Análisis en torno a la auditoría de seguridad en tecnologías de la información y las comunicaciones

Por el Dr. Javier Areitio Bertolín

Prof. Dr. Javier Areitio Bertolín – E. Mail: jareitio@eside.deusto.es

Catedrático de la Facultad de Ingeniería. ESIDE.

Director del Grupo de Investigación Redes y Sistemas. Universidad de Deusto.

En el presente artículo se analiza un concepto muy amplio que abarca múltiples aspectos, se trata de la auditoria de la seguridad que se relaciona con los sistemas de detección de intrusiones (o IDS), con la valoración y análisis de vulnerabilidades, con los procesos forenses, etc. Permite básicamente determinar con precisión quién (persona o máquina) hizo qué, donde lo hizo y cuando lo llevó a cabo.

Introducción

Hoy en día se observa una importancia creciente en temas de auditoria en todo tipo de sistemas donde existan implantadas tecnologías de la información y comunicaciones. La seguridad de red no sólo es tecnología, es una combinación de muchas cosas y la auditoria trata de averiguar el estado de una red o sistema observando muchos frentes con indicios registrados.

Hoy en día se observa un incremento de adquisición de recursos en seguridad de la información por parte de las empresas debido a diversas causas como por ejemplo, el crecimiento del comercio electrónico, el impacto de las legislaciones sobre seguridad de la información, los riesgos de los negocios y requisitos de seguridad de la red de negocios. El entender y especificar el tipo de seguridad que necesita una organización es una tarea difícil. Para poder priorizar de forma racional los requisitos de seguridad se debe vincular los requisitos de seguridad (como confidencialidad, integridad, disponibilidad, etc.) a la visión de negocios de la empresa (los objetivos de negocios residen en el nivel estratégico y son mejora de eficiencia operacional, gestión de recursos financieros, cumplimiento con leyes y regulaciones, etc.) a través de los factores de impacto críticos (como pérdida financiera, pérdida de productividad, daño en la reputación, responsabilidad legal, etc.).

Este artículo se enmarca en las actividades desarrolladas dentro del proyecto LEFIS-APTICE: Legal Framework for the Information Society II (financiado por Socrates 2005. European Commission).

Concepto de auditoría. Usos. Problemas.

La auditoria consiste en la evaluación independiente de las actividades y registros del sistema con vistas a asegurar la adecuación de los controles del sistema, el cumplimiento con las políticas, procedimientos y estándares establecidos y recomendar los cambios necesarios en dichos controles. El *logging* registra los eventos o estadísticas para proporcionar información sobre el uso y rendimiento del sistema. La auditoria analiza los registros de *log* para presentar la información sobre el sistema de una manera clara y entendible.

La auditoria presenta diferentes usos:
(1) Describe el estado de seguridad. Determina si el sistema entra en un estado no autorizado.
(2) Evalúa la efectividad de los mecanismos de protección. Determina que mecanismos son apropiados y operan. Disuade de ataques debido a la presencia de un registro.

Se pueden identificar algunos problemas:
(1) ¿Qué se registra? Se buscan violaciones de una política, de modo que se registra al menos lo que demostraría tales violaciones.
(2) ¿Qué se audita? No se necesita auditar todo. Se debe auditar aquello en lo que esta implicada la política.

Estructura de un sistema de auditoria

Básicamente la arquitectura de un sistema de auditoría consta de tres elementos:
(1) *Logger o registrador*. Registra información, normalmente controlado por ciertos parámetros. Tipifica cantidad de información registrada controlada por los parámetros de configuración del sistema o programa. Puede ser leíble o no por las personas, en caso negativo normalmente se suministran herramientas de visualización. Son características a tener en cuenta el espacio disponible y la portabilidad influenciada por el formato de almacenamiento. Examinemos algunos ejemplos del elemento *logger*:

(i) Para el sistema operativo Windows NT. Los diferentes *logs* para los distintos tipos de eventos son:

(a) Logs de eventos del sistema. Registra las averías del sistema, los fallos de componentes y otros eventos del sistema.

(b) Logs de eventos de aplicación. Registra eventos que las aplicaciones solicitan ser registradas.

