Seguridad en Redes

Carlos J. Mateos Orozco
Alejandro Carrasco Muñoz

Capítulo 1

Introducción a la seguridad informática

1.1. ¿Qué es la seguridad informática?

No existe una definición consensuada de seguridad informática. Existen mil y una formas de definirla. Una "bastante completa" podría ser la siguiente:

*"Un concepto global de seguridad informática sería aquel definido como el **conjunto de sistemas, métodos, herramientas, procedimientos y actuaciones** encaminados a conseguir la protección de la información y la garantía de funcionamiento de los sistemas informáticos, obteniendo **eficacia**, entendida como el cumplimiento de la finalidad para el que estaba establecido, manteniendo la **integridad**, entendida como la inalterabilidad del sistema por agentes externos al mismo, y **alertando** la detección de actividad ajena, entendida como el control de la interacción de elementos externos al propio sistema. Si conseguimos todo esto, podremos decir que disponemos de un sistema seguro."*

Analizando y leyendo entre líneas este párrafo podemos sacar las siguientes ideas:

→ La seguridad, con todos sus componentes hardware, software, etc., **es un sistema** dentro de sistemas mayores.

→ Más que un producto o conjunto de ellos, más que una o varias tecnologías, la seguridad **es un proceso**, que hace intervenir a todas las tecnologías, todos los productos y, especialmente, el sentido común de los seres humanos que la gestionan.

La mayoría de los conceptos e ideas de la seguridad informática explicadas en este documento son exportables a cualquier otro mundo diferente del de los sistemas informáticos, aunque éste sea el tema que nos concierne ahora.

Antes de continuar definiremos algunos términos que, a lo largo del documento, veremos con bastante frecuencia:

- **Vulnerabilidad**: Debilidad.
- **Amenaza**: Probabilidad de que ocurra un hecho que puede provocar daño.
- **Ataque**: Acción cuyo objetivo es provocar daño. Suele aprovechar la existencia de una vulnerabilidad.
- **Riesgo**: Posibilidad de que, ante un ataque, nos veamos perjudicados.

1.1.1. Seguridad física v.s Seguridad lógica

El estudio de la Seguridad Informática podríamos plantearlo desde dos enfoques distintos aunque complementarios:

La **Seguridad Física**: Es la parte que asociada a la protección ante amenazas físicas como incendios, inundaciones, robo de material corporativo, control de accesos de personal autorizado, etc. Esta protección se aplica a edificios, cables, personas, etc.

La **Seguridad Lógica**: Asociada con la protección de la información en su propio medio de su revelación, del acceso a la misma y de su monitorización.

Y a medio camino de ambos conceptos está la **Gestión de la Seguridad**. Nos referimos a las políticas de seguridad, los planes de contingencia, las normativas, los planes de formación del personal, etc.

1.1.2. Principios de la seguridad informática

1. PRINCIPIO DE ACCESO MÁS FACIL

"La cadena es tan fuerte como su eslabón más débil"

¿Cuáles son los puntos débiles de un sistema informático?

El intruso estudiará la manera mas sencilla de acceder, y posteriormente, atacar al sistema.

Existirán múltiples frentes desde los que puedan producirse un ataque. Sin embargo, un atacante normalmente aplicará la filosofía del atacar contra el punto más débil.

2. PRINCIPIO DE CADUCIDAD DE LA INFORMACIÓN

"Los datos confidenciales deben protegerse sólo hasta que el secreto pierda su valor como tal"

¿Cuánto tiempo deberá protegerse un dato?

Se habla pues de **caducidad del sistema de protección**: tiempo durante el que debe mantenerse la confidencialidad o el secreto de la información.

3. PRINCIPIO DE EFICIENCIA

"Las medidas de control se implementan para ser utilizadas de forma efectiva. Deben ser eficientes, fáciles de usar y apropiadas al medio"

Esto significa que:

- Deben funcionar en el momento oportuno.
- Deben hacerlo optimizando los recursos del sistema.
- Deben pasar desapercibidas para el usuario normal, es decir, ser transparentes.

1.1.3. Objetivos de la seguridad informática

La seguridad de sistemas de información tiene por objetivo evitar que se violen los siguientes cuatro puntos. Su cumplimiento proporciona a un sistema la categoría de "**sistema seguro**":

• **Autenticidad**: Requiere que un sistema informático sea capaz de verificar la identidad de los usuarios que hacen uso de ellos de forma legítima.

• **Confidencialidad**: Exige que la información de un sistema informático sea accesible solamente para aquellas partes autorizadas. Este tipo de acceso incluye impresión, visualización y otras formas de revelación, incluyendo el simple conocimiento de la existencia de un objeto.

• **Integridad**: Establece que los elementos de un sistema informático puedan ser modificados sólo por partes autorizadas. La modificación incluye escritura, reemplazo, cambio de estado, borrado y creación.

• **Disponibilidad**: Exige que los elementos de un sistema informático estén siempre disponibles para las partes autorizadas.

Los actos que van en contra de la autenticidad, confidencialidad, la integridad y la disponibilidad son la falsificación, la revelación, la modificación y la denegación de servicio.

1.1.4. ¿Qué queremos proteger?

La seguridad persigue proteger lo que en lo sucesivo identificaremos como **ACTIVOS.** En resumidas cuentas, puede considerarse como un activo:

- La **información**.
- Los **equipos** que la soportan.
- Los **usuarios** que hacen uso de ella.

La información. Archivos de programas, documentos, informes, libros, manuales, correspondencias, patentes, información de mercado, código de programación, líneas de comando, reportes financieros, archivos de configuración, plantillas de sueldos de empleados, plan de negocios de una empresa, etc.

El interés por la seguridad con respecto a la información y los datos es amplio. Debemos preocuparnos por asegurar la disponibilidad, el secreto y la integridad de ésta.

Posibles amenazas: Robos de documentos, pérdida de archivos de configuración, acceso a archivos de logs, bases de datos, etc.

Equipos que la soportan. Claramente habrá que distinguir entre:

• **Software**. Este grupo contiene todos los programas que se utilizan para la automatización de procesos, es decir, accesos, lectura, escritura, tránsito y almacenamiento de la información. Entre ellos se pueden citar, aplicaciones generales, programas específicos y sistemas operativos, entre otros.

Las aplicaciones deberán estar protegidas para que su interacción con los datos, otras aplicaciones y los usuarios se realice de forma segura.

Posibles amenazas: Bugs, malas configuraciones, etc.

• **Hardware**. Estos activos representan toda la infraestructura tecnológica que brinda soporte a la información durante su uso, tránsito y almacenamiento. Entre los equipos que se encuentran en este grupo podemos citar cualquier equipo en el cual se almacene, procese o transmita la información. Los más importantes son: las computadoras, servidores, equipos portátiles, mainframes, medios de almacenamiento y elementos de interconexión por donde transita la información.

Posibles amenazas: Fallas eléctricas que dañen los equipos, inundaciones en centros de cómputo, robo, ataques a las redes, etc.

Usuarios. Personas que utilizan la estructura tecnológica y de comunicación de la empresa, y que manejan la información. El enfoque de la seguridad en los usuarios está orientado a la toma de conciencia de formación del hábito de la seguridad para la toma de decisiones y acción por parte de todos los empleados de una empresa. Esto engloba desde la dirección hasta los usuarios finales, incluyendo los grupos que mantienen en funcionamiento la estructura tecnológica, como técnicos, operadores y administradores.

Posibles amenazas: Olvido de contraseñas, descuido en el manejo de la información y en general falta de concienciación por parte de los usuarios en materia de seguridad.

El triángulo de debilidades

A partir de los elementos citados anteriormente se forma el llamado "triángulo de debilidades", compuesto por los principales elementos de un sistema informático que son susceptibles de ser atacados:

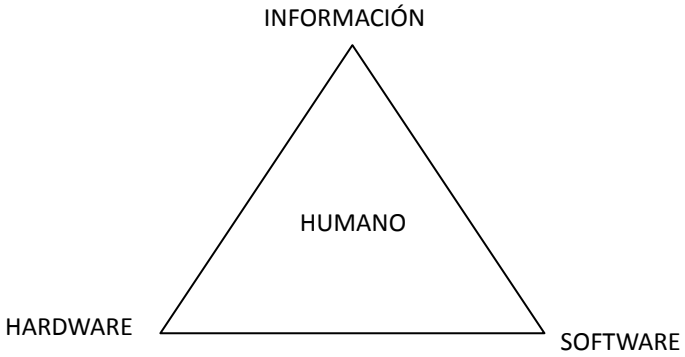

Fig. 1.1: Triángulo de debilidades

Todos los activos que se pretenden proteger están íntimamente relacionados. Por ejemplo, si se daña el hardware podemos perder la información almacenada y el software instalado. Es muy posible que un ataque que busque comprometer a alguno de ellos, se impliquen también a los restantes. Debemos concentrar nuestros esfuerzos en protegerlos ya que, por el "principio de acceso más fácil", un atacante siempre aprovechará cualquier vulnerabilidad en cualquiera de ellos para atacar.

En el centro de todas las vulnerabilidades se encuentra el factor humano. El usuario es considerado el punto más débil para cualquier atacante. Por mucho que protejamos nuestros sistemas, software o hardware, si un usuario con acceso privilegiado no es precavido, todas medidas de protección serán inútiles.

1.1.5. ¿De qué nos queremos proteger?

La seguridad informática tiene como objetivo la protección de los sistemas informáticos y de telecomunicaciones ante los riesgos y amenazas. Los riesgos se cuantifican en función de varios factores:

• Las amenazas existentes para los activos a proteger.

