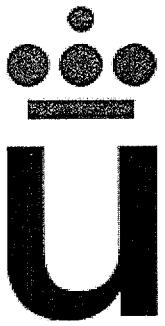


Universidad
Rey Juan Carlos

INGENIERÍA INFORMÁTICA

Curso Académico 2004/2005

Proyecto de Fin de Carrera

EVALUACIÓN DE SEGURIDAD EN SISTEMAS INFORMÁTICOS

Autor: Lucas Nieto Rodríguez

Tutor: Antonio Guzmán Sacristán

AGRADECIMIENTOS

A mis padres y hermana.

A Marta y Antonio por toda la ayuda prestada.

A mis compañeros de proyecto, Alberto, Carlos y Héctor.

A todos los compañeros y amigos que a lo largo de estos seis años han formado parte en alguna ocasión de mi vida.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

CAPÍTULO 1: Introducción

1.1 - Introducción

Uno de los aspectos de la Seguridad Informática que debe ser mejorado es la escasa existencia de productos para realizar auditorías lógicas completas en empresas, universidades, organizaciones, etc.

En la actualidad existen abundantes programas para realizar auditorías, pero cada uno de ellos se centra en realizar la auditoría sobre una parte concreta del sistema, por lo que para realizar una auditoría completa hay que utilizar varias herramientas por separado. Esto lleva a que la información obtenida mediante las auditorías realizadas al sistema esté en diferentes formatos, por tanto es más difícil y costoso obtener una valoración global de la seguridad del sistema.

También existen varios programas (Internet Scanner [25], Retina, NetRecon [26], SAINT [27], Nessus [28]) que realizan estudios de seguridad a los sistemas informáticos, pero sólo se preocupan de escanear las vulnerabilidades que puedan tener, como por ejemplo escaneos de puertos, búsqueda de agujeros de seguridad frente a determinados ataques, búsqueda de contraseñas débiles, etc., pero no realizan auditorías lógicas completas por lo que hay que utilizar otras herramientas para completar sus funcionalidades.

Por todo esto es interesante proponer una solución para poder realizar auditorías lógicas completas a organizaciones de forma fácil y sencilla. De esta forma las organizaciones podrán saber el nivel de seguridad que tiene su sistema informático y podrán tomar las medidas más adecuadas para solucionar posibles agujeros de seguridad o reforzar la arquitectura del sistema de seguridad implementada.

Todo esto nos lleva a realizar un estudio sobre la seguridad en los sistemas informáticos y a analizar las diferentes maneras de cuantificar su nivel de seguridad.

Particularmente este proyecto se centra en realizar un estudio sobre los diferentes métodos y tipos de herramientas relacionadas con la auditoría en sistemas informáticos y en implementar una aplicación que sirva de base para aplicaciones de auditoría de seguridad más avanzadas y completas.

Para crear esta aplicación es necesario estudiar las diferentes aplicaciones existentes y los principios básicos de las mismas así como los de la seguridad informática.

1.2 - Entorno de trabajo

El entorno ideal para este proyecto se basa en la creación de una empresa de seguridad ficticia dedicada a la realización de auditorías de seguridad a otras empresas. Para crear esta empresa se presentan una serie de problemas que serán divididos en cuatro partes o proyectos individuales. Cada uno de estos proyectos es llevado y realizado por una persona diferente. El objetivo de cada proyecto individual es implementar una parte fundamental de la infraestructura de la empresa ficticia desde el punto de vista de la seguridad informática, como son la arquitectura de red, el servidor Web con la aplicación Web, el servidor de la base de datos con la base de datos y la herramienta para realizar las auditorías de seguridad.

Aunque cada proyecto podría realizarse sin tener en cuenta a los demás proyectos, sí es necesario para realizar el proyecto global ponerse de acuerdo en algunos aspectos de la implementación, para que las cuatro implementaciones puedan comunicarse entre ellas y trabajar en un entorno común.

El entorno común de trabajo estará formado principalmente por un servidor de bases de datos en donde se almacenarán la base de datos con los datos que necesiten cada una de las aplicaciones implementadas. Otra máquina realizará la función de servidor Web, y se encargará de contener la aplicación Web necesaria para que los clientes puedan acceder a los servicios de auditoría que ofrece la empresa. También son necesarias una o más máquinas que hagan la función de cortafuegos o "firewall", esas máquinas se encargarán de controlar el flujo de datos de la red. Por último habrá otras máquinas para poder implementar las aplicaciones y realizar pruebas independientemente de las máquinas principales.

La red estará conectada a la red interna de la universidad y a Internet, aunque en los procesos de implementación de los proyectos y de pruebas de los mismos se utilizará una red sin conexión al exterior de la sala para así evitar que las posibles pruebas puedan afectar al resto de redes externas.

1.3 - Metodología

A la hora de afrontar un proyecto de estas características lo primero que hay que hacer es buscar información y adquirir los conocimientos básicos necesarios, centrados en la Seguridad Informática, para llevar a cabo el proyecto con garantía.
Principalmente hay que conocer el estado actual en el que se encuentra la seguridad en el ámbito de la informática: cuáles son las necesidades, cuáles son las técnicas utilizadas, qué tipo de amenazas nos podemos encontrar hoy en día, así como el estudio de los posibles ataques que se puedan realizar sobre el sistema de nuestra empresa y sus posibles soluciones o contramedidas.

Después de tener una idea general de cómo está la situación en la Seguridad Informática hay que plantearse qué estructura se va implantar en nuestra empresa, tanto a nivel físico como a nivel lógico. Para poder realizar esta planificación es necesario tener claros los objetivos a alcanzar y los servicios que va a ofrecer nuestra empresa.
Cuando los objetivos a llevar a cabo están claros es el momento de planificar la metodología de trabajo. Esta metodología de trabajo se ha dividido en cuatro partes, o proyectos individuales:

- Diseño e Implantación de Arquitecturas Seguras: Este proyecto se encarga de planificar e implementar la arquitectura del sistema y de la red de la empresa. Para ello hay que tener en cuenta las necesidades de la empresa y sobre todo la seguridad de la empresa a nivel de red.
  Es necesario hacer un intenso estudio de las amenazas y posibles ataques que se pueden realizar, a la arquitectura planteada, en cualquiera de los cuatro niveles del modelo OSI (Interconexión de Sistemas Abiertos). También el correspondiente estudio de cómo evitar y defenderse de estos ataques y qué estructuras sería necesario implementar para mantener segura la arquitectura de la empresa.

- Fortificación de Servidores: En este proyecto se encarga de implementar el servidor Web y de correo electrónico de la empresa. También diseñar e implementar la aplicación Web que representará a la empresa y que permitirá el acceso a la herramienta de auditoría a las empresas cliente.
  Todo esto debe tener una seguridad adecuada, por lo que hay que realizar un estudio de los posibles ataques a servidores que hay en la actualidad y sus posibles contramedidas e implantar las medidas de seguridad adecuadas tanto en la aplicación Web como en el servidor.

- Metodología para la fortificación de bases de datos: Este proyecto se centra en el diseño e implementación de la base de datos de la empresa teniendo en cuenta las necesidades de los demás proyectos. Al igual que en los dos proyectos anteriores se tendrá que realizar un estudio de los diferentes ataques a bases de datos existentes y de las mediadas necesarias para evitarlos.

- Evaluación de Seguridad en Sistemas Informáticos: El último proyecto consiste en el desarrollo de la aplicación utilizada para hacer las auditorías de seguridad que ofrecerá la empresa. Se utilizará la base de datos de ataques implementada en el proyecto de "Metodología para la fortificación de bases de datos" para obtener y guardar información referente a las auditorías. La aplicación será accesible vía Web, mediante el servidor fortificado creado en el proyecto Fortificación de Servidores. Además será necesario documentarse sobre las diferentes métricas de seguridad existentes y diseñar una propia para medir cuantitativamente la seguridad del sistema de las empresas cliente.

Estos cuatro proyectos tienen puntos en común que habrá que discutir para llegar a un acuerdo, y así adaptar cada proyecto a las necesidades particulares de los demás proyectos y del proyecto global.

En este proyecto individual dedicado a la evaluación de la seguridad en sistemas informáticos la metodología planteada es la siguiente:

1- Estudiar los principios básicos de la Seguridad Informática y recolectar información sobre los diferentes sistemas, métodos, técnicas y métricas dedicados a la evaluación de seguridad, detección de intrusos y defensa de ataques en los sistemas informáticos.

2- Evaluar cuáles son los mejores métodos, técnicas y métricas y proponer una metodología completa para evaluar la seguridad de un sistema informático.

3- Implementar una herramienta que sirva de base para futuras herramientas de auditoría más avanzadas y completas.

4- Realizar pruebas para validar su funcionamiento.

5- Documentar la métrica y la herramienta.

1.4 - Objetivos globales

Teniendo en cuenta los cuatro proyectos descritos anteriormente el objetivo global es hacer un estudio exhaustivo sobre la seguridad de los sistemas informáticos.

Otros objetivos más concretos del proyecto global son:

- Estudiar las diferentes técnicas de ataque que pueden sufrir las empresas.
- Estudiar las técnicas de detección de intrusiones y de análisis de los sistemas informáticos.
- Estudiar las contramedidas y las técnicas de defensa existentes en la actualidad para poder evitar las intrusiones.
- Estudiar las diferentes herramientas existentes dentro del campo de la seguridad informática en la actualidad tanto para ataque, como para defensa y auditoría y detección.

1.5 - Objetivos particulares del proyecto

Los objetivos del proyecto "Evaluación de Seguridad en Sistemas Informáticos" son:

- Estudiar los diferentes sistemas de detección de intrusos y los tipos de servicios de auditorías que hay en el mercado.
- Estudiar las métricas de seguridad existentes en la actualidad y la definición de una métrica de seguridad que permita certificar el grado de seguridad del sistema informático de una organización de forma cuantitativa.
- Desarrollar una herramienta para realizar las auditorías de seguridad a los sistemas de las organizaciones y obtener el valor de esta métrica.
- Integrar en esta herramienta los procedimientos necesarios para determinar la seguridad a diferentes niveles de abstracción en los sistemas utilizando para ello tanto técnicas pasivas como activas.

1.6 - Estructura del documento

El documento se divide en seis capítulos.

En el segundo capítulo se explica qué es la seguridad informática y en qué consiste, exponiendo conceptos básicos como los tipos de amenazas y sus posibles soluciones, la criptografía y los tipos de atacantes y de ataques.

En el capítulo tres se habla de técnicas de defensa y detección centrándose en las herramientas y técnicas que más útiles resultan para realizar auditorías de seguridad, como son los detectores de intrusos y los escáneres de vulnerabilidades. También se realiza un estudio exhaustivo sobre las diferentes métricas de seguridad utilizadas en la actualidad, tanto cualitativas como cuantitativas.

En el cuarto capítulo se explica detalladamente la aplicación implementada y las partes que la componen, tanto a nivel de desarrollo como a nivel de funcionamiento.

Por último, en el capítulo cinco se ofrecen las conclusiones finales y se habla de los trabajos futuros.

CAPÍTULO 2: Estado del Arte

Desde siempre la seguridad de la información ha sido para la humanidad un foco de preocupaciones. En la actualidad esta preocupación va aumentando cada día. Esto es debido a que todo está controlado por computadoras (cuentas bancarias, satélites, automóviles, el sistema judicial y sanitario, etc.) por lo que la informática es utilizada continuamente y los datos sensibles, como los personales, son tratados y manipulados diariamente.

Sin los mecanismos de seguridad adecuados en el ámbito de la informática el derecho constitucional a la intimidad del individuo no está garantizado y tampoco se puede tener la certeza de que este derecho no haya sido quebrantado.

La principal fuente de ataques a la intimidad está en los sistemas de comunicación electrónicos y sobre todo en Internet. La mayoría de usuarios de Internet no son conscientes del riesgo que se asume al enviar datos personales a través de las redes o simplemente de la cantidad de información privada que pueden obtener y usar usuarios con intenciones maliciosas por el simple hecho de navegar o visitar páginas Web de poca confianza.

Por esto es necesario tener, por parte de los usuarios, unas nociones básicas sobre qué es y en qué consiste la seguridad informática y por parte de las organizaciones realizar estudios de seguridad informática e implantar las medidas de protección adecuadas.

2.1 - Conceptos básicos de Seguridad Informática

2.1.1 - Definición de Seguridad Informática

La seguridad se define como la certeza, garantía de que algo va a cumplirse.

Si se traslada esta definición al ámbito de la informática se podría decir que la seguridad es la certeza o garantía de que los sistemas informáticos hagan lo que tienen que hacer o que funcionen correctamente. Este funcionamiento correcto debería incluir hoy en día confidencialidad, autentificación, integridad y disponibilidad.

Si se tiene en cuenta que no es posible la certeza absoluta, el elemento de riesgo está siempre presente independientemente de las medidas de seguridad que tomemos, por lo que debemos hablar de niveles de seguridad. Como nunca se podrá alcanzar la seguridad absoluta lo importante es alcanzar unos niveles altos de seguridad en los sistemas informáticos. Estos niveles se alcanzan mediante un conjunto de técnicas y mediante la planificación u organización de las mismas.

En la seguridad informática hay que tener en cuenta que aunque todos los componentes hardware y software están expuestos a ataques, son la información y los datos los principales objetivos a proteger por las técnicas de seguridad.

Teniendo en cuenta todo lo expuesto anteriormente la seguridad informática se puede definir como la disciplina que vela por el cumplimiento de los siguientes principios: confidencialidad, integridad y disponibilidad.

- Confidencialidad: Requiere que la información sea accesible únicamente por las entidades autorizadas. Existen varias técnicas para conseguir confidencialidad, una de ellas es el cifrado. También existen multitud de ataques para romper la privacidad, especialmente la de los datos. La mayoría de estos ataques se basan en interceptar la información que se envía por la red o la intrusión directa en los sistemas donde se almacena la información.([1], [2])

- Integridad: Se refiere a la seguridad de que la información no haya sido modificada por entidades no autorizadas durante la transmisión de la misma o en el propio equipo donde se encuentra. La modificación incluye cualquier operación posible sobre la misma como borrado, copia, escritura, creación, etc. La integridad de los datos asegura que los datos recibidos no han sido modificados, esto se puede conseguir mediante, por ejemplo, un "hash" criptográfico con firma. También es necesario mantener la integridad de la secuencia de datos para asegurar que la información no se repita o se pierda y que la secuencia de bloques de datos recibidos no haya sido alterada. Normalmente cuando un atacante intenta obtener una información y no lo consigue al tener técnicas de confidencialidad implantadas lo mas normal es que intercepte la información o la borre por esto también es importante tener medidas de integridad. ([1], [2])

- Disponibilidad: Requiere que la información o los recursos informáticos del sistema estén accesibles en todo momento por las entidades autorizadas. Esto implica evitar la denegación de servicio (bloqueos o pérdidas debido a ataques, malas gestiones o situaciones de fuerza mayor). ([1], [2])

Otros servicios importantes que deben ser tenidos en cuenta para la seguridad de los sistemas son:

- La autenticación: Se refiere a la prevención de suplantaciones, se basa en requerir una identificación correcta que asegure que el individuo o la entidad que haya enviado el mensaje o requerido un servicio sea quien dice ser. La técnica mas utilizada para proporcionar autentificación es la firma digital, también es utilizada la biométrica pero en menor medida. ([1], [2])

- El no repudio: Se refiere a la protección de un individuo o entidad frente a la negación de otro individuo o entidad de haber enviado o recibido una determinada información o haber realizado una cierta comunicación. Existen dos clases: el no repudio de origen que protege al receptor de la negación del emisor de no haber enviado el mensaje; y el no repudio de recepción que

protege al emisor de la negación del receptor de no haber recibido el mensaje. Esta protección se realiza mediante mecanismos que evidencian que la comunicación fue realizada, uno de ellos es la firma digital. ([1], [2])

- El control de acceso: Se refiere a que el acceso a los recursos de los sistemas sea controlado y restringido por los mismos sistemas, para que solo puedan acceder a los recursos las entidades autorizadas, evitando así la manipulación de la información por entidades no autorizadas. La técnica más común para el control de acceso es la utilización de contraseñas. ([1], [2])

- La tolerancia a fallos: Se puede definir como la correcta ejecución de un algoritmo en presencia de fallos, es decir la capacidad de respuesta de un sistema frente a un suceso inesperado, como puede ser un fallo de suministro eléctrico o un fallo de hardware. Algunas de las técnicas para conseguir tolerancia a fallos son los sistemas redundantes y los sistemas de alimentación ininterrumpida. ([1])

- Propiedad de la información: Se refiere a la posibilidad de poder verificar la procedencia de la información. ([1], [2])

2.1.2 - Amenazas a la seguridad de los sistemas

Una amenaza o ataque se puede definir como una acción o un acontecimiento que puede atentar contra la seguridad; o como la violación en potencia de la seguridad de un sistema. También se pueden definir como cualquier acción que suponga una violación de la seguridad de los sistemas.

Los sistemas informáticos están expuestos a tres tipos básicos de amenazas: intencionadas, no intencionadas y naturales o de fuerza mayor.

- Intencionadas: Pueden producirse por usuarios no autorizados externos o internos al sistema. Estos usuarios se pueden clasificar según sus intenciones: curiosos y maliciosos. Los usuarios curiosos intentarán acceder a los sistemas por simple curiosidad de saber si pueden hacerlo o por diversión, normalmente no tienen objetivos concretos y simplemente ojean la información. Los maliciosos intentan acceder a los sistemas con intenciones dañinas. Estas intenciones dañinas pueden ir desde el robo de información con fines económicos hasta la destrucción o modificación de información o recursos. ([6])

- No intencionadas: Se producen normalmente a partir de usuarios inexpertos que ya sea por ignorancia, negligencia o descuido pueden borrar información, crear agujeros de seguridad al no actualizar los programas debidamente o facilitar sus contraseñas personales. También está implicado en estas amenazas el personal dedicado al proceso de datos e información que por no seguir los procedimientos de seguridad establecidos pueden

facilitar el acceso a información relevante o pueden crear agujeros de seguridad en pequeños programas que pueden llegar a afectar a la aplicación global o aplicaciones con las que comparta información. ([6])

- Naturales o de fuerza mayor: Estas amenazas incluyen cualquier desastre imaginable tanto naturales (terremotos, incendios, inundaciones, etc.) como por fallos de los equipos (cortocircuitos, cortes del sistema eléctrico, etc.). ([6])

También hay que tener en cuenta los virus como una amenaza, aunque son considerados como una amenaza diferente ya que, aunque son creados por personas, son autónomos y se pueden evitar siguiendo unas normas básicas y utilizando antivirus actualizados.

2.1.3 - Soluciones básicas a las amenazas a la seguridad de los sistemas

La seguridad de los sistemas debe ser tratada como un problema global y flexible, ya que el nivel de seguridad de los sistemas es igual al de su punto más débil. Es decir, si se fortifica una parte del sistema pero se deja otra débilmente protegida el nivel de seguridad del sistema será igual al de la parte menos protegida.

Para crear un programa de seguridad efectivo hay que contar con un participación activa de los usuarios internos que utiliza el sistema. Esto es debido en parte a la existencia de la llamada Ingeniería Social que consiste en conseguir de los usuarios autorizados información crítica, como contraseñas, mediante engaños. Por tanto, la educación de los usuarios es una parte fundamental en la seguridad informática.

También es posible ayudarse de fuentes externas para adquirir conocimientos y experiencia específica, pero se debe tener en cuenta que el programa de seguridad debe ser gestionado y dirigido internamente involucrando en él a todos los usuarios del sistema.

Hay varias técnicas o métodos con las que se pueden evitar la mayoría de los daños causados por cada tipo de amenaza. ([6])

- Contramedidas para amenazas intencionadas: Las contramedidas que se pueden utilizar contra este tipo de amenazas se pueden dividir en dos grupos:
    - Para amenazas que vienen de exterior del sistema: Para evitar este tipo de amenazas hay que mantener siempre actualizados con las últimas versiones los programas instalados en el sistema. Es recomendable estar informado de los últimos agujeros de seguridad que se encuentran. También es importante tener instalado un cortafuegos para controlar el tráfico que va de la red interna a la externa y viceversa.

- Para amenazas que vienen del interior del sistema: Para evitar las amenazas internas se puede encriptar la información, especialmente las contraseñas y establecer una política de contraseñas. También es recomendable utilizar un cortafuegos interno para controlar el tráfico de información que va y viene de la zona más sensible del sistema. Y contratar personal de seguridad para vigilar los equipos más críticos.

- Contramedidas para amenazas no intencionadas: Para evitar los problemas que causan estas amenazas hay que restringir los permisos de cada usuario para que no puedan realizar acciones que no les correspondan. Así como la mencionada implantación de un cortafuegos interno.

- Contramedidas para amenazas naturales o de fuerza mayor: La implantación de medidas de seguridad físicas como sistemas ante incendios y sistemas de alimentación ininterrumpida (UPS y grupos electrógenos) y baterías son las principales soluciones para este tipo de amenazas.

Otras soluciones o medidas generales para cualquier tipo de amenaza es la utilización de sistemas de respaldo (backup), que consisten en hacer copias de seguridad de la información almacenada en el sistema y la utilización de sistemas redundantes (RAID), que pueden aportar mas rendimiento y aumentan el nivel de seguridad al guardar por duplicado la información en unidades de disco diferentes. Es aconsejable la utilización de los dos sistemas simultáneamente ya que por separado no llegan a ofrecer un nivel de seguridad aceptable. ([1])

2.2 - Criptografía

Una de las técnicas más usadas en la seguridad informática y que se aplica a todos los niveles en sistemas informáticos es la criptografía, de esta técnica se habla en este punto.

2.2.1 - Introducción al problema de la criptografía

La Real Academia de la Lengua [34] define la criptografía como el arte de escribir con clave secreta o de un modo enigmático.

A partir de los años 50 con los trabajos de Shannon se empezó a considerar la criptografía como un conjunto de técnicas en vez de un arte. En la actualidad la criptografía está al alcance de la mayoría de la gente y es utilizada para obtener confidencialidad y privacidad en las comunicaciones. La criptografía sin embargo está generando controversia en algunos gobiernos que consideran la criptografía como una amenaza en manos de terroristas, que hace más difícil la implantación de sus planes de seguridad nacional, y no como una herramienta para asegurar la intimidad de los ciudadanos y ayudar a preservar el derecho a la intimidad.

2.2.2 - Clases de criptografía

Existen dos tipos de criptografía:

- Criptografía de clave simétrica: Se basa en la utilización de una sola clave para cifrar y para descifrar. Los algoritmos mas utilizados son el DES e IDEA basados en cifrado por bloques. Uno de los métodos clásicos más conocidos es el de transposición. El problema principal de la utilización de los métodos clave simétrica es cuando hay muchos miembros dentro del grupo donde se utiliza la clave, esto implica que todos deben saber la clave y que haya más posibilidades de que sea capturada por extraños al grupo. Otro caso parecido sería cuando se cambia la clave, en este caso hay que notificar la clave a todos los miembros corriendo el riesgo de que caiga la clave en manos no autorizadas. Por este problema es por lo que se desarrolló la criptografía de clave pública. ([7])

- Criptografía de clave pública: También llamada de clave asimétrica, se basa en el uso de un par de claves. Con cada una de estas claves se puede descifrar lo que con la otra se ha cifrado. Una de las claves se llama clave privada y sólo debe ser conocida por el propietario. La otra se llama clave pública y es utilizada por quien quiera comunicarse con el propietario de la clave privada de forma segura. En la actualidad hay muy pocos algoritmos eficientes o útiles de clave asimétrica. El principal problema es que necesitan claves muy grandes, de unos 1024 bits, para considerarse seguros por lo que la complejidad de cálculo hace que sean más lentos que los de clave simétrica. Estos métodos, en la práctica, se utilizan principalmente para codificar la clave de sesión (simétrica) cuando se hace una conexión. También se puede usar para autentificar mensajes. El algoritmo mas utilizado es el RSA, aunque hay otros muchos como el Rabin, ElGama y McElice. ([7])

En la actualidad los algoritmos de cifrado más utilizados son:

- DES: Es el algoritmo simétrico más extendido mundialmente. Nació en los años 70 en Estado Unidos. Trabaja codificando bloques de 64 bits empleando claves de 56 bits, por lo que con un ataque de fuerza bruta es viable, en la actualidad, obtener la clave y decodificar la información. Esto es debido a la corta longitud que emplea en la clave. A pesar de esto se sigue utilizando ya que desde el punto de vista teórico y de diseño es perfectamente válido. Para intentar solucionar el problema de la longitud de la clave se han creado diferentes variantes como el Tripe-DES que codifica dos veces por lo que la clave se dobla a 112 bits. ([7], [8])

- IDEA: Surgió a partir del DES en 1992, se considera uno de los mas seguros algoritmos simétricos en la actualidad. Trabaja codificando bloques de 64 bits de longitud y claves de 128 bits. Usa diversas técnicas de confusión y difusión utilizando operaciones básicas como el XOR. ([7], [8])

- RSA: Es un algoritmo famoso por su sencillez y a la vez su robustez aunque necesita una longitud de clave bastante larga. Al utilizar claves duales se puede utilizar tanto para codificar como para autentificar. ([7], [8])

- Protocolos SSL y TLS: Son protocolos asimétricos. El protocolo TLS (Transport Layer Security) es una mejora de su padre el protocolo SSL (Secure Sockets Layer). Los dos protocolos están diseñados para establecer conexiones seguras a través de Internet de formas sencilla y transparente. Básicamente su funcionamiento consiste en interponer una capa que codifica los mensajes antes de ser enviados y que decodifica los mensajes, cuando son recibidos, antes de ser mostrados o enviados a la aplicación destinataria. ([7])

2.2.3 - Esteganografía

Se podría definir como la técnica de ocultar el hecho de enviar un mensaje.

Normalmente esto se consigue enviando el mensaje oculto dentro de otro mensaje. Hay multitud de métodos de aplicar esta técnica, por ejemplo se puede utilizar el bit menos significativo del color de cada píxel de una imagen para mandar la información que se desea ocultar, el mensaje que se reciba será una imagen por lo que un observador externo nunca sospechará que se envía también un mensaje. ([7])

2.2.4 - Certificados X.509

Un certificado está formado esencialmente una clave pública y un identificador de usuario, avalados con la firma digital de una autoridad de certificación. Los certificados se utilizan para demostrar que una clave pública pertenece a un usuario concreto. El formato X.509 es el estándar de certificación más extendido y común en la actualidad. ([7])

2.2.5 - PGP

El PGP (Pretty Good Privacy) nació como una herramienta para poder usar la encriptación de forma sencilla y potente. Con el paso de los años se ha convertido en un estándar internacional, gracias sobre todo a su fiabilidad y popularidad.

PGP utiliza criptografía de clave asimétrica. Uno de sus puntos fuertes es la gran facilidad con la que se gestionan las claves públicas y privadas.