(c) Logs de eventos de seguridad. Registra eventos de seguridad críticos como entradas y salidas del sistema, el acceso a los ficheros del sistema y otros eventos. Los logs son binarios y se utiliza un *visor de eventos* para poder verlos. Si el *log* se llena, se puede, bien apagar el sistema, o bien inhabilitar el *logging* o bien sobre escribir los logs.

(ii) En RACF, un paquete de mejora de seguridad para los sistemas operativos MVS/VM de IBM. Registra los intentos de acceso fallidos, el uso de privilegios para cambiar niveles de seguridad y si se desea las interacciones RACF; los eventos se visualizan con los comandos LISTUSERS.

(2) *Analizador*. Analiza la información registrada en busca de algo. Se encarga de analizar uno o más registros o *logs*. Los *logs* pueden proceder de diversos sistemas o de un único sistema. Puede conducir a cambios en el *logging* o bien al informe de un evento. Examinemos algunos ejemplos del elemento analizador:

(i) Utilizar *swatch* se pueden encontrar instancias de *telnet* de los logs *tcpd*: /telnet/&!/localhost/&!/*.site.com

(ii) El motor de análisis de un sistema de detección de intrusiones recibe datos de los sensores y determina si ocurre una intrusión.

(iii) Establecimiento de control de solapamiento de interrogaciones o *queries* en bases de datos. Si existen demasiados solapamientos entre la interrogación corriente y *queries* pasadas no responder.
(3) *Notificador*. Informa los resultados del análisis. Puede reconfigurar el análisis y/o logging en base a los resultados obtenidos. Examinemos algunos ejemplos del elemento notificador: (i) Utilizar *swatch* para notificar de *telnets*: /telnet/&!/localhost/&!/*.site.com mail staff.
(ii) Tres login fallidos seguidos inhabilitan la cuenta de un usuario, el notificador inhabilita la cuenta y notifica al administrador del sistema.
(iii) Establecimiento de control de solapamiento de interrogaciones o *queries* en bases de datos. Previene que la respuesta se produzca si suceden demasiadas solapamientos.

Diseño de un sistema de auditoria. Cuestiones de implementación y sintácticas.

Un sistema de auditoria es un componente esencial de todo mecanismo de seguridad y sus objetivos determinan lo que se registra. Los auditores desean detectar violaciones de política, que proporciona un conjunto de restricciones que debe satisfacer el conjunto de posibles acciones.

Examinemos un ejemplo: En el modelo de seguridad BLP (Bell-LaPadula) la condición de seguridad simple y propiedad * es:
(1) El sujeto S lee el objeto O => autorización de S mayor o igual que la autorización de O (L(S) >= L(O)).
(2) El sujeto S escribe el objeto O => autorización de S menor o igual que la autorización de O (L(S) <= L(O)).
Para comprobar las violaciones en cada lectura y escritura se debe registrar L(S), L(O), acción (lectura, escritura) y el resultado (éxito, fallo). No se necesita registrar ni S ni O, en la práctica lo que se hace es identificar el objeto O del intento de violación y el usuario S que intenta la violación.

Algunas cuestiones de implementación son:
(1) Demostrar bien que no hay seguridad o bien encontrar violaciones, en el primer caso se requiere registrar el estado inicial así como los cambios.
(2) Definir las violaciones, por ejemplo escribir incluye *append* y crear directorio.
(3) Múltiples nombres para un objeto. El *logging* funciona por objeto y no por nombre. Las representaciones pueden afectar esto, si se lee discos en bruto se leen ficheros, ¿puede el sistema de auditoria determinar que fichero?

Algunas cuestiones sintácticas son:
(1) Los datos que se registran pueden ser ambiguos. Por ejemplo en BSM (Basic Security Module, mejora la seguridad del SunOS de Solaris) se utilizan dos campos de texto opcionales seguidos por dos campos de texto obligatorios; si existen tres campos, ¿cual de los campos opcionales se ha omitido? La solución es utilizar una gramática para asegurar la sintaxis bien definida de los ficheros de log. El formato de las entradas del fichero de log debe definirse de forma no ambigua, el mecanismo de auditoria debe explorar e interpretar las entradas sin confusión.
(2) Contexto. Supongamos que un usuario desconocido utiliza el ftp anónimo para recuperar el fichero /etc/passwd. El problema que se plantea es qué fichero /etc/passwd, uno del directorio system/etc o un directorio ftp anónimo /var/ftp/etc y cuando ftp piensa en /var/ftp es el directorio raíz, etc/passwd se refiere a /var/ftp/etc/passwd.