• Las vulnerabilidades de estos activos.

• La sensibilidad del activo.

No se puede realizar una clasificación sin un criterio. En el caso de los ataques y las amenazas existen varios, dependiendo de lo que se quiera analizar. Algunos criterios para clasificar ataques y amenazas pueden ser:

- El origen del ataque
- La complejidad
- El objetivo

1.1.5.1. Clasificación de ataques según el origen del ataque

Desde el punto de vista del origen desde el que se produce el ataque podemos distinguir entre:

Externos: El ataque es originado desde el exterior de la organización víctima, ya sea Internet ó una red externa en la que se confía pero que ha sido comprometida.

Internos: El ataque procede desde dentro de la organización víctima. Puede haber ataques profesionales internos, los cuales son muy peligrosos, pero también pueden producirse ataques desde el interior por culpa de una mala aplicación de la política de seguridad (o una mala política de seguridad en si misma). Incluso puede haber ataques no maliciosos de usuarios internos cuyos equipos han sido comprometidos o simplemente "prueban herramientas inconscientemente".

1.1.5.2. Clasificación de ataques según la complejidad

En este sentido se puede hablar de ataques:

No estructurados: No se define un objetivo específico. Suelen tener por objetivo la realización de pruebas con herramientas de hacking contra objetivos aleatorios. Suelen ser facilmente reconocibles.

Estructurados: Son ataques que se enfocan como un proyecto. Se dirigen contra un objetivo muy concreto. Se estudian todos los detalles y debilidades del objetivo, intentando además evitar dejar huellas tras el ataque.

1.1.5.3. Clasificación de ataques según el objetivo

Por último clasificaremos los ataques según el objetivo buscado por el atacante.

Esta clasificación se contempla teniendo en cuenta la función del sistema como un suministrador y receptor de información. En general, en un sistema informático se genera un flujo de información desde un origen y hacia un destino.

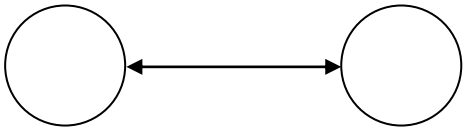

Fig. 1.2: Flujo normal de comunicación

A continuación se muestran cuatro categorías muy generales de ataques según su objetivo:

Interrupción: El objetivo es hacer inaccesible un elemento del sistema.
- Es un ataque a la disponibilidad (hardware o software).
- Su detección puede ser inmediata.

Como ejemplos podemos citar la destrucción de una pieza hardware, como un disco duro, el corte de una línea de comunicaciones, la inutilización del sistema de gestión de archivos o la saturación de un servidor.

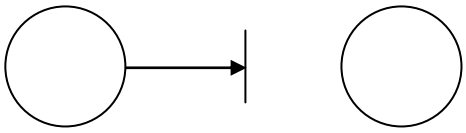

Fig. 1.3: Ataque de Interrupción

Interceptación: Tiene lugar cuando una parte no autorizada consigue acceder a un elemento durante la comunicación, y obtener privilegios para leer y supervisar el tráfico. La parte no autorizada puede ser una persona, un programa o un computador.
- Este es un ataque a la confidencialidad.
- Su detección es difícil. A veces no deja huellas.

Como ejemplos se incluyen la intervención de las conexiones telefónica, el "fisgoneo", el sniffing y la copia ilícita de archivos o programas desde el sistema víctima.

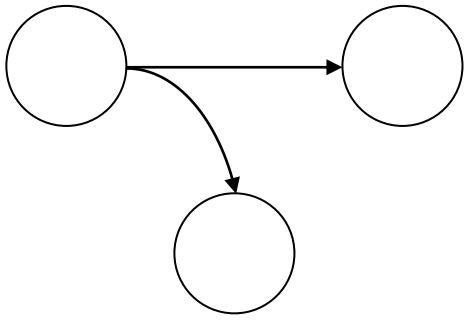

Fig. 1.4. Ataque de Interceptación

Modificación: Se produce cuando una parte no autorizada no sólo consigue acceder a información no autorizada, sino que también la modifica en tránsito. Estos ataques se conoce como "**Man in the middle**".

- Es un ataque a la confidencialidad y a la integridad.
- Su detección es puede ser fácil o difícil según las circunstancias.

Como ejemplos tenemos el cambio de valores en un archivo de datos, la alteración de un programa para que se comporte de manera diferente y la modificación de datos transmitidos en una red (Ej: ARP Poisoning).

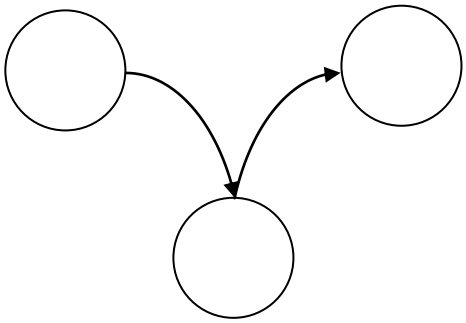

Fig. 1.5: Ataque de Modificación

Invención o Generación: Una parte no autorizada inserta objetos falsos en el sistema, suplantando a un emisor legítimo.

- Ataque a la autenticidad.
- Su detección es difícil. Engloba delitos de falsificación y suplantación de identidad.

Como ejemplos tenemos el phishing.

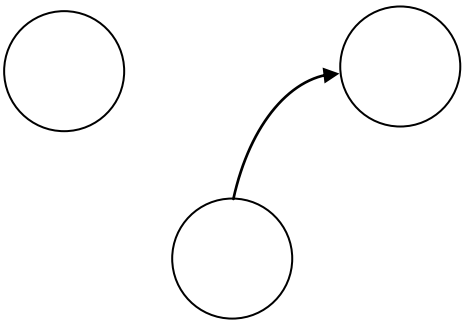

Fig. 1.6: Ataque de Generación

En la siguiente figura se muestra cómo afectan los ataques anteriores a cada uno de los vértices del triángulo de debilidades.

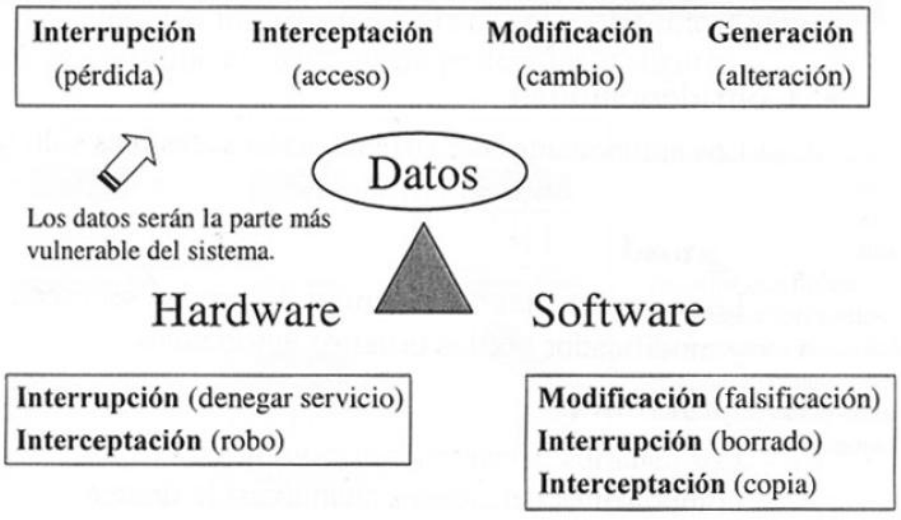

Fig. 1.7: El triangulo de debilidades. Amenazas

1.2. Gestión de la seguridad

La gestión de la seguridad de una organización puede ser algo infinitamente complejo dado el número de departamentos o de áreas que pueden conformarla. Hasta hace relativamente poco las preocupaciones en materia de seguridad de una organización se centraban en aspectos principalmente técnicos.

A partir de finales de los 90 es cuando comienza la seguridad a ir más allá de las cuestiones técnicas y se empieza a hablar de **gestión de la seguridad** como algo crítico para cualquier organización, igual de importante que los sistemas de calidad o las líneas de desarrollo de la propia organización.

1.2.1. Políticas de seguridad

El término **política de seguridad** se suele definir como el conjunto de requisitos, definidos por los responsables directos o indirectos de una organización (o de un sistema) que indican, en términos generales, que está y que no está permitido.

Una política de seguridad puede ser:

- **Prohibitiva**: Todo lo que no está expresamente permitido se deniega.

- **Permisiva**: Todo lo que no está expresamente prohibido se permite.

Una política de seguridad debe conseguir asegurar los objetivos de seguridad definidos anteriormente en un grado adecuado a los objetivos de la empresa. Para cubrir dichos objetivos de forma adecuada una política se suele dividir en puntos más concretos llamados **normativas**. El estándar **ISO-27001** define las siguientes líneas de actuación:

- **Seguridad organizativa**: Aspectos relativos a la gestión de la seguridad dentro de la organización (cooperación con elementos externos, outsourcing, estructura del área de seguridad…).

- **Clasificación y control de activos**: Inventario de activos y sus mecanismos de control, así como etiquetado y clasificación de la información corporativa.

- **Seguridad del personal**: Formación en materia de seguridad, cláusulas de confidencialidad, reporte de incidentes, monitorización del personal, etc.

- **Seguridad física y del entorno**: Aspectos relativos a la seguridad física del recinto, donde se encuentran los diferentes recursos de la organización y de los sistemas en si.

- **Gestión de comunicaciones y operaciones**: Engloba aspectos de la seguridad relativos a la interoperabilidad de los sistemas y las telecomunicaciones. Controles de redes, protección frente al malware, gestión de copias de seguridad o intercambio de software dentro de la organización.