Cuando el emisor quiere mandar un mensaje o iniciar una conexión primero se crea una clave simétrica (clave de sesión) aleatoria con la cual se codifica el mensaje, luego se codifica la clave de sesión con la clave pública del receptor y se envía todo junto al receptor. Cuando el receptor recibe la clave de sesión la descodifica con su clave privada y obtiene la clave simétrica para descifrar los mensajes de la conexión. Con este método se consigue establecer una conexión segura. ([7])

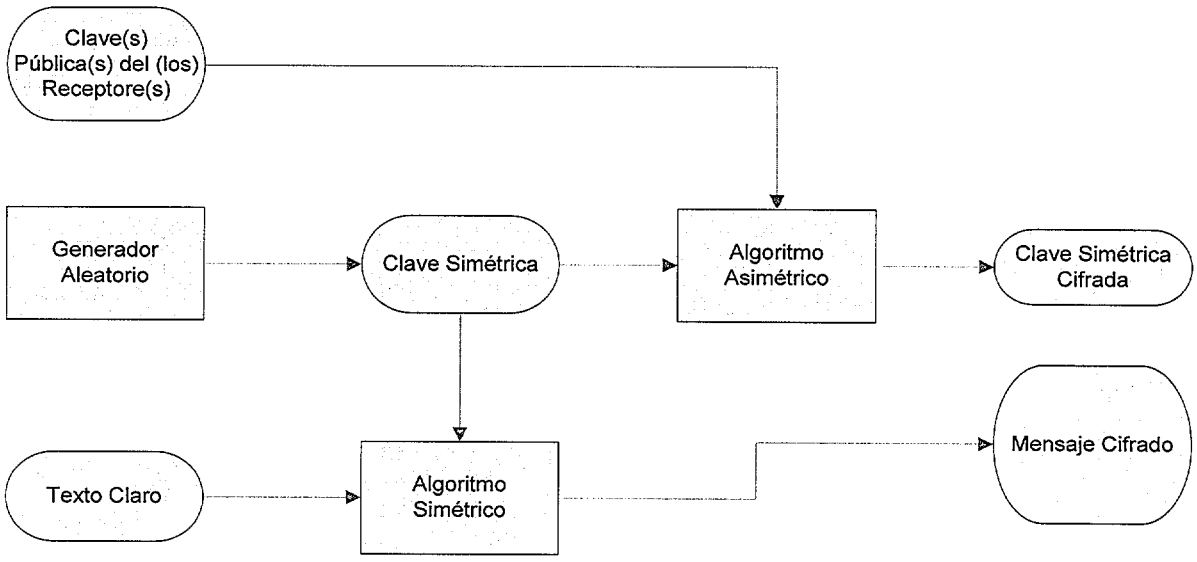


Figura 1: Codificación de un mensaje PGP

2.3 - Tipos de atacantes y de ataques

2.3.1- Ataques

Se pueden clasificar por los efectos que causan en los sistemas:

- Interrupción: Un recurso del sistema es destruido o se vuelve no disponible. Es un ataque contra la disponibilidad. Un claro ejemplo de estos ataques son los Nukes, que hacen que los equipos queden fuera de servicio. Otros ataques posibles serían la destrucción de elementos hardware, como un disco duro, la deshabilitación del sistema de gestión de ficheros o como el corte de la línea de comunicación. ([4])

- Intercepción: Una entidad no autorizada consigue acceso a un recurso. Es un ataque contra la confidencialidad. El Soopfing y la utilización de troyanos para la obtención de datos de forma ilícita, o la obtención de la identidad de los usuarios de una comunicación mediante la intercepción y estudio de los paquetes de la comunicación son ejemplos claros de este tipo de ataques. ([2], [4])

- Modificación: Una entidad no autorizada no sólo consigue acceder a un recurso, también es capaz de manipularlo. Es un ataque contra la integridad. Los troyanos y virus realizan este tipo de ataques. Otros ejemplos serían la modificación de los datos de los archivos, la alteración del funcionamiento de los programas y la modificación del contenido de mensajes que se estén transmitiendo por la red. ([2], [4])

- Fabricación: Una entidad no autorizada inserta objetos falsificados en el sistema. Es un ataque contra la autenticidad. La inserción de mensajes falsos en una red o la adicción de datos a un archivo son ejemplos de este tipo de ataques. ([2], [4])

Otra posible clasificación de estos ataques es:

- Ataques pasivos: Estos tipos de ataques se basan en la obtención de información sin alterar la comunicación ni los datos. Para ello se escuchan o monitorizan las comunicaciones para obtener y analizar la información que está siendo transmitida. Estos ataques son muy difíciles de detectar, ya que no provocan ninguna alteración de los datos, aunque existen formas sencillas de evitarlos como con la encriptación. Un ejemplo de ataque de este tipo es el Sniffing. ([2])

- Ataques activos: Realizan modificaciones de los datos o crean datos falsos. Se pueden dividir en cuatro categorías [2]:

  1- Suplantación de identidad: El atacante se hace pasar por una identidad (por ejemplo como un usuario o una máquina) para poder entrar en el sistema con los privilegios de la identidad suplantada.
  2- Reactuación: Consiste en la captación y repetición de uno o varios mensajes legítimos para producir un efecto no deseado, como por ejemplo ingresar dinero repetidas veces en una cuenta dada.
  3- Modificación de mensajes: Se modifican partes de los mensajes o la secuencia de los mismos para conseguir efectos distintos de los originales.
  4- Degradación del servicio: Impiden o deterioran el uso normal de los servicios o de las comunicaciones. Un ataque característico de esta clase de ataques es la denegación de servicio, consiste en paralizar temporalmente el servicio que ofrece un servidor.

2.3.1.1 Ataques típicos

SPOOFING: Este ataque consiste en suplantar la identidad de una persona o máquina desde otra máquina consiguiendo así comunicarse como la identidad suplantada. De esta forma se puede conseguir acceso a los recursos de las máquinas víctima teniendo en cuenta que esas máquinas tienen algún tipo de confianza basado en el nombre, dirección IP o dirección MAC. Existen variaciones de este ataque según en que se base la suplantación: IP Spoofing, ARP Spoofing, Web Spoofing, DNS Spoofing y SMTP Soopfing. ([18])

CROSSITE SCRIPTING: Este ataque se beneficia de la pobre verificación por parte de los sitios Web de las cadenas de entrada enviadas por los usuarios a través de formularios, o directamente a través del URL. Estas cadenas pueden contener "scripts" maliciosos. Cuando esta entrada se le muestra dinámicamente a un usuario dentro de una página Web, el "script" malicioso se ejecutará en el navegador del usuario dentro del contexto de seguridad de la página Web visitada. Como consecuencia, podrá realizar

en el ordenador del usuario todas las acciones que le sean permitidas a ese sitio Web, como por ejemplo interceptar entradas del usuario víctima o leer sus "cookies". ([5])

PHISHING: El término "phishing" viene de la contracción de "password harvesting fishing" (cosecha y pesca de contraseñas). El ataque consiste en duplicar una página Web para hacer creer al visitante que se encuentra en la página original en lugar de la copiada. Con este método se podrá obtener los datos que el usuario víctima introduzca en la página falsa. Este ataque se ayuda de la ingeniería social (por ejemplo mediante el envío de correos electrónicos indiscriminadamente) para conseguir que los usuarios entren en las páginas falsas e introduzcan sus datos. ([21])

SQL-INJECTION: Afecta a cualquier aplicación que utilice llamadas a bases de datos. Cuando se realiza una consulta sin haber tratado correctamente los datos que forman parte de ella, se puede lograr que la consulta produzca resultados no previstos. Dependiendo de diversos factores, el problema puede suponer desde autorizar un acceso no permitido, pasando por obtener la base de datos o modificar los datos de la misma hasta ejecutar código no previsto en el servidor. ([20])

DOS (Denial of Service): Las negaciones de servicios son ataques dirigidos contra un recurso informático (una máquina, una red, una impresora, etc.) con el objetivo de degradar total o parcialmente los servicios prestados por ese recurso a sus usuarios legítimos. Son ataques sencillos de implementar, difíciles de evitar y en entornos donde la disponibilidad sea esencial son muy perjudiciales. ([18])

BUFFER OVERFLOW: Consiste en escribir en un buffer más datos de los que es capaz de contener. Con esto se puede conseguir corromper la pila de ejecución de un programa escribiendo más allá de los limites del buffer declarado en una función, causando que la dirección de retorno de dicha función sea una dirección aleatoria o que se ejecute algún código no permitido. ([19])

TROYANOS: Son programas que aparentemente realizan una función útil para quién lo ejecuta, pero que también, o sólo, realizan una función que el usuario desconoce. Un troyano por si mismo no es dañino, normalmente son programas que se instalan en la máquina víctima (como servidor) facilitando el acceso a la misma desde otras máquinas (como clientes). Por lo tanto, el peligro reside en el uso malévolo de estos programas. ([18])

VIRUS: Es una secuencia de código que se inserta en un fichero ejecutable denominado anfitrión, de forma que al ejecutar el programa también se ejecuta el virus. Generalmente esta ejecución implica la copia del código del virus, o una modificación del mismo, en otros programas. El virus necesita un programa donde insertarse para poderse ejecutar por lo que no se puede considerar un programa o proceso independiente. ([18])

WORMS: Son programas capaces de viajar por sí mismos a través de redes de computadores para realizar cualquier actividad una vez alcanzada una máquina. Aunque esta actividad no tiene por qué ser peligrosa, los gusanos pueden instalar virus en los sistemas alcanzados, atacar el sistema como lo haría un intruso, o simplemente consumir recursos de la red afectada. ([18])

ROOTKIT: Son herramientas que permiten manejar una máquina de forma ilegítima, permaneciendo invisibles para los usuarios de la máquina y para todo el software que se ejecuta en ella. Además, proveen al atacante de vías de acceso ocultas para utilizar nuevamente el sistema en futuras oportunidades. ([22])

EXPLOIT: Es código escrito con el fin de aprovechar un error de programación para obtener diversos privilegios como el ingreso en el sistema. La única manera de protección contra este tipo de ataque es tener actualizado el sistema.

2.3.2 - Atacantes

Los tipos de atacantes que existen en la actualidad se definen por sus motivaciones, intereses, nivel de conocimientos y forma de actuar. Se pueden separar en estos grupos:

HACKERS: Este tipo de atacantes son expertos en sistemas informáticos y de comunicaciones. Dominan la programación y la electrónica. Se dedican a explorar (o introducirse en) sistemas sin causar daños y suelen ser los que informan de los fallos en los programas comerciales. Tienen una ética basada en el aprendizaje y en la información libre y abierta. ([23])

WHACKER: Tienen los mismos conocimientos que los Hackers pero no comparten la misma ética. Estos atacantes se mueven por dinero o por fama y suelen causar daños graves. ([17])

CRACKERS: Estos atacantes se dedican principalmente a romper las medidas de seguridad de los sistemas y del software. Suelen difundir por Internet la forma de romper la protección ("cracks"), esto último es donde radica el problema de este tipo de atacantes. ([23])

CRASHER: Este grupo se dedica a atacar sistemas para provocar que dejen de funcionar. Normalmente utilizan ataques de denegación de servicio. ([23])

LAMERS: Este grupo es el más numeroso y con mayor presencia en la red. Suelen tener un conocimiento nulo de informática y apenas saben lo que es un ordenador, pero conocen las oportunidades que les brinda Internet. Se dedican obsesivamente a buscar información sobre "hacking" para poder entra en algún sistema remoto o poder modificar algún fichero de otro ordenador remoto. Utilizan las múltiples herramientas software de "hackeo" que ofrece Internet para bombardear correos, obtener contraseñas, etc. Suelen destruir su propio ordenador y otros de la red, y cuando esto sucede se enorgullecen de sus logros. ([23])

PHREAKER: Inicialmente este tipo de atacantes se conocían en la red por sus conocimientos de telefonía. En la actualidad conocen también los sistemas de prepago y deben de tener amplios conocimientos de informática. Se aprovechan de forma ilegal de servicios de privados. Como televisión de pago, servicios de telefonía o servicios de compañías de suministros. ([23])

SCRIPT KIDDIES: Suelen ser usuarios habituales de Internet con algún conocimiento básico sobre seguridad y "hacking". Suelen ser devotos de estos temas aunque no llegan a comprenderlos. Se dedican a recopilar información sobre estos temas. También buscan programas de "hacking" para luego ejecutarlos sin haberse leído las instrucciones de uso. De esta forma suelen propagar virus y dañar sus propios equipos. ([23])

NEWBIES: Son novatos o principiantes en los temas de "hacking". Tienen conocimientos básicos de seguridad. Al contrario que los "Lamers" y los "Script Kiddies" prueban los programas de "hacking" que hay por Internet con cuidado y adquieren conocimientos poco a poco. ([23])

CAPÍTULO 3: Técnicas de detección y defensa

Dentro de las técnicas usadas en los sistemas informáticos para evitar o defenderse de los ataques o intrusiones existen dos tipos de herramientas: estáticas y dinámicas. Una herramienta que utiliza un enfoque estático, o basado en intervalos, trabaja en intervalos de tiempo y no de forma continua como una herramienta con enfoque dinámico. Para aclarar el concepto, se podría decir, que un ejemplo estático, es una imagen, mientras que uno dinámico es como un vídeo. Una cámara fotográfica puede captar imágenes a intervalos periódicos de tiempo, mientras una cámara de video permite ver lo que pasa en tiempo real. Este capítulo se centra en dos tipos de herramientas, uno que utiliza medidas dinámicas y otro que utiliza solo medidas estáticas. Estos tipos de herramientas son los sistemas de detección de intrusos y los escáneres de vulnerabilidades respectivamente.

Como se verá a continuación, cabe destacar otra diferencia importante entre estos dos tipos de herramientas: su forma de actuar. Un sistema de detección de intrusos es reactivo porque actúa cuando detecta una intrusión mientras sucede o después de haber sucedido. En cambio, un analizador de vulnerabilidades es proactivo, determina la susceptibilidad a las intrusiones antes de que los sistemas sean atacados.

Estos dos tipos de herramientas junto con las métricas de seguridad son parte esencial en la realización de las auditorías de seguridad en sistemas informáticos: los detectores de intrusos y los escáneres de vulnerabilidades son utilizados para obtener datos sobre los sistemas que pueden ser utilizados en las métricas de seguridad. Por estas razones en los siguientes puntos del capítulo se analizan profundamente.

3.1 - Sistemas de Detección de Intrusos (IDS)

3.1.1 - Definiciones

Un intruso es un atacante o usuario que intentan acceder al sistema o a partes del mismo sin autorización, o que utiliza sus privilegios malignamente con fines poco éticos.

Una intrusión se define como un "conjunto de acciones que comprometen la integridad, confidencialidad o disponibilidad de un recurso" [9].

Los IDS son sistemas software o hardware que automatizan el proceso de monitorizar los eventos que suceden en un sistema o en una red, analizándolos en busca de actividades no autorizadas o sospechosas.

Los Sistema de Detección de Intrusiones se pueden definir como "los elementos que detectan, identifican y responden a actividades no autorizadas o anormales" [9].

3.1.2 - Un poco de historia

Año 50:

- Bell Telephone System desarrolla el EDP (Procesamiento Electrónico de Datos) para realizar auditorías mediante ordenadores.

Años 70:

- El departamento de Defensa de EEUU crea la Iniciativa de Seguridad, que define los requisitos de seguridad de los Sistemas de Confianza.

Años 80:

- James P. Anderson elabora el primer documento en que se habla sobre la detección de intrusiones.

- Dorothy Denning y Perter Newmann desarrollan el IDES (Sistemas Experto de Detección de Intrusiones.

- Se crean numerosas iniciativas en materias de detección de intrusiones: Haytack, MIDAS, NADIR, NSM, Wisdom and Sense, etc.

Años 90:

- Detección de intrusiones de red y primeros productos comerciales. ([9])

3.1.3 - Introducción

La mayoría de los atacantes utilizan técnicas documentadas que se basan en vulnerabilidades ampliamente conocidas. Estas vulnerabilidades no siempre pueden ser corregidas ya sea por problemas de soporte del software utilizado, por errores en las configuraciones por parte de los usuarios o del administrador del sistema, o por requerimientos operacionales de protocolos o de los sistemas utilizados. Por esto la utilización de sistemas de detección de intrusos es importante para mantener y mejorar la seguridad del sistema.

Estos sistemas se utilizan para:

- Detectar ataques o violaciones de la seguridad del sistema que no están cubiertas por otras medidas de seguridad implantadas en el sistema.

- Detectar los pasos previos a los ataques para poder tomar las contramedidas adecuadas para poder evitarlos.

- Como medida disuasoria para prevenir los comportamientos maliciosos o problemáticos e incrementar el riesgo percibido por los atacantes de ser descubiertos y castigados.

- Comprobar y controlar la calidad del sistema de seguridad implantado y de la administración del mismo.

- Recoger información sobre las intrusiones sufridas en el sistema para poder documentarlas y poder tomar las medidas adecuadas para evitarlas en el futuro.

La mayoría de los sistemas de detección de intrusos permiten detectar gran cantidad de ataques ya sean a nivel de red o de aplicación:

- Exploración de la red o de un sistema:

    - Topología de la red.
    - Tipo de tráfico permitido a través de un cortafuegos.
    - Máquinas activas en una red.
    - SO de los ordenadores activos.
    - Servidores activos.
    - Versiones del software.
    - Búsquedas de vulnerabilidades.

- Ataques de denegación de servicio:

    - Ataques lógicos ("Flaw attacks").
    - Ataques de inundación ("Flood attacks").

- Penetración en sistemas:

    - Adquisición no autorizada de privilegios, recursos o datos.

3.1.4 - Tipos y características de Sistemas de Detección de Intrusos

Los IDS se pueden diferenciar mediante las características de los tres componentes fundamentales que la mayoría tienen en común. El primero es la fuente de información que es el medio de donde los IDS obtienen los datos para detectar las intrusiones. El segundo componente es el sistema de análisis que se define como el mecanismo con el cual detectan las intrusiones. Y por último la respuesta que es la forma que tienen de reaccionar cuando es detectada una intrusión.

- Fuente de información:

    - Red: En este tipo de fuentes se obtiene la información a partir de los paquetes que circulan por la red.

- Máquina ("Host"): La información que se obtiene de este tipo de fuente la proporciona el estado de la máquina donde está instalado el IDS.

- Aplicación: En este tipo de fuentes la información es obtenida directamente de las aplicaciones instaladas en la máquina donde está instalado el IDS.

- Sistema de análisis:

  - Detección de anomalías: Este tipo de análisis se basa en estudiar cuál es el comportamiento normal del sistema para poder definir que alteraciones en este comportamiento se pueden identificar como anómalas (comportamientos extraños).

  - Detección de usos indebidos o no autorizados: Este tipo de análisis consiste en estudiar y crear patrones de las intrusiones conocidas para así poder detectarlos cuando ocurran.

- Respuesta:

  - Medidas activas: Este tipo de respuesta implica alguna acción en particular como el bloqueo de la conexión o el cierre inmediato de la cuenta.
  - Medidas pasivas: Normalmente se suelen limitar a presentar informes o a registrar las intrusiones ocurridas.

Independientemente del tipo de IDS del que se trate, las características principales de cualquier sistema de detección de intrusos son:

- Arquitectura: Se refiere a la organización de los diferentes componentes de los IDS.

  - Conjunta: Antiguamente el sistema en el que se ejecuta el IDS y el sistema de monitorización estaban juntos, lo que reducía el coste, pero presentaba problemas desde el punto de vista de la seguridad.

  - Separada: En la actualidad el sistema que monitoriza el IDS se separa del sistema que ejecuta el IDS. De esta forma se consigue mayor seguridad al poder ocultar la presencia del IDS a los posibles atacantes.

- Objetivos:

  - Registro de los ataques: Detectan y determinan quién es el atacante, el tipo de ataque, el objetivo del ataque, etc.

  - Respuesta a los ataques: Detectan y actúan bloqueando el ataque sin importar quién lo realiza.

- Estrategia de control:
    - Centralizado: Todos los sistemas monitor del IDS son controlados e informan a un sistema central.
    - Parcialmente distribuido (jerárquico): Cada sistema (agente) informa a un sistema (agente) superior que coordina la información e informa a su superior.
    - Completamente distribuido (sistemas multiagente): Cada agente del IDS se coordina e informa al resto de agentes.
- Tiempo de procesado:
    - Continuo (tiempo real): Se analiza continuamente los datos que va proporcionando la fuente información.
    - Flujo discontinuo de información (proceso a intervalos): La información se obtiene y analiza en intervalos de tiempo.

3.1.5 - Fuentes de información de los IDS

La forma más extendida de clasificar los IDS es mediante el tipo de fuente de información que utilizan. En este caso se clasifican teniendo en cuenta la información que utilizan y cómo la obtienen. ([10])

3.1.5.1 Basados en máquina (HIDS)

En los últimos años este tipo de IDS ha ido perdiendo importancia a favor de los sistemas de detección de intrusos basados en red (NIDS, Network Intrusion Detection System), aunque en la actualidad los HIDS (Host Intrusion Detection System) se están empezando a considerar de nuevo. Esto es debido a un aumento del número de redes de alta velocidad conmutadas. Estas redes son un obstáculo significativo para que los detectores de intrusos basados en red sean eficaces. También está influyendo en el resurgimiento de los HIDS el hecho de que en la actualidad hay demasiados IDS basados en la red y pocos basados en "host".

Las ventajas de estos sistemas son:

- Pueden analizar el sistema con gran precisión y fiabilidad determinando qué procesos y usuarios están involucrados en un determinado ataque.
- Pueden observar el desenlace de los ataques producidos.
- Detectan ataques que no son vistos por los NIDS.

- No tienen problemas al trabajar en redes conmutadas ni en entornos donde se intercambie información cifrada.

- Pueden detectar troyanos y otros ataques si se maneja la información que proporciona el núcleo del sistema operativo.

Las desventajas:

- Se tiene que configurar y gestionar la información de cada uno de las máquinas que estemos monitorizando.

- Suelen ser los sistemas de detecciones de intrusos más difíciles de configurar.

- Requieren constante atención sobre la política de auditoría.

- Al estar la fuente de información en el mismo equipo que recibe los ataques el IDS puede ser afectado por estos mismos ataques.

- Los equipos pueden ser deshabilitados mediante ataques de denegación de servicio.

- Son poco apropiados para sondeos de red o para ataques de sondeos de redes enteras, ya que la información que se procesa no contiene registros del comportamiento de bajo nivel de la red.

- Pueden producir cantidades de información muy elevadas.

- Reducen el rendimiento de las máquinas donde residen.

Los detectores de intrusos basados en máquina ("host") se pueden dividir en dos categorías principales, pudiendo haber HIDS con elementos de ambas categorías.

3.1.5.1.1 IDS Basados en máquina tradicionales (HIDS)

Los HIDS basan su sistema de monitorización en el análisis de la información que obtienen del estado de la máquina ("host") donde éste reside. Emplean un agente que reside en cada máquina a supervisar. Este agente escudriña los registros de los eventos del sistema, los registros del núcleo, los ficheros críticos del sistema y los demás recursos auditables buscando cambios desautorizados o patrones sospechosos de cualquier actividad. Siempre que cualquier cosa se salga de lo normal se notifica y se levantan automáticamente las alarmas o las trampas de SNMP (Simple Network Management Protocol).

Por ejemplo, un HIDS monitoriza o supervisa el Registro para los accesos no autorizados, los registros del núcleo para detectar cuándo se inician procesos inadecuados, o toma nota de cuando se intenta acceder a una cuenta con una contraseña incorrecta. Si hay muchos intentos para conectarse a una cuenta en un período corto de tiempo el sistema puede interpretar que alguien está intentando acceder ilegalmente y levantar una alarma.

Los HIDS tradicionales son muy buenos en la detección de amenazas internas y proporcionan generalmente una extensa valoración del daño y datos para posteriores análisis forenses de los ataques. Hay que tener en cuenta que las amenazas internas no se refieren siempre a los empleados. Es posible, por ejemplo, que un atacante acceda a los sistemas internos robando o consiguiendo mediante ingeniería social el nombre de usuario y la contraseña de algún usuario legítimo. En este punto, el atacante tendría todos los derechos y privilegios asociados a ese usuario, y sería mucho más difícil de detectar. ([9], [10], [11])

Ejemplos de HIDS son Aide [36] con licencia GPL (GNU General Public License), LogCheck [38] y LogWatcher [39].

3.1.5.1.2 Comprobadores de integridad de ficheros (FIA)

Los FIA (File Integrity Assesment) supervisan el estado de los archivos del sistema y de la aplicación, o del Registro. Son herramientas que utilizan un enfoque estático y suelen estar incluidas en los IDS basados en máquina ("host"), por lo que también se suelen considerar como un caso especial de los IDS.

Para que los FIA funcionen correctamente primero se debe hacer una fase inicial que consiste en crear y almacenar una imagen del sistema cuando está limpio de intrusiones, utilizando funciones resumen u otros métodos de cifrado robustos, generalmente en la forma de "hashes" criptográficos de los objetos supervisados. De esta forma la imagen del sistema limpio queda asegurada contra posibles intentos de modificación. Después, cada cierto tiempo, se compara la imagen del sistema limpio con la imagen del sistema actual.

Así si un intruso o troyano intenta acceder al sistema y realiza cambios en ficheros críticos, altera o elimina ficheros para no dejar evidencias de su actividad o incluso deja una puerta trasera para poder volver a entrar en el sistema, el FIA lo detectará y lanzará una alarma. Esto hace a los FIA muy adecuados para determinar el grado de daño real causado por un ataque realizado con éxito.

El inconveniente principal de los FIA es que los escaneos deben de ser frecuentes y periódicos, más bien en tiempo real para poder detectar ataques y poder tomar medidas, normalmente cuando ya se ha realizado el ataque. Por esto su fuerza está en análisis forense después de que un ataque haya sido realizado, facilitando la identificación del ataque o método utilizado, por lo que importan poco las alertas en tiempo real.

También son útiles para devolver a la normalidad al sistema, optimizando el proceso de restauración. ([9], [10], [11])

Algunos ejemplos de HIDS con comprobadores de integridad de ficheros son Tripwire [35] y SamHain [37].

3.1.5.2 Basados en red (NIDS)

Los NIDS (Network Intrusión Detection System) basan su sistema de monitorización en la información recolectada de la red en tiempo real. La información que se captura son los paquetes que circulan por la red. Normalmente se capturan mediante mecanismos de "sniffing" y se analizan detalladamente, para detectar posibles ataques antes de que los paquetes alcancen su destino y hagan daño.

Los posibles ataques son detectados comparando unos o varios paquetes contra una base de datos de firmas de ataques o mediante protocolos de desciframiento para detectar anomalías, o ambas formas.

Cuando se detecta una actividad sospechosa, los NIDS son capaces de lanzar alarmas y de terminar la conexión utilizada para el ataque inmediatamente (al igual que algunos HIDS). Algunos también se integran con el cortafuegos, definiendo automáticamente nuevas reglas para evitar el ataque concreto en futuro.

La mayoría de los "Network-based IDS" hasta la fecha trabajan en modo promiscuo. Esto significa que examinan todos los paquetes del segmento de la red local donde estén instalados, sin detenerse en si esos paquetes son destinados a la máquina que tiene el IDS (como un monitor de la red o como un Sniffer). Dado que tienen mucha carga de trabajo al examinar cada paquete y seguir las sesiones activas, requieren generalmente una máquina dedicada debido a que suelen usar bastantes recursos del sistema.

Como la mayoría de los ataques no se basan en el contenido de un solo paquete, sino que se componen de varios, enviados a veces en períodos de tiempo muy largos, es necesario que los NIDS tengan un buffer interno para almacenar los paquetes recibidos y así poder llevar al día las sesiones establecidas y poder comparar grupos de paquetes con la base de datos de firmas de ataques. De esta forma los NIDS pueden comparar los paquetes que se van recibiendo contra la base de datos de firmas en el contexto de una sesión concreta, en vez de examinar cada paquete aisladamente.

Los NIDS suelen estar formados por un conjunto de "hosts" que se sitúan en lugares estratégicos de la red y que actúan como sensores, cada uno de ellos con un único propósito. La información obtenida por estos equipos se envía a una consola de control central. Normalmente se necesita un sensor de NIDS por segmento de la red, puesto que no pueden ver a través de los "routers" o "switches". ([9], [10], [11])

Las principales ventajas de los NIDS son:

- Como sólo se ejecuta el IDS en los equipos montados para este propósito se pueden proteger mejor frente a ataques.

- Con pocos NIDS bien situados se puede controlar una red de gran tamaño.

- La infraestructura de los NIDS tiene poco impacto en el rendimiento de la red.

- La seguridad de estos sistemas es bastante buena ya que se pueden hacer casi invisibles a los atacantes.

Algunas desventajas son:

- En redes grandes con mucho tráfico es muy difícil analizar toda la información capturada, por lo que la eficiencia no es óptima. Se pueden utilizar implementaciones hardware de los NIDS para intentar aumentar la eficiencia. También se puede limitar el número de paquetes capturados o limitar la capacidad de proceso para cada ataque, pero esto reduce la eficiencia de la detección.