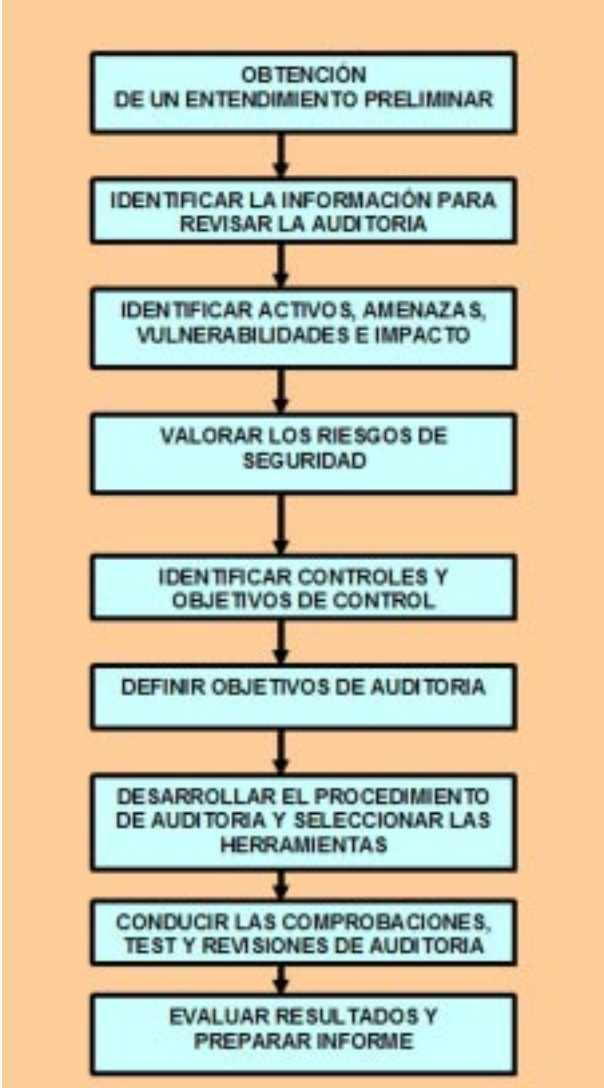

Figura 1. Proceso de Auditoría de seguridad en la información

Ocultación de log. Organización. Reconstrucción. Generación de pseudónimos.

Veamos en que consiste la ocultación de log, sea U el conjunto de usuarios y P la política que define el conjunto de información C(U) que los usuarios de U no pueden ver. La ocultación de log se da cuando toda la información de C(U) se hace invisible del log. Existen dos tipos de P:
(1) C(U) no puede dejar el sitio. Las personas dentro del sitio son con-