- **Controles de acceso**: Definición y gestión de puntos de acceso a los recursos informáticos de la organización: contraseñas, **seguridad perimetral**, monitorización de accesos, etc.

- **Desarrollo y mantenimiento de sistemas**: Seguridad en el desarrollo y en las aplicaciones, cifrado de datos, controles de software, etc.

- **Gestión y continuidad del negocio**: Definición de continuidad del negocio, análisis de impactos, simulacros de catástrofes, planes de contingencia, etc.

- **Requisitos legales**: Una política ha de cumplir con la normativa vigente en el país donde se aplica.

Podemos resumir diciendo que una buena política de seguridad debe cumplir las siguientes normas generales:

- Debe poder implantarse.
- Debe entenderse.
- Debe cumplirse.
- Debe definir responsabilidades.
- Debe permitir que siga realizándose el trabajo normal.
- Debe ser exhaustiva (tener en cuenta todos los componentes que ha de proteger).
- Debe incluir mecanismos de respuesta.
- Debe tener mecanismos de actualización.

1.2.2. La rueda de la seguridad

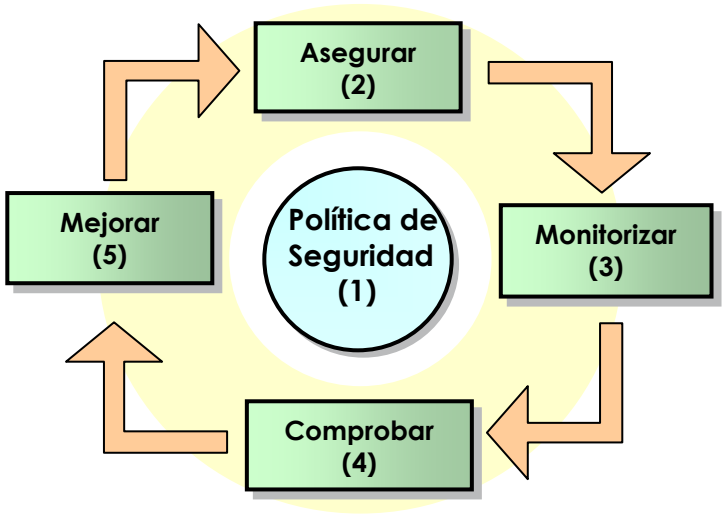

Fig. 1.8: Rueda de la seguridad

La "rueda de la seguridad" representa una metodología destinada a verificar que se han implementado correctamente las medidas de seguridad y que funcionan correctamente.

Se trata de un proceso continuo que invita a probar y a aplicar las medidas de seguridad actualizadas constantemente.

Este proceso se compone de las siguientes fases:

1. **POLÍTICA DE SEGURIDAD**

   Definir una política de seguridad es el punto de partida de esta metodología y el eje sobre el que se sustentan los pasos restantes.

2. **ASEGURAR**

   En esta fase el objetivo es detener y evitar el acceso de los intrusos y las actividades no autorizadas. Las soluciones de seguridad deben permitir y asegurar que ninguno de los cuatro pilares susceptibles de amenazas sean comprometidos. Para ello existen múltiples soluciones:

   - **PKI (Public Key Infraestructure)**: Es la forma común de referirse los sistemas de criptografía basados en el intercambio de clave pública y privada. Es la base fundamental en la gestión de certificados digitales y aplicaciones de la firma digital.

- **Firewalls/Proxys**: Filtran el tráfico de la red para permitir sólo aquel el tráfico y los servicios validados por la política de seguridad de la empresa.

- **IPSec**: Conjunto de protocolos, cuya función es asegurar las comunicaciones sobre el protocolo IP autenticando y/o cifrando cada paquete IP en un flujo de datos. IPSec también incluye protocolos para el establecimiento de claves de cifrado.

- **VPN**: Redes privadas virtuales que oculten el contenido del tráfico para evitar que los individuos no autorizados o malintencionados lo descubran.

3. MONITORIZAR y REACCIONAR

Implica métodos activos y pasivos de detección de violaciones en tiempo real:

- **IDS (Intrusión Detection System)**: Estos dispositivos pueden detectar las violaciones de la seguridad en tiempo real. Se pueden configurar para que respondan de forma automática antes de que el intruso pueda causar algún daño.
- **Monitor de logs**
- **Honeypots**: También conocidos como señuelos. Son sistemas cuyo objetivo es atraer a atacantes simulando ser sistemas vulnerables para así recoger información de los atacantes y estudiar sus métodos.

4. COMPROBAR

Debemos asegurar de forma proactiva la funcionalidad de las soluciones de seguridad implementadas en el paso 2 y los métodos de monitorización del paso 3.

En este punto incluimos auditorías y pruebas de intrusión (pentesting).

5. GESTIONAR Y MEJORAR

En ésta fase se analizan los datos recopilados durante las fases de monitorización y prueba. Con esta información se desarrollan e implementan mecanismos de mejora que realimentan la política de seguridad.

Para mantener nuestra red segura este proceso debe repetirse continuamente.

1.2.3. Análisis de riesgos

Para tratar de enfocar el problema con acierto, cuanto más conocimiento tengamos de qué se puede perder, qué se quiere proteger, de quién se quiere proteger, cómo pueden ser los ataques, cuales pueden ser las defensas y cuanto se tiene que invertir en ello, mejor.

El análisis de riesgos es un proceso de identificación y evaluación del riesgo a sufrir un ataque, y como consecuencia perder datos, tiempo y horas de trabajo, comparándolo con el coste que significa la prevención del suceso. Su análisis no solo nos lleva a establecer un nivel adecuado de seguridad, sino que nos permite conocer mejor el sistema que vamos a proteger.

Una vez conozcamos y evaluemos los riesgos a los que nos enfrentamos podremos definir políticas e implementar soluciones prácticas.

En España existe una metodología estándar para el análisis de riesgos denominada **MAGERIT**.

Ecuación básica del Análisis de Riesgo: **RIESGO vs CONTROL vs COSTE**

$$B < P * L$$

B: Es la carga o **coste** que significa implantar medidas de prevención ante un riesgo específico.

Ejemplo: El coste necesario a invertir para que un sistema informático minimice el riesgo de que sus servidores sean atacados desde fuera incluye la instalación de software y hardware adecuados, un cortafuegos, un sistema de detección e intrusos, una configuración de red segura, una política de seguimiento de accesos y de passwords, personal técnico cualificado, etc.

L: Es el impacto o coste total que supone una cierta pérdida. Es difícil de evaluar. Incluye daños a la información, a los equipos, pérdidas por reparación, pérdidas por horas de trabajos, etc. Siempre tendrá una parte de valoración subjetiva. No hay que olvidar que la pérdida de datos puede llevar a una pérdida de oportunidades por el llamado efecto cascada. En la organización debe existir una comisión interna o externa especializada que sea capaz de evaluar todas las posibles pérdidas y cuantificarlas.

P: Probabilidad de que una vulnerabilidad sea aprovechada por un atacante y provoque una pérdida. Este valor está relacionado con la determinación del impacto total **L** y depende del entorno en el que se sitúe el riesgo. Como este valor es difícil de cuantificar, dicha probabilidad puede asociarse a una frecuencia conocida.

Una vez se conoce **P** para un **L** dado, se obtiene la probabilidad de **pérdida relativa** de la ocurrencia **P*L**, que se comparará con **B**.

- Si B ≤ P * L: El coste que supone perder el activo es mayor que el coste que supone implantar una medida de prevención. Hay que implementar una medida de prevención o mejorar la existente.

- Si B > P * L: No es necesaria una medida de prevención ya que el coste de implantarla es mayor que el coste que supone perder el activo que se quiere proteger.

En definitiva: **el coste necesario para implantar las medidas de prevención ha de ser menor que el coste que supone perder el activo que se protege.**

El verdadero problema radica en muchos casos en la dificultad para estimar de forma precisa el impacto económico que puede suponer el hecho de que ocurra la amenaza en concreto.

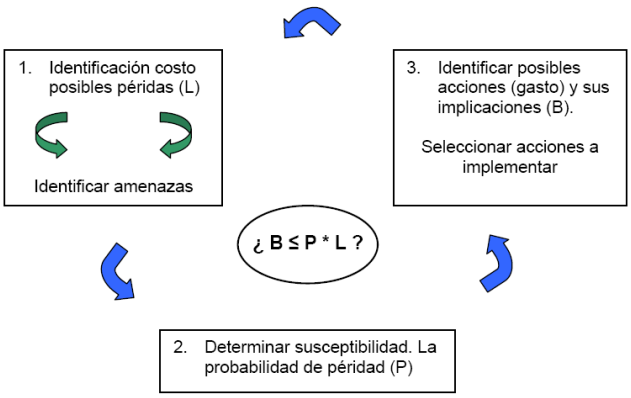

Fig. 1.9: Análisis de riesgos

1.3. Nuestro objetivo: La seguridad en redes

A lo largo del capítulo se han presentado una serie de ideas relacionadas con el mundo de la seguridad de los sistemas informáticos y con la protección de la información.

Es un mundo amplio que afecta a campos muy heterogéneos, unos más técnicos, otros de carácter estratégico, organizativos, políticos, etc. Esto significa que se tiene abordar desde diferentes frentes y en lo que respecta a esta asignatura nos centraremos en aspectos principalmente técnicos como punto de partida.

Dentro de los aspectos técnicos nuestro punto de mira serán las redes de computadores ya que protegerlas, al fin y al cabo, implicará proteger los sistemas que la conforman, las máquinas que conectan y la información que se desplaza de unas a otras.