- No son apropiados para redes conmutadas. Los "switches" segmentan las redes por lo que es difícil capturar todo el tráfico. Es necesario trabajar con "switches" con un puerto que reciba toda la información que atraviesa el "switch". Pero esto no asegura que se reciba toda la información que atraviesa la red.

- Tienen problemas con la fragmentación y no pueden analizar información cifrada, por ejemplo en redes privadas virtuales (VPNs).

- No pueden informar si un ataque ha tenido éxito o no.

Algunos ejemplos de sistemas de detectores de intrusos basados en red son RealSecure de ISS [40], Cisco IDS [41], Enterasys Dragon [42] y Snort [43] un NIDS de código abierto.

3.1.5.3 NIDS de nodo de red (NNIDS)

Los sistemas de identificación de intrusos de nodo de red son un tipo de agente híbrido de IDS que supera algunas de las limitaciones de los NIDS.

El agente de los NNIDS ("Network Node IDS") trabaja de una manera similar a los NIDS, captura los paquetes de la red y realiza un análisis de protocolo o los compara contra una base de datos de firmas. Pero en este caso el agente solamente recoge los paquetes que van al nodo de red en el cual reside. Al estar instalado dentro de la pila de protocolo de la máquina se le llama a veces IDS basado en pila (Stack-based IDS).

Como los NNIDS no examinan todos los paquetes de la red pueden ser mucho más rápidos y necesitan menos recursos de sistema que los NIDS, y esto permite que sean instalados en los servidores ya implantados sin tener un deterioro en el rendimiento

considerable. También son adecuados para las redes con gran carga de tráfico, redes conmutadas o para una VPN (Red Privada Virtual) con tráfico cifrado, básicamente en todas las áreas donde los NIDS tienen problemas.

Obviamente es necesario instalar uno de estos agentes NNIDS por cada servidor que se quiera proteger, y todos tendrán que comunicarse con una consola central que será la que informe sobre las alarmas.

Un ejemplo de configuración de un sistema informático en la que se utilizan los NNIDS es una red con una granja de servidores. Los NNIDS se instalan en cada uno de los servidores conmutados de la granja de servidores, y para el resto de la red o para segmentos de la red con menos carga de tráfico se pueden utilizar NIDS, donde un sólo IDS puede proteger una gran cantidad de máquinas. ([10])

Algunas herramientas que incluyen este tipo de IDS son BlackICE Agent [90] y el cortafuegos Tiny CMDS [91].

3.1.5.4 Basados en aplicación

Se suelen considerar un subconjunto especial de los IDS basados en máquina ("host"). Estos IDS se instalan en el sistema que se quiere proteger y se configuran según las aplicaciones que se deseen proteger.

Estos IDS no analizan el tráfico de la red, sino que se centran en el análisis de los eventos que se generan dentro de las aplicaciones, como las llamadas al sistema o los intentos de conexión de la aplicación, también analizan la memoria. Este enfoque permite evitar problemas con las posibles deficiencias de las aplicaciones como, por ejemplo, el desbordamiento de buffer. También sirven para detectar comportamientos sospechosos relacionados con usuarios que excedan sus permisos.

Con estos IDS es necesario realizar una primera fase de creación de perfil de sistema, similar al que se hace en los métodos detección de anomalías (explicado mas adelante en los sistemas de análisis de los IDS). En esta fase se registra la actividad de la aplicación y se elabora un modelo de comportamiento que sirve para detectar, junto con una serie de políticas, las posibles intrusiones.

Suelen ser utilizados en combinación con NIDS y HIDS ya que estos sistemas solo se centran en aplicaciones determinadas. ([9], [11])

Ventajas:

- Hacen análisis más minuciosos y personalizados al estar instalados en cada máquina a proteger.
- Pueden trabajar en entornos que empleen cifrado.
- Si se detecta alguna acción no permitida se activa la alarma y se bloquea la acción.

- Por lo tanto no sólo detecta sino que evita la supuesta acción dañina.

Desventajas:

- Son más vulnerables a los ataques que los HIDS porque las aplicaciones están menos protegidas que los núcleos de los sistemas operativos.
- No pueden detectar troyanos porque trabajan al nivel de abstracción del usuario.
- Para que funcione correctamente hay que definir todos los comportamientos normales de la aplicación.
- Si se actualiza la aplicación se deberá hacer, posiblemente, de nuevo el proceso de creación del perfil.

Algunos ejemplos de IDS basados en aplicación son: Store wath [30] y Entercept [31].

3.1.6 - Sistemas de análisis de los IDS

Una vez obtenidos los datos de las distintas fuentes de información, se les somete a diversas técnicas de estudio realizadas por los sistemas de análisis. Este componente de los IDS se encarga de la detección de los ataques que se producen en el sistema informático. Más específicamente se encarga del preproceso, clasificación, y posproceso de la información que les llega de las fuentes de información.

3.1.6.1 Detección de usos indebidos

Esta técnica consiste en identificar ataques conocidos mediante la forma que tienen de producirse. Es la técnica más utilizada en los IDS actuales.
Se parte de una base de datos en donde están documentados los ataques conocidos. Cada ataque tiene un determinado patrón al que se le denomina firma, con la cual se identifica a cada ataque (a este tipo de análisis también se le denomina análisis basado en firma).

Normalmente a cada patrón se le asigna una única firma, pero hay IDS que agrupan determinados ataques en una sola firma. A este tipo de análisis se les denomina análisis basado en estado y son bastante más complicados de implementar.

Las principales ventajas del análisis basado en firma son que:

- Generan un porcentaje bajo de falsas alarmas siendo muy efectivos en la detección.
- Permiten detectar y determinar la utilización de herramientas específicas para realizar ataques.
- La información que producen es sencilla y facilitan el manejo de los incidentes producidos.

Las desventajas son:

- Solo detectan ataques conocidos, por lo que hay que actualizar las bases de datos con las firmas de los nuevos ataques.

Hay que tener en cuenta que las firmas de los ataques son bastante estrictas por lo no detectan variantes de ataques conocidos. Sin embargo con los análisis basados en estado sí se pueden detectar variaciones de ataques conocidos al agrupar en una sola firma ataques con características similares.

En definitiva un detector de usos indebidos, es a grandes rasgos, un comparador de patrones. Su estructura general debe contar con una base de conocimiento con patrones fiables, una serie de eventos para poder ser analizados y un eficaz motor de análisis.

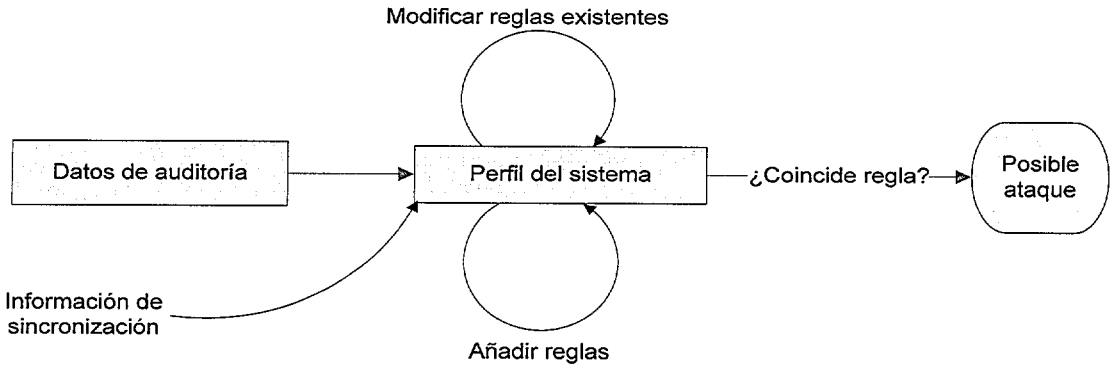


Figura 2: Modelo general de un detector de usos indebidos

Al principio la mayoría de los NIDS utilizaban búsquedas de correspondencias contra una base de datos de firmas conocidas. Después se empezó a investigar y a crear nuevos métodos totalmente diferentes que se basaban en hacer un análisis de protocolo completo del flujo de datos. Otros empezaron a usar heurísticas o análisis de anomalías para determinar cuando había tenido lugar un determinado ataque. Hoy en día la mayoría de los IDS utilizan una mezcla de diferentes métodos, aunque se suelen concentrar más en un método en concreto.

A continuación se describen diferentes métodos basados en usos indebidos centrados en los sistemas de detección de intrusos basados en red.

3.1.6.1.1 Búsqueda de correspondencias

Este método esta basado en la búsqueda de una secuencia fija de bytes en un solo paquete. Como su nombre indica es un enfoque bastante rígido pero fácil de emplear. En la mayoría de los casos el patrón es comparado solamente si el paquete sospechoso es asociado con un servicio o una dirección IP, o una red en particular o, más concretamente, con una entrada o una salida destinada a un puerto en particular. Esto ayuda a disminuir el número de inspecciones hechas para cada paquete. Sin embargo, esto se complica en sistemas que trabajan con protocolos que no tienen bien definidos los puertos que utilizan y, en particular, cuando se trata de troyanos, y de su tráfico asociado, que se puede mover generalmente a voluntad.
La estructura básica de una firma desde el enfoque de la búsqueda de correspondencia puede ser algo así: si el paquete es IPv4, el protocolo es TCP, el puerto destino 6666 y el código del paquete contiene la cadena "coco", lanzar una alarma.
También es posible indicar el punto de partida y el punto final de la inspección dentro del paquete, así como los TCP flags que deben ser considerados. No obstante esta técnica es la más simple y primitiva que existe para la detección de intrusos. ([10], [13])

Ventajas:

- Es el método más simple para detectar las intrusiones.
- Permite la comparación directa de un posible ataque con el patrón.
- Al ser tan específico, las alarmas de estos métodos son fiables con los patrones especificados.
- Es aplicable sobre todos los protocolos.

Inconvenientes:

- Pueden darse muchos falsos positivos si el patrón no está bien definido o si el patrón no es tan distinto al resto como asumió el creador de la firma.
- Cualquier modificación del ataque puede resultar en falsos negativos.
- Puede requerir que múltiples firmas traten una vulnerabilidad concreta.
- Este método está limitado normalmente a la inspección de un solo paquete y no se puede aplicar a flujos de tráfico de la red como puede ser el de HTTP. Esta limitación hace que sea fácil evitar la detección de ataques con técnicas sencillas de implementar.

3.1.6.1.2 Búsqueda de correspondencias con estados

La búsqueda de correspondencia con estados brinda un enfoque más sofisticado. Añade el concepto de secuencia de paquetes y tiene en cuenta el contexto en el que se encuentran los paquetes, es decir, que analiza cada paquete en conjunto con los demás paquetes que corresponden a la misma sesión. Para poder realizar este análisis los sistemas que lo implementen deben considerar el orden de llegada de los paquetes en el flujo TCP y deben de ajustar los patrones con los límites del paquete.

En vez de buscar el patrón en cada paquete, el sistema mantiene la información sobre el flujo TCP que se esta supervisando en cada estado. Un ejemplo simple del funcionamiento de este método sería: si por ejemplo, el ataque que se busca es mediante una conexión de un cliente a un servidor por el puerto 66666 y la cadena que busca el patrón para lanzar la alarmar es "cocola". El atacante podría dividir la cadena en varios paquetes diferentes conteniendo un paquete la cadena "coco" y otro paquete la cadena "la", con lo que el método de búsqueda de correspondencia simple o sin estado no detectaría el ataque. Sin embargo la búsqueda de correspondencia con estados guardaría los paquetes de la conexión, ordenaría la secuencia de paquetes detectando la palabra "cocola" y lanzaría la alarma correspondiente. ([10], [13])

Ventajas:

- El empleo de este método requiere sólo un poco más de esfuerzo que con la búsqueda de correspondencias simple.
- Como en la búsqueda de correspondencias simple este método permite la comparación directa de los ataques con los patrones.
- Al ser tan especifico, las alarmas de estos métodos son fiables con los patrones especificados.
- Es aplicable sobre todos los protocolos.
- Con este método es más difícil para el atacante evitar su detección.

Inconvenientes:

- Puede conducir a un alto número de falsos positivos si el patrón no es tan raro como el creador de la firma asumió.
- Cualquier modificación del ataque puede resultar en falsos negativos.
- Puede requerir que múltiples firmas traten una vulnerabilidad concreta.

3.1.6.1.3 Análisis basado en decodificación de protocolo

Las firmas de este método son en muchos sentidos extensiones inteligentes de las búsquedas de correspondencias con estado. Esta clase de la firma funciona descifrando varios elementos de manera semejante a como lo hace un cliente o un servidor en una conversación. Lo primero que se debe hacer es identificar los elementos del protocolo, después los IDS aplican las reglas definidas por el RFC (Request For Comments) para buscar violaciones. Las reglas del RFC ayudan a detectar anomalías como por ejemplo datos binarios en una solicitud HTTP o como una larga cadena de datos, en algún sitio donde no debiera estar, sospechosa de ser una tentativa de un ataque de desbordamiento de buffer. En algunos casos, estas violaciones se encuentran con indicios de algún patrón dentro de algún campo específico del protocolo, y pueden requerir técnicas más avanzadas que expliquen variables tales como la longitud de un campo o el número de argumentos.

Para explicar el funcionamiento del método se puede coger el mismo ejemplo explicado en los métodos anteriores. Supongamos un protocolo ficticio llamado BGS bajo el cual se este realizando el ataque, y más específicamente, asumimos que el ataque requiere un argumento ilegal que contiene la cadena "coco" que se tiene que enviar en el campo del Tipo del BGS. Se puede complicar más la situación, asumiendo que el campo Tipo es precedido por un campo de la longitud variable llamado Opciones. La lista válida de Opciones es "cocoh", "mocoh", "tcormer", y "buildco". Si se usa el algoritmo de búsqueda de correspondencias simple o con estado este caso conduce a falsos positivos porque la cadena "cocoh" contiene el patrón que se está buscando. Además, al ser las longitudes del campo variables, ese imposible limitar los falsos positivos especificando localizaciones del comienzo y de finalización de la búsqueda. La única manera de estar seguro que el "coco" se está pasando dentro del de argumento Tipo es descifrar completamente el protocolo.

Si por algún motivo no se realizase completamente la decodificación del protocolo pueden también darse falsos negativos si el protocolo permitiese comportamientos que los algoritmos de búsqueda de correspondencias tuvieran dificultad en tratar. Por ejemplo, si el protocolo BGS permitiera que hubiese bytes nulos (NULL) en la cabecera del BGS, los métodos de búsquedas de correspondencias no podrían ver cx00ox00cx00ox00. Sin embargo el motor de análisis del método de decodificación de protocolo activado quitaría los bytes nulos y lanzaría una alarma, suponiendo que la cadena "coco" está en el campo Tipo.

Un ejemplo simple para comparar el funcionamiento de estos dos métodos sería el registro de entrada en el Telnet para detectar los múltiples nombres de usuarios que suelen dejar abiertos los "rootkits" conocidos en el sistema. Un sistema con búsqueda de correspondencia podría escanear todo el tráfico del Telnet para todos los patrones conocidos, y cuantos más patrones más lento sería el sistema (aunque no siempre es así, esta suposición nos sirve para el ejemplo). En contraste, un sistema con decodificación de protocolo decodificará el protocolo de Telnet, extraerá el nombre del usuario y hará una búsqueda del nombre sobre un árbol binario o una tabla "hash". De esta forma, este método se puede adaptar mucho mejor a la aparición de nuevas firmas. No obstante no hay mucha diferencia entre los dos métodos ya que las soluciones con búsqueda de correspondencias no suelen utilizar la fuerza bruta.

Puede parecer que el método de búsqueda de correspondencias y el método de decodificación de protocolo son mutuamente excluyentes pero esto no es así. La búsqueda de correspondencias se ha utilizado para hacer búsquedas en subconjuntos de firmas dentro de los métodos de decodificación de protocolo.

Hay que tener en cuenta que este método se puede referenciar de diferentes maneras: decodificación de protocolo, detección de anomalías de protocolo o validación de protocolo. Cada uno de estos métodos puede tener ligeras diferencias respecto a los demás. Por ejemplo la validación de protocolo o la detección de anomalías de protocolo más estrictas se centran solo en comprobar las reglas del RFC y si no se cumplen lanzar una alarma. ([10], [13])

Ventajas:

- Minimiza los falsos positivos si esta bien definido y se utiliza correctamente.
- Permite la comparación directa de un "exploit".
- Se adapta mejor a las posibles diferencias y variaciones que pueden surgir en un entorno concreto de un ataque, de forma más amplia y general.
- Las alarmas lanzadas sobre violaciones de protocolo definidas son completamente fiables.

Inconvenientes:

- Si el RFC es ambiguo puede darse un alto número de falsos positivos y deben de ser los programadores los que interpreten e implementen la solución adecuada. Estas zonas grises del método son bastante comunes.
- Este método requiere tiempos de desarrollo más largos para poder poner correctamente el programa de análisis de protocolo en funcionamiento.
- Si surge un nuevo tipo de "exploit" lo más probable es que el desarrollador del IDS tenga que crear código nuevo para manejarlo. En cambio con otros métodos el administrador del IDS puede desarrollar firmas optimizadas para el entorno donde esté trabajando.

3.1.6.1.4 Análisis basado en heurística

Las firmas basadas en heurísticas usan un tipo de lógica algorítmica sobre la cual basan sus decisiones de lanzar o no la alarma. Estos algoritmos son a menudo evaluaciones estadísticas del tipo de tráfico que están analizando.

Un buen ejemplo de este tipo de firma sería una firma que sirviera para detectar un barrido de puertos. Esta firma buscaría la presencia de un número máximo de puertos consultados o tocados en una única máquina. También puede restringirse más especificando en qué tipo de paquetes se está interesado (en este ejemplo serían

paquetes SYN). Adicionalmente, se puede requerir que todas las pruebas se originen desde una única fuente, e incluso que los paquetes SYN ACK válidos sean devueltos por la máquina que esta siendo probada.

Las firmas de este tipo reaccionan de forma diferentes en redes diferentes, esto puede ser el origen de falsos positivos si no se adapta el sistema adecuadamente, requiriendo algunas modificaciones conforme a la utilización de los patrones en la red que se está supervisando. Este tipo de firma se puede utilizar para buscar relaciones muy complejas así como sencillas, como en el ejemplo visto. ([10], [13])

Ventajas:

- Algunos tipos actividades maliciosas o sospechosas de serlo sólo se pueden detectar con este método.

Inconvenientes:

- Los algoritmos podrían requerir una modificación o afinación para ajustarse al tráfico de la red y así evitar falsos positivos.

3.1.6.2 Detección de anomalías

Los sistemas basados en anomalías se encargan de buscar o identificar comportamientos que se desvían de la normalidad en un sistema o en el tráfico de una red. La detección se basa en la suposición de que cuando se produce un ataque la actividad del sistema es diferente de la actividad normal del mismo y podemos, por tanto, identificar esas diferencias.

El mayor problema de esta metodología está en definir qué es lo normal. Algunos sistemas tienen definiciones fuertemente codificadas de lo que es normal, y en este caso podrían ser considerados como sistemas basados en heurística. Otros sistemas se han construido para aprender qué es lo normal, pero el principal problema de estos sistemas consiste en la eliminación de la posibilidad de clasificar incorrectamente el comportamiento anormal como normal. Otro problema es cómo el sistema debe lidiar con cómo diferenciar, si tenemos un patrón del tráfico que se ha aprendido y se ha asumido como normal, entre las desviaciones permisibles y ésas no permitidas o que representan tráfico de algún ataque.

El trabajo en este área se ha limitado sobre todo al ámbito académico, aunque hay algunos productos comerciales que utilizan métodos de detección basado en anomalías. Una subcategoría de este tipo de detección son los métodos de detección basados en perfiles. Estos sistemas basan sus alarmas en cambios en la manera que los usuarios o los sistemas interactúan en la red. Incurren en muchas de las mismas limitaciones y problemas que la categoría principal tiene para intentar deducir el cambio en el comportamiento. La información que estos sistemas proporcionan es generalmente muy poco específica y requieren una extensa investigación para poder adaptarla al contexto apropiado.

Un subtipo de la detección basada en anomalía es la basada en protocolo. Este método está relacionado con el de decodificación de protocolo. En este caso las anomalías no requieren ser aprendidas porque las definiciones de los protocolos son detecciones del protocolo bien definidas. Un ejemplo de una anomalía de protocolo sería si un campo en el protocolo tiene un valor inesperado.

En algunos casos, las fronteras entre las metodologías están borrosas, porque la mayoría de los motores de análisis de decodificación de protocolo alertan al usuario de la presencia de violaciones de protocolo que no se relacionan directamente con ningún ataque conocido, pero que es anómalo (por ejemplo, la detección del desbordamiento de buffer basada en longitud). Por lo tanto, en este ejemplo, el motor tiene cualidades de un sistema basado en anomalía.

Finalmente otro subtipo es la detección de anomalías mediante estadísticas. Se pueden identificar en la red mediante aprendizaje o enseñanza de las normas estadísticas para ciertos tipos de tráfico, por ejemplo, los sistemas que detectan las inundaciones del tráfico, tales como UDP, TCP o inundaciones ICMP. Estos algoritmos comparan el índice actual de la llegada del tráfico con una referencia histórica; de acuerdo con esto, los algoritmos alertarán de las desviaciones estadísticas significativas respecto de la media histórica. Normalmente, el usuario puede proporcionar el umbral estadístico para las alarmas.

Estos sistemas se consideran generalmente como la solución definitiva para los IDS, pero siguen siendo actualmente sujeto de numerosas investigaciones académicas con resultados no muy alentadores, que demuestran un éxito limitado.
También pueden ser utilizados para crear patrones de ataques que luego pueden ser utilizados por los detectores basados en firmas. ([10], [13])

En definitiva, el proceso general en un sistema de detección de anomalías parte de la utilización de diversas técnicas para crear perfiles de comportamiento normal, que sirvan de modelos de comportamiento. Las desviaciones, que resultan de comparar estos modelos con los comportamientos que se van sucediendo en el sistema, son sometidas a técnicas que deciden si ha habido o no indicios de alguna intrusión. Los perfiles que se crean están compuestos por conjuntos de métricas. Estas métricas están basadas en aspectos concretos del comportamiento del usuario, de la máquina o de la red.

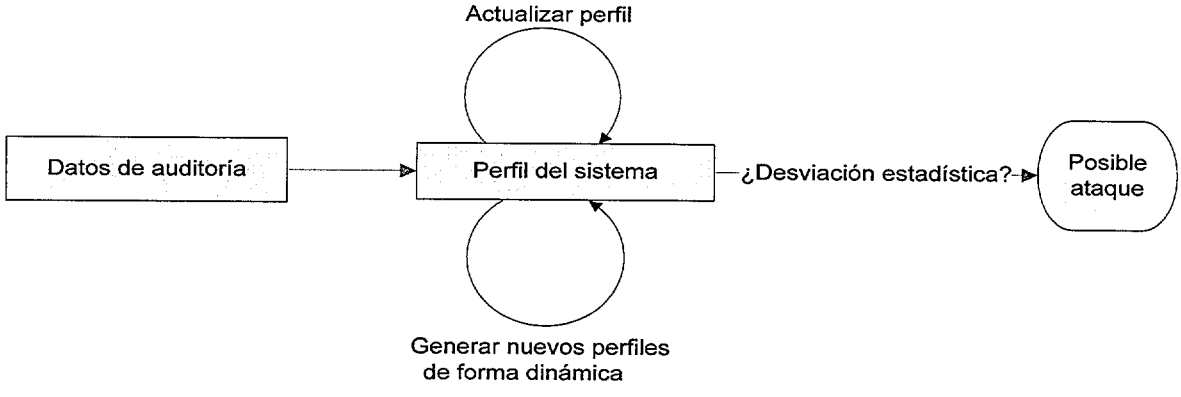


Figura 3: Modelo general de un detector de anomalías

Ventajas:

- Si este método es implementado correctamente puede detectar ataques desconocidos al adaptarse y aprender de la conducta del sistema.
- Tiene un mantenimiento menos costoso porque no se tienen que desarrollar nuevas firmas para los nuevos ataques.
- Producen información que se puede utilizar para crear nuevas firmas de ataques.

Inconvenientes:

- En general, estos sistemas no pueden dar datos de la intrusión con ninguna granularidad, es decir, no suelen identificar el tipo de ataques o intrusiones que detectan, ni dar información sobre ellos.
- Pueden producir un gran número de falsas alarmas causadas por el impredecible comportamiento de los usuarios y de las redes.
- Son difíciles de implementar y requieren entrenamiento para poder definir el comportamiento normal del sistema.
- Este método es altamente dependiente del ambiente en el cual los sistemas aprenden qué es lo normal.
- Los perfiles de conducta pueden ser gradualmente educados. Un atacante que sepa que sus actividades están siendo monitorizadas, puede cambiar paulatinamente su forma de comportamiento a lo largo del tiempo, para que cuando cometa una intrusión no sea reconocida como tal. Esta técnica es conocida como "session creep" (deslizamiento, o movimiento sigiloso de sesión).

3.1.6.3 Conclusión: ¿Qué método de detección es mejor?

La respuesta a la pregunta de qué método es el mejor depende de lo que se esté tratando de hacer y de las prioridades que se tengan respecto a lo que se quiere detectar. Por lo tanto para poder contestarla se debe hacer primero un estudio de las necesidades del sistema donde se quiera implantar el IDS.

En la actualidad se tiende a la utilización de varios métodos conjuntamente, de esta forma se pueden compensar las deficiencias de unos con las ventajas de los otros. En bastantes casos, como se menciona en las definiciones de los métodos, hay una falta de definición en las fronteras que separan las diferentes metodologías, llegando al punto de ser casi indistinguible. Como ya se ha explicado, el método de decodificación de protocolo utiliza para detectar los ataques de desbordamientos de buffer técnicas de

detección basadas en anomalías. También, los sistemas que utilizan el método de decodificación de protocolo utilizan técnicas de búsqueda de correspondencias y viceversa.

Por todo esto la conclusión que se saca es que en la actualidad no hay un método de detección mejor y que dependiendo de lo que se quiera detectar es mejor uno u otro. Por lo tanto la mejor solución es combinar todos los métodos existentes para poder detectar la mayor cantidad de intrusiones posibles.

En un futuro cercano los sistemas de detección de intrusos tendrán totalmente integrados los métodos análisis de usos indebidos (búsqueda de correspondencias, decodificación de protocolo y análisis heurístico) y las técnicas de análisis de anomalías.

3.1.7 - Técnicas de proceso de datos usadas en IDS

Dependiendo del tipo de sistema de análisis tomado para la detección de intrusos, se emplean varios mecanismos de proceso de los datos que son revisados por un IDS. En las siguientes líneas, se describen varias técnicas brevemente:

- *Sistemas expertos ("Expert systems"):* Trabajan sobre un sistema de reglas previamente definido que describe ataques. Toda la seguridad relacionada con los eventos incorporados en la auditoría se traduce en reglas de la forma if-then-else. Un ejemplo es el ComputerWatch de AT&T. ([45], [12], [44])

- *Análisis basado en firma ("Signature analysis"):* Es semejante a la técnica anterior. Este método se basa en el conocimiento previo de la forma del ataque. Transforman la descripción semántica de un ataque en el formato apropiado para realizar la auditoría. Así, las firmas del ataque se pueden encontrar en registros o secuencias de datos de entrada de una manera directa. Un escenario de un ataque se puede describir, por ejemplo, como secuencia de los eventos de una auditoría que un ataque concreto genera o como un modelo de los datos investigables que se capturan en el proceso de la auditoría. En este método se utilizan equivalencias abstractas de los datos obtenidos de la auditoría. La detección es lograda usando mecanismos que comparan secuencias de texto comunes. Es una técnica de gran alcance y como tal es empleada muy a menudo en sistemas comerciales (por ejemplo Stalker [46], Real Secure [40], NetRanger [48] y Emerald eXpert-BSM [49]). ([12], [44])

- *Redes De Petri Coloreadas ("Colored Petri Nets"):* Las redes de Petri coloreadas se utilizan para obtener las características comunes de los ataques almacenados en bases de conocimiento avanzadas y crear un patrón general para esos ataques. También se utilizan para representar los ataques gráficamente. El sistema IDIOT de Purdue University utiliza las redes de Petri coloreadas. Con esta técnica, es fácil que los administradores de sistemas agreguen nuevas firmas al sistema. Sin embargo, emparejar una firma compleja con los datos obtenidos de una auditoría puede que no sirva para nada. Esta técnica no se utiliza en sistemas comerciales.