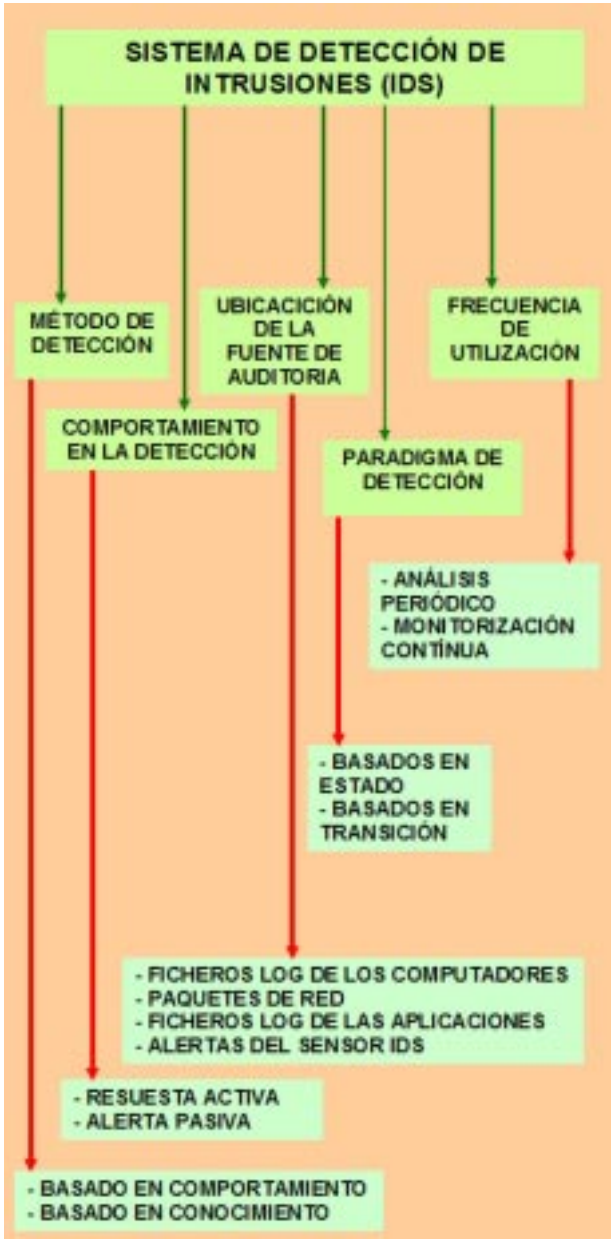

Figura 2. Clasificación de los IDS en relación a los sistemas de Auditoría

fiables y la información no es sensible a ellas.
(2) C(U) no puede dejar el sistema. Las personas de dentro del sitio no son confiables o más comúnmente la información es sensible a ellas. No se registra esta información sensible.

Existen dos esquemas de organización del *looging*:

(1) La salida del sistema de *logging* se pasa directamente al log que se hace pasar por un proceso de ocultación y finalmente se presenta a los usuarios. Este enfoque previene que la información deje el sitio. La privacidad de los usuarios no se protege de los administradores del sistema y de otro personal administrativo.

(2) La salida del sistema de *logging* se pasa por un proceso de ocultación antes de pasar al log que por último se presenta a los usuarios. Este enfoque previene que la información deje el sistema, los datos se protegen por ocultación antes de ser registrados.

El proceso de reconstrucción puede ser de dos tipos:

(1) El ocultador por anonimato no puede anularse y no hay forma de recuperar los datos.

(2) El ocultador por pseudónimo puede anularse y el log original puede reconstruirse. La reconstrucción es importante ya que el análisis de seguridad requiere acceso a la información que se ocultó. La ocultación de log debe preservar las propiedades para el análisis de la seguridad. Si se añaden nuevas propiedades debido a que cambie el análisis se puede tener que re-ocultar la información, esto requiere de la ocultación por pseudónimo o el log original.

Examinemos un ejemplo, una empresa desea que su espacio de direcciones IP sea secreto pero quiere que un consultor analice los log en busca de un ataque de exploración de direcciones, por ejemplo conexiones al puerto 25 en las direcciones IP:

185.67.23.12,185.67.23.13,
185.67.23.14, 185.67.23.15,
185.67.23.16, 185.67.23.17.

Para ello se oculta bien con direcciones IP aleatorias o con direcciones IP secuenciales, en este último caso se puede ver el recorrido con direcciones IP consecutivas.

La generación de pseudónimos puede realizarse de dos formas:

(1) Se idea un conjunto de pseudónimos para reemplazar la información sensible, se sustituye los datos por pseudónimos y se mantiene una tabla de correspondencias entre pseudónimos y datos.
(2) Se utiliza una clave aleatoria para cifrar los datos sensibles y se utiliza el esquema de compartición de secretos tipo Shamir para compartir la clave (se requiere que k de las N personas para poder leer los datos). Se utiliza cuando el personal de dentro no puede ver los datos ocultos, pero lo necesitan los de fuera por ejemplo ordenado por un juez.

Tipos de logging

El *logging* puede clasificarse atendiendo a diferentes criterios, por ejemplo atendiendo a qué entidad lo realiza, podemos diferenciar:

(1) *Logging* de aplicación. La realizan las aplicaciones. Las aplicaciones controlan cual es el logging. Normalmente se utilizan abstracciones de alto nivel como su: urrutia to root on /dev/ttyp0. No incluye información de nivel de llamada al sistema como resultados, parámetros, etc. El logging de aplicación se en foca en los eventos de la aplicación, como fallos al introducir la contraseña adecuada y causas subyacentes como, ¿cuál fue la razón del intento de acceso?.