NOTA: A partir del siguiente capítulo se explicarán cuestiones cuya comprensión necesitan de sólidos conocimientos de redes TCP/IP.

Seguridad en Redes

Carlos J. Mateos Orozco
Alejandro Carrasco Muñoz

Capítulo 2

Peligros, amenazas y defensas

2.1. Peligros y modos de ataque

"Los ataques son cada vez mas sofisticados, y a la vez se requieren menos conocimientos técnicos para llevarlos a cabo"

Podríamos definir ataques como todas aquellas acciones que supongan una violación de la seguridad de nuestros sistemas, es decir, comprometan la autenticidad, confidencialidad, integridad o disponibilidad de los mismos.

Antes de continuar hay que realizar un comentario muy importante: **Es improbable que se conozcan todos los tipos de ataques posibles**.

Hay que pensar si es posible que exista alguien capaz de abarcar todo lo hecho y lo que se podrá hacer mas adelante. Un buen profesional se puede conformar con conocer el mayor número posible de ataques y estar bien informado sobre todos los nuevos que se van publicando.

Cuanto mas conocimiento técnico se tenga (de sistemas operativos, programación, protocolos, redes, etc) más fácil será aprovecharse de cada una de las distintas vulnerabilidades que van apareciendo, e incluso aumenta la posibilidad de descubrir vulnerabilidades nunca publicadas.

Si buscamos en Internet una taxonomía de tipos de ataques nos encontraremos con que no existe una única clasificación. Es este documento se ha optado por clasificarlos en función de cuál es el objetivo del ataque:

- Ataques a la confidencialidad.
- Ataques a la autenticidad.
- Ataques a la disponibilidad.
- Ataques a la integridad.

2.1.1. Ataques a la confidencialidad

→ **Objetivo:** Obtener información privilegiada.

Ataques para obtener información sensible o acceso a sistemas privilegiados. Pueden estar basados en herramientas no diseñadas específicamente para un ataque. Un ejemplo sería el comando **ping**. Éste nos revela qué dirección IP tiene una máquina (además de conocer si llegamos hasta ella), o **tracert** para trazar una ruta desde un host a otro. Veamos con un poco mas de detalle algunas de estas técnicas y herramientas.

ESCANEO DE PUERTOS

El escaneo (o rastreo) como método para descubrir canales de comunicación susceptibles de ser explotados lleva en uso mucho tiempo. La idea es recorrer (escanear) tantos puertos de escucha como sea posible, y guardar información de aquellos que sean receptivos o de utilidad para cada necesidad en particular.

La idea básica es simple y para su mejor entendimiento vamos a realizar un símil con las llamadas telefónicas. Por ejemplo, se llama a un número y si da tono es que está conectado, entonces grabamos el número. En otro caso comunica, entonces se cuelga el teléfono y se llama al siguiente número. Escanear puertos implica estas mismas técnicas de fuerza bruta. Se envía una serie de paquetes para varios protocolos a un rango de direcciones y se deduce qué servicios y aplicaciones están "escuchando" (y por tanto levantadas en la máquina objetivo) por las respuestas recibidas o no recibidas. Como ejemplo de un escáner de puertos muy extendido tenemos **nmap**.

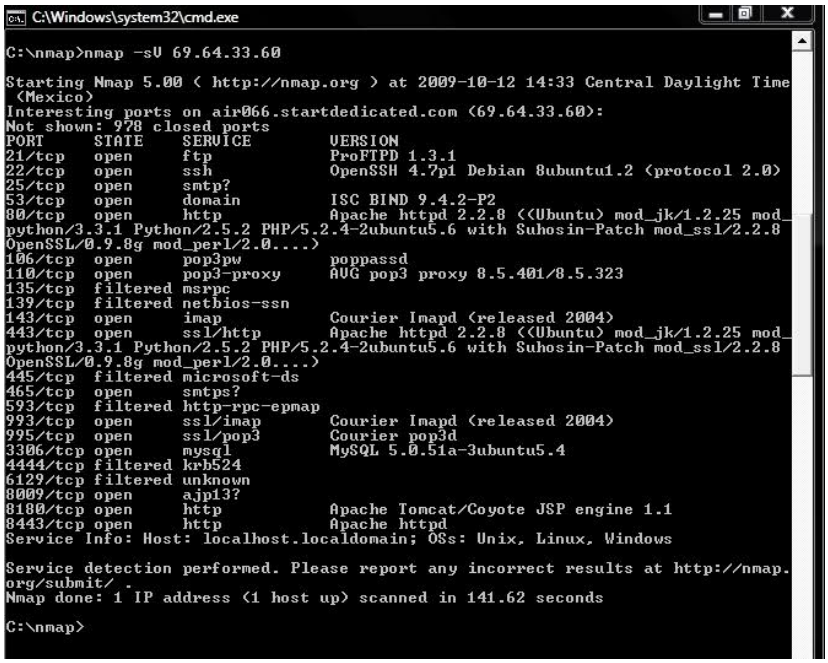

Fig. 2.1: Escaneo de puertos con nmap

***Contramedidas**: Filtrado de puertos, gestión de logs y alarmas.*

Existen diversos tipos de escaneo según técnicas, puertos y protocolos explotados:

- **TCP Connect Scan**: Esta es la forma mas básica que existe para escanear puertos TCP. Se implementa haciendo una llamada al sistema de tipo **connect( )**. Nos permite saber rápidamente si el puerto está o no abierto en la máquina destino:
    - Si se recibe un paquete con los flags **SYN/ACK** activados, el puerto está escuchando y se devolverá una respuesta de éxito (**ACK**) estableciendo conexión.
    - Si se recibe cualquier otra cosa (**RST/ACK**) significa que el puerto no está abierto o no se puede establecer conexión con él.

  Las ventajas que caracterizan esta técnica es que no necesita de privilegios especiales y su gran velocidad.

  Su principal desventaja es que este método es fácilmente detectable. El administrador del sistema verá un gran número de conexiones para los servicios con los que el atacante se ha conseguido conectar y mensajes de error para aquellos con los que no ha tenido éxito. Se verá como una máquina (la que lanza el scanner) se conecta e inmediatamente se desconecta, una y otra vez.

- **TCP SYN Scan**: Cuando dos procesos establecen una comunicación usan el modelo **cliente/servidor** para establecer la conexión. La aplicación del servidor "escucha" todo lo que ingresa por los puertos que está escuchando. El cliente establece la conexión con el servidor a través del puerto disponible para luego intercambiar datos. El establecimiento de dicha conexión se realiza mediante un protocolo llamado **Three-Way Handshake** (ver ANEXO).

  La técnica "TCP SYN Scan", implementa un escaneo de "**media-apertura**", dado que nunca se abre una sesión TCP completa. Se envía un paquete SYN (como si se fuera a usar una conexión real) a un puerto, y se espera la respuesta. Al recibir un SYN/ACK se envía, inmediatamente, un RST para terminar la conexión y se registra, el puerto en cuestión, como abierto. Si el servidor no devuelve nada ante un SYN significará que el puerto está cerrado.

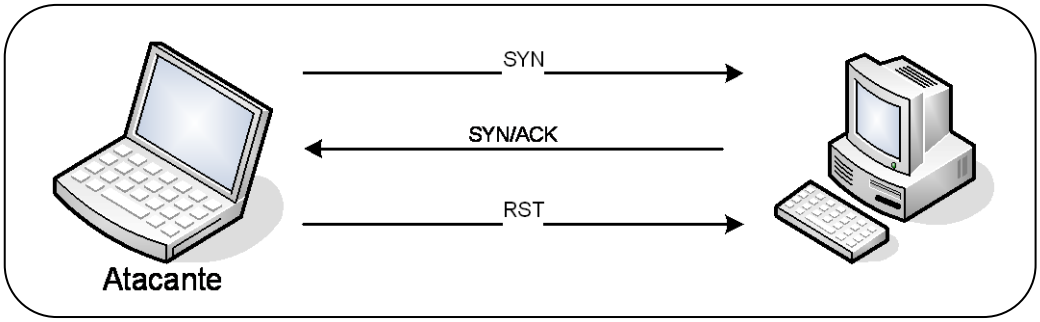

Fig. 2.2: TCP SYN Scanning

- **TCP FIN Scan**: Hay veces en que incluso el escaneo SYN no es lo suficientemente "clandestino" o limpio. Algunos sistemas (firewalls y filtros de paquetes) monitorizan la red en busca de paquetes SYN que tengan como objetivo puertos restringidos. Para subsanar este inconveniente esta técnica usa los paquetes FIN. Estos podrían ser capaces de pasar sin ser advertidos.

  Este tipo de escaneo está basado en la idea de que los puertos cerrados tienden a responder a los paquetes FIN con un RST. Los puertos abiertos, en cambio, suelen ignorar el paquete en cuestión.

  Este es un comportamiento correcto del protocolo TCP, aunque algunos sistemas (entre los que se hallan los de Microsoft) no cumplen con este requerimiento, enviando paquetes RST siempre, independientemente de si el puerto está abierto o cerrado. Como resultado, no son vulnerables a este tipo de escaneo. Sin embargo es posible realizarlo contra sistemas Unix.

  Este último es un ejemplo en el que se puede apreciar que algunas vulnerabilidades se presentan en la aplicación de tecnologías (en este caso el protocolo TCP nacido en los años 70) y no sobre sus implementaciones. Es más, se observa que una implementación incorrecta (la de Microsoft) soluciona el problema.

  "*Muchos de los problemas globales de vulnerabilidades son inherentes al diseño original de algunos protocolos*".