- *Análisis de transiciones de estado ("State-transition analysis"):* En esta técnica un ataque se describe con el conjunto de las metas y de las transiciones que debe alcanzar un intruso para comprometer un sistema. Hacen uso de grafos o autómatas finitos para representar y utilizar patrones de ataque. ([12], [44])

- *Acercamiento al análisis estadístico ("Statistical análisis approach"):* Es un método frecuentemente usado (por ejemplo en el proyecto de SecureNet [50]). El comportamiento del usuario o del sistema es traducido a conjuntos de atributos y es medido por un número de variables en un cierto período de tiempo. Estas variables pueden ser: el login del usuario, el registro de estado de la máquina, el número de archivos accedidos en un período de tiempo, el uso del espacio del disco, de la memoria o de la CPU, etc. La frecuencia con la que se actualizan estas variables puede variar desde unos minutos hasta, por ejemplo, un mes. Los sistemas de almacenamiento guardan valores de cada variable que son usados para detectar si es superado el umbral predefinido.

Hoy en día, este enfoque simple no puede ajustarse a un típico modelo de comportamiento de usuario. Los enfoques que combinaban perfiles de usuario individuales con las variables de grupo también han resultado ineficientes. Por lo tanto, se usa un modelo más sofisticado que consiste en usar perfiles de usuario a corto y largo plazo. Estos perfiles se actualizan regularmente para adaptarlos a los cambios de comportamiento del usuario. Los métodos estadísticos se utilizan a menudo en implementaciones de IDS basados en perfiles de comportamientos de usuario normales. ([12], [44])

- *Redes Neuronales ("Neural Networks"):* Utilizan sus algoritmos de aprendizaje para aprender sobre la relación entre los vectores de entrada y de salida y para generalizarlos para extraer nuevas relaciones entre la entrada y la salida. El propósito principal de las redes neuronales en la detección de intrusos es aprender el comportamiento de los agentes en el sistema (por ejemplo usuarios y demonios). Algunos métodos estadísticos comparan parcialmente redes neuronales. La ventaja de usar redes neuronales sobre estadísticas reside en tener una manera simple de expresar relaciones no lineales entre las variables, y en aprender sobre las relaciones automáticamente.

Se han hecho experimentos con la predicción de una red neuronal sobre comportamientos del usuario. De los resultados se ha obtenido que el comportamiento de los super-usuarios de UNIX ("roots") es predecible (debido a que el funcionamiento de los procesos automáticos del sistema es muy regular). Con pocas excepciones, el comportamiento de la mayoría de los demás usuarios es también bastante predecible. Sin embargo las redes neuronales siguen siendo una técnica de cómputo intensiva, y no se utilizan extensamente en la comunidad de los sistemas de detección de intrusos.([12])

- *Identificación de la intención del usuario ("User intention identification"):* Esta técnica (utilizada solamente en el proyecto de SecureNet [50]) modela el comportamiento normal de los usuarios con el conjunto de tareas de alto nivel que tienen que realizarse en el sistema (en referencia a las funciones de cada usuario). Estas

tareas se toman como una serie de acciones, que alternativamente se comparan con los datos apropiados obtenidos de la auditoría. El analizador guarda un conjunto con las tareas que son aceptables para cada usuario. Siempre que se encuentre un emparejamiento erróneo, se produce una alarma. ([12])

- *Inmunología de la computadora ("Computer immunology"):* Las analogías con la inmunología conducen al desarrollo de una técnica que consiste en construir un modelo del comportamiento normal de los servicios de red de UNIX, más que de usuarios individuales. Este modelo consiste en secuencias cortas de llamadas de sistema hechas por los procesos. Los ataques que explotan defectos en el código de las aplicaciones normalmente realizan secuencias de ejecución inusuales. Primero, se recoge un conjunto de los datos de la auditoría para utilizar como referencia y así representar el comportamiento correcto o normal de los servicios. Luego, a la base de conocimiento creada se le añaden todas las secuencias de llamadas del sistema antes mencionadas. Estos patrones se utilizan para controlar continuamente las llamadas del sistema y comprobar si la secuencia generada está registrada en la base de conocimiento, si no se generará una alarma.

Esta técnica tiene potencialmente un número de falsos positivos muy bajo, pero para esto la base de conocimiento debe de ser bastante amplia y buena. Su principal desventaja es la imposibilidad de detectar errores en la configuración de los servicios de red. Si un atacante utiliza acciones legítimas sobre el sistema para obtener acceso no autorizado, no se generará ninguna alarma. ([12])

- *Aprendizaje de la máquina ("Machine learning"):* Es una técnica basada en inteligencia artificial, almacena la secuencia de comandos que va introduciendo el usuario de forma vectorial y se utiliza como referencia para crear el perfil de comportamiento del usuario. Los perfiles se agrupan en una librería de comandos de usuario que tienen ciertas características en común. ([12], [44])

- *Minería de datos ("Data mining"):* Generalmente son un conjunto de técnicas que extraen modelos de datos de grandes cantidades de información. Entre las técnicas más usadas en la minería de datos está la llamada de Clasificación, esta técnica se asocia a los árboles de decisión y es usada en la detección de intrusos. Los árboles de decisión se utilizan para detectar anomalías en los datos de auditorías.

Otra técnica utilizada es la Segmentación, usada para extraer patrones de ataques desconocidos. Esta técnica consiste en comparar los patrones extraídos de un conjunto de auditorías con los ataques desconocidos almacenados.

La última de las técnicas más usadas en la minería de datos consiste en encontrar reglas de asociación. Esta técnica identifica relaciones y correlaciones en el cuerpo de los datos permitiendo extraer características desconocidas de los nuevos ataques o construir patrones de comportamiento normales.

La detección de anomalías genera, a menudo, falsos positivos. Con la minería de datos es fácil correlacionar los datos relacionados con las alarmas con los datos de extraídos de la auditoría, reduciendo de este modo considerablemente el índice de falsos

positivos. También, este método destaca por poder procesar grandes cantidades de información sin problemas. Sin embargo son menos útiles para el análisis del tráfico de la red. ([9], [12], [44])

3.1.8 - Tipos de respuesta de los IDS frente a los ataques

Cuando los IDS detectan en la fase análisis un posible ataque o comportamiento anómalo es cuando entra en funcionamiento el componente de respuesta del IDS. Dependiendo de la forma de respuesta se pueden clasificar en dos tipos: pasivos y activos. La mayoría de los IDS actuales permiten la utilización de ambos tipos de respuesta. ([9], [11])

3.1.8.1 Pasivos

Este tipo de respuesta se limita a informar, registrar o anunciar del evento acaecido. Se pueden dividir en dos subtipos:

- Alarmas y notificaciones: Pop-ups, notificaciones a teléfonos móviles o buscas, notificación por e-mail (desaconsejada por que el atacante podría interceptar el e-mail).

- SNMP traps: Se envían mensajes de alarmas a consolas centrales remotas de gestión de red. La mayor ventaja de este sistema es que al estar centralizado el control de la red la respuesta al posible ataque aplicarse a toda la infraestructura de la red más rápidamente.

3.1.8.2 Activos

Las respuestas activas además de informar sobre el evento actúan dependiendo del tipo de ataque detectado. Se dividen en tres tipos de respuesta activa:

- Recopilar información adicional: Se crean informes y se sensibiliza más el nivel de las fuentes.

- Detener el ataque: Se intenta detener el ataque mediante contramedidas como bloquear las direcciones IP de procedencia del ataque, inyectar paquetes TCP RST para terminar la conexión, reconfigurar el mecanismo de filtrado de paquetes o en casos extremos, deshabilitar las interfaces de red de donde proceda el ataque, etc.

- Contraatacar: Esta respuesta no es aconsejable porque puede tener consecuencias legales, se puede atacar a inocentes y hacer que el atacante endurezca sus acciones. No obstante si se implementa este tipo de respuesta debería contar con la supervisión humana.

3.1.9 - Casos especiales y complementos

Existe una serie de herramientas de seguridad que complementan a los detectores de intrusos haciendo más efectiva la detección de las intrusiones. Por otro lado, algunos sistemas de seguridad ofrecen soluciones tan similares que se pueden considerar como casos especiales de los sistemas de detección de intrusos.

3.1.9.1 Honeypots

Un "honeypot" (o tarro de miel) es un sistema trampa que ayuda y complementa a los sistemas de detección de intrusos aportando información muy útil para mejorar los métodos de detección y para crear nuevos patrones de ataques.

Los sistemas trampa están diseñados para reclamar la atención de los posibles intrusos, estudiar sus actividades y así poder aprender sus métodos. Suelen imitar el comportamiento de aquellos sistemas que pueden ser de interés para un intruso, por ejemplo conteniendo información aparentemente real e importante.

Los "honeypots" también cuentan con mecanismos para que un atacante no pueda acceder al resto de la red y para monitorizar las actividades de los atacantes. Lógicamente el intruso no debe percatarse de que esta en un sistema trampa ni de que está siendo monitorizado. ([9])

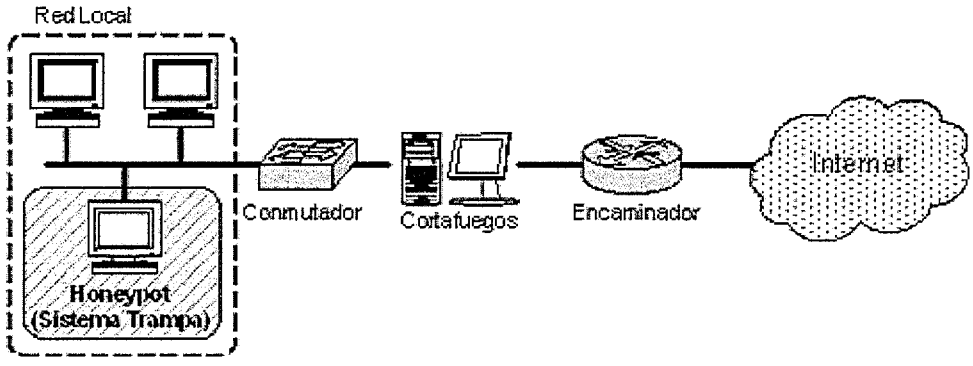


Figura 4: Ejemplo de un "Honeypot" (Sistema trampa)

Ventajas:

- Atraen a los atacantes a sistemas en los que no pueden causar daños.
- Las acciones de los atacantes pueden ser más fácil y extensamente monitorizadas pudiendo utilizar estos resultados para refinar los modelos de ataques y mejorar las protecciones de los sistemas.

- Registran poco volumen de datos pero de mucho valor. Al estar dedicado únicamente a monitorizar las intrusiones registra poca información, pero toda está información suele ser útil porqué está relacionada con actividades hostiles.

- Utilizan pocos recursos porque solo se dedican a recopilar información de las actividades que se realizan en el sistema y no la analizan.

- Son muy sencillos de utilizar, en la mayoría de estas herramientas basta con instalarlas y esperar.

Desventajas:

- Si un sistema trampa no es atacado carece de valor. Por lo tanto si un atacante logra identificar uno de estos sistemas puede anular toda su efectividad evitándolos.

- Si un sistema trampa es atacado con éxito puede ser utilizado por el intruso para acceder al resto de sistemas de la red. Cuanto más sencillo sea el sistema trampa menores riesgos implica.

- No esta demostrado que sean una tecnología de seguridad útil a gran escala.

- Tienen implicaciones legales no muy claras.

3.1.9.2 Padded Cell

Los sistemas "padded cell" o células de aislamiento tienen una metodología parecida a los sistemas trampa pero mantienen algunas diferentas. Funcionan de forma conjunta con dispositivos que cuentan con capacidades de enrutamiento y detección de intrusiones. Al detectar una intrusión estos dispositivos redirigen la intrusión hacia una máquina especial llamada célula de aislamiento.

Las células de aislamiento ofrecen al atacante un entorno idéntico a uno real. Sin embargo, al estar protegido del resto de la red, no causa daños. En muchas ocasiones estas máquinas aisladas consisten en espejos de sistemas de producción reales, para proporcionar un escenario más creíble. Al igual que los sistemas trampa se pueden utilizar para estudiar los métodos utilizados por los intrusos.

Un producto que esta enfocado para trabajar con células de aislamiento es el "Bait and Switch", desarrollado por J. Whitsitt y A. Gonzalez [24]. Se instala en un sistema con tres interfaces de red. Cuando llega tráfico hostil se envía a la célula de aislamiento (que normalmente es una copia parcial del sistema real con datos ficticios) y el resto del tráfico se trata normalmente.

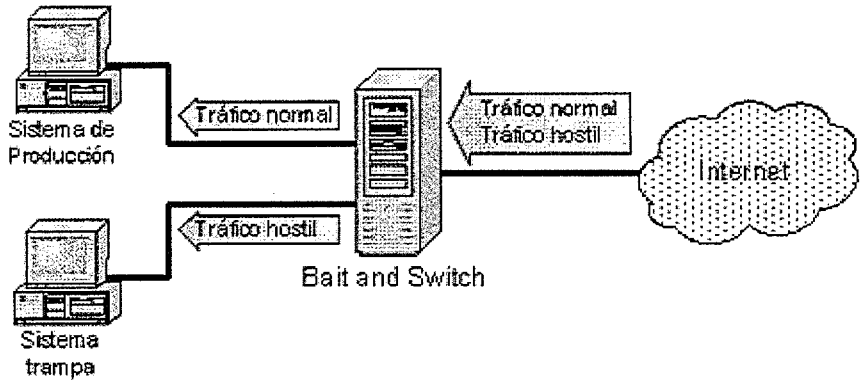


Figura 5: Procedimiento general de "Bait and Switch"

Las ventajas e inconvenientes de los "padded cell" se pueden considerar iguales que las de los "honeypots" ya que son sistemas similares. ([9])

3.1.9.3 Sistemas de prevención de intrusos

Las carencias en los productos de seguridad actuales para defenderse de los ataques han conducido a desarrollar una nueva clase de productos de seguridad conocidos como Sistemas de Prevención de Intrusos (IPS). Estos sistemas reúnen la capacidad de bloqueo de un cortafuegos y la capacidad de análisis de un IDS.

Son sistemas que tienen mecanismos de defensa preactiva (los IDS son reactivos), diseñados para detectar los paquetes dañinos dentro del tráfico normal de la red (los cortafuegos no detectan este tipo de paquetes). Es decir, detienen cualquier ataque antes de que pueda causar algún daño.

Los IPS son considerados un caso especial de los IDS porque ambos comparten la misma metodología básica. La única diferencia es que se les ha añadido las capacidades de un cortafuegos. De hecho, algunos expertos ya los consideran como un producto diferente, como la siguiente generación de IDS.

Atendiendo a la fuente de datos que utilizan, los IPS se dividen en tipos: basados en máquina (HIPS) y basados en red (NIPS). Los NIPS a su vez tienen varios modelos con algunas diferencias en la implementación.

3.1.9.3.1 IPS basados en máquina

Como con los sistemas detectores de intrusos basados en máquina, los IPS basados en máquina instalan los agentes directamente en la máquina protegida. Analizan detalladamente el núcleo del sistema operativo y los servicios, supervisando e interceptando las llamadas del sistema al núcleo de sistema operativo o las APIs para prevenir ataques o registrarlos.

Pueden también supervisar secuencias de datos y el entorno específico de una aplicación particular (por ejemplo las localizaciones de los archivos o la configuración del registro de un servidor Web) para protegerlas de ataques genéricos para los cuales no existe ninguna firma.

Una desventaja potencial con este acercamiento es que, dado la integración que necesariamente tiene con el sistema operativo de la máquina protegida, las mejoras o actualizaciones del sistema operativo pueden causar problemas, causando el mal funcionamiento del IPS.

Puesto que un agente de un HIPS intercepta todas las peticiones que se hacen sobre el sistema que protegen, deben cumplir ciertos requisitos como: ser muy confiable, no afectar negativamente al funcionamiento de la máquina, y no bloquear el tráfico legítimo. Cualquier HIPS que no cumpla estos requisitos mínimos no es recomendable instalarlo, aunque detecte y detenga con eficacia los ataques.

3.1.9.3.2 IPS basados en red

Este tipo de IPS también son conocidos como "Gateway IDS" (GIDS), están formados al menos por dos interfaces de red, una para monitorización interna y otra para externa, e integran características de cortafuegos y de IDS.

Entre los modelos más extendidos destacan cuatro: IDS en red en modo "in-line", conmutador de nivel siete, conmutador híbrido y aplicación engañosa.

3.1.9.3.2.1 IDS basado en red, en modo "In-line"

Es el modelo más común de implementación de los NIDS. Se utilizan dos dispositivos de red, uno para interceptar el tráfico de la red y otro para hacer las labores de gestión y de administración.

Con la interfaz de red que se utiliza para la monitorización se pincha el segmento de red deseado para poder capturar el tráfico. Esta interfaz no suele tener asignada ninguna dirección IP, con esto se consigue que ningún elemento de la red le envíe paquetes o que el NIDS pueda responder. Así se disminuye la posibilidad de ser detectado.

Con el modo "in-line" (en línea) el NIDS se sitúa entre la red que se desea proteger y el resto (como un puente). Utiliza tres interfaces de red, una para recibir el tráfico externo, otra para enviar el tráfico a la red a proteger y la tercera para las labores de administración y de gestión. De esta forma el IPS puede controlar todo el tráfico de red que pasa por el segmento decidiendo que hacer con él, e incluso gestionar el ancho de banda. En este caso la detección de ataques depende directamente de los métodos utilizados por el NIDS. ([9])

3.1.9.3.2.2 Conmutador de nivel siete

Estos dispositivos son "switches" adaptados para trabajar con grandes anchos de banda. Se suelen utilizar para balancear la carga de una aplicación entre varios servidores. Para ello, examinan la información de la aplicación (HTTP, FTP, DNS, etc.) para tomar decisiones de encaminamiento.

Los fabricantes de estos dispositivos les han añadido la capacidad de protegerse de los ataques de denegación de servicio y de los de denegación de servicio distribuida. El método utilizado para detectar los ataques es similar al utilizado por los NIDS por lo que presentan las misma desventajas. Estos IPS trabajan con facilidad en redes de alta velocidad, detectan y bloquean ataques de denegación de servicio con mayor efectividad que otros IPS. Otra de las ventajas de estos dispositivos, que no se encuentra en otros IPS, es que permiten redundancia. Pueden ser implementados en modo de espera en caliente (si el dispositivo principal falla se activa el secundario) o en modo de balance de carga (se puede repartir el trabajo entre varios dispositivos). ([9])

3.1.9.3.2.3 Conmutador híbrido

Este tipo de dispositivos son una combinación de los IDS basados en aplicación y los conmutadores de nivel siete explicados en el punto anterior. Son dispositivos hardware instalados de la misma forma que los conmutadores de nivel siete, pero no utilizan conjuntos de reglas como los NIDS, sino un método de detección basado en políticas similar al de los IDS basados en aplicación, con el cual analizan el tráfico de la red en busca de información definida en las políticas aplicadas.

Los conmutadores híbridos tienen conocimientos sobre el servidor que protegen (servidor FTP, Web, SMTP, etc.), como un conmutador de nivel siete, pero también tienen conocimiento de las aplicaciones que hay instaladas sobre él. Además, como con los IDS basados en aplicación bloquea todas las acciones no definidas como permitidas. Estos IPS pueden combinarse con un conmutador de nivel siete para desviar solo la información que se considere maliciosa al conmutador híbrido, y así evitar una sobrecarga de trabajo.

Una ventaja a destacar en estos productos es que se pueden configurar importando los resultados de un analizador de vulnerabilidades utilizado contra el sistema a proteger. Esto permite implementar de forma rápida y efectiva este tipo de sistemas. ([9])

3.1.9.3.2.4 Aplicación engañosa

Los IPS basados en aplicación engañosa ("Deceptive application") tienen un particular enfoque que comprende dos fases. La primera consiste en la monitorización del tráfico de la red para crear un modelo de la actividad normal (explicado en el análisis basado en anomalía). Durante la segunda fase, si el IPS observa algún intento

de conexión a algún servicio que no existe, devuelve una respuesta falsa hacia el atacante. La respuesta es marcada por el IPS, de tal forma que cuando el posible atacante decida volver a intentar alguna conexión, el IPS reconocerá su marca y lo bloqueará. De esta manera, el ataque es detectado y anulado antes de que ocurra. También se pueden introducir marcas en el campo de datos de los paquetes, permitiendo la detección de ataques contra servicios que existen.

Uno de los principales problemas de estos sistemas es que el atacante puede descubrir el la forma en que el IPS marca los paquetes y así desmarcarlos antes de realizar el ataque, para evitar la protección. ([9])

3.2 - Escáneres de vulnerabilidades

Un escáner de vulnerabilidades ("Vulnerability scanner") es una herramienta que realiza un conjunto de pruebas (generalmente ataques) para determinar si una red o una máquina tienen fallos de seguridad.

Como ya se mencionó en la introducción del capítulo los escáneres de vulnerabilidades son proactivos y tienen un enfoque estático, a diferencia de los detectores de intrusos que tienen un enfoque dinámico. A pesar de esto los escáneres de vulnerabilidades y los detectores de intrusos mantienen muchas características en común y es por esto que se pueden considerar los escáneres de vulnerabilidades como un caso especial de los detectores de intrusos aunque más bien se complementan.

En general los escáneres de vulnerabilidades sólo pueden detectar las vulnerabilidades que tiene almacenadas en su base de datos y en el momento en que se ejecuta el escáner sobre el sistema objetivo. La información obtenida después de cada sesión es muy útil para mejorar la seguridad de los sistemas.

3.2.1 - Sistemas de análisis de vulnerabilidades

La principal forma de clasificar los sistemas de análisis de vulnerabilidades es mediante la fuente de datos al igual que con los IDS. Pueden ser:

- Análisis basados en máquina.
- Análisis basados en red.

También se pueden clasificar mediante el nivel de confianza del que hace uso el analizador de vulnerabilidades. Es decir, mediante el hecho de utilizar o no credenciales de sistema (tales como contraseñas u otro tipo de identificación y autentificación que concedan el acceso a partes internas del sistema) durante el proceso de análisis. Existen

dos tipos:

- Con acreditaciones ("credentialed")
- Sin acreditaciones ("non credentialed")

En los siguientes puntos se describen los sistemas de análisis mediante la primera clasificación basada en la fuente de datos.

3.2.1.1 Análisis de vulnerabilidades basado en máquina

Este tipo de análisis fue el primero en utilizarse en la evaluación de vulnerabilidades. Utiliza elementos tales como ajustes de configuración, contenidos de ficheros u otros tipos de información del sistema para la detección de vulnerabilidades. Esta información se puede obtener normalmente mediante consultas al sistema o a través de la revisión de los diferentes atributos del sistema. Mientras esta información es recopilada se asume que el analizador de vulnerabilidades tiene acceso autorizado al sistema, por lo que también es llamado análisis con acreditaciones. También se denomina evaluación pasiva.

Dependiendo del sistema operativo que se utilice la información es obtenida de diferente forma, por ejemplo en los sistemas UNIX se obtiene a nivel de "host" o de dispositivo. En cambio en sistemas Windows se permiten realizar llamadas nativas de forma local o remota, según las credenciales utilizadas. Por lo que la correspondencia entre sistemas basados en máquina y con acreditaciones no siempre se cumple.

Las vulnerabilidades que se suelen encontrar con la evaluación basada en máquina están relacionadas con ataques de escalada de privilegios. Estos ataques buscan obtener permisos de super usuario ("root") en sistemas UNIX o de administrador en sistemas Windows.

Uno de los principales inconvenientes en los analizadores de vulnerabilidades basados en máquina es que el motor del analizador está muy relacionado con el sistema operativo que evalúa, esto hace que su mantenimiento sea costoso y complica su administración en sistemas heterogéneos. Otro inconveniente es que al utilizar credenciales éstas deben de ser protegidas convenientemente, al igual que la información accedida mediante las mismas, para evitar que sean objeto de ataques.

Un ejemplo de este tipo de analizador de vulnerabilidades es el "Cerberus Internet Scanner (CIS) ([32]). Es un programa gratuito, simple y eficaz que comprueba una lista de vulnerabilidades conocidas para sistemas Windows e informan en poco tiempo de las vulnerabilidades existentes.

3.2.1.2 Análisis de vulnerabilidades basado en red

Los análisis de vulnerabilidades basados en red han ido aumentando su popularidad desde su aparición hace ya algunos años. Este tipo de analizador requiere una conexión de red con la máquina a ser evaluada, mediante la cual obtiene la información necesaria. Realiza diversos ataques contra la máquina objetivo y registra las respuestas obtenidas o simplemente realiza sondeos sobre los diversos objetivos para deducir debilidades de las respuestas obtenidas. Estos ataques o sondeos no necesitan tener acceso al sistema de la máquina objetivo por lo que a este tipo de análisis se le suele denominar análisis sin acreditaciones. Además, al basarse en ataques y sondeos sobre el objetivo también recibe el nombre de evaluación activa. No obstante, del mismo modo que pasa con los sistemas de análisis basados en máquina y los autenticados, la correspondencia entre los sistemas basados en red y los no autenticados no siempre se cumple.

Existen dos técnicas o métodos que se suelen utilizar para la evaluación de vulnerabilidades basadas en red:

- Prueba por explotación ("Testing by exploit"): Esta técnica consiste en lanzar ataques reales contra el objetivo. Estos ataques están programados normalmente mediante guiones de comandos o lenguajes de "scripting". Una vez lanzado un ataque se espera a recibir una señal que indica si el ataque ha tenido éxito o no. Estos métodos no llegan a aprovechar las vulnerabilidades para acceder al sistema, pero al ser técnicas muy agresivas sí pueden afectar al funcionamiento del sistema, sobre todo cuando se prueban ataques de denegación de servicio.

- Método de inferencia ("Inference Methods"): Con este método el sistema no explota vulnerabilidades, sino que busca indicios que indiquen que se han realizado ataques. Es decir, busca resultados de posibles ataques en el objetivo. Este método es menos agresivo que el anterior pero también es menos específico a la hora de dar los resultados. Algunos ejemplos de técnicas de inferencia son la comprobación de las versiones del sistema y de las aplicaciones para determinar si existe alguna vulnerabilidad, la comprobación del estado de determinados puertos para descubrir cuáles están abiertos, y la comprobación de conformidad de protocolo mediante solicitudes de estado.

El ejemplo mas conocido de este tipo de analizadores de vulnerabilidades es Nessus [29]. Es un escáner de vulnerabilidades de Software Libre con licencia GNU ("General Public License"). Su predecesor fue SATAN [33] otro escáner de vulnerabilidades del que mantiene la forma de trabajar basada en el modelo cliente-servidor.

3.2.1.3 La técnica "Password cracking"

Otro método utilizado por los escáneres de vulnerabilidades es el conocido como "Password cracking". No tiene nada que ver con los demás métodos explicados en los puntos anteriores por lo que se puede comentar separadamente.

Esta técnica se basa en intentar romper contraseñas con el propósito de determinar el grado de calidad de las mismas. En el proceso se utilizan funciones relacionadas con el sistema de autentificación de usuarios del sistema operativo. Para intentar adivinar las contraseñas se puede utilizar la fuerza bruta (técnica basada en ir probando todas las combinaciones una a una) o la búsqueda por diccionario que va probando palabras cogidas de una lista.

La calidad de las contraseñas depende de la longitud de la misma así como de la variedad de caracteres utilizados (mayúsculas, minúsculas, números, símbolos especiales, etc.). Cuanto más larga y mayor sea la variedad de caracteres utilizados en la contraseña más difícil será romperla y por lo tanto de mayor calidad será.

Está técnica esta incluida y es utilizada por la mayoría de los productos de escáneres de vulnerabilidades actuales.