(2) *Logging* del sistema. Registra eventos del sistema tales como acciones del kernel (un evento del kernel es abrir un fichero). No incluye abstracciones de alto nivel como carga de librerías. El *logging* del sistema se enfoca en eventos del sistema como mapping de memoria o accesos a ficheros y causas subyacentes ¿por qué falló el acceso?. Los logs del sistema son normalmente mucho mayores que los log de aplicaciones. Se puede realizar ambos y tratar de correlacionarlos.

Objetivos de la auditoria. Formas de auditorias: basadas en estado y basadas en transición.

Los objetivos de la auditoria son:

(1) Detectar cualquier violación de una política conocida. Se trata de preguntarse si el sistema entra o no en un estado no permitido. El enfoque se centra en la política y las acciones diseñadas para violar la política, las acciones específicas pueden no ser conocidas.

(2) Detectar las acciones conocidas que son parte de un intento de hacer brecha en la seguridad. Se enfoca en acciones específicas que han sido determinadas para indicar ataques.

Pueden identificarse dos formas de auditoria en torno a la detección de violaciones de política conocida:

(1) Auditoria basada en estado. Se observa el estado actual del sistema. Se registra información sobre el estado y se determina si el estado esta permitido. La suposición es que se puede obtener una instantánea del estado del sistema. La instantánea necesita ser consistente. Los sistemas no distribuidos necesitan ser inactivos. Los sistemas distribuidos pueden utilizar el algoritmo Chandy-Lamport para conseguirlo. Examinemos un ejemplo, las herramientas de auditoria del sistema de ficheros.

(2) Auditoria basada en transición. Se miran las acciones que posibilitan la transición del sistema de un estado a otro. Se registra información sobre las acciones y se examina el estado corriente y la transición propuesta para determinar si el nuevo estado debería no permitirse. Sólo el analizar la transición puede no ser suficiente y se puede necesitar el estado inicial. Se tiende a utilizar esto cuando las transiciones específicas requieren siempre análisis, por ejemplo cambio de privilegios. Examinemos un ejemplo, el mecanismo de control de acceso TCP intercepta conexiones TCP y comprueba con una lista de conexiones a bloquear, obtiene la dirección IP del origen de la conexión, registra la dirección IP, puerto y resultado (permitido o bloquedo) en un fichero log. Se basa en transición, analiza el estado actual no todos.

Test y estudios de penetración. Niveles de test.

Los defectos de seguridad y vulnerabilidades son fallos de políticas, procedimientos y controles de seguridad que permiten que un sujeto cometa una acción que viola la política de seguridad. El sujeto se denomina atacante y utiliza el fallo para violar la política explotando o forzando la vulnerabilidad. El test de penetración utilizado en auditoria puede probar la presencia de vulnerabilidades, pero no la ausencia de ellas. Los estudios de penetración son test para evaluar la fuerza y efectividad de todos los controles de seguridad del sistema, se denominan también *ataques tiger team o ataques red team*. El objetivo es violar la política de seguridad del sitio, intentar violar limitaciones específicas en la seguridad, encontrar algún número de vulnerabilidades dentro de un período de tiempo.

Los test se pueden estructurar en niveles:

(1) Ataque externo sin conocimiento del sistema, localiza el sistema, aprende lo suficiente para poder accederlo.

(2) Ataque externo con acceso al sistema, puede hacer *log in* y acceder a los servidores de red, a menudo intenta ampliar el nivel de acceso.

(3) Ataque interno con acceso al sistema, los verificadores o tester son usuarios autorizados con cuentas restringidas, son como usuarios ordinarios, el objetivo es obtener privilegios o información no autorizada. La utilidad del estudio de penetración se basa en la documentación, las conclusiones indican si los defectos son o no endémicos, no proceden del éxito o fallo en la penetración intentada.

