismos de seguridad tradicionales. Estos elementos no deben ser vistos como una alternativa, sino como un complemento necesario para poder garantizar la seguridad de una red TCP/IP.

La gran cantidad de formas de abordar el problema de la detección de ataques e intrusiones ha dado lugar a numerosas y variadas propuestas y soluciones. La mayor parte de estas propuestas basan su capacidad de detección en la recogida de información desde una gran variedad de fuentes de auditoría de sistema, analizando posteriormente estos datos de distintas formas. Algunas consisten en comparar los datos recogidos con grandes bases de datos de firmas de ataques ya conocidos, otros tratan de encontrar problemas relacionados con usuarios autorizados que sobrepasan sus acciones permitidas en el sistema, o incluso mediante el análisis estadístico, buscando patrones que indiquen actividad anormal y que no se hayan tenido en cuenta en los pasos anteriores.

Existen también mecanismos de seguridad que tratan de mejorar el problema de la seguridad de una red desde un punto de vista mucho más activo. Tanto los mecanismos de protección de la información como los mecanismos de prevención y detección tradicionales son utilizados para proteger los recursos de la red, detectando deficiencias en la seguridad y reaccionando más tarde para solventar estos inconvenientes. Como novedad, estos nuevos elementos cambian las reglas del juego, ofreciendo la posibilidad de tomar la iniciativa utilizando técnicas de monitorización para registrar y analizar las acciones de los atacantes para aprender de sus conocimientos.

Una tercera categoría de elementos de detección que hemos visto trata de unir la capacidad de bloqueo de los mecanismos de prevención con la capacidades de análisis de los sistemas de detección. Conocidos como *sistemas de prevención de intrusos*, estos nuevos elementos son considerados como la evolución lógica de los sistemas de detección de intrusos tradicionales.

Por último, un caso de especial interés para los sistemas de detección son los ataques distribuidos. Estos ataques no se pueden detectar de forma aislada, sino que es necesario poner en correlación múltiples indicios encontrados en diferentes equipos de una red. Dos de las propuestas más utilizadas para poder construir sistemas capaces de detectar este tipo de ataques son la utilización de nodos dedicados (mediante una arquitectura centralizada o jerárquica) y la utilización de nodos distribuidos (mediante una arquitectura basada en código móvil o mediante la cooperación de nodos mediante un paso de mensajes).

Glosario

Escáner de vulnerabilidades: herramienta que permite comprobar si un sistema es vulnerable a un conjunto de problemas de seguridad.

***Exploit*:** aplicación, generalmente escrita en C o ensamblador, que fuerza las condiciones necesarias para aprovecharse de un error de programación que permite vulnerar su seguridad.

Explotación de un servicio: actividad realizada por un atacante para conseguir privilegios de administrador abusando de alguna deficiencia del sistema o de la red.

Ocultación de huellas: actividad ejecutada por un atacante (una vez producida la intrusión) para pasar desapercibido en el sistema.

Política de seguridad: resultado de documentar las expectativas de seguridad de una red, tratando de plasmar en el mundo real los conceptos abstractos de seguridad

***Rootkit*:** recopilación de herramientas utilizadas en un ataque de intrusión para garantizar la ocultación de huellas, garantizar futuras conexiones, realizar otros ataques al sistema, etc.

Seguridad perimetral: seguridad basada únicamente en la integración en la red de sistemas cortafuegos y otros mecanismos de prevención tradicionales.

Cortafuegos: elemento de prevención que realizará un control de acceso para separar la red de los equipos del exterior (potencialmente hostiles). En inglés, *firewall*.

Vigilancia de una red: actividad realizada por el atacante para tratar de aprender todo lo que pueda sobre la red que quiere atacar, especialmente servicios vulnerables y errores de configuración.

Bibliografía

[1] Cheswick, W. R.; Bellovin, S. M.; Rubin, A. D. (2003). *Firewalls and Internet Security: Repelling the Wily Hacker, 2nd ed.* Addison-Wesley Professional Computing.

[2] González, D. (2002). *Sistemas de Detección de Intrusiones.*

[3] Northcutt, S. (2000). *Network Intrusion Detection. An analyst's handbook.* New Riders.

[4] Proctor, P. E. (2001). *The practical intrusion detection handbook.* Prentice-Hall.

[5] Spitzner, L. (2001). *Honeypots: Tracking Hackers.* Addison-Wesley.