3.2.2 - Ventajas e inconvenientes

Los escáneres de vulnerabilidades tienen valiosas características de las que carecen los enfoques más dinámicos. Son herramientas muy útiles a la hora de ampliar la seguridad de un sistema. Pero, como es habitual en el ámbito de la seguridad informática, no es la solución definitiva, por lo que siempre es recomendable utilizar otra herramienta o sistema que complemente sus carencias.

3.2.2.1 Ventajas

- Mejoran de forma significativa la seguridad de un sistema, especialmente en entornos en los que no se cuenta con un sistema de detección de intrusos.

- Proporcionan las pruebas necesarias para documentar el estado de la seguridad de los sistemas en el comienzo de un proyecto o auditoría de seguridad y sirven para restablecer la base inicial del sistema de seguridad cuando ocurren cambios importantes.

- Si los escáneres de vulnerabilidades se utilizan regularmente, pueden detectar los cambios en el estado de la seguridad de un sistema, informando a los encargados de la seguridad de los problemas que requieren corrección.

- Facilitan a los encargados de la seguridad y los administradores de sistema la comprobación con minuciosidad de cualquier cambio que realicen en el sistema, asegurando que en la corrección del conjunto de problemas de la seguridad del sistema, no crean otro conjunto de problemas.

- Un escáner de vulnerabilidades reduce eficazmente los fallos de seguridad más comunes de un sistema. Alarma de forma precisa de muchos problemas de configuración que se le pueden pasar por alto a un administrador de sistemas o a un gestor de seguridad.

- Los analizadores basados en máquina, que son dependientes del sistema operativo, están mejor adaptados a su objetivo. Esto les permite identificar de forma más eficaz que otros sistemas, más generales, ataques o signos de intrusiones particulares.

3.2.2.2 Inconvenientes:

- Los analizadores basados en máquina, debido a su dependencia del sistema operativo que evalúan, son más costosos y complicados de gestionar.

- Los analizadores de vulnerabilidades basados en red son independientes de la plataforma, pero también son menos exactos y propensos a emitir falsas alarmas en sus resultados.

- Si se utilizan escáneres de vulnerabilidades en sistemas informáticos en los cuales se está ejecutando algún sistema de detección de intrusos, los análisis de vulnerabilidades pueden ser bloqueados por el IDS. En estas situaciones se corre el riesgo de entrenar erróneamente a los detectores de intrusos para que ignoren ataques reales.

- Algunas pruebas basadas en red, como los ataques de denegación de servicio pueden llevar a provocar la caída del objetivo. Este tipo de pruebas deben hacerse de forma controlada, conociendo de antemano los posibles efectos negativos que pueden tener.

- Las organizaciones que utilizan escáneres de vulnerabilidades deben asegurar que sus pruebas están limitadas a los sistemas dentro de sus límites políticos o de control de la dirección. Los asuntos de privacidad deben ser tenidos en cuenta, especialmente cuando los datos personales del empleado o del cliente se incluyen en las fuentes de información.

3.2.3 Herramientas de escáner de vulnerabilidades

Entre las herramientas dedicadas a analizar vulnerabilidades existen un grupo que destacan sobre el resto. Estas herramientas se caracterizan por detectar gran cantidad de vulnerabilidades, presentar informes claros y completos, tardar poco tiempo en analizar los sistemas y ser relativamente sencillas de usar. Entre estas aplicaciones están ISS Internet Scanner [25], GFI LANguard Network Security Scanner (N.S.S.) [74], Retina Network Security Scanner [26], Nessus [29] o SAINT [28].

ISS Internet Scanner [25] empezó como una herramienta de libre con código abierto en 1992. En la actualidad es uno de los mejores y más completos analizadores de vulnerabilidades y también de los más caros del mercado.

El N.S.S. [74] es un de los analizadores de vulnerabilidades más extendidos, destaca por tener un conjunto de herramientas que complementan las tareas de análisis de vulnerabilidades y por poder ser modificado para adaptarlo a las características de cada sistema informático.

Retina [26] destaca por utilizar una gran base de datos de firmas digitales de vulnerabilidades que es actualiza frecuentemente y por ser unos de los escáneres de vulnerabilidades más rápidos. También tiene múltiples opciones para poder adaptarlo a los sistemas que analiza.

Nessus [29] al igual que las anteriores herramientas analiza todas las máquinas de una red e informa de cualquier vulnerabilidad encontrada. Destaca por ser una herramienta libre, de código abierto y multiplataforma lo que la ha llevado ser uno de los escáneres de vulnerabilidades más conocidos. Está herramienta se explica con más detalle en el siguiente capítulo.

SAINT [28] sólo está disponible para plataformas UNIX, a diferencia de Internet Scanner [25] o Retina [26] que sólo están disponibles para las plataformas Windows. Este escáner de vulnerabilidades esta basado, al igual que SARA [47] (un escáner bastante difícil de utilizar) en SATAN [33] (uno de los primeros escáneres de vulnerabilidades). Destaca por ofrecer uno de los más completos informes entre este tipo de herramientas y por realizar un análisis que consiste en ir aprendiendo y guardando información de la máquina objetivo en los primeros pasos del análisis, para luego utilizar esta información en los pasos posteriores.

Las siguientes herramientas aunque no llegan al nivel de las anteriores presentan características interesantes por lo que merecen la pena ser mencionadas:

- NeWT [75] es una versión de Nessus [29] sólo para plataformas Windows. Tiene dos versiones una gratuita y otra de pago. Se caracteriza por ser muy sencilla de utilizar y por ser bastante potente aunque bastante lento en el análisis de las vulnerabilidades.

- TyphonIII [76] es un escáner de vulnerabilidades muy completo que contiene varios módulos que otras soluciones comerciales no tienen, como análisis de ataques de SQL-injection, SSL Web, XSS, POP3, SMTP, etc.

- NetIQ Vulnerability Manager [77] requiere instalar agentes distribuidos en todas las máquinas objetivo. Utiliza una extensa base de datos de vulnerabilidades y presenta un informe donde recoge todos los datos descubiertos durante el escaneo.

Dentro de los escáneres de vulnerabilidades existen herramientas que sólo se centran en escanear servidores Web. Dentro de este grupo de herramientas se encuentra Whisker [78] que utiliza una librería propia (Libwhisker) que se centra vulnerabilidades de servidores http, particularmente en "scripts CGI". Esta librería es también utilizada por otro escáner de vulnerabilidades llamado Nikto [79]. Nikto [79] es más potente, está más actualizado y puede ser optimizado con mayor facilidad.

N-Stealth Security Scanner [80] es una herramienta comercial que es actualizada con mayor frecuencia que los escáneres de vulnerabilidades libres tales como Whisker [78] y Nikto [79]. Tiene una base de datos de más de 30.000 vulnerabilidades y "exploits" para escáner remotamente los servidores y es solo para plataformas Windows.

Por último hay escáneres que sólo se centran en escanear una red para dar información sobre los puertos de las máquinas de la red, la ruta hasta una máquina remota o de las máquinas que están conectadas a un servidor. Estos escáneres se llaman "portscanners" escáneres de puertos y suelen ser utilizados por administradores expertos para detectar vulnerabilidades. Un ejemplo de este tipo de escáner es NScan [81].

3.3 - Métricas de Seguridad

En los últimos tiempos se han popularizado las métricas de seguridad por la exigencia desde la alta dirección de las organizaciones de disponer de visibilidad del estado de seguridad informática y su evolución, un cuadro de mando de la seguridad. El cuadro de mando refleja los valores asociados a cada uno de los parámetros relacionados con la seguridad, o bien un valor final ponderado que representaría el nivel de seguridad de la organización. De esta forma podrá monitorizarse el estado de la seguridad de la empresa y saber, no sólo que en un determinado momento es bueno o malo sino, lo que es más importante, medir las variaciones que a lo largo del tiempo se puedan producir y que harán que dicho nivel de seguridad aumente o disminuya.

3.3.1 - Introducción

La Real Academia de la Lengua [34] define métrica como el conjunto de preceptos y reglas necesarios para hacer algo.

Teniendo en cuenta la definición de métrica de la RAE las métricas de seguridad se pueden definir como el conjunto de preceptos y reglas necesarios para poder medir de forma real el nivel de seguridad de una organización.

Para poder definir más ampliamente una métrica de seguridad lo primero que hay que preguntarse es como se mide la seguridad informática de una organización. Las respuestas más sencillas e inmediatas a esta pregunta pueden ser, por ejemplo, medir el número de ataques sufridos en un período de tiempo concreto, o el número de veces que un virus ha entrado en la organización. Pero con estas medidas solo obtenemos un valor que no nos responde a las preguntas que nos interesan como: ¿Qué valor es el que proporciona seguridad a la organización?, ¿Qué número de intrusiones son aceptables en un determinado período de tiempo? Y por otro lado: ¿Qué tipo de herramientas o dispositivos son más seguros o añaden más seguridad a la empresa? Las respuestas a todas estas preguntas dependerán en gran medida del tipo de organización, de los criterios de seguridad de la misma, de los sistemas potencialmente afectados, del tiempo de reacción para restaurarlos, del impacto producido por la intrusión, etc.

Teniendo en cuenta el párrafo anterior se pueden sacar diversos factores a medir como: vulnerabilidades, riesgo, impacto, tecnología implantada, etc.

La definición de las métricas deberá tener en cuenta aspectos que se detallan a continuación a partir de tres preguntas clave:

- ¿Para qué queremos la métrica?:
    - Para medir la evolución de la seguridad de un sistema o producto.
    - Para medir la efectividad de una aproximación de protección y prevención.
    - Para medir la capacidad o habilidad de la organización en las tareas de seguridad.
    - Para tener datos elocuentes sobre el estado de seguridad de la organización.
    - Para saber dónde hay que hacer hincapié en la mejora de la seguridad.
- ¿Qué queremos medir?:
    - La cantidad y tipo de amenazas.
    - La calidad de las respuestas a las amenazas y ataques.
    - Los incidentes y sus impactos.
    - Las vulnerabilidades y puntos débiles.
- ¿Cómo podemos medirlo?:
    - ¿Cómo convertimos la información en datos válidos?
    - ¿Qué fuentes y herramientas necesitamos?

La dificultad en la definición de una métrica está precisamente en la dificultad de concretar estos tres factores.

Esta dificultad se debe en gran medida, a que existen diferentes áreas funcionales dentro de la seguridad informática, para las cuales se deben tener en cuenta diferentes aspectos que determinan el grado de seguridad en las mismas. Algunos ejemplos de métricas sobre estas áreas son:

- *Evaluación de riesgos:* Se comprueba el porcentaje de sistemas de la organización a los que la administración ha realizado una revisión formal de riesgos documentándola y revisándola.

- *Prueba de vulnerabilidades:* Se comprueba el porcentaje de sistemas cuyos controles de seguridad fueron probados y evaluados el año anterior.

- *Respuesta a incidentes:* Se mide el tiempo promedio que transcurre desde que se descubre una vulnerabilidad o punto débil hasta que se implanta una acción correctiva. Refleja la capacidad de respuesta del equipo para actuar después de que se descubran las vulnerabilidades.

- *Protección de la infraestructura:* Se refiere al porcentaje de sistemas que tienen instalados los últimos parches u otras protecciones.

- *Control de acceso:* Se comprueba el porcentaje de sistemas que han definido e implantado políticas de control de acceso y determinan si estas políticas cumplen la política de la entidad.

- *Entrenamiento en Seguridad Informática:* Se mide el porcentaje de empleados con responsabilidades importantes en seguridad, como privilegios de acceso a los sistemas o a la información, que han realizado comprobaciones en segundo plano y han recibido entrenamiento especializado.

- *Cumplimiento de las reglamentaciones:* Comprobación del cumplimiento de las métricas activas en la organización y número de incidentes relacionados con las mismas.

Básicamente, para cada área se debe decidir lo que se va a comprobar, cómo se va a hacer y la frecuencia con que se van a presentar los informes correspondientes teniendo en cuenta los recursos disponibles.

3.3.2 - Justificación de la necesidad de métricas de seguridad

La necesidad de definir e implantar métricas es algo habitual en distintos ámbitos dentro de los Sistemas Informáticos. Hasta hace algunos años esta necesidad se ha mantenido un poco al margen del ámbito de la seguridad informática lógica, aunque sin duda debe ser tenida en cuenta. Existen motivos claros que así lo justifican, como por ejemplo:

- Para conocer el estado de la seguridad de una forma clara y concisa y permitir el crecimiento dinámico de los sistemas de seguridad de acuerdo con las necesidades reales de la organización.

- Para anticiparse a las necesidades, de forma que puedan preverse las inversiones necesarias para garantizar, al menos, el cumplimiento de los objetivos de seguridad de la organización.

- Para justificar de cara a los departamentos financieros de las organizaciones el gasto en seguridad informática, mostrando de una forma clara la relación existente entre dicho gasto y el aumento en la seguridad de los activos de la empresa.

- Para disponer de herramientas que permitan a los responsables de la seguridad informática realizar informes detallados del estado de sus sistemas y detectar las variaciones que puedan producirse en la seguridad.

Con estos argumentos, justificar la necesidad de implantar métricas de seguridad adaptadas al entorno concreto de una organización puede parecer algo sencillo. Pero cuando llega el momento de realizarlo y nos encontramos con las dificultades derivadas de la definición de esas métricas siguiendo los esquemas más tradicionales ya no es tan sencillo como parecía al principio.

Existen distintas aproximaciones reconocidas de forma "oficial" (por ejemplo NIST - National Institute of Standards and Technology [51]) para la definición de métricas de seguridad, el problema radica en que en la mayor parte de los casos dichas aproximaciones pueden resultar demasiado complejas para su implantación en el ámbito de la pequeña o mediana empresa.

Por lo tanto, se hace necesario dotar a los administradores de seguridad de soluciones que permitan una rápida implantación de los modelos de métricas de seguridad simplificando la definición de las mismas y aportando modelos sencillos de implantar, que permitan obtener resultados de forma rápida.

3.3.3 - Tipos de métricas

Hoy en día se utilizan comúnmente dos medidas para determinar la seguridad de un sistema: a nivel del código ("code level") que se basa en contar el número de fallos ("bugs") encontrados en los sistemas (o los fallos encontrados desde una versión a la siguiente), y a nivel de sistema ("system level") que cuenta el número de veces que una versión del sistema se menciona en las advertencias del CERT [53] , boletines de seguridad de Microsoft [54], MITRE Common Vulnerabilities and Exposures (CVEs) [55], etc.

A nivel del código, muchos estudios se centran en contar y analizar los fallos (por ejemplo [56], [57], [58], [59]). Este método de contar los fallos puede valer como métrica de seguridad pero tiene desventajas significativas como que en el proceso de detección de los fallos se pueden omitir algunos, como también crear falsos positivos. Otra desventaja es que se da la misma importancia a todos los fallos, aunque algunos fallos sean más fáciles de explotar que otros.

Por estas desventajas la utilización de medidas a nivel de código no es recomendable y los estudios sobre métricas de seguridad se suelen centrar en medidas a nivel de sistema como las que a continuación se explican.

Muchas organizaciones como CERT [53] y MITRE [55] y sitios Web como SecurityFocus [60] siguen la pista de vulnerabilidades encontradas en varios sistemas y utilizan como métrica el número de vulnerabilidades de un sistema. Pero contar el número de veces que un sistema aparece en estos boletines no es una métrica ideal porque ignora la configuración específica del sistema que dio lugar a la vulnerabilidad, y no capta la probabilidad futura de un sistema de ser atacado.

También existen métricas bastante sencillas que se pueden utilizar para medir la seguridad a nivel de sistema, aunque los resultados que se obtienen de estas métricas se quedan cortos a la hora de poder valorar la seguridad de un sistema informático completo. No obstante, sirven para obtener medidas que se pueden utilizar para formar métricas más complejas y completas. Una de estas métricas puede ser utilizar los IDS (sistemas de detección de intrusos) para medir el número de veces que suceden determinados problemas. Esta métrica mide el número de alarmas mostradas por el IDS. Paradójicamente se debe intentar que el IDS no emita demasiadas alarmas, a la vez que se debe intentar que no bajen las alarmas de un umbral mínimo.

Buscando métricas más complejas podemos encontrar la propuesta de Shawn A. Butler [6] que usa un método de valoración de riesgos multi-atributos para obtener una lista priorizada de las amenazas para una organización. El método junta amenazas tales como el escaneo de puertos, el DDoS("Distributed Denial of Service") y la captación de contraseñas basándose en la frecuencia de la amenaza y el resultado previsto. El método se centra en las amenazas existentes para una organización más que en los sistemas de software usados en la organización.

En "A Trend Analysis of Exploitations" [62] se plantea un modelo matemático para reflejar la frecuencia con la cual los incidentes que implican explotaciones de vulnerabilidades son expuestos en el CERT. En "Timing the Application of Security Patches for Optimal Uptime" [63] se propone un modelo para encontrar el momento apropiado para aplicar los parches de seguridad a un sistema para conseguir una disponibilidad óptima. Ambos estudios se centran en las vulnerabilidades con respecto a cuando son descubiertas y en el tiempo de explotación y de arreglo, en vez de en las vulnerabilidades de un sistema concreto.

Hay varios trabajos que plantean una solución para modelar cuantitativamente la seguridad de un sistema. S. Brocklehurst y otros [64,65] miden la seguridad operacional de un sistema estimando el esfuerzo que un atacante debe de hacer para causar una brecha en la seguridad del sistema y la recompensa asociada a dicha brecha. En "Assessing Computer Security Vulnerability" [66] utilizan el Índice de las Vulnerabilidades del Sistema ("System Vulnerability Index") como medida de vulnerabilidad del sistema informático. Estos índices son obtenidos evaluando factores tales como las características del sistema, los actos potencialmente negligentes y los actos potencialmente malévolos. J. Voas y otros [67] proponen la métrica MTTI, "minimum-time-to-intrusion" (mínimo tiempo a la intrusión), basado en el período del tiempo predicho antes de que cualquier intrusión simulada pueda ocurrir. MTTI es una métrica relativa que permite comparar diversas versiones del mismo sistema. R. Ortalo,

Y. Deswarte y M. Kaâniche [68] modelan el sistema como un gráfico de privilegios [69] mostrando sus vulnerabilidades y estimando el esfuerzo que realiza el atacante para atacar el sistema con éxito explotando estas vulnerabilidades. El esfuerzo estimado es una medida de la seguridad operacional del sistema.

En [71] se plantea una curiosa forma de medir la seguridad. La clave de este planteamiento radica en que el concepto del tiempo está fuertemente ligado a la mayoría de modelos de negocio. Este modelo de seguridad basado en tiempo identifica el tiempo de exposición de nuestros sistemas utilizando el coste asociado a dicha exposición. Los otros dos conceptos clave que se barajan son el tiempo de detección y el tiempo de reacción, relacionados con el concepto de tiempo de exposición de la siguiente forma:

T. de exposición >= T. de detección + T. de reacción

El problema de este modelo radica en la falta de información que suministran los fabricantes de software y hardware.

En "An Attack Surface Metric" [52] proponen una métrica que está entre las medidas a nivel de código y a nivel de sistema. Dan importancia a la facilidad de realizar un ataque concreto y tienen en cuenta la relación que tienen las vulnerabilidades con las configuraciones específicas de los sistemas. Esta métrica esta pensada para determinar si una versión de un sistema es más seguro que otro. Mide la seguridad relativa de un sistema respecto de otro, si A y B son versiones de un sistema mide si la versión A es más segura que la versión B con respecto a su "attack surface" (superficie de ataque).

La superficie de ataque de un sistema son las vías por las que el sistema puede ser atacado con éxito. Definen la superficie de ataque en los términos de los recursos de sistema, porque un atacante utiliza estos recursos para atacar al sistema. De esto se deduce que si un atacante dispone de más recursos de sistema, mayor es la superficie de ataque del mismo.

Esta métrica utiliza la probabilidad de los recursos del sistema de poder ser atacados en vez de centrarse solo en las vulnerabilidades como medida de la seguridad, como sucede en las métricas explicadas anteriormente. Por este motivo se puede deducir la probabilidad futura de los sistemas de ser atacados. Los métodos que utilizan las vulnerabilidades como medida de seguridad no pueden deducir la esta probabilidad futura ya que no pueden deducir las futuras vulnerabilidades.

También esta métrica destaca sobre las anteriores en que se concentra en el diseño un sistema, por lo que se puede utilizar por los diseñadores y desarrolladores de sistemas para mejorar su seguridad.

Otras métricas interesantes se basan en de la utilización de herramientas como árboles y mallas de dependencias y ventanas de oportunidad [72]:
Un árbol o una malla de dependencias consiste en un grafo compuesto por los distintos objetos que componen nuestra instalación informática, tanto a nivel físico (máquinas, discos, etc.) como a nivel conceptual (procesos, usuarios, etc.). Estos objetos se asocian

entre sí mediante las relaciones de dependencia que tiene unos con otros. Con los datos obtenidos a partir del grafo construido (como el grado de dependencias) se pueden crear varias métricas de seguridad.

La segunda herramienta mencionada es la Ventana de Oportunidad. Esta herramienta consiste en un período de tiempo durante el cual se ve mermada nuestra capacidad de proteger nuestros sistemas de información. Las ventanas de oportunidad vienen dadas por la aparición de "exploits" inéditos (ataques de día cero) para los que no existe todavía una medida correctiva, o bien por la ejecución de procesos que debiliten la seguridad del sistema (como cambios de configuración temporales, etc.). Cuando se da una de estas ventanas de oportunidad se deben recoger dos datos: la duración de la ventana (tiempo de exposición) y la causa por la que se ha producido la oportunidad. Estos datos pueden servir para la creación de métricas de seguridad más complejas.

Otra típica forma de dividir las métricas de seguridad es basándose en la forma en que analizan y dan los resultados: cualitativamente o cuantitativamente.

- Un enfoque cualitativo es el que tiene por objetivo descubrir y separar cada activo (elemento en el que influye la seguridad). Por ejemplo un acercamiento cualitativo sería preguntarse: ¿Qué valor tiene la reputación de la organización?

- Un enfoque cuantitativo es el que da valores numéricos a cada activo para poder medir los riesgos en la seguridad.

Las ventajas y desventajas de estos métodos son:

- Ventajas:

  - Métricas cuantitativas:

    - Los resultados se basan en medidas objetivas por lo que se puede medir el riesgo y la seguridad mediante valores monetarios.
    - Los cálculos sobre las medidas de seguridad y las medidas de gestión de riesgos pueden ser llevados al día.
    - Los recursos disponibles se pueden centrar en las áreas de seguridad más importantes o de alto riesgo.
    - El rendimiento de la inversión sobre las soluciones de seguridad puede ser medido diariamente.
    - Con estas métricas se puede realizar un análisis del impacto del negocio.
    - Se pueden comparar los resultados con los estándares de la industria.
    - La dirección de la organización puede comprender los gastos en seguridad sobre la base del riesgo.

  - Métricas cualitativas:

    - Los cálculos son sencillos, en el caso de que los haya.

- El valor monetario no tiene que ser utilizado, ni es necesario medir las amenazas y las contramedidas. Como tampoco es necesario medir el coste de las contramedidas en comparación con las amenazas.
- Se pueden hacer y acatar generalizaciones de riesgos rápidamente.

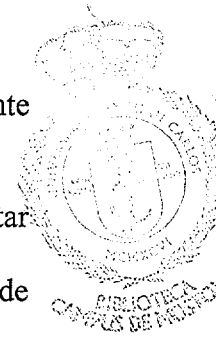
- Desventajas:

  - Métricas cuantitativas:

    - Los cálculos pueden llegar a ser muy complicados.
    - Se requiere tiempo y esfuerzo significativo por adelantado para asignar grados de riesgo a los activos.
    - Los estándares de la industria sobre el riesgo y las inversiones necesarias todavía no se han fijado.

  - Métricas cualitativas:

    - Los resultados que se obtienen con estas métricas son puramente subjetivos.
    - Los recursos pueden ser desplegados en el área equivocada.
    - No hay ninguna manera de calcular el valor monetario de implementar las medidas de seguridad ni de los resultados.
    - No se puede analizar la relación entre coste y beneficios ni se puede analizar el impacto en el negocio.
    - No hay ninguna manera objetiva de identificar el riesgo.



En el pasado se solían utilizar solo métricas cualitativas como si la cantidad de personal de seguridad en relación con el presupuesto anual es buena o mala o si tiempo que se tardaba en solucionar una vulnerabilidad desde que se daba a conocer era aceptable o no. Se demostró que estas métricas no eran eficientes y no servían para ahorra los costes producidos por las intrusiones al utilizar métodos reactivos y no cuantificables.

En la actualidad la utilización en solitario de métricas cualitativas esta desestimada. Por lo que se suelen utilizar solo métricas cuantitativas o métricas que utilizan métodos cualitativos y cuantitativos conjuntamente.

Por ejemplo, una de las formulas más conocidas en el ámbito de las métricas de seguridad es:

Riesgo ("Risk")= Amenaza ("Threat") x Vulnerabilidad ("Vulnerability") x Pérdidas previstas ("Expected Loss") ([73]).

El principal problema de esta formula es cómo cuantificar cada parte con valores significativos. ¿Cómo expresar numéricamente una amenaza? ¿Cuál es el coste de una vulnerabilidad? ¿Cómo calcular las pérdidas esperadas?

Para poder valorar cada parte de la formula correctamente es necesario determinar lo que se va a medir. Para conseguirlo, es recomendable dividir el estudio de la seguridad en diferentes elementos con los que poder cuantificar las diferentes partes de la fórmula como el valor de las posesiones (activos) o el valor de las pérdidas potenciales.

Un ejemplo de métrica cuantitativa y cualitativa es la propuesta de FoundStone [70]. En esta empresa de seguridad proponen una métrica con la que miden cuantitativamente el riesgo que corre una organización basándose en tres factores:

- El valor de los activos y de las contramedidas para proteger estos activos.
- Las amenazas existentes contra los activos.
- Las vulnerabilidades que pueden ser aprovechadas por las amenazas.

Dependiendo de si la métrica es utilizada sobre la red interna o externa de la organización en FoundStone [70] proponen mediciones distintas. Esto se debe a que el riesgo de las vulnerabilidades o de las exposiciones que existen en una red interna puede totalmente diferente a los de una red externa.

También tienen en cuenta lo críticos que son los activos, por ejemplo si una vulnerabilidad considerada de riesgo medio se encuentra en un activo que ha sido considerado como crítico, como puede ser un servidor de contabilidad, tendrá más puntuación en la métrica que si la vulnerabilidad fuese encontrada en un activo no critíco como un servidor de pruebas.

Teniendo en cuenta lo explicado en los párrafos anteriores en FoundStone [70] valoran el riesgo de la siguiente forma:

Para cada red (interna y externa) la métrica se divide en dos componentes (vulnerabilidad y exposición) a los que se les otorga diferente valor respecto el total:

En la red interna:

- Las vulnerabilidades puntúan 70 sobre 100. Para obtener el valor adecuado se combinan las puntuaciones individuales de cada vulnerabilidad según la gravedad de estas (alta, media o baja).
- La exposición puntúa 30 sobre 100. La valoración de la exposición de la red a la amenazas de Internet se basa en principios se seguridad generalmente aceptados.

Las características del entorno que son tenidas en cuenta para determinar la clasificación final conjunta (vulnerabilidades + exposición) son:

- ¿El entorno posee vulnerabilidades que pueden ser explotadas por atacantes para dañar los sistemas y/o obtener acceso no autorizado?

- ¿Hay puntos de acceso inalámbricos en el entorno que puedan ser comprometidos?
- ¿Hay puertos de Troyanos abiertos sobre alguna máquina que esté en el entorno?
- ¿Hay algún atacante interno o aplicaciones prohibidas (por ejemplo aplicaciones de mensajería instantánea, P2P ("peer to peer"), aplicaciones de compartición de datos, etc.) ejecutándose en el entorno?

En la red externa:

- Las vulnerabilidades puntúan 50 sobre 100. Al igual que para la red interna el valor adecuado se obtiene combinando las puntuaciones individuales de cada vulnerabilidad según la gravedad de estas (alta, media o baja).
- La exposición puntúa 50 sobre 100. Del mismo modo que en la red interna la valoración de la exposición de la red a la amenazas de Internet se basa en principios se seguridad generalmente aceptados.