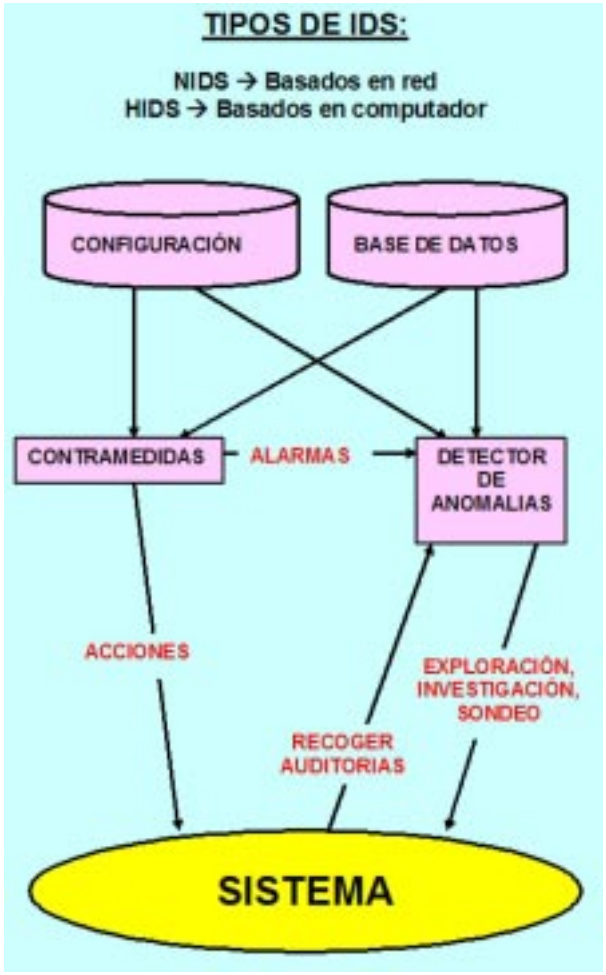

Figura 3. Arquitectura de bloques de un IDS y su vinculación con los sistemas automáticos de Auditoría

Una posible metodología consta de las siguientes etapas:

(1) Recoger información del funcionamiento del sistema. Idear el modelo del sistema y/o componentes, buscar discrepancias en los componentes, considerar los interfaces entre componentes, diseñar documentos, observar como el sistema gestiona los usuarios con privilegios.

(2) Establecer hipótesis en cuanto a vulnerabilidades. Examinar políticas y procedimientos, pueden ser inconsistencias para explotar, examinar implementaciones, utilizar modelos de vulnerabilidades para ayudar a localizar problemas potenciales, utilizar manuales, intentar exceder límites y