- **Fragmentation Scan**: En lugar de enviar paquetes completos de sondeo, éstos se particionan en pequeños fragmentos IP. Así se logra partir una cabecera IP en distintos paquetes para hacerlo más difícil de monitorizar por los filtros que pudieran estar ejecutándose en la máquina objetivo.

  Sin embargo, algunas implementaciones de estas técnicas tienen problemas con la gestión de este tipo de paquetes tan pequeños, causando una caída de rendimiento en el sistema del intruso o en el de la víctima. Problemas de esta índole convierte en detectables a este tipo de ataque.

SNIFFING

El sniffing (también conocido como "eavesdropping" ó intercepción pasiva) consiste en la escucha del tráfico que cursa por una red, sin intervenir activamente en la conexión (no hay modificación). Esto se realiza con unos programas denominados **packet sniffer** ó **sniffer** a secas. Éstos monitorizan los paquetes que circulan por la red. Los sniffers pueden situarse tanto en una estación de trabajo conectada a la red, como en un router o en un gateway de Internet, y esto puede ser realizado por un usuario con acceso legítimo (p.ej: el administrador de red), o por un intruso que tiene acceso a las comunicaciones. Los sniffers nos permiten, si se dispone de un enlace a una red, obtener todo el tráfico que circula por ésta sin interferir en la conexión. Esto incluye todos los datos, encriptados y no encriptados.

En la cabecera de los paquetes enviados a través de una red, entre otros datos, se tiene la dirección del emisor y la del destinatario. Cada maquina conectada a la red (con una dirección MAC única) verifica la dirección destino del paquete. Si la dirección destino del paquete coincide con la suya asume que el paquete enviado es para ella. En caso contrario reenvía el paquete.

Un sniffer básicamente lo que hace es establecer la interfaz de red del equipo en el que está funcionando en un modo llamado **promiscuo**, el cual desactiva el filtro de verificación de direcciones, lo que provoca que todos los paquetes enviados a la red sean capturados, aunque no sea destino del paquete.

Inicialmente este tipo de software era únicamente utilizado por los administradores de redes. Con el tiempo llegó a convertirse en una herramienta muy usada por los intrusos.

Actualmente existen sniffers para capturar cualquier tipo de información específica. Por ejemplo contraseñas de un recurso compartido o de acceso a una cuenta, que viajen sin encriptar al ingresar en sistemas de acceso remoto. El análisis de tráfico puede ser utilizado también para determinar relaciones entre organizaciones e individuos. Para realizar estas funciones se analizan las tramas de un segmento de red y se presentan al usuario sólo las que interesan mediante filtros.

Normalmente los buenos sniffers no se pueden detectar, aunque la inmensa mayoría, y debido a que están demasiado relacionados con el protocolo TCP/IP, si pueden ser detectados con algunos métodos. Algunos sniffers conocidos son:

- Wireshark
- tcpdump
- DSniff
- Darkstat
- Ettercap
- Cain & Abel
- WinDump
- Airodump-ng

Contramedida: *Cifrado de conexiones.*

2.1.2. Ataques a la autenticidad

→ **Objetivo:** Engañar a un sistema víctima haciéndose pasar por alguien de confianza o legítimo.

Veamos algunas de sus formas de ataque y herramientas.

Ingeniería social

El término ingeniería social es equivalente a engañar o mentir para conseguir que otra persona haga cosas que la parte que ataca quiere que haga. Es difícil de parar, al menos técnicamente, pues no usa métodos informáticos. Va a uno de los eslabones más débiles de la cadena de la seguridad: el factor humano.

Un ejemplo de ataque de ingeniería social sería realizar una simple llamada telefónica en la que el atacante se hace pasar por algún técnico de la compañía proveedora de Internet o por cualquier otro individuo de forma ilegítima, y solicita al usuario sus credenciales de acceso a algún sistema, o cualquier otro tipo de información sensible.

***Contramedidas**: Uso de firma digital, formación en seguridad, etc.*

SPOOFING. Suplantación de identidad

El término **spoofing** puede traducirse como "hacerse pasar por otro". El objetivo de esta técnica es actuar en nombre de otros usuarios para que, una vez conseguido el engaño, realizar otro tipo de acciones maliciosas. Una forma común de spoofing es conseguir el nombre y password de un usuario legítimo para, una vez ingresado al sistema, realizar acciones en nombre de él.

Muchos ataques de este tipo comienzan con técnicas de ingeniería social aprovechándose de algunos usuarios que, por falta de cultura u otro motivo, facilitan a extraños sus identificaciones dentro del sistema.

Algunos ataques de spoofing mas conocidos son el IP Spoofing, el DNS Spoofing, el phishing (Mail Spoofing o Web Spoofing) y el ARP Spoofing.

- **IP Spoofing:** Se busca usurpar la dirección IP de una máquina que permita al atacante ocultar el origen de su ataque o para beneficiarse de una relación de confianza entre dos máquinas.

  El atacante crea sus propios paquetes IP (con una herramienta de manipulación de paquetes, como **Scapy**), les cambia la dirección IP origen (campo **from**) y éstos son aceptados por el destinatario del paquete. Su utilización más común es enviar los paquetes con la dirección origen de un tercero, de forma que la víctima lo que podría ver es un ataque que procede de esa dirección y no la dirección real del intruso.

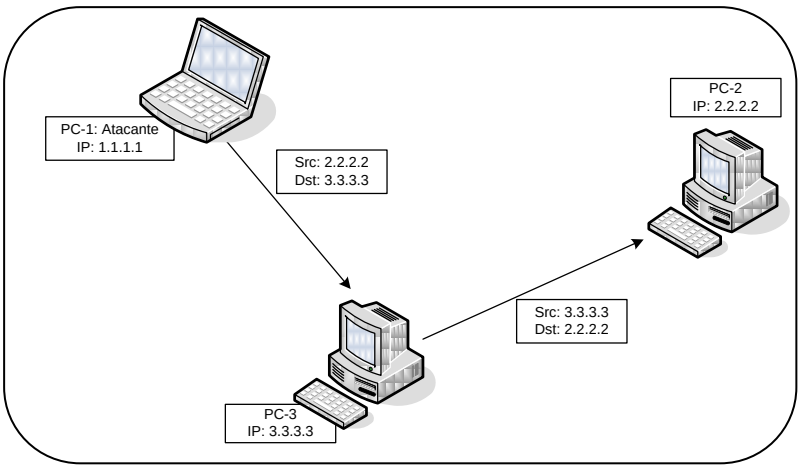

Fig. 2.3: IP Spoofing

En el ejemplo, nótese que si la victima (PC-3) descubre el ataque verá al PC-2 como su atacante y no el verdadero origen (PC-1). Este ataque se hizo famoso al usarlo Kevin Mitnick.

- **DNS Spoofing:** El protocolo DNS (Sistema de Nombres de Dominio) convierte un nombre de dominio (www.dte.us.es) en su dirección IP (ej: 150.214.141.196) y viceversa. Este ataque se consigue mediante la manipulación de paquetes UDP para comprometer el servidor DNS.

  Para realizar el ataque se usan respuestas falsas a las peticiones DNS enviadas por una potencial victima hacia el servidor. El nombre DNS se asocia con la IP del atacante de forma que el intruso puede presentarse como un servidor DNS resolviendo nombres con su IP (**DNS ID Spoofing**) ó modificando la caché de resolución local de nombres (**envenenamiento del caché DNS** ó **DNS Poisoning**).

  En la siguiente figura se muestra como funciona un ataque DNS Poisoning.

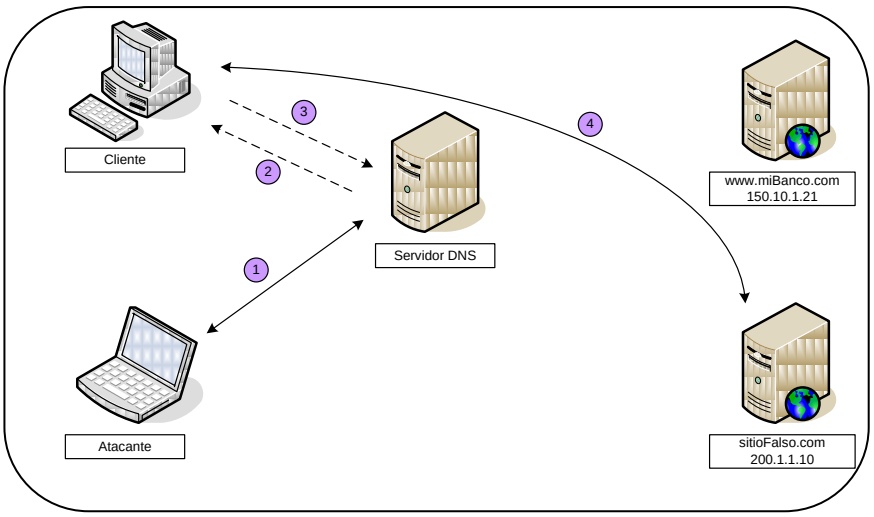

Fig. 2.4: DNS Spoofing

1. El atacante toma el control del servidor DNS usado por el cliente y añade ó altera las entradas para una URL (ej: www.miBanco.com), cambiando la IP almacenada (150.10.1.21) por la de un sitio falso (200.1.1.10).
2. El cliente realiza una petición al servidor DNS, por ejemplo: "¿Cuál es la IP de www.miBanco.com?".
3. El servidor DNS responde al cliente con: "La IP de www.mibanco.com es 200.1.1.10", es decir, la IP de un sitio falso.
4. El cliente conecta con el sitio falsificado del atacante

- **Mail spoofing**: El envío de falsos e-mails es una forma de spoofing conocida como **phishing**. Aquí el atacante envía e-mails a nombre de otra persona o entidad con el fin de engañar a la víctima que recibe el mensaje y que ésta le proporcione cierta información.