Las características del entorno que son tenidas en cuenta para determinar la clasificación final conjunta (vulnerabilidades + exposición) son:

- ¿El entorno posee vulnerabilidades que pueden ser explotadas por atacantes para dañar los sistemas y/o obtener acceso no autorizado?
- ¿El entorno posee servicios de red prescindibles que incrementan la posibilidad de que se produzca una intrusión o una brecha en la seguridad?
- ¿Hay computadoras en el entorno que efectúen alguna función no necesaria para soportar las operaciones normales de Internet?
- ¿Se permite el tráfico de UDP ("User Datagram Protocol") entrante a la red (aparte del tráfico de DNS (Domain Name System) sobre 53 de puerto)?
- ¿Se permite ICMP (Internet Control Message Protocol) entrante a la red?

La evaluación de la red con los criterios anteriores proporciona una medida cuantitativa del riesgo de la seguridad en el entorno. Para dar una puntuación final a la red en FoundScore [70] dan al principio 100 puntos a todas las redes. Para cada violación se restan un determinado número de puntos. Así, cuanto la puntuación sea más baja más débil será la seguridad y más riego correrá. Los grados cualitativos que se les asigna a cada rango de valores son:

0 – 26 Pésimo.
26 – 50 Por debajo de la media
51 – 70 Medio.
71 – 85 Por encima de la media.
85 – 100 Excelente.

3.3.4 - Conclusión

Las métricas de seguridad en los sistemas informáticos son una herramienta de gran utilidad tanto para el administrador del sistema como para el gerente de seguridad de cualquier organización. No obstante aunque todavía se esta muy lejos de poder dar una valoración concreta (por ejemplo de 7.4 sobre 10) a la seguridad de un sistema, la implantación de una métrica de seguridad en la organización permite ofrecer datos a los directivos para poder negociar y justificar con ellos la inversión necesaria en medidas de seguridad, ya que no suelen estar familiarizados con la seguridad informática.

Por otra parte, la información existente sobre métricas de seguridad es bastante escasa, sobre todo si nos queremos centrar (como en este proyecto) en medir cuantitativamente el nivel de seguridad de una organización. La mayoría de estudios sobre métricas de seguridad se centran en métodos cualitativos y ninguno ofrece métricas cuantitativas para medir de forma general el nivel de seguridad de las organizaciones.

El futuro está en combinar y estudiar los métodos cualitativos y cuantitativos para poder crear una métrica que cuantifique cada elemento relacionado con la seguridad de una organización de forma general.

CAPÍTULO 4: Implementación

Después de haber analizado con detalle los diferentes sistemas de detección de intrusos y de escáneres de vulnerabilidades, y las métricas de seguridad informática actuales, en este capítulo se explica detalladamente la implementación de la herramienta creada para realizar las auditorías de seguridad informática.

Con dicha herramienta se podrá valorar de forma rápida y sencilla el nivel de seguridad de las organizaciones, ofreciendo un informe final claro y detallado que servirá para ayudar a los encargados de seguridad informática de las organizaciones a tomar las decisiones oportunas para mejorara las seguridad en sistemas.

La herramienta, bautizada como AuditTool, esta formada por tres módulos independientes, cada uno con una función concreta. Estos módulos están separados por el tipo de tarea que realiza: pasiva, activa y evaluación de la métrica.

En los siguientes puntos del capítulo se explicaran las características y el funcionamiento de la herramienta, de los diferentes módulos que la forman y de la métrica utilizada para cuantificar la seguridad de las organizaciones.

4.1 - La herramienta AuditTool

Uno de los mayores inconvenientes en la realización de las auditorías de seguridad es el arduo trabajo que supone la obtención de los datos necesarios para realizarlas. Por esto la aplicación implementada en este proyecto es un primer acercamiento para poder auditar los sistemas informáticos de una organización de forma clara, fácil y rápida.

Antes de empezar a implementar la aplicación fue necesario, como queda plasmado en el capítulo tres, estudiar los diferentes métodos y dispositivos que hay en la actualidad relacionados con las auditorías de seguridad y elegir entre todos los más adecuados para realizar este proyecto.

El funcionamiento de la aplicación se dividió en tres módulos: un módulo pasivo, otro activo y la evaluación de la métrica de seguridad que da paso a la creación del informe final. En los dos primeros módulos se recogen datos sobre el sistema a auditar que luego son utilizados por el tercero para evaluar la métrica y dar una valoración de la seguridad de la organización mediante la creación de un informe sencillo y detallado.

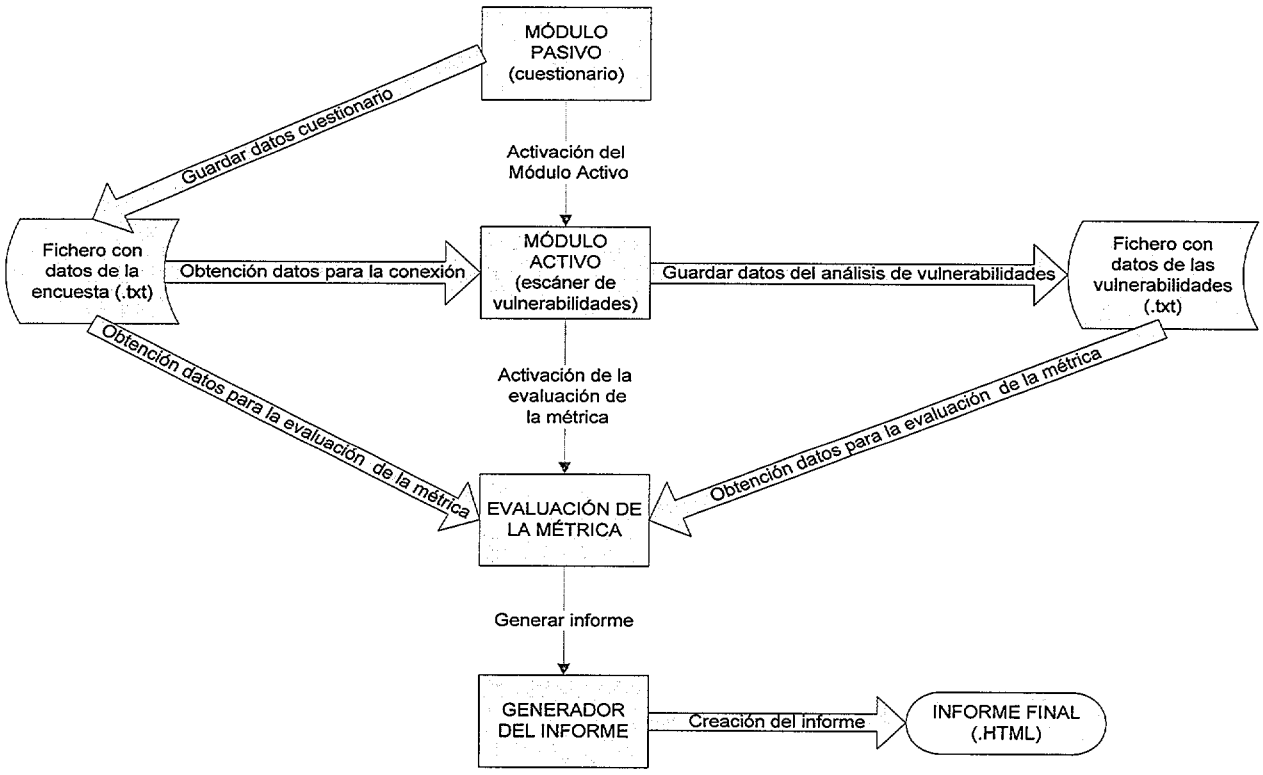


Figura 6: Estructura de AuditTool

El módulo pasivo se encarga de recopilar, mediante un cuestionario, la información necesaria sobre la organización a auditar (información general e información concreta sobre la configuración de la red) y también la necesaria para configurar el módulo activo (por ejemplo en este acercamiento la dirección IP del servidor y el nombre y la clave del usuario son necesarios para ejecutar el escáner de vulnerabilidades utilizado). Después de recopilada, la información es guardada en el fichero .txt. Al mismo tiempo se lanza el módulo activo, que recoge del fichero creado por el modulo pasivo los datos necesarios para configurarse y ejecutarse. Justo antes de terminar la ejecución, el módulo activo crea un informe sobre las vulnerabilidades de las máquinas de la red de la organización y lo guarda en un fichero .txt. Tras guardar el informe de las vulnerabilidades se activa la evaluación de la métrica. En este módulo se obtienen los datos necesarios (datos de la configuración y de las vulnerabilidades de las máquinas de la red de la organización) para la evaluación de la métrica de los dos ficheros creados por el módulo pasivo y por el módulo activo. Una vez evaluada la métrica se genera el informe final en formato .html.

La elección de esta estructura para AuditTool fue pensando principalmente en el futuro de la herramienta, concebida como un primer acercamiento para futuras herramientas de auditoría. La estructura se ha planteado para poder ser adaptada y mejorada con poco esfuerzo a las necesidades de otras herramientas más avanzadas de auditoría de sistemas informáticos que utilicen como base esta herramienta.

Esta estructura también proporciona la posibilidad de cambiar los módulos por otros sin tener que modificar el resto de módulos. Se podrían crear e intercambiar diferentes módulos pasivos dependiendo de las preguntas que se planteen, diferentes módulos activos según el escáner de vulnerabilidades escogido, diferentes métricas o

diferentes formatos de informe final.

A continuación se detallaran las particularidades de las implementaciones escogidas para la primera versión.

4.2 - Módulo pasivo

El módulo pasivo de AuditTool consiste en la recogida de información relativa a la organización que se va a auditar y de los datos necesarios para ejecutar el módulo activo. La información se recoge con un cuestionario que debe contestar el administrador o el encargado de la seguridad del sistema de la organización a auditar. Este cuestionario consta de un ajustado grupo de preguntas sencillas, que sin ser exhaustivo sobre la configuración del sistema de la organización, si da una visión suficientemente clara sobre la estructura del sistema y sus características. No obstante aunque las preguntas sean sencillas y concretas se debe tener un conocimiento claro del sistema que se desea auditar para poder contestar correctamente el cuestionario. Por lo que debe rellenar el cuestionario una persona con suficientes conocimientos del sistema informático, como por ejemplo el administrador del sistema, como se menciona en las líneas anteriores.

4.2.1 Metodología

La primera cuestión que se planteó en esta parte de la aplicación fue qué datos necesitaba la parte activa y qué tipo de información era necesaria para evaluar la métrica. Para resolver esta cuestión fue necesario estudiar y plantear las necesidades de la parte activa y de la métrica como queda reflejado en los puntos 4.3 y 4.4.

Después de aclarar esta cuestión hubo que estudiar la forma de recoger esta información de la forma más concreta y sencilla posible.

4.2.2 Creación del cuestionario

Antes de decidirse por una determinada forma de recoger los datos necesarios para implementar la herramienta se plantearon varias maneras de obtenerlos. En un primer intento se planteó la recogida de la información integrando un cuestionario basado en ventanas dentro del módulo activo. La ventaja de esta propuesta era la mayor automatización a la hora de ejecutar la aplicación. Pero se desestimó porque al integrar la parte pasiva en la parte activa se perdía la modularidad de la herramienta y se alejaba del concepto de herramienta que se había planteado inicialmente partiendo de las premisas de fácil modificación y mejora. La siguiente propuesta que surgió fue separar el cuestionario de la parte activa e implementarlo por separado dentro de una aplicación de Visual C++. Este acercamiento tuvo como inconvenientes la dificultad de crear cuestionarios dentro de una aplicación de Visual C++ y principalmente que, aunque en menos medida que el anterior intento, las parte activa y pasiva no eran independientes entre sí.

Después de plantear y probar los acercamientos explicados en el párrafo anterior, se llegó a la conclusión de que la mejor forma de recoger los datos era mediante un cuestionario creado con el leguaje de marcado HTML y con el lenguaje de "scripts" JavaScript. Se eligió esta opción por que proporcionaba total independencia a la parte pasiva al crear un módulo pasivo y un módulo activo independientes entre sí. De esta forma podría ser mejorada y adaptada con mayor facilidad respecto a las necesidades de futuros proyectos. Otra ventaja de esta propuesta es la posibilidad de poder adaptar la herramienta fácilmente para ser utilizada remotamente mediante una conexión (por Internet o una red interna) desde navegador Web a un servidor Web.

Una vez seleccionados los lenguajes y obtenida el tipo de información necesaria hubo que diseñar el formulario Web de forma que se pudiese obtener la información ordenadamente. El orden de recogida de la información se dividió en dos partes. En la primera se pregunta por los datos generales de la organización y por los datos requeridos por el escáner de vulnerabilidades utilizado en el módulo activo. Al ejecutarse el módulo activo el escáner de vulnerabilidades requiere hacer una conexión con su servidor. Para realizar esta conexión se necesita el nombre ("login") y la contraseña ("password") de un usuario previamente registrado y autorizado en el servidor para utilizar AuditTool. También será necesario recoger la dirección IP del servidor con el que se desea realizar la conexión.

Otra de las preguntas que se realizan en la primera parte es el número de cortafuegos existente con el que se obtiene en número de redes de la organización a auditar, como se explica más adelante. Una vez obtenido en número de redes de la organización se recoge, en la segunda parte del cuestionario, la información de cada red. La información recogida de cada red se divide a su vez en dos partes. En una parte hacen preguntas generales sobre la configuración de la red y en la otra parte se pregunta más concretamente sobre las máquinas que forman dicha red. El orden de recogida y el tipo de los datos queda de la siguiente manera:

- *Primera parte:*

  1- Primero se recogen los datos necesarios para realizar la conexión entre el cliente y el servidor del escáner de vulnerabilidades. Estos datos son nombre y contraseña del usuario que realiza la auditoria y dirección IP de servidor con el que se desea realizar el escaneo de vulnerabilidades del sistema auditado.

  2- Después se pregunta por los datos generales de la organización como: nombre y dirección de la organización, e-mail de contacto, etc. Estos datos serán utilizados para crear el informe final y por ejemplo, el e-mail, para tener la posibilidad enviar el informe final al administrador o encargado de la seguridad de la organización auditada.

  3- En tercer lugar se pregunta por el número de cortafuegos instalados en la organización mediante el cual se obtiene el número de redes existentes. Esta forma de dividir las redes de la organización basándose en el número de cortafuegos surgió a partir de la estructura de red propuesta en el proyecto "Diseño e Implantación de Arquitecturas Seguras" [88]. Esta

estructura divide las redes siempre mediante cortafuegos, por lo que se pueden contar las redes por el número de cortafuegos del sistema informático.

4- Por último se recoge la información relativa a las medidas de seguridad generales de la organización. Estas medidas son, por ejemplo, la utilización de cámaras de vigilancia, la existencia de guardias de seguridad o la instalación de alarmas.

- *Segunda parte:*

1- Lo primero que se pregunta es la dirección IP de la máquina que funciona como cortafuegos y que separa la subred de las demás subredes de la organización.

2- Después se obtiene el número de máquinas de cada tipo que hay en la red. Las máquinas se han dividido en siete tipos basándose en el tipo de actividad que realizan. Por ejemplo máquinas de trabajo, servidores Web, servidores de aplicaciones, etc. Estos tipos se enumeran unas líneas más adelante.

3- En tercer lugar se recogen las direcciones IP de cada máquina separándolas por los tipos de las máquinas. Dichas direcciones IP serán utilizadas posteriormente por el módulo activo para realizar es el análisis de vulnerabilidades del sistema.

4- En cuarto lugar se pregunta por el nivel de importancia que tiene la información almacenada en los servidores de bases de datos y en las bases de datos de la organización. El nivel de importancia de los datos se establece mediante los tres niveles propuestos por la legislación vigente sobre protección de datos personales. El encargado de realizar el cuestionario deberá saber en que nivel se encuentra la información contenida en cada base de datos y servidor de bases de datos.

5- Por último se pregunta por los dispositivos de seguridad implantados en la red. Estos dispositivos son, por ejemplo, utilización de medidas de control de acceso a las máquinas, copias de seguridad, sistemas de detección de intrusos, etc.

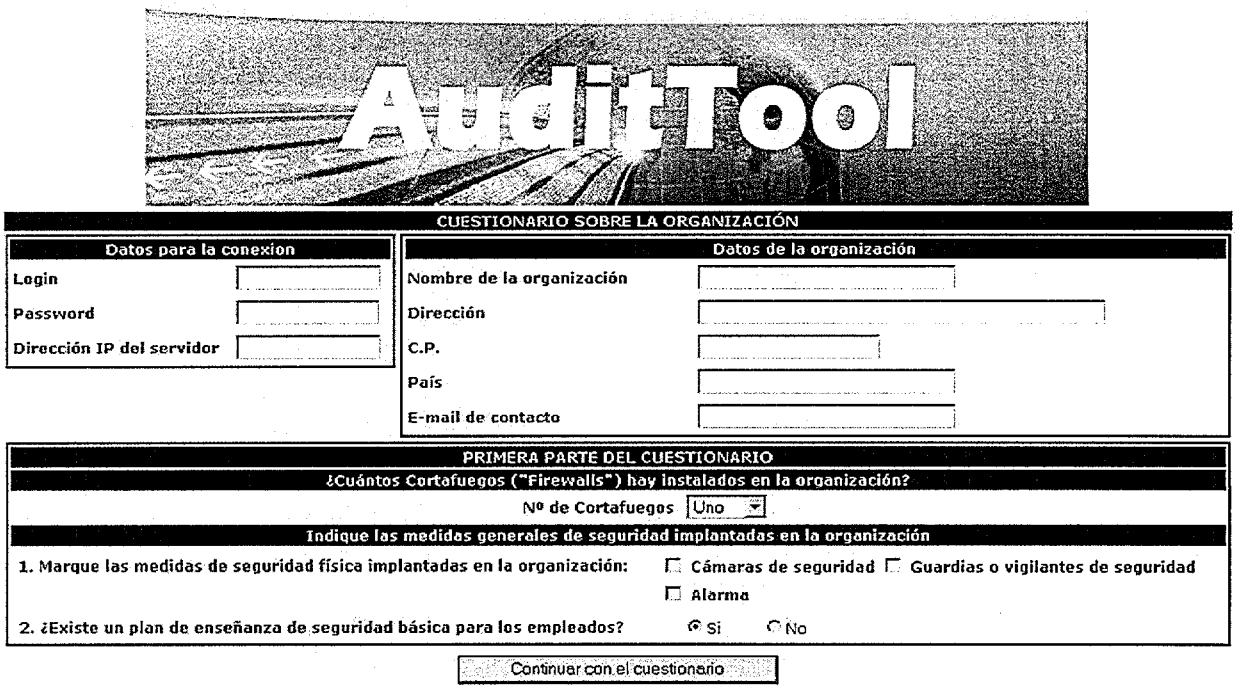


Figura 7: Modelo de la primera parte del cuestionario

Todas las preguntas y el orden explicado anteriormente se basan en un el estudio previo (realizado para la definición de la métrica) de los elementos y aspectos que afectan a la seguridad de una organización. Partiendo de los más concretos hasta los más generales son:

- Tipo de cada máquina: Los tipos se han dividido en cortafuegos, máquinas de trabajo sin privilegios, máquinas de trabajo con privilegios, servidores Web, servidores de aplicaciones, servidores de bases de datos y bases de datos.

- Importancia de la información contenida en cada servidor de bases de datos y cada base de datos teniendo en cuenta la legislación vigente sobre datos personales.

- Dispositivos de seguridad existentes en cada red o subred: Se han tenido en cuenta siete: IDS, IPS, medidas de control de acceso a las máquinas: contraseñas o dispositivos de identificación personal, dispositivos de almacenamiento extraíbles (disquetes, CDs, etc.), plan de restauración de los sistemas (copias de seguridad) y vigilancia física de las máquinas de la red.

- Importancia de la red: Esta importancia depende de cuantas redes dependan de ella para mantener un funcionamiento óptimo o por decirlo de otro modo dependerá de la profundidad de la red: lo lejos o cerca que esta de la red externa.

- Medidas generales de seguridad de la organización: Estas medidas están relacionas con la seguridad física general: guardias o vigilantes de seguridad, cáramas de vigilancia y dispositivos de alarma. Y con medidas de educación para los empleados como un plan de enseñanza sobre seguridad básica.

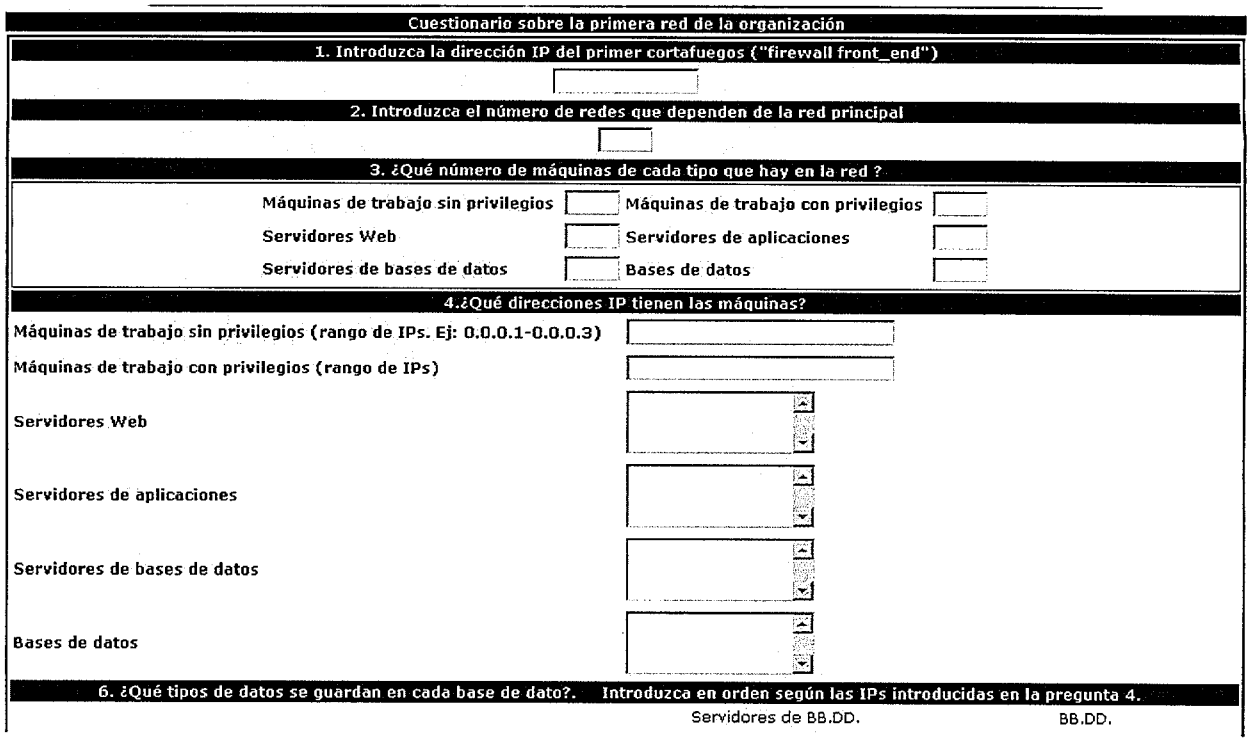


Figura 8: Modelo de cuestionario sobre la configuración de las redes

Tras introducir la información en cada formulario y cuando se pulsa el botón "Aceptar" se crean dos ficheros de texto plano (.txt), uno con todos los datos de los formularios y otro con las direcciones IP de las máquinas de la organización. El primero de los ficheros será utilizado por los siguientes módulos de AuditTool para obtener los datos de conexión del escáner de vulnerabilidades y para la evaluación de la métrica. El segundo será utilizado en la parte activa para obtener las direcciones IP de las máquinas objetivo del escáner de vulnerabilidades.

4.3 - Módulo activo

Esta parte de la aplicación consiste en detectar y evaluar las vulnerabilidades de las máquinas que se encuentran dentro de la red de la organización y presentar un informe con el número y tipo de vulnerabilidades de cada máquina teniendo en cuenta la información obtenida en el módulo pasivo. El análisis de las vulnerabilidades de cada máquina se realiza utilizando el cliente NessusWX [84] con algunas modificaciones para adaptarlo a las necesidades de la herramienta AuditTool.

Esta adaptación así como las características y funcionamiento de Nessus [29] se explican en los siguientes puntos.

4.3.1 Metodología

El plan de trabajo propuesto consistió primero en la búsqueda de una herramienta que facilitase la tarea de analizar las máquinas de la red. Entre los tipos de herramientas de seguridad estudiados se eligió un escáner de vulnerabilidades frente a la posibilidad de elegir un detector de intrusos. Esta elección es bastante obvia, con un escáner de vulnerabilidades se obtienen los datos de forma proactiva y no reactiva como sucede con los IDS como se explicó en el capitulo 3. Además con un escáner de vulnerabilidades se obtienen más datos y más relevantes para la realización de una auditoría de seguridad de sistemas informáticos que con un detector de intrusos.
El escáner de vulnerabilidades elegido fue Nessus [29]. Las principales razones para su elección fueron dos: por ser software de libre distribución y de código abierto, lo que proporciona la posibilidad de adaptar el código a las necesidades de la herramienta del proyecto (AuditTool). Y por que está disponible para plataformas Windows, una cualidad indispensable ya que este proyecto tiene como base dichas plataformas.

También el hecho de que sea uno de los escáneres de vulnerabilidades más reconocidos en el mundo de la seguridad ha sido un factor importante para su elección.
La elección de Nessus [29] condicionó a que el lenguaje con el que se ha programado la aplicación sea C++ ya que NessusWX [84] (el cliente de Nessus [29] para Windows) está programado en C++. Otra condición que supuso la elección de NessusWX [84], aunque esta vez fue más por comodidad, fue la utilización del entorno de desarrollo Visual C++ de Microsoft, ya que el código fuente del cliente esta realizado con este entorno de desarrollo y cambiar de entorno de desarrollo supondría adaptar el código al nuevo entorno.

Después de seleccionar la herramienta adecuada hubo que estudiarla profundamente tanto a nivel de aplicación como a nivel de código para poder entender su funcionamiento a la perfección. Para estudiar la herramienta a nivel de código fue necesario aprender el funcionamiento del entrono de trabajo Visual C++ y familiarizarse con el leguaje de programación C++.

Una vez realizado el estudio sobre la herramienta hubo que adaptarla para que se pudiera comunicar con los demás módulos del proyecto. Es decir: el módulo activo utiliza la información proporcionada por el módulo pasivo y proporciona información para evaluar la métrica y para crear el informe.

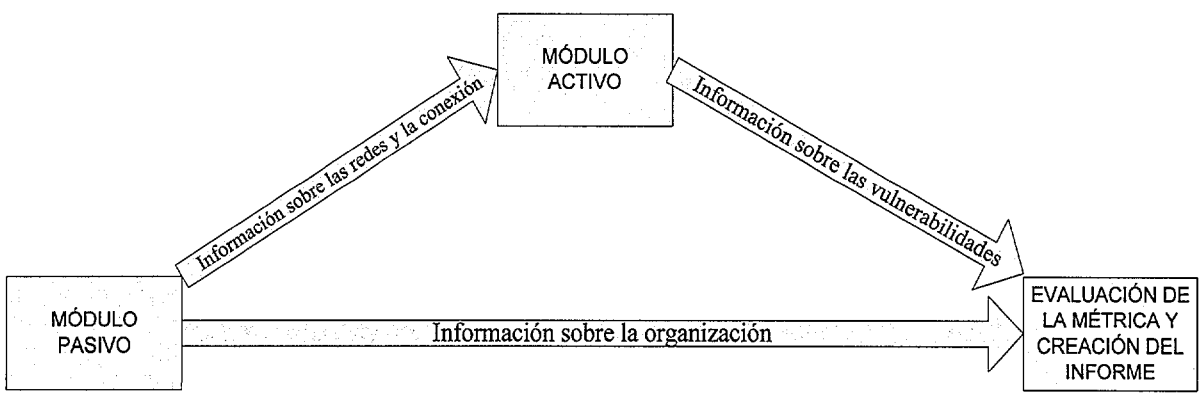


Figura 9: Paso de datos entre módulos

4.3.2 Nessus

Nessus [29] una herramienta de escáner de vulnerabilidades de libre distribución y de código abierto. Nació de la mano de Renaud Deraison en 1998 para intentar reemplazar a los escáneres de vulnerabilidades de libre distribución que se habían quedado estancados y para contrarrestar el aumento de los precios de los escáneres de vulnerabilidades comerciales. Es una herramienta robusta apropiada para evaluar la seguridad de la red de cualquier tipo de organización. También cuenta con una gran comunidad de usuarios en continuo aumento que permite tener actualizadas las vulnerabilidades y mantiene el proyecto al día y en continua mejora.