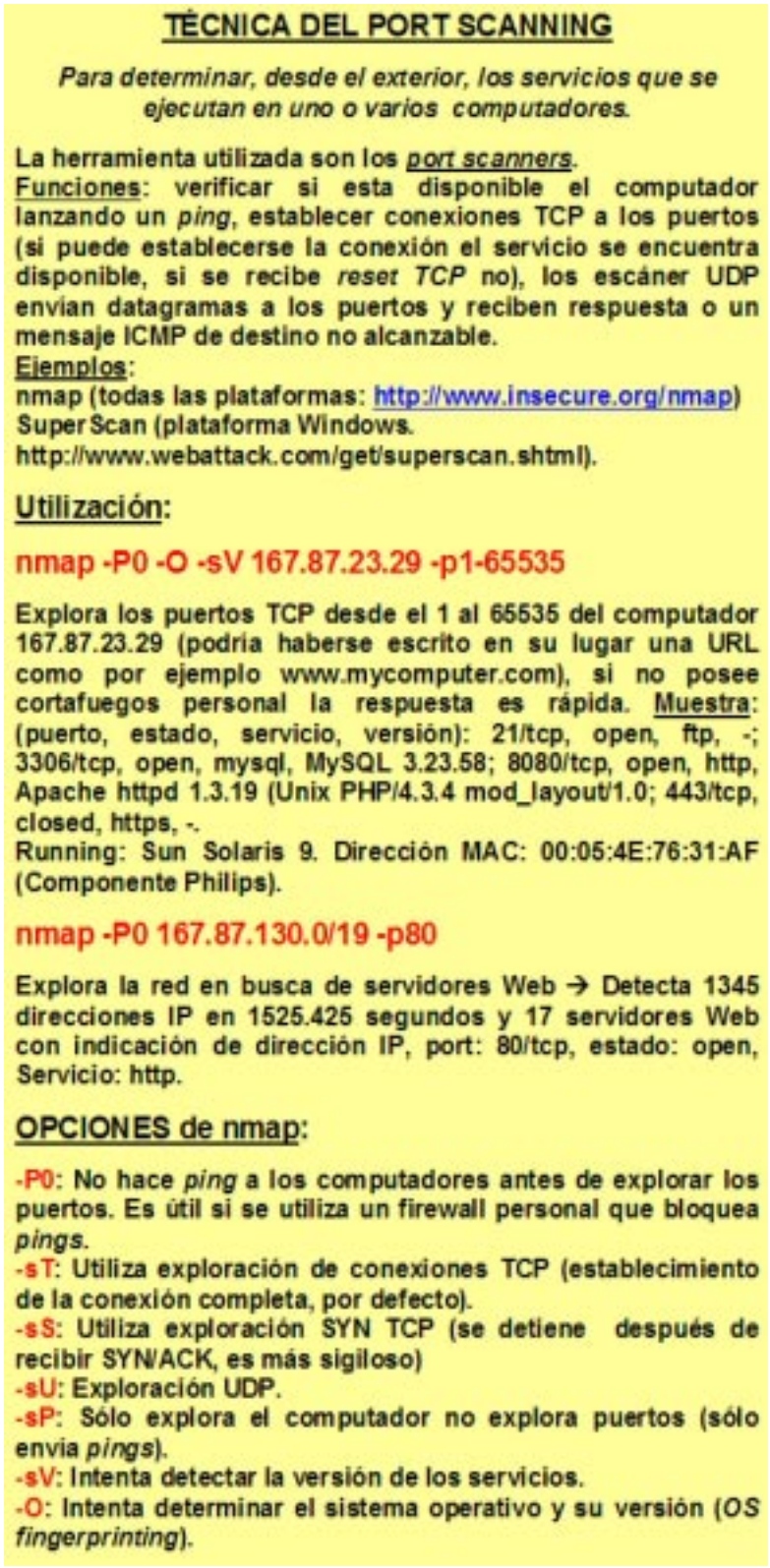

Figura 4. Técnica de exploración de puertos (port scanning) utilizado en los test de penetración

restricciones, intentar omitir pasos en procedimientos. Identificar estructuras, mecanismos que controlan el sistema, son los que utilizarán los atacantes.
(3) Testear vulnerabilidades. Encontrar los principales problemas de diseño e implementación enfocándose en vulnerabilidades críticas de sistema potenciales, encontrar vulnerabilidades a atacantes externos, enfocándose en programas y protocolos de acceso.
(4) Determinar la posibilidad de que se combinen las vulnerabilidades.
(5) Eliminación, de forma optativa, las vulnerabilidades encontradas.

Política de seguridad. Propiedades de confidencialidad, integridad y disponibilidad

La política de seguridad describe lo que esta permitido y el mecanismo de seguridad la forma en que se ejecuta la política, es decir un mecanismo de seguridad es una entidad o procedimiento que hace cumplir alguna parte de la política de seguridad (por ejemplo, impedir a los usuarios que accedan con *pen-drives* a un centro de cálculo ya que podrían copiar información sensible o descargar software no autorizado. Un mecanismo de control de acceso discrecional o DAC es aquel en el que el usuario permite o deniega el acceso a los objetos. Un mecanismo de control de acceso mandatario o MAC es aquel en el que el sistema controla el acceso a los objetos y los individuos no pueden alterar dicho acceso. Un mecanismo de control de acceso controlado por el originador u ORCOM es aquel en el que el creador de la información es quien puede acceder a la información. La confianza se encuentra debajo de todo.

La política de seguridad particiona los estados del sistema en:
(1) Estados autorizados (seguros). Son estados donde puede entrar el sistema.

(2) Estados no autorizados (no seguros). Si el sistema entra en cualquiera de estos estados se produce una violación de seguridad. Un sistema se dice que es seguro si comienza en un estado seguro y nunca entra es estados no autorizados. Dado un conjunto X de entidades y una información I. Se dice que I posee la propiedad de confidencialidad con respecto a X si ningún elemento x de X puede obtener información de I. Así mismo, I puede ser revelado a otros, por ejemplo X es el conjunto de alumnos de un curso de una academia, I es la clave de respuestas del examen final. I es confidencial con respecto a X si los estudiantes no pueden obtener la clave de respuestas del examen final.