- **Web Spoofing:** En el caso del Web Spoofing (también llamado phishing) el atacante crea un sitio web completo, falso y similar al que la víctima desea entrar. Los accesos a este sitio están dirigidos por el atacante, permitiéndole monitorizar todas las acciones de la víctima durante su navegación por el mismo, desde sus datos hasta las contraseñas, números de tarjeta de créditos, etc. El atacante también es libre de modificar cualquier dato que se esté transmitiendo entre el servidor original y la víctima o viceversa.

- **ARP Spoofing**: Se redirecciona el trafico de una, o varias máquinas de la red hacia la del atacante. El atacante falsifica los paquetes ARP proporcionando su MAC en relación a la IP suplantada. Los equipos que reciban dichos paquetes ARP falsificados modificarán su tabla ARP apuntando al atacante.

Por ejemplo, mediante este ataque podríamos suplantar la dirección MAC del gateway de salida a Internet de una red. De esta forma los equipos conectados a esta red asociarían en sus tablas ARP la IP del gateway a nuestra MAC y todo su tráfico hacia Internet sería enviado hacia nosotros. Para evitar ser descubiertos, una vez capturado el tráfico de los usuarios deberíamos reenviarlo hacia el gateway real para que todo ocurra de forma transparente.

Existe una herramienta muy conocida llamada **Social Engineer Toolkit (SET)** dedicada al estudio de este tipo de ataques en los que se persigue engañar a una víctima.

HIJACKING. Secuestro de sesiones

Se produce cuando un atacante consigue interceptar una sesión ya establecida. El atacante roba una conexión después de que el usuario haya superado con éxito el proceso de identificación ante el sistema. El único método seguro para protegerse contra este tipo de ataques es el uso de encriptación.

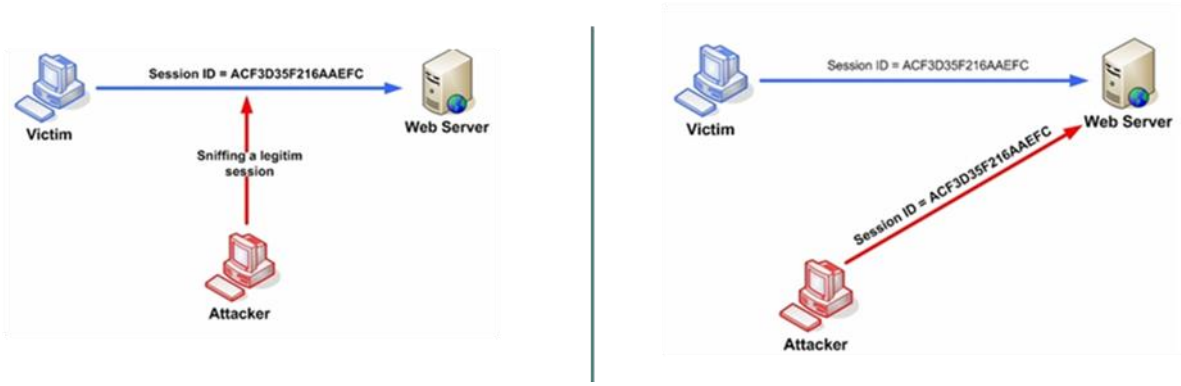

Fig. 2.5: Hijacking

BACKDOORS

Las puertas traseras o backdoor son trozos de código dentro de un programa que permiten, a quien las conoce, saltarse los métodos usuales de autentificación para realizar ciertas tareas. Habitualmente son insertados por los programadores del sistema para agilizar la tarea de probar código durante la fase de desarrollo.

Esta situación se convierte en una falla de seguridad si se mantiene, involuntaria o intencionalmente, una vez terminado el producto ya que cualquiera que conozca la existencia del agujero o lo encuentre en su código podrá saltarse los mecanismos de control.

Existen algunos malwares llamados comúnmente backdoors debido a que tras infectar a un sistema abren una puerta trasera de forma intencionada, dejando vía libre a los atacantes.

FUERZA BRUTA

Este método busca la obtención de aquellas claves que permiten ingresar a los sistemas, aplicaciones, cuentas, etc, probando todas las posibilidades. Muchos passwords de acceso son obtenidos fácilmente porque involucran el nombre u otro dato familiar del usuario. Además, ésta nunca (o rara vez) se cambia. En este caso, el ataque se simplifica e involucra algún tiempo de prueba y error. Otras veces se realizan ataques sistemáticos (incluso con varias computadoras a la vez) con la ayuda de programas especiales (ej: Medusa, Hydra) que prueban millones de posibles claves, con la ayuda de un diccionario, hasta encontrar la password correcta.

Los **diccionarios** son archivos con millones de palabras. Estos archivos son utilizados para descubrir passwords en pruebas de fuerza bruta. Actualmente es posible encontrar diccionarios de gran tamaño orientados, incluso, a un área específica de acuerdo al tipo de organización que se este

atacando. Incluso para casos en los que las claves están encriptadas, es posible encontrar diccionarios que en vez de contener posibles claves lo que contienen son los "hash" para claves encriptadas con un cierto algoritmo de encriptación (ej: MD5). Si tenemos la cadena encriptada y el hash que se ha usado para encriptarla, podemos obtener la cadena original sin encriptar. Este tipo de diccionarios se conocen como **tablas rainbow.**

2.1.3. Ataques a la disponibilidad. DoS

→ **Objetivo:** Inhabilitar el acceso al sistema víctima o a los servicios ofrecidos por éste.

La realidad indica que es más fácil desorganizar el funcionamiento de un sistema que acceder al mismo. Los ataques de denegación de servicio (**DoS – Denial of Service**) buscan impedir que los usuarios puedan hacer uso de los servicios atacados. Dentro de esta categoría existen muchas variantes de ataques que son bastantes destructivos.

Distinguiremos dos tipos de denegación de servicio. Por un lado aquellos que explotan un fallo en una aplicación y por otro lado los que se basan en la mala implementación o en la debilidad de un protocolo.

A) Denegación de servició de aplicación

Si las vulnerabilidades de una aplicación pueden llevar a la posibilidad de tomar el control de una máquina (ej. Buffer Overflow), pueden también llevar a la denegación de servicio. La aplicación se volverá innaccesible por bloqueo de los recursos de la máquina o por un bloqueo completo de ésta.

B) Denegación de servicio de red

Buscan impedir que los usuarios de cierta red puedan hacer uso de los servicios que en ella se brindan. Existen distintos tipos de denegación de servicio en función de las características de la pila TCP/IP que se deseen explotar.

FLOODING – JAMMING

Este tipo de ataque busca saturar los recursos del sistema. Por ejemplo, un ataque de este tipo puede consistir en consumir toda la memoria o espacio en disco disponible, o enviar tanto tráfico a la red para saturarla y conseguir así que nadie más pueda utilizarla.

El atacante puede saturar el sistema con mensajes en los que se requiere que se establezca conexión. Además, para ocultarse como originante del ataque, en vez de proveer la dirección IP del emisor se pueden crear mensajes que contengan falsas direcciones IP en el campo origen (usando Spoofing). El sistema atacado intentará responder al mensaje, pero al no recibir respuesta (ya que está enviando paquetes SYN-ACK a una máquina que no le ha enviado previamente ningún paquete SYN) acumulará la información de las conexiones abiertas en sus buffers, no dejando lugar a las conexiones legítimas.

Muchos hosts e ISPs (proveedores de Internet) han sufrido bajas temporales del servicio por ataques que explotan el protocolo TCP como el "ping de la muerte" (una antigua versión del comando ping para ataques). Mientras que el ping normal simplemente verifica si un sistema está enlazado a la red, el ping de la muerte causa el bloqueo instantáneo del equipo. Esta vulnerabilidad ha sido ampliamente utilizada en el pasado pero aún hoy pueden encontrarse sistemas vulnerables.

Otra acción común es enviar millones de e-mails sin sentido a todos los usuarios posibles de forma continuada, saturando los sistemas de correo destinos.

Dentro de esta modalidad se pueden diferenciar muchísimas variantes como el SYN Flooding, UDP Flooding, Connection Flooding o el Net Flooding. Veamos algunos.

- **SYN Flooding**: Para entender este tipo de ataques debemos recordar el mecanismo de conexión TCP en tres etapas. En éste, por cada petición de conexión entre un cliente y un servidor, el servidor crea una tabla donde lleva la cuenta de sesiones semi-establecidas de TCP. Sólo cuando le llega al servidor un mensaje de aceptación del cliente (paso 3 la conexión TCP), se borra la entrada, en caso contrario la conexión permanece en un estado semi-abierto. SYN Flooding explota este mecanismo.

  La idea es dejar en la máquina objetivo un número elevado de conexiones TCP en espera. Para hacer esto se envían una gran cantidad de peticiones de conexiones (paquetes SYN) a la víctima y ésta enviará un SYN-ACK de vuelta para responder al SYN recibido. El atacante no le responderá con el ACK que la víctima espera puesto que previamente habrá falseado el origen de los paquetes enviados y la víctima estará enviando los SYN-ACK a otra dirección. Así, para cada SYN recibido, la máquina objetivo tendrá una conexión abierta. Estas conexiones medio abiertas usan recursos de memoria. Después de un tiempo acumulando conexiones semi-abiertas la máquina se saturará y no podrá aceptar más conexiones. Este tipo de denegación de servicio sólo afecta a la máquina objetivo.