En la actualidad Nessus [29] es utilizado por muchas organizaciones y ha servidor como referencia para nuevos proyectos relacionados con el análisis de las redes como éste.

4.3.2.1 Características

El punto fuerte de Nessus [29] es su arquitectura basada en el modelo cliente-servidor. Esta arquitectura y el hecho de haber tenido en cuenta cada elemento del ciclo de vida de la seguridad informática desde el principio de la implementación, proporciona a la herramienta mucha flexibilidad para poder adaptarla a cualquier red de cualquier empresa. ([82])

Los componentes principales de la arquitectura de Nessus [29] son: el cliente, el servidor, los "plugins" y la base de conocimiento.

4.3.2.1.1 El cliente y el servidor

Al principio los escáneres de vulnerabilidades no estaban basados en el modelo cliente-servidor. Esto hacía que para analizar un red primero había que buscar el mejor sitio de la red para instalar el escáner de vulnerabilidades o para conectar un portátil con el escáner de vulnerabilidades instalado, con la consecuencia de tener que ir cambiando de sitio para poder analizar la red correctamente.

El servidor de Nessus [29] se encarga de escanear las vulnerabilidades y de realizar el análisis de la red. Contiene la base de datos actualizada con los diferentes ataques y pruebas necesarios para realizar el análisis de las redes. Y soporta toda la carga de proceso de la aplicación.

El cliente se conecta al servidor y se encarga de crear y configurar las diferentes sesiones que son guardadas en el cliente. Cuando se quiere analizar una red el cliente manda los datos de configuración de la sesión con las máquinas que se desean escanear y espera a recibir los datos que va mandando el servidor sobre el procesamiento de la sesión.

Con esto modelo se evita tener que estar moviéndose por toda la red para analizarla. Del mismo modo se centra toda la carga de trabajo en el servidor, liberando el cliente de la misma y pudiéndolo utilizar para otras tareas al mismo tiempo que se realiza el análisis de la red.

El cliente esta disponible para plataformas UNIX y Windows en los dos casos con el código abierto, pero el servidor sólo esta disponible para plataformas UNIX por lo que siempre se tiene que tener instalado un sistemas UNIX, lo que resta un poco de movilidad a Nessus [29].

4.3.2.1.2 Los "plugins"

Los "plugins" son programas que se agregan a la aplicación y que realizan funciones determinadas, en este caso relacionadas con la seguridad informática.

Una de las características más destacables de Nessus es el potencial que tiene al tener un lenguaje de "script" propio para crear nuevos "plugins" que implementen ataques y las comprobaciones necesarias para analizar las máquinas en busca de vulnerabilidades. Este lenguaje se llama NASL, es parecido al leguaje C, esta diseñado específicamente para realizar programas centrados en la seguridad informática, se comunica con las máquinas pasando solo un argumento y no ejecuta ningún comando local del sistema. Además es fácil de utilizar por lo que cualquier analista de seguridad puede crear "plugins" para una red específica, por ejemplo para protocolos y servicios concretos de la red. Otra cualidad de NASL es que fue construido para poder compartir información entre las diferentes pruebas de seguridad mediante la utilización de la base de conocimiento de Nessus [27].

4.3.2.1.3 La base de conocimiento

La base de conocimiento permite a los "plugins" utilizar la información obtenida por otros "plugins" que hayan sido ejecutados con anterioridad. Por ejemplo, si un primer "plugin" obtiene la versión de una determinada aplicación en un sistemas operativo concreto, ésta se guardará en la base de conocimiento y podrá ser utilizada por otros "plugins" para efectuar comprobaciones más concretas en el sistema sabiendo desde el principio la versión de la aplicación y del sistema. Esta característica sirve para extender la capacidad de Nessus [29] y acelerar el rendimiento de futuros "plugins" que pueden buscar en la base de conocimiento los datos requeridos en vez de tener que analizar la red para obtenerlos.

4.3.2.2 Instalación

La instalación de Nessus [29] es bastante sencilla y no se tarda más de 30 minutos en instalar, configurar y ejecutar el primer análisis. Es recomendable dividirla en dos partes: la instalación del servidor y la instalación del cliente.

4.3.2.2.1 Instalación y configuración del servidor

Como ya se ha señalado, el servidor sólo está disponible para plataformas UNIX, por lo que se necesita tener instalada en una máquina alguna distribución de Linux o UNIX. Dependiendo de la distribución que se tenga instalada los pasos a realizar para instalar el servidor serán diferentes. En este caso el servidor se ha instalado en una distribución Ubuntu [85] de Linux, basada en Debian [86].

Lo primero que hay que hacer es descargar de la página oficial de Nessus [29] los archivos del servidor. En este caso no es necesario porque al utilizar una distribución basada en Debian[86] se puede directamente descargar e instalar automáticamente el servidor y el cliente desde el repositorio de Debian [86] (para las demás distribuciones se puede consultar la página oficial [29] o [82] y otras múltiples páginas en Internet).
Para bajar e instalar el servidor del repositorio de Debian [86] hay que entrar como super usuario ("root") en el sistema e introducir en la consola: "*apt-get install nessusd*".
Es recomendable para obtener la última versión de Nessus [29] tener actualizados los archivos de configuración del comando "apt-get".

Cuando termina la instalación lo primero que se requiere es añadir un usuario. Un nuevo usuario puede ser añadido mediante el comando: "*nessus-adduser*". Al ejecutar el comando lo primero que habrá que introducir será en nombre del nuevo usuario, después se pedirá que elija entre el método de autentificación que desee. Se puede elegir entre una autentificación mediante clave ("pass") o mediante certificado ("cert"). En este caso se elegirá la autentificación mediante clave ya que es la más sencilla de usar. En el siguiente paso se preguntará por las reglas que se aplicarán al nuevo usuario. En este apartado se podrá especificar por ejemplo las direcciones IP que podrá escanear el usuario, pero en este caso no se introducirá ninguna regla dejando en blanco este apartado.

En caso de querer personalizar el servidor habría que modificar los parámetros que están guardados en el fichero *"/etc/nessus/nessusd.conf"* pero en este caso la configuración por defecto del servidor no necesita ser modificada.

Si se quiere también instalar el cliente en Ubuntu [85] habría que introducir en la consola, después de la instalación del servidor, el comando: "*apt-get install nessus*".

4.3.2.2.2 Instalación del cliente

Después de haber instalado el servidor el siguiente paso deberá ser la instalación del cliente de Nessus [29].

El cliente que se utilizará en este proyecto es la versión para Windows, se puede descargar de la página oficial de Nessus [29] o desde la página de creador de la versión del cliente de Nessus [29] para Windows llamada NessusWX [84]. El cliente NessusWX [84] está disponible en dos formas: en ficheros binarios para plataformas Intel o en código fuente que deberá ser compilado después de ser descargado. La forma más rápida y sencilla es descargar los ficheros binarios listos para ejecutar. Una vez descargados el cliente se ejecutará el archivo "*NessusWX.exe*" para lanzar la aplicación.

4.3.2.3 Ejecución y funcionamiento

Una vez instalados el servidor y el cliente en sus respectivas máquinas se deberán ejecutar por orden: primero el servidor y luego el cliente.

Para ejecutar el servidor habrá que introducir en la consola de comandos: "*nessusd*". El servidor se quedará esperando a recibir las ordenes del cliente. Después se deberá ejecutar la aplicación del cliente mediante el ejecutable "*NessusWX.exe*".

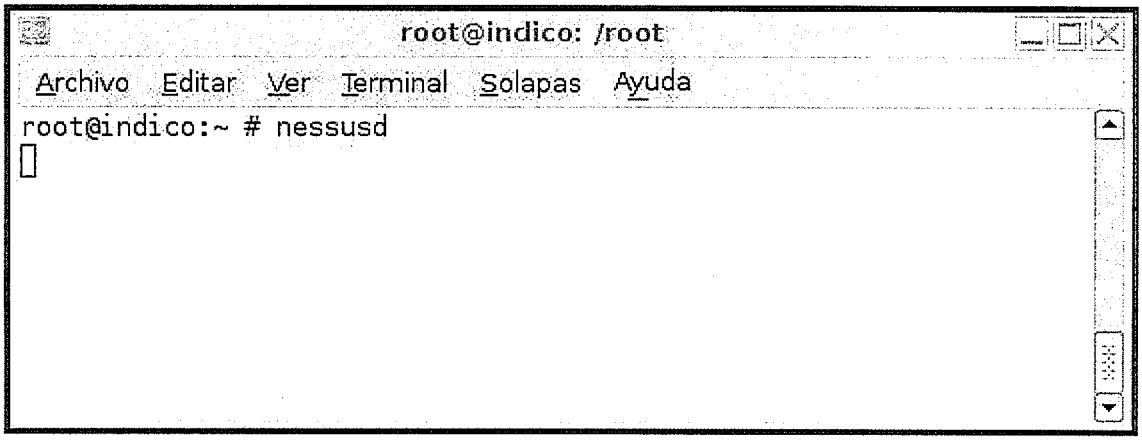


Figura 10: Ejecución del servidor de Nessus

Después de ejecutar el cliente se abrirá la ventana principal de la aplicación y se pedirá que elija donde desea crear y guardar la bases de datos de Nessus [29]. Por defecto se creará en "*C:\NessusDB*". Después se deberá elegir la opción "Connect" dentro del apartado "Communications"del menú principal. Una vez realizado este paso se abrirá la ventana para realizar la conexión, en donde habrá que introducir la dirección IP de la máquina donde este instalado el servidor, el nombre ("Login") del usuario así como su clave ("Password") y presionar el botón "Connect" (si no se introduce la clave del usuario ésta será requerida en el proceso de conexión con el servidor).

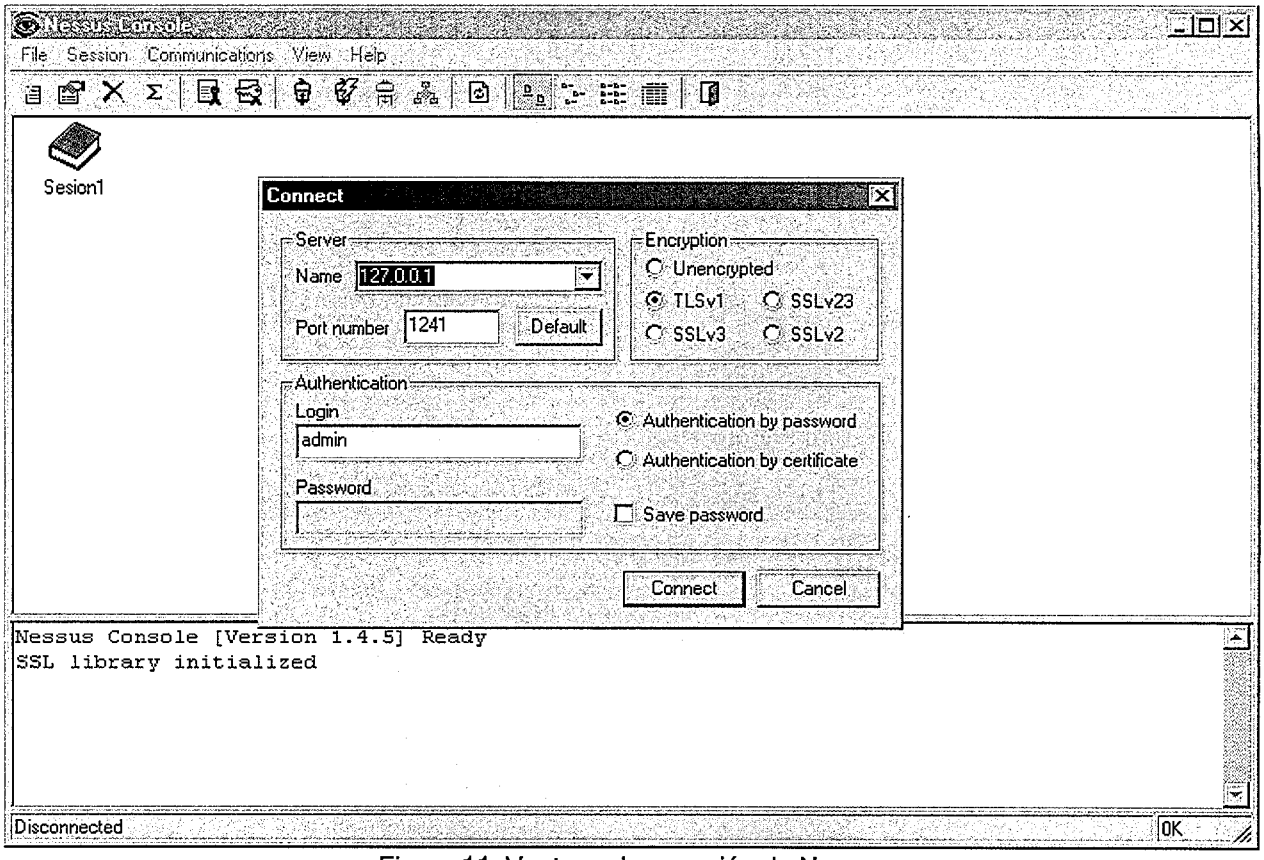


Figura 11: Ventana de conexión de Nessus

Tras la realización de la conexión con el servidor se podrá realizar el primer escaneo. Para poder llevarlo a cabo se deberán seguir los siguientes pasos:

1- Se deberá crear una nueva sesión. Dentro del apartado del menú "session" se elegirá "New".

2- Después de poner nombre a la sesión se deberán añadir las direcciones IP de las máquinas que se desee analizar. Para esto en la ventana "Session Properties" en la pestaña "Targets" se deberá pulsar el botón "Add" y añadir las direcciones IP de las máquinas objetivo. Dentro de la ventana "Session Properties" se podrá configurar la sesión a gusto del usuario aunque para este primer escaneo las opciones por defecto son suficientes.

3- Una vez configurada la sesión es hora de ejecutarla. Para ejecutar una sesión se debe pulsar la opción "Execute" dentro del apartado del menú "Session".

4- Después de realizar el paso tres aparecerá una nueva ventana en la que se pulsará directamente el botón "Execute" para pasar directamente a la ventana del estado del escaneo "Scan Status", en donde se podrá observar el proceso del escaneo en cada uno de las máquinas objetivo.

5- Cuando termine la ejecución del escaneo se pulsará el botón "Close" y se abrirá una nueva ventana en donde se podrá ver detalladamente los resultados del escaneo en cada máquina así como globalmente. También en esta ventana se podrá crear informes de la ejecución en diferentes formatos como .pdf, .html o .txt.

4.3.3 Adaptación de Nessus e implementación del módulo activo

Después de estudiar el código de NessusWX [84] y su funcionamiento había que adaptarlo para que utilizase la información obtenida en el módulo pasivo, analizara las máquinas en busca de vulnerabilidades y guardara la información obtenida automáticamente, de tal forma que se pudiera utilizar para evaluar la métrica propuesta y generar el informe final.

El principal problema que apareció al estudiar el funcionamiento y el código de NessusWX [84] fue que no tiene la posibilidad de ejecutar escaneos mediante línea de comandos (función que si tiene la versión del cliente de Nessus [29] para plataformas UNIX), por lo que siempre se tiene que hacer a través del entorno gráfico. Para solucionar este problema hubo que modificar todas las funciones de creación de ventanas de diálogo ("Dialog Box") que forman parte de una ejecución normal, desde la conexión y creación de una nueva sesión hasta la generación de informes y cierre de la aplicación, para que obtuviesen los datos necesarios en cada caso e hiciesen su función automáticamente. En algunos casos se eliminó el paso por algunas ventanas de diálogo innecesarias. También se añadió código para que se generase el informe de las vulnerabilidades y se cerrase la aplicación automáticamente, borrando los datos de la sesión ejecutada.

La única ventana que se ha dejado visible es la que muestra el proceso de análisis de la red, en la que se va mostrando la evolución del escaneo y el número de vulnerabilidades de cada tipo de cada máquina objetivo.

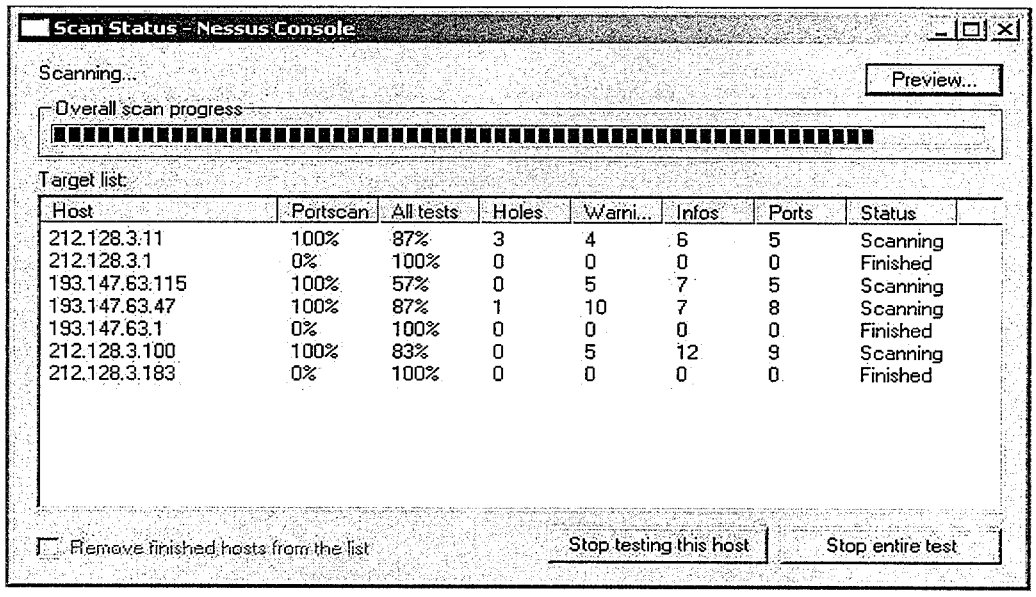


Figura 12: Ventana de proceso del análisis de vulnerabilidades

Por otra parte, para generar el informe de las vulnerabilidades de cada máquina de la red se modificó la generación de informes de NessusWX [84] para obtener un informe sencillo únicamente con los datos utilizados por la métrica. Este informe es guardado en un fichero de texto plano (".txt") y contiene las direcciones IP de cada máquina seguidas por el número de vulnerabilidades de cada tipo. Estos tipos los proporciona Nessus [29], son tres y se basan en la gravedad de cada vulnerabilidad. Pueden ser vulnerabilidades graves, vulnerabilidades medias y vulnerabilidades leves. Las graves y medias se consideran vulnerabilidades críticas y que sean de un tipo u otro depende del escenario donde se encuentren y de la posibilidad de explotación en el mismo. Dichas vulnerabilidades están relacionadas con: la ejecución de código maligno, la obtención de privilegios, desbordamiento de buffer, etc. Las vulnerabilidades leves suelen estar relacionadas con la posibilidad de obtener información sobre el sistema, acceso a la memoria, información sobre las versiones de las aplicaciones, información relativa a la configuración de red o información sobre que usuarios están registrados en el sistema.

El resultado de la adaptación de Nessus[84] es el fichero ejecutable "AuditTool.exe" acompañado por las librerías necesarias para ejecutarlo que realiza las siguientes tareas:

- Recoge la información capturada en el módulo pasivo.
- Analiza las vulnerabilidades de las máquinas del sistema.
- Crea un informe en formato .txt con el la dirección IP de cada máquina analizada y el número de vulnerabilidad de cada tipo detectada.
- Por último y después de incluir la evaluación de la métrica explicada en el siguiente punto generará un informe final.

4.4 - Métrica de seguridad

Una vez obtenidos los datos del módulo pasivo y del módulo activo se han de combinar para conseguir el objetivo propuesto: evaluar de manera cuantitativa y precisa la seguridad de las organizaciones.

4.4.1 Estudio y definición de la métrica

Después de haber analizado y estudiado los tipos de métricas de seguridad que existen en la actualidad se observa que hay una carencia de métricas cuantitativas o de un método general y efectivo para realizar evaluaciones de la seguridad informática de una organización. En esta parte del proyecto se realiza un primer acercamiento a una métrica cuantitativa que sirva de base de futuras métricas de seguridad que valgan para auditar cualquier tipo de organización.

En un principio toda métrica se basa en datos subjetivos condicionados por la experiencia del creador de la métrica o del administrador del sistema. Se da valor a cada elemento (activo) según el creador lo considere más o menos importante para la seguridad de una organización. También hay que tener en cuenta que dependiendo del tipo y de las singularidades de la organización que se esté auditando el valor de cada activo varía.

En este proyecto la definición de la métrica está basada en la empresa ficticia que tienen como objetivo conjunto los cuatros proyectos mencionados en el primer capítulo. Más concretamente en el modelo general de la arquitectura de la red, creado en el proyecto "Diseño e Implantación de Arquitecturas Seguras" [88]. De esta forma se simplifica notablemente la realización del cuestionario del módulo pasivo al reducir así las posibles variaciones y combinaciones que se podrían dar si se tuviesen en cuenta todos los posibles tipos de organizaciones y de redes. Esta decisión se tomó para centrarse más en la definición de la métrica que en la recolección de los datos necesarios para evaluarla, que de otro modo podría haberse complicado enormemente esta primera aproximación.

La definición de la métrica partió de la utilización del escáner de vulnerabilidades Nessus [29], a partir del cual se obtiene el número y el tipo de las vulnerabilidades de las máquinas de una red. Una vez obtenido el punto de partida se tomó como principales puntos de referencias la propuesta de FoundStone [70] y la fórmula explicada en [73]:

Riesgo = Amenaza x Vulnerabilidad x Pérdidas previstas

De estas referencias se obtuvo el concepto de riesgo como medida del nivel de seguridad de una organización. El riesgo se entiende en este contexto como la contingencia o proximidad de que se produzca un daño en la organización con las medidas de seguridad y a la estructura de los sistemas informáticos que tiene la organización en el momento de hacer la auditoría.

Después de haber definido como medida inicial en número y tipo de las vulnerabilidades de las máquinas y el riesgo como medida de evaluación del resultado final de la métrica, había que buscar la forma de evaluar conjuntamente esta medida inicial y los demás elementos y medidas que afectan a la seguridad de una organización, ya explicados y definidos en el punto 4.2.2 del módulo pasivo para obtener una valoración del riesgo.

Como resultado del estudio realizado se formuló una ecuación con la que poder medir el riesgo al que está sometida una organización:

Riesgo de la organización = Media del riesgo de las redes de la organización – Valor de las medidas generales de seguridad existentes. *Riesgo de una red* = (Valor de la red * Riesgo de los activos de la red) – Valor de los dispositivos de seguridad de la red. *Riesgo de los activos de una red* = (Valor del activo * Valor de las vulnerabilidades del activo) / Número de activos.

Tabla 1: Fórmula para la evaluación de la métrica

Como se puede apreciar la métrica evalúa el riesgo que corre la organización auditada partir del riesgo de sus redes. Y a su vez, el riesgo de cada red depende del valor de las vulnerabilidades detectadas en sus activos. Este riesgo depende de los siguientes factores:

- *Número de activos:* Los activos son las máquinas que tiene la red, por lo tanto el número de activos es el número total de máquinas que tiene la red. Estas máquinas son estaciones de trabajo que pueden realizar diferentes funciones como por ejemplo de cortafuegos, de servidores Web, de bases de datos, de puestos de trabajo, etc.

- *Valor de las vulnerabilidades del activo:* El valor de las vulnerabilidades que tiene cada máquina se calcula sumando el valor asignado a cada tipo de vulnerabilidad por el número vulnerabilidades de ese tipo. Una vez obtenido el valor de las vulnerabilidades es adaptado a una escala de 0 a 10.

- *Valor del activo:* El valor de cada activo depende del tipo de máquina que sea. Por ejemplo, un puesto de trabajo sin privilegios siempre tendrá menos valor que un servidor de bases de datos. Esto es así porque las pérdidas ocasionadas por la exposición de una máquina sin privilegios es menor que las que puede ocasionar la exposición de un servidor de bases de datos.

- *Riesgo de los activos de la red:* El riesgo de los activos de una red es el valor medio de todos los activos de la red.

- *Valor de los dispositivos de seguridad de la red:* El valor de los dispositivos de la red es la suma de los valores asignados a cada dispositivo.

- *Valor de la red:* El valor de la red dependerá del número de redes que dependan de ella.

- *Riesgo una red:* El riesgo de la red es el riesgo de los activos de la red por el valor de la red menos el valor de los dispositivos de seguridad. Si de una red dependen otras redes el valor de dicha red será mayor por lo que se correrá más riesgo al tenerla. Por otra parte hay que restar al riesgo de la red el valor de la protección que aportan los dispositivos de seguridad ya que contra más protección se tenga menos riesgo se corre.

- *Valor de las medidas generales de seguridad existentes:* El valor de las medidas generales de seguridad existentes es la suma de los valores asignados a cada medida de seguridad general de la organización.
- *Media del riesgo de las redes de la organización:* La media del riesgo de las redes es la suma del riesgo de cada red partido entre el número de redes de la organización.

- *Riesgo de la organización:* El valor final obtenido se pasa a una escala de 0 a 100. Un nivel de riesgo aceptable valdría menos de 30, de 30 a 60 la organización correría un riesgo moderado y a partir de 60 la organización correría alto riesgo.

Una vez definidos estos factores lo único que queda por definir son los valores que se asignarán a cada tipo de activo, medidas de seguridad, dispositivos de seguridad y a cada tipo de vulnerabilidad. En el apartado 4.4.2.2 se proponen unos valores, pero se podrían definir otros ajustados a las prioridades de cada organización.

4.4.2 Incorporación de la métrica a AuditTool

Después de estudiar y plantear las bases de la métrica es hora de incorporar su evaluación a la herramienta de auditoría AuditTool. La evaluación se ha incluido dentro del código modificado del NessusWX [84] para automatizar y sincronizar con mayor facilidad la ejecución del escaneo de vulnerabilidades de la red, la evaluación de la métrica y la creación del informe final.

La evaluación de la métrica consta de tres partes que están implementadas dentro de la misma función y son: recogida de datos, evaluación de la fórmula de la métrica y creación del informe.

4.4.2.1 Recogida de datos

La recogida de datos se ejecuta antes que las dos partes restantes de la evaluación de la métrica. Consiste en obtener los datos guardados en los ficheros de texto plano creados en la parte pasiva y en la parte activa y guardarlos en diferentes estructuras para que luego puedan ser tratados por la evaluación de la fórmula de la métrica y por la parte de la creación del informe final.

Lo primero que se guarda es la información general de la organización obtenida del fichero creado por el módulo pasivo. Después para cada red de la organización se guarda los tipos de dispositivos de seguridad que tiene, el número de redes que dependen de ella y por otra parte las direcciones IP de las máquinas de la red con el tipo de máquina, el número de vulnerabilidades graves, medias y leves y, si son servidores de bases de datos o bases de datos, también se guarda el tipo de los datos. Estos datos son obtenidos de los ficheros creados por el módulo pasivo y por el módulo activo.