Se dice que I tiene la propiedad de integridad con respecto a X si todos los elementos de X confían en la información I. Pueden identificarse diferentes tipos de integridad:
(a) Integridad de datos. Confianza en I, en su transporte y protección.
(b) Integridad de origen / autenticación. Información I tiene un origen e identidad segura.
(c) Aseguramiento del recurso de I. Significa que el recurso funciona como debería.

Se dice que I posee la propiedad de disponibilidad con respecto a X si todos los elementos x de X pueden acceder a I. Se pueden identificar diversos tipos de disponibilidad:
(a) Tradicional. X obtiene o no acceso.
(b) Calidad de servicio. Promete un nivel de acceso, por ejemplo, un nivel específico de ancho de banda.

Modelos y tipos de políticas de seguridad

Un modelo de seguridad es una descripción abstracta de una política o clase de políticas.

Los puntos de interés en políticas son: *los niveles de seguridad* en modelos de seguridad multi-nivel como BLP (Bel LaPadula), *la separación de obligaciones* en el modelo Clark-Wilson, *el conflicto de intereses* en el modelo Chinese Wall. En la política CISS se combina integridad y confidencialidad. Se pueden identificar cuatro tipos de políticas de seguridad:
(1) Política de seguridad militar o gubernamental, la política protege primariamente la confidencialidad.
(2) Política de seguridad comercial, la política protege primariamente la integridad.
(3) Política de confidencialidad, la política protege sólo la confidencialidad.
(4) Política de Integridad, la política protege sólo la integridad.

Analicemos la integridad y las transacciones, se comienza en un estado consistente (consistente significa definido por una especificación), se realiza una serie de acciones (denominada transacción) de modo que las acciones no se pueden interrumpir, si se completan las acciones el sistema esta en un estado consistente, si las acciones no se completan el sistema regresa al estado de comienzo consistente. Examinemos la confianza en torno al hecho de que un administrador instala un parche de seguridad (por ejemplo en un sistema operativo como Windows):
(1) Confianza de que el parche viene del fabricante (por ejemplo Microsoft) y no ha sido modificado en tránsito.
(2) Confianza en el fabricante que verificó el parche a fondo.
(3) Confianza en el entorno de verificación del fabricante que corresponde a un entorno local. (4) Confianza en que el parche se instale correctamente.

Consideraciones finales

Actualmente nuestro grupo de investigación trabaja en el contexto de la auditoria y su relación con los sistemas de detección y prevención de intrusiones, identifica entre otros diferentes componentes como:
(1) Software de *logging,* aplicaciones que recojan eventos de log con información detallada.
(2) Normalizadores, agentes que transformen los log existentes para que puedan ser incorporados en un esquema común de sistema de auditoría.
(3) Subsistemas de recogida, los logs de auditoria se recogen por un sistema de recogida segura.
(4) Repositorio de base de datos, los logs de auditoria se envían a través de la red normalizados, para ser archivados, están disponibles para ser interrogados por una interfaz de usuario.
(5) Herramientas forenses, permiten pedir y procesar los datos de auditoría para encontrar problemas y responder a preguntas como: cual es la lista de sitios Web que el usuario A accedió la semana pasada, o la lista de ficheros de datos abiertos por el usuario B la pasada semana, o la lista de usuarios que utilizaron la cuenta compartida ZZ sobre el recurso K durante el día de ayer, o si puede monitorizar el IDS tráfico sobre puertos que supuestamente se encuentran cerrados por el servicio de firewall corporativo.

Bibliografía

-Areitio, J. "Tipificación de amenazas, identificación de contramedidas de seguridad en el ámbito de redes y sistemas". REE Nº 613. Dic.. 2005.
-Areitio, J. "Análisis en torno a la seguridad de los cortafuegos para la seguridad de red perimetral". REE . Nº 592. Marzo 2004.
-Areitio, J. "Identificación y análisis del control de acceso para la seguridad de TIC". Revista Española de Electrónica. Nº 591. Febrero 2004.
-Douligeris, C. and Serpanos, D.N. "Network Security: Current Status and Future Directions. Wiley- IEEE Press. 2005.
-Cole, E., Krutz, R.L. and Conley, J. "Network Security Bible". John Wiley & Sons. 2005. ❑