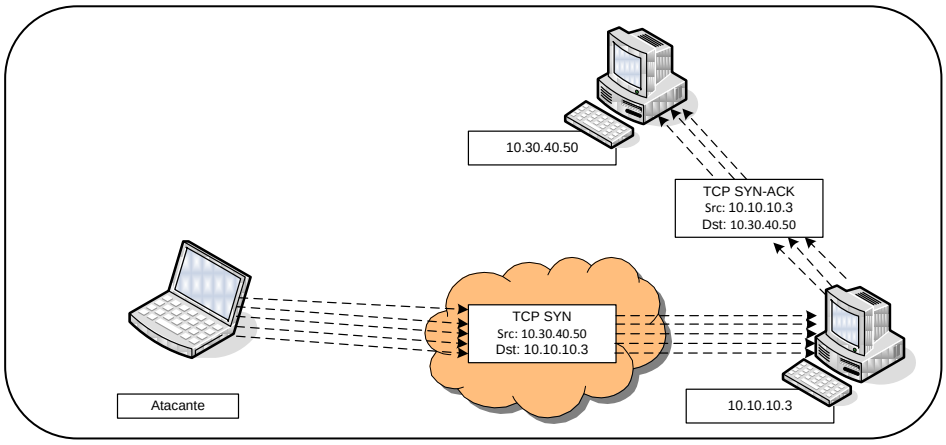

Fig. 2.6: SYN-Flood

  El problema es que muchos sistemas operativos tienen un límite muy bajo en el número de conexiones "semi-abiertas" que pueden manejar en un momento determinado. Si se supera ese límite, el servidor sencillamente dejará de responder a las nuevas peticiones de conexión que le vayan llegando. Las conexiones "semi-abiertas" van caducando tras un tiempo, liberando "huecos" en la tabla para nuevas conexiones. Sin embargo, mientras el atacante mantenga el Syn Flood, la probabilidad de que una conexión recién liberada sea capturada por un nuevo SYN malicioso es muy alta.

  La potencia de este ataque reside en que muchos sistemas operativos fijan un límite del orden de 5 a 30 conexiones "semi-abiertas", y que éstas caducan al cabo de un par de minutos. Para mantener el servidor fuera de servicio, un atacante sólo necesita enviar un paquete SYN cada 4 segundos. Como hemos visto, este ataque suele combinarse también con el IP Spoofing para ocultar el origen del ataque (como se muestra en la figura).

  Para generar los paquetes SYN el atacante usa un **inundador de SYNs** (como el **synk4, neptune,** o creando un simple script), indicando el puerto TCP destino y utilizando una IP de origen aleatoria para evitar que la máquina del atacante sea identificada.

  Este ataque ha sido la principal arma usada por grupos hacktivistas como Anonymous.

- **UDP Flooding**: Esta denegación de servicio explota la forma de trabajo que tiene el protocolo UDP (sin conexión). La idea es crear una "tormenta" de paquetes UDP y lanzarla sobre una, dos o varias máquinas. Esto provocará la congestión de la red sobre la que estas máquinas trabajan, así como la saturación de los recursos. La base de este ataque está en que el tráfico UDP tiene prioridad sobre el TCP.

El protocolo TCP tiene un mecanismo para controlar la congestión en el caso de que se conozca que un paquete llega después de un "largo tiempo". Este mecanismo adapta la frecuencia con la que se envían los paquetes TCP. El protocolo UDP no posee este mecanismo. Después de un tiempo enviando paquetes UDP se utilizará todo el ancho de banda dejando una pequeñísima parte al tráfico TCP.

El caso más conocido de inundación UDP es el "***Chargen Denial of Service Attack***". La implementación de este ataque es bastante sencilla. Se basa en establecer una comunicación con el servicio CHARGEN de una máquina y el servicio ECHO de otra. El servicio CHARGEN (como su nombre indica) genera caracteres mientras que el servicio ECHO reenvía los datos que recibe. El atacante envía paquetes UDP al puerto 19 (chargen) de una de las víctimas proporcionando la dirección IP y el puerto de origen de otra. En este caso, el puerto origen UDP 7 (echo). La inundación UDP lleva la saturación del ancho de banda entre ambas máquinas. Con este ataque se puede conseguir saturar una red al completo.

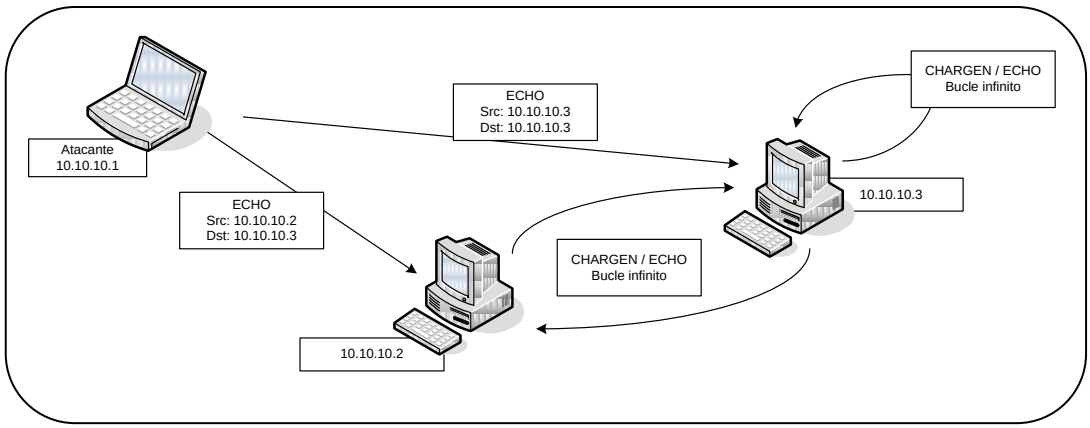

Fig. 2.7: UDP-Flood

Otro caso más simple de UDP-Flooding es el FRAGGLE que sólo recae en el servicio ECHO.

FRAGMENTACIÓN DE PAQUETES

Se trata de un tipo de ataque bastante antiguo, lo cual implica que los firewalls actuales son capaces de resolverlos. Estos ataques recaen sobre los mecanismos de protección de filtrado IP. Este ataque aprovecha una vulnerabilidad de la pila TCP/IP en lo concerniente a la defragmentación IP (reensamblado de fragmentos IP). Existen dos métodos para llevar a cabo este ataque:

a. **Pequeños fragmentos**: Cuando se recibe un paquete fragmentado, los filtros IP aplican la misma regla de filtrado a todos los fragmentos del paquete. Un paquete malicioso podría contener en su primer fragmento solamente los 8 primeros bytes de la cabecera TCP (origen, destino y número de secuencia) lo cual podría denotar que es un paquete como otro cualquiera y se le dejaría pasar a él y al resto de fragmentos. Una vez reconstruido en el destino, éste pasa a la capa TCP con todo los fragmentos sin ningún tipo de control.

b. **Superposición de fragmentos**: En TCP/IP, si dos fragmentos IP se superponen, el segundo sobrescribe al primero. Aprovechando esta cualidad, el ataque se basa en enviar un paquete dividido en dos fragmentos. Un primer fragmento que no solicita ninguna conexión y un segundo que si solicita una conexión. El filtro IP dejará pasar el primer fragmento pues no ve conexión alguna, pero puesto que el segundo se considera Offset del primero también lo dejará pasar. Así, cuando se reconstruya el paquete, el segundo sobrescribirá al primero consiguiendo una conexión válida en la máquina destino.

   Un ataque conocido que usa este método es el TearDrop (I y II). En el tiempo de

defragmentación, algunos sistemas no gestionan esta excepción y ésto les lleva a una denegación de servicio. Hay variantes de este ataque: **bonk, boink y newtear** por ejemplo. La denegación de servicio "ping de la muerte" explota la mala administración de la defragmentación ICMP, enviando más datos que el tamaño máximo de un paquete IP. Estos tipos diferentes de denegación de servicio consiguen "colgar" la máquina objetivo.

SMURFING

Se basa en explotar el protocolo ICMP. Cuando se envía un ping (mensaje ICMP ECHO) a una dirección de broadcast (por ejemplo 10.255.255.255), se envía un ping a cada máquina de la red. El principio del ataque es inundar de paquetes ICMP ECHO REQUEST enviados usando como IP de origen la máquina del objetivo. El atacante envía un flujo continuo de ping´s a la dirección broadcast de la red y todas las máquinas responderán con un mensaje ICMP ECHO REPLY a la víctima. El flujo se multiplica por número de host en la red. En este caso, toda la red en la que se encuentra el objetivo se verá afectada por la denegación de servicio, ya que el gran tráfico que se genera con este ataque producirá una congestión en la red.

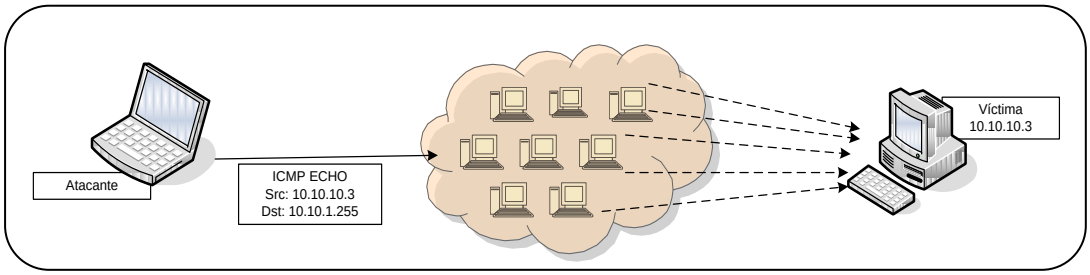

Fig. 2.8: SMURF

E-Mail BOMBING – SPAMMING

El E-Mail Bombing consiste en enviar muchas veces un mensaje idéntico a una misma dirección, saturando así el servidor de correo entrante (MDA) del destinatario. El Spamming, en cambio se refiere a enviar el e-mail miles de usuarios, hayan estos solicitados el mensaje o no (normalmente es que no). Es muy utilizado por empresas para publicitar sus productos. El Spamming esta siendo actualmente tratado por las leyes europeas como una violación de los derechos de privacidad del usuario.

DENEGACIÓN DE SERVICIO DISTRIBUIDA (DDoS)

Podemos definir los ataques de denegación de servicio distribuido (DDoS) como un ataque de denegación de servicio (DoS) dónde existen múltiples focos, distribuidos y sincronizados, que dirigen su ataque hacia un mismo destino.

El proceso es el siguiente:

- El atacante busca una serie de sistemas vulnerables.
- Se realiza un ataque sobre esos nodos y se les instala algún software malicioso (un troyano) que le permita al atacante controlarlos. Estos nodos son los "masters" (o botmasters). Los masters tienen conexión directa con el atacante y son los que llevarán a cabo la primera fase del ataque.
- A través del software instalado, estos nodos buscan un segundo nivel de nodos (slaves, daemons o bots) que serán los encargados de realizar el ataque final.
- El atacante, cuando considere oportuno, dará la orden de atacar de manera sincronizada para que todos los nodos masters ordenen a sus esclavos que ataquen al sistema víctima.

Estas redes de ordenadores "zombies" al servicio del atacante se conocen como **botnets** (redes de bots).

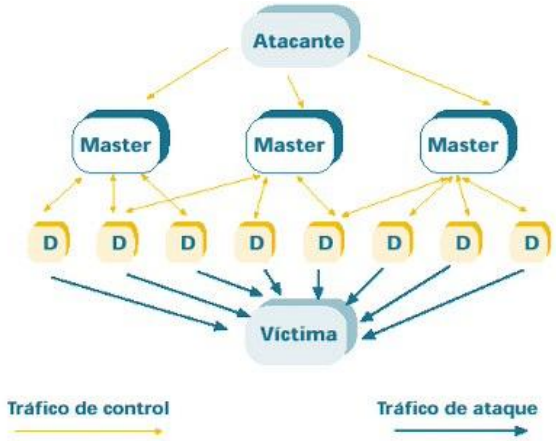

Fig. 2.9: DDOS. Jerarquía de nodos atacantes

Las **botnets** constituyen una de las principales amenazas para la seguridad de los sistemas informáticos de todo el mundo. Algunas de las más conocidas son ZeuS, SpyEye o Rustock.

2.1.4. Ataques a la integridad

→ **Objetivo:** Modificar o destruir información y/o aplicaciones.

Constituyen el siguiente paso tras un ataque en el que conseguimos acceso al sistema víctima. En esta categoría nos referimos a ataques que causan la modificación desautorizada de datos o software instalado en el sistema víctima (incluyendo el borrado de archivos). Son particularmente serios cuando el atacante ha obtenido privilegios de administrador (a través de otros métodos), con capacidad para disparar cualquier comando, y por ende, alterar o borrar cualquier información. Puede incluso provocar la baja total del sistema (DoS).

BORRADO DE HUELLAS

El borrado de huellas es una de las tareas mas importantes que debe realizar el intruso después de ingresar en un sistema, ya que si se detecta su ingreso, el administrador buscará como conseguir "tapar el hueco" de seguridad, evitar ataques futuros e incluso rastrear al atacante. Las "huellas" son todas aquellas tareas que realizó el intruso en el sistema, que por lo general, son almacenadas en algún log por el sistema operativo. Los archivos logs son una de las principales herramientas con las que cuenta un administrador para conocer los detalles de las tareas realizadas en el sistema y la detección de intrusos. Los logs son el principal enemigo del atacante.

ATAQUES A APLICACIONES

→ **Objetivo:** Aprovechar fallos en el diseño y/o en librerías relacionadas con una aplicación.

Existen un gran número de tipos de ataques para provocar modificaciones o daños que comprometan nuestro sistema mediante el aprovechamiento de vulnerabilidades en aplicaciones como Java Applets, JavaScript, Active-X, etc. Quizás este grupo sea el más extenso en cuanto a variedad de ataques debido a la gran cantidad de servicios y aplicaciones que usamos cuando estamos conectados a la red. Por este motivo no vamos a entrar en detalle con esta categoría. Lo que si mostraremos a continuación serán los principales factores y fallas que los atacantes suelen aprovechar para llevar a cabo los ataques a las aplicaciones.

1. **El problema de la configuración**

Uno de los problemas clásicos de seguridad encontrados en las aplicaciones procede de malas configuraciones o por culpa de las configuraciones por defecto. El software, por ejemplo de servidores webs, con la instalación por defecto instalan ficheros que con frecuencia proporcionan sites de ejemplo que pueden ser usados por atacantes para

acceder a información confidencial.

2. Bugs

La mala programación del software da lugar a los bugs. Los bugs no son más que defectos o fallos del software cometidos durante alguna de las etapas de su desarrollo. Éstas son vulnerabilidades que, si un atacante las descubre y con la ayuda de algún exploit, pueden permitirle ejecutar comandos no autorizados, obtener el código fuente de páginas dinámicas, hacer que un servicio no esté operativo, tomar el control de la máquina, etc.

3. Buffer overflow

Este ataque intenta aprovechar la falta de controles existente en ciertos programas a la hora de pasarles argumentos. Tiene lugar cuando ciertas variables pasadas como argumentos a una función se copian en un buffer en el que no se realiza comprobación de tamaño.

Un atacante puede colocar, o enviar por la red, una secuencia de bytes mayor que el espacio de memoria reservado para esa variable. Esto provocará una sobrescritura en las siguientes posiciones de memoria. La siguiente figura muestra esto.

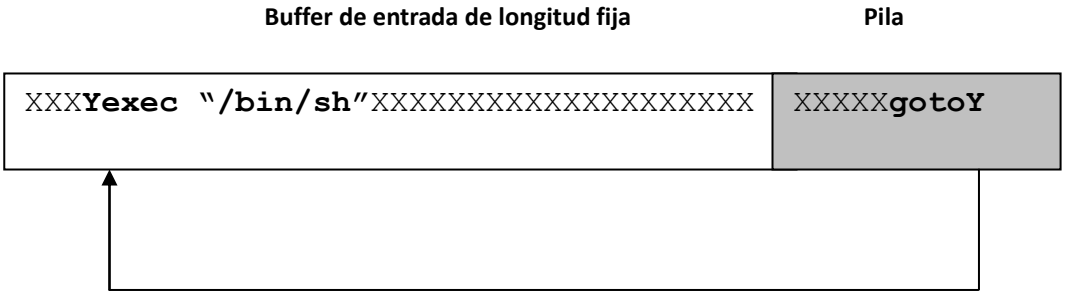

Fig. 2.9: Ataque por buffer overflow

En tales posiciones, a menudo, están los valores de punteros a pila, que indican por qué parte del programa hay que continuar la ejecución. Adecuando los bytes de información de entrada, el atacante puede redirigir al programa para que ejecute instrucciones embebidas en los datos de entrada. En la figura se muestra como se redirige el salto del puntero de pila a la instrucción "exec /bin/shell" que proporcionará una consola de comandos al atacante.

Un atacante puede tener el control de la máquina y controlarla remotamente si se hace con los permisos del programa atacado.

Este ataque es bastante antiguo pero, aun así, hoy día sigue siendo uno de los más utilizados y peligrosos.

EXPLOITS

Como última parada de esta sección, es imprescindible que citemos a los famosos exploits.

Es muy frecuente ingresar a un sistema explotando agujeros en los algoritmos de encriptación utilizados, en la administración de las claves por parte la empresa, o simplemente encontrando un error en los programas utilizados.

Los programas para explotar estos "agujeros" reciben el nombre de exploits. Los exploits aprovechan la debilidad, fallo o error hallado en el sistema (hardware o software) para ingresar al mismo. Normalmente los exploits son pequeños scripts programados en cualquier tipo de lenguaje como C, Perl, Python o incluso ensamblador.

Todos los días se publican nuevos exploits (http://www.exploit-db.com). En cuanto una vulnerabilidad es publicada, inmediatamente es publicado un exploit que sirve para

aprovecharse de ella. Por tanto, mantenerse informado de los mismos y de las herramientas para combatirlos es de vital importancia.

2.2. Atacantes

Hacker

Centrándonos en el mundo de los sistemas de información, un hacker es una persona con amplios conocimientos técnicos, bien puede ser informática, electrónica o comunicaciones, que se mantiene permanentemente actualizado y conoce a fondo temas complejos. Se trata de un investigador nato que se inclina ante todo por conocer, y en al ámbito de la seguridad, por estudiar las posibilidades de acceder a cualquier tipo de "información segura".

Un hacker suele difundir sus conocimientos para que personas interesadas se enteren de cómo funciona realmente la tecnología, dando a conocer las debilidades de sus propios sistemas de información.

Usualmente este grupo está conformado tanto por adolescentes como por adultos, en su mayoría estudiantes o profesionales de la informática, con una característica común: las ansias de conocimiento.

Características

1. Amplios conocimientos
2. Se mantiene actualizado
3. Investigador
4. Con experiencia
5. Deseos de conocer siempre mas
6. Discreto

Cracker

Un cracker, al igual que un hacker, es un hábil conocedor de la tecnología, software, hardware o de cualquier otro tipo, pero a