4.4.2.2 Evaluación de la métrica

Una vez guardados los datos en las estructuras oportunas se aplica la fórmula de la métrica plateada en el punto 4.5.1. Pero antes de poder aplicarla hay que dar valores a los diferentes activos implantados en la organización y en cada red. Estos valores, como se explicó en el punto 4.4.1, están condicionados por la experiencia del creador de la métrica o del administrador del sistema. Para intentar ponderar el valor de cada activo de la forma más objetiva y real posible se hicieron varias simulaciones y pruebas sobre diferentes topologías de red en un entorno controlado. De estas pruebas y simulaciones se obtuvo unas valoraciones de los activos que fueron matizadas con la experiencia de Alberto Luís Corrales y Carlos Rubio obtenida en la realización de los proyectos "Diseño e Implantación de Arquitecturas Seguras" [88] y "Metodología para la fortificación de bases de datos" [89], comentados en el primer capítulo.

Los valores asignados a los activos son:

- Máquinas: Las máquinas son valoradas de 0 a 100 puntos. Los puntos indican la importancia que tiene cada máquina dentro de un sistema informático. Cuanta más puntuación más importancia tienen y se corre más riesgo al tenerlas dentro de la organización.
    - Cortafuegos: 70
    - Máquinas de trabajo sin privilegios: 30
    - Máquinas de trabajo con privilegios: 85
    - Servidores Web: 60
    - Servidores de aplicaciones: 40
    - Servidores de bases de datos: 60
    - Bases de datos: 50

- Tipos de datos: El valor de los tipos de datos es sumado al valor del servidor de bases de datos o de la base de datos. Cuanto mayor es el valor de los datos mayor es la importancia de los mismos y mayor es el riesgo que se corre al tenerlos en los servidores de bases de datos o en las bases de datos.

    - Nivel 1: +0
    - Nivel 2: +20
    - Nivel 3: +35

- Vulnerabilidades: Las vulnerabilidades se puntúan según la importancia de las mismas, cuanto más grave sea la vulnerabilidad mayor será su puntuación.

    - Graves: 8
    - Medias: 3
    - Leves: 1

- Dispositivos de seguridad de la red: El total de los dispositivos de la red suma 100 puntos quitando los dispositivos de almacenamiento externos que resta puntos. También la ausencia de algunos dispositivos resta puntuación. La puntuación indica el nivel de seguridad o de protección que aporta el dispositivo dentro de la red de la organización. Sumar 100 puntos en este apartado sería tener la red protegida de la mejor forma posible.

    - IDS: +25
    - IPS: +30
    - Control de acceso mediante contraseñas: +10, en ausencia -10
    - Dispositivos de identificación personal: +15
    - Dispositivos de almacenamiento extraíbles: -10
    - Plan de restauración de los sistemas: +15
    - Vigilancia física de la red: +5

- Medidas de seguridad generales de la organización: Las medidas de seguridad generales de la organización suman un total de 100 puntos. La puntuación indica el nivel de seguridad o de protección que aporta cada medida a la organización. Sumar 100 puntos en este apartado sería tener la organización protegida de la mejor forma posible.

    - Guardias de seguridad: +30
    - Cámaras de vigilancia: +15
    - Alarma: +20
    - Plan de enseñanza básica de seguridad para los empleados: +35

4.4.2.3 Creación del informe

Tras haber realizado los cálculos oportunos y haber obtenido el valor del riesgo en la organización auditada se crea un informe en formato HTML (.html). En el informe final se muestran en la mitad superior los datos generales de la organización y la fecha de realización de la auditoría. Y en la mitad inferior se muestra la valoración de cada red de la organización (según el orden introducido en el cuestionario de la parte pasiva) y la valoración total de la seguridad de la organización.

INFORME DEL NIVEL DE SEGURIDAD DE LABGAAP 11.09.2005

by AuditTool S.A.

ORGANIZACIÓN: LABGAAP **e-mail de contacto:** correo@labgaap.es	**Dirección:** Mostoles 33 C.P. 28339 **País:** Spain

Número de redes auditadas: 3

REDES	PUNTUACIÓN
Red Principal:	77.142860
Red 2:	30.457332
Red 3:	42.694310

Puntuación total del riesgo de la organización (de 0 a 100): 40.153971 *(Riesgo: menos de 30 aceptable, de 30 a 60 moderado, más de 60 alto.)*

Figura 13: Informe final de AuditTool

CAPÍTULO 5: Conclusiones y trabajo futuro

5.1 Conclusiones

Con AuditTool, la herramienta creada en este proyecto para realizar auditorías de seguridad a las organizaciones, se ha conseguido cumplir los objetivos expuestos en el primer capítulo relacionados con el desarrollo de una herramienta para que analizara la seguridad de los sistemas informáticos. Estos objetivos consistían en la implementación de dicha herramienta y en la definición de una métrica de seguridad que permitiese certificar el grado de seguridad informática de una organización de forma cuantitativa y que fuese utilizada por la herramienta de auditoría implementada.

AuditTool ofrece la posibilidad de evaluar el riesgo que corren las organizaciones de forma rápida y sencilla. Una vez auditada la organización AuditTool proporciona un informe general claro y conciso sobre el nivel de riesgo que corren sus redes.

Además de estas conclusiones generales, el trabajo desarrollado a lo largo de este proyecto ha permitido llegar a otras conclusiones más particulares.

Uno de los objetivos del proyecto era estudiar los diferentes sistemas de detección de intrusos y de los tipos de servicios de auditorías que hay en el mercado, así como estudiar las métricas de seguridad existentes en la actualidad. De estos estudios la primera conclusión que se ha sacado es que hay una gran falta de información, sobre todo en castellano, sobre métricas de seguridad y escáneres de vulnerabilidades. Si se busca información sobre detectores de intrusos se podrán encontrar varios estudios, pero excepto uno ([9]) los demás están en inglés u otro idioma diferente al castellano. Este hecho agrega valor al proyecto ya que en pocas publicaciones se podrá encontrar la recopilación de información contenida en él.

Centrándose más en el estudio de los IDS se obtiene la conclusión de que el futuro de las técnicas de defensa pasa por utilizar sistemas de prevención de intrusiones, que aunque en la actualidad no están muy desarrollados sí serán en el futuro unas herramientas indispensables para proteger los sistemas informáticos. Tampoco hay que olvidar que por si solos los IDS o IPS no son suficientes para proteger adecuadamente los sistemas informáticos, por lo que hay que utilizarlos en combinación de otros sistemas de seguridad como los escáneres de vulnerabilidades.

Por otra parte los escáneres de vulnerabilidades son herramientas esenciales si se quiere mejorar la seguridad de los sistemas o controlar los cambios en la seguridad. La multitud de herramientas de análisis de vulnerabilidades que existen abarcan precios y características para cualquier presupuesto u organización.

Con el estudio de las métricas de seguridad se llegó a la conclusión de que hay muy pocos estudios realizados y menos aún sobre métricas cuantitativas. La mayoría de los estudios sobre métricas de seguridad están basados en métodos cualitativos o en combinaciones de métodos cualitativos y cuantitativos. Aunque en este proyecto se hace un primer acercamiento a una métrica cuantitativa, en la actualidad, todas las métricas de seguridad están basadas en valoraciones del riesgo al que están expuestos los sistemas informáticos, y siempre son subjetivas a la experiencia del creador de la métrica o del administrador del sistema auditado. Aún queda mucho para poder valorar la seguridad de cualquier organización de forma general, sencilla y cuantitativa. En definitiva, aunque se está empezando a tener conciencia de la importancia de las métricas de seguridad en las organizaciones, todavía queda mucho por hacer en el campo de las métricas de seguridad informática.

En lo que se refiere a la herramienta, para cumplir el objetivo de integrar en una única herramienta los procedimientos necesarios para determinar la seguridad a diferentes niveles de abstracción en los sistemas informáticos utilizando técnicas pasivas y activas se planteó una estructura basada en módulos. En esta estructura se separaron, en diferentes módulos, las técnicas pasivas y las técnicas activas de tal forma que el módulo pasivo fuese independiente del módulo activo, y en el módulo activo sólo se tuviese que adaptar la recogida de datos ofrecidos por el módulo pasivo. De esta forma, además de cumplir el objetivo propuesto, se ha conseguido crear una herramienta fácilmente ampliable, mejorable, adaptable y modificable que puede ser utilizada como base para futuras herramientas de auditoría más avanzadas.

De la implementación del módulo pasivo se han obtenido varias conclusiones. Se puede empezar por decir que la mejor forma de obtener información sobre las características de las organizaciones es mediante un cuestionario. Además, en la realización de este cuestionario se debe tener cuidado con el número de preguntas que se realizan, ya que un número grande podría complicarlo innecesariamente y con un número pequeño los datos recogidos podrían no ser suficientes. Por último es importante que el cuestionario sea rellenado por un administrador experto para que las preguntas sean contestadas correctamente.

El hecho de elegir un escáner de vulnerabilidades para recoger datos sobre los sistemas informáticos en el módulo activo fue por la capacidad de estas herramientas de obtener datos sobre la seguridad de forma proactiva, por lo que no es necesario que se produzcan intrusiones para obtener información relativa a la seguridad.

La métrica definida para evaluar el riesgo que corren las organizaciones es una primera aproximación para futuras métricas cuantitativas y está creada para ser evaluada de forma rápida y efectiva. Aunque esta métrica está basada en la arquitectura de red propuesta en [88] cumplir con el objetivo conjunto de los cuatro proyectos mencionados en el punto 1.3, sirve como base para futuras métricas cuantitativas más generales.

En definitiva AuditTool ha cumplido los objetivos marcados pudiéndose utilizar por los administradores o encargados de la seguridad de los sistemas informáticos para saber qué configuraciones de la red de la organización aportan una mayor seguridad o un menor riesgo y para saber cómo afectan los cambios realizados en los sistemas. También los resultados obtenidos con la evaluación de la métrica creada han sido suficientemente satisfactorios como para tenerla en cuenta en futuros estudios sobre

métricas cuantitativas y utilizarla como base en nuevos proyectos sobre métricas de seguridad.

5.2 Trabajo futuro:

Este proyecto deja muchas puertas abiertas para futuros estudios o proyectos. Una de las principales líneas de trabajo que se pueden seguir es la de las métricas de seguridad. Se pueden plantear nuevas métricas cuantitativas más generales o centrándose en otros tipos de arquitecturas de red.

Por otra parte, otra posible línea de trabajo se basaría en la herramienta AuditTool. Se podría mejorar optimizando la transferencia de información entre módulos o ampliándola para que se pudiesen hacer auditorías de seguridad más exhaustivas. Sería interesante probar otro tipo de escáneres de vulnerabilidades o implementar uno nuevo e incluirlos en el módulo activo. También se podrían realizar estudios sobre el comportamiento de AuditTool en sistemas reales, haciendo baterías de pruebas exhaustivas con administradores de sistemas experimentados.

Respecto a Nessus, se podrían plantear varios trabajos como su optimización para diferentes tipos de arquitectura de red o realizar una optimización más exhaustiva para un tipo de arquitectura de red concreto. Para esto se podría estudiar y utilizar NASL (el lenguaje de "script" propio de Nessus) para crear los "plugins" necesarios para analizar específicamente las máquinas de cada red.

Por último se podría plantear una herramienta similar a AuditTool con la que se pudiese auditar las organizaciones desde plataformas UNIX. O implementar o modificar el servidor de Nessus para que funcionase bajo Windows.

ANEXO I

INSTALACIÓN Y EJECUCIÓN DE AUDITOOL

En este anexo se explica como instalar la herramienta AuditTool, como configurarla y como ejecutar la primera auditoría.

1. Instalación

La instalación de AuditTool es recomendable dividirla en dos partes: la instalación y configuración del servidor de Nessus [29] y la instalación de la herramienta AuditTool.

1.2 Instalación y configuración del servidor de Nessus

El servidor de Nessus sólo está disponible para plataformas UNIX, por lo que se necesita tener instalada en una máquina alguna distribución de Linux [87] o UNIX. Dependiendo de la distribución que se tenga instalada los pasos a realizar para instalar el servidor serán diferentes. En el caso de tener una distribución Linux basada en Debian [86] se puede directamente descargar e instalar automáticamente el servidor desde el repositorio de Debian (para las demás distribuciones se puede consultar la página oficial y otras múltiples páginas en Internet). Los archivos del servidor de Nessus se encuentran en el directorio del CD-ROM "\AuditTool\ServidorNessus"

Para bajar e instalar el servidor del repositorio de Debian hay que entrar como super usuario ("root") en el sistema e introducir en la consola: "apt-get install nessusd". Es recomendable para obtener la última versión de Nessus tener actualizados los archivos de configuración del comando "apt-get".

Cuando termina la instalación lo primero que se requiere es añadir un usuario. En la ejecución de AuditTool se requerirán los datos de uno de los usuarios creados en el servidor de Nessus. Un nuevo usuario puede ser añadido mediante el comando: "nessus-adduser". Al ejecutar el comando lo primero que habrá que introducir será en nombre del nuevo usuario, después se pedirá que elija entre el método de autentificación que desee. Se puede elegir entre una autentificación mediante clave ("pass") o mediante certificado ("cert"). En este caso se deberá de elegir la autentificación mediante clave ya que AuditTool es la que utiliza. En el siguiente paso se preguntará por las reglas que se aplicarán al nuevo usuario. En este apartado se puede especificar por ejemplo las direcciones IP que podrá escanear el usuario, pero en este caso no se introducirá

ninguna regla dejando en blanco este apartado. Si se quiere personalizar el servidor habría que modificar los parámetros que están guardados en el fichero "/etc/nessus/nessud.conf" pero de nuevo en este caso la configuración por defecto del servidor no necesita ser modificada.

1.3 Instalación de AuditTool

La instalación de AuditTool requiere tener instalado Windows XP [83] en la máquina donde se desee instalar. Una vez instalado el Windows XP bastará con copiar el directorio con nombre "\AuditTool" del CD-ROM en el directorio raíz "C:\" del sistema operativo. Una vez realizados estos pasos la herramienta estará lista para ejecutarse.

1.3.1 Compilación del código fuente de AuditTool

Otra forma de instalar AuditTool es compilando el código fuente del módulo activo y del de evaluación de la métrica que se proporciona en el directorio "\AuditTool_cf\AuditTool" del CD-ROM. Antes de poder compilar el código fuente de AuditTool se necesita instalar algunas herramientas y librerías que se detallan a continuación:

1- Visual C++: Para compilar el código fuente de AuditTool sin tener que hacer ninguna modificación en el código se debe compilar con el entorno de desarrollo Visual C++, ya que AuditTool fue creado con esta herramienta y sus librerías MS VC. Es posible compilarlo con GCC, pero se necesitan hacer algunas modificaciones.

2- PdfLib: Estas librerías son utilizadas por el cliente NessusWX [89] en el que se basa el módulo activo de AuditTool para generar informes en formato .pdf. Los ficheros .h y .lib necesarios de estas librerías se encuentran en el directorio "\AuditTool_cf\pdflib" del CD-ROM. También se pueden descargar los archivos binarios de http://www.pdflib.org y posteriormente ser compilados.

3- OpenSSL: Estas librerías son utilizadas por NessuWX para comunicarse con el servidor de Nessus bajo una conexión SSL. Las librerías de OpenSSl se encuentran en el directorio "\AuditTool-cf\openssl" del CD-ROM y también se pueden descargar de http://www.openssl.org.

4- HTML Help Workshop: Estas librerías son utilizadas para por NessusWX para implementar la ayuda de la aplicación. Se pueden encontrar en la página Web de Microsoft o instalarlas desde el directorio del CD-ROM "\AuditTool_cf\HTML Help Workshop".

5- Una vez instaladas las librerías y el entrono de desarrollo Visual C++ se tendrá que copiar el directorio "\AuditTool_cf" del CD-ROM en el disco duro. Luego se deberá abrir el proyecto AuditTool con el fichero "\AuditTool_cf \AuditTool\AuditTool.dsw", incluir las librerías requeridas y compilar la herramienta.

Una vez compilado el código fuente de la herramienta AuditTool se deberán copiar los archivos resultantes de la compilación y los formularios .html que se encuentran en el directorio "\AuditTool" del CD-ROM a directorio "C:\ AuditTool" del disco duro en donde se desee instalar la herramienta.

2. Ejecución y funcionamiento

Lo primero que hay que hacer es poner en funcionamiento el servidor de Nessus. Para ejecutar el servidor habrá que introducir en la consola de comandos: "nessusd". El servidor se quedará esperando a recibir las órdenes del cliente.

Una vez este el servidor esperando se deberá ejecutar el archivo "AuditTool.html" para lanzar el módulo pasivo de AudiTool.

Tras ejecutar el fichero se abrirá el cuestionario en donde habrá que introducir la dirección IP de la máquina donde esté instalado el servidor, el nombre ("Login") del usuario así como su clave ("Password") y contestar a las demás preguntas sobre la organización a auditar.

Después de rellenar correctamente las preguntas del primer formulario se pulsará el botón "Continuar". A continuación, y dependiendo del número de redes de la organización se abrirán un número determinado de formularios (un formulario por cada red). Se deberán contestar cada uno de ellos adecuadamente y pulsar al terminar el botón "Guardar". Sólo se deberán tener en cuenta que el botón "Continuar" del cuestionario de la red principal no se deberá pulsar hasta que se hayan contestado y guardado los cuestionarios de las demás redes.

Después de pulsar el botón "Continuar" del cuestionario de la red principal se ejecutará el módulo activo de AuditTool y aparecerá en el monitor una ventana que mostrará el proceso de análisis de vulnerabilidades de las máquinas de las redes introducidas en los cuestionarios.

Tras finalizar el análisis de las máquinas se guardará en el escritorio de Windows XP el fichero inforFinal.html que contendrá el informe final de la auditoría mostrando el nivel de riesgo que corre la empresa entre otros datos.

REFERENCIAS

[1] Eurologic. http://www.eurologic.es/conceptos/conbasics.htm

[2] CSIC. http://www.iec.csic.es/criptonomicon/seguridad

[3] Conceptos y Prácticas de Seguridad Informática de Antonio Sanz.
http://catedratelefonica.unizar.es/tecnologias_de_red/seguridad.pdf

[4] http://webs.ono.com/usr026/Agika2/3internet/ataques.htm

[5] Instituto Seguridad Internet. http://www.instisec.com/publico/xss.asp

[6] Alerta-Antivirus. http://alerta-antivirus.red.es/seguridad/ver_pag.html?tema=S&articulo=4&pagina=1

[7] Criptografía y Seguridad en Computadores escrito por Manuel José Lucena López. 2002.
http://www.dragones.org/Biblioteca/arti-cri.htm

[8] Auditoria aplicada a la seguridad en redes de computadores de Julián Gutiérrez Melo.
http://www.monografias.com/trabajos10/auap/auap.shtml#cri

[9] Sistemas de Detección de Intrusiones escrito por Diego González Gómez. 2003.
http://www.dggomez.arrakis.es/secinf/ids/html/index.htm

[10] NSS Group. http://www.nss.co.uk/ips/edition3/index.htm

[11] Informática64.
http://www.informatica64.com/materialSeminarios/II_Seminario_Seguridad_I64.ppt#97

[12] WindowSecurity.com. http://windowsecurity.com/articles/IDS-Part2-Classification-methods-techniques.html

[13] The Science of Intrusion Detection System Attack Identification de Cisco Systems.
http://www.cisco.com/warp/public/cc/pd/sqsw/sqidsz/prodlit/idssa_wp.htm

[14] Intrusion Detection de Prabhaker Mateti.
http://www.cs.wright.edu/~pmateti/InternetSecurity/Lectures/IntrusionDetection/index.html

[15] IDS basado en Mapas Autoorganizados de P. Cortada, G. Sanromà, P. García, A. Arenas y R. Rallo, 2002. http://www.etse.urv.es/~rrallo/papers/ids_slides.pdf

[16] Windows vs. Linux Web Server Role Security Research Study de SecurityInnovation.
http://www.securityinnovation.com/resources/linux_windows.shtml

[17] World Wide School.
http://www.worldwideschool.org/library/books/tech/computers/TheHackersDictionaryofComputerJargon/chap55.html

[18] Seguridad en UNIX y redes de Antonio Villalón Huerta, 2002.
http://es.tldp.org/Manuales-LuCAS/doc-unixsec/unixsec-html/unixsec.html

[19] Searchsecurity.com. http://searchsecurity.techtarget.com/

[20] SecuriTeam. http://www.securiteam.com/securityreviews/5DP0N1P76E.html

[21] Honeynet Proyect. http://www.honeynet.org/papers/phishing/

[22] Rookit. http://www.rootkit.com/

[23] Hackers 3: Defensa y Ataque Autor: Claudio Hernández

[24] Whitsitt Jr, John y Alberto Gonzalez. The Bait and Switch Honeypot.
http://baitnswitch.sourceforge.net

[25] Internet Scanner de Internet Security Systems. http://www.iss.net

[26] Retina de eEye Digital Security's. http://www.eeye.com/html/

[27] Symantec's NetRecon de Symantec. http://www.symantec.com/index.htm

[28] SAINT de SAINT. http://www.saintcorporation.com

[29] Nessus de Nessus. http://www.nessus.org

[30] Storm watch de OKENA. http://www.okena.com

[31] Entercept de Entercept. http://www.entercept.com

[32] Cerberus' Internet Scanner (CIS) de Cerberus Information Security, Ltd.
http://www.cerberus-infosec.co.uk/cis.shtml

[33] SATAN creado por Venema, Wietse y Dan Farmer. http://www.porcupine.org/satan/

[34] Real Academia de la Lengua. http://www.rae.es

[35] Tripwire de Tripwire. http://www.tripwire.com/

[36] Advanced Intrusion Detection Environment (AIDE). http://www.cs.tut.fi/~rammer/aide.html

[37] SamHian de SamHian Labs. http://la-samhna.de/samhain/

[38] LogCheck. http://logcheck.org/ ó http://www.psionic.com/abacus/logcheck

[39] LogWatcher de Acrasoft. http://www.acrasoft.com/lw.html

[40] RealSecure Network de Internet Security Systems.
http://www.iss.net/products_services/enterprise_protection/rsnetwork/sensor.php

[41] Cisco IDS de Cisco. http://www.cisco.com/en/US/products/sw/secursw/ps2113/index.html

[42] Dragon Intrusion Detection System de Enterasys. http://www.enterasys.com/products/ids

[43] Snort de Sourcefire Network Security. http://www.snort.org/

[44] Modern Intrusion Detection, Data Mining, and Degrees of Attack Guilt de Steven Noel, Duminda Wijesekera y Charles Youman. http://www.cse.msu.edu/~wuming/papers/IDS%20chapter.pdf

[45] ComputerWatch de AT&T Bell Labs. http://www.bell-labs.com/

[46] Stalker de Haystack Labs. http://www.haystack.com

[47] SARA - Security Auditor's Research Assistant de The Advanced Research Corporation.
http://www.www-arc.com/sara/

[48] NetRanger de Cisco Systems. http://www.cisco.com/univercd/cc/td/doc/product/iaabu/netrangr/

[49] Emerald eXpert-BSM de EMERALD. http://www.sdl.sri.com/projects/emerald/releases/eXpert-BSM/

[50] SecureNet de Intrusion. http://www.intrusion.com/products/securenet/

[51] NIST - National Institute of Standards and Technology. http://www.nist.gov

[52] "An Attack Surface Metric" de Pratyusa Manadhata y Jeannette M. Wing.
http://reports-archive.adm.cs.cmu.edu/anon/2005/CMU-CS-05-155.pdf

[53] CERT Advisories. http://www.cert.org/advisories/

[54] Microsoft Security Bulletins. http://www.microsoft.com/technet/security/current.asp

[55] MITRE CVEs. http://www.cve.mitre.org

[56] Andy Chou, Junfeng Yang, Benjamin Chelf, Seth Hallem and Dawson Engler, An Empirical Study of Operating System Errors, Proceedings of ACM Symposium on Operating System Design and Implementation (2001).
[57] J. Gray, A Census of Tandem System Availability between 1985 and 1990, IEEE Transactions on Software Engineering 39,4 (1990) p. 409-418.
[58] I. Lee and R. Iyer, Faults, Symptoms, and Software Fault Tolerance in the Tandem GUARDIAN Operating System, Proceedings of International Symposium on Fault-Tolerant Computing (1993).
[59] M. Sullivan and R. Chillarge, Software Defects and their Impact on System Availability- A Study of Field Failures in Operating Systems, Proceedings of International Symposium on Fault-Tolerant Computing (1991).
[60] SecurityFocus. http://www.securityfocus.com/archive/1
[61] Shawn A. Butler, Security Attribute Evaluation Method: A Cost-Benefit Approach, Proceedings of International Conference on Software Engineering (2002).
[62] Hilary Browne, John McHugh, William Arbaugh and William Fithen, A Trend Analysis of Exploitations, Proceedings of IEEE Symposium on Security and Privacy (2001).
[63] Steve Beattie, Seth Arnold, Crispin Cowan, Perry Wagle, Chris Wright and Adam Shostack,Timing the Application of Security Patches for Optimal Uptime, Proceedings of LISA: Systems Administration Conference (2002).
[64] S. Brocklehurst, B. Littlewood, T. Olovsson and E. Johsson, On Measurement of Operational Security, Proceedings of Annual Conference on Computer Assurance (1994).
[65] Andy Chou, Junfeng Yang, Benjamin Chelf, Seth Hallem and Dawson Engler, An Empirical Study of Operating System Errors, Proceedings of ACM Symposium on Operating System Design and Implementation (2001).
[66] J. Alves-Foss and S. Barbosa, Assessing Computer Security Vulnerability, ACM SIGOPS Operating Systems Review 29,3 (1995) p. 3-13.
[67] J. Voas, A. Ghosh, G. McGraw, F. Charron, and K. Miller, Defining an Adaptive Software Security Metric from a Dynamic Software Failure Tolerance Measure, Proceedings of Annual Conference on Computer Assurance (1996).
[68] R. Ortalo, Y. Deswarte, M. Ka^aniche, Experimenting with Quantitative Evaluation Tools for Monitoring Operational Security, IEEE Transactions on Software Engineering 25,5 (1999) p.633-650.
[69] M. Dacier and Y. Deswarte, Privilege Graph: An extension to the Typed Access Matrix Model, Proceedings of European Symposium on Research in Computer Security (1994).
[70] FoundStone. http://www.foundstone.com/resources/whitepapers/wp_securitymetrics.pdf
[71] Schwartau, Winn Time Based Security , Interpact Press 1999.ISBN 0-9628700-4-8
[72] Seguridad Basada en Métricas de Iñigo González Ponce, 2000.
http://exocert.com/Docs/repository/disc2000-metricas-de-seguridad.pdf
[73] "Security: Measuring Up" de Pete Lindstrom.
http://informationsecurity.techtarget.com/magPrintFriendly/0,293813,sid42_gci1052390,00.html
[74] GFI LANguard Network Security Scanner (N.S.S.) de GFi. http://www.gfi.com/lannetscan/
[75] NeWT de Tenable Network Security. http://www.tenablesecurity.com/products/newt.shtml

[76] TyphonIII de NGS Software. http://www.nextgenss.com/typhon.htm

[77] NetIQ Vulnerability Manager de NetIQ. http://www.netiq.com/products/vsm/default.asp

[78] Whisker de rfp labs . http://www.wiretrip.net/rfp/

[79] Nikto de CIRT.net. http://www.cirt.net/code/nikto.shtml

[80] N-Stealth Security Scanner de N-Stalker. http://www.nstalker.com/eng/products/nstealth/

[81] NScan de NS Netfactory. http://www.nscan.org/

[82] Nessus Network Auditing de SensePost, Raven Alder, Jimmy Alderson, Andy Johnston, George A. Theall. SYNGREESS.

[83] Microsoft. http://www.microsoft.com/

[84] NessusWX - Nessus Client for Win32. http://nessuswx.nessus.org

[85] Ubuntu. http://www.ubuntulinux.org

[86] Debian. http://www.debian.org

[87] Linux. http://www.linux.org

[88] "Diseño e Implantación de Arquitecturas Seguras" realizado por Alberto Luís Corrales. ESCET-URJC.

[89] "Metodología para la fortificación de bases de datos" realizado por Carlos Rubio. ESCET-URJC.

[90] BlackICE Agent de NetworkICE.
http://www.digitalriver.com/dr/v2/ec_dynamic.main?SP=1&PN=12&sid=26412

[91] Tiny CMDS. - http://www.programmi.com/articolo.asp?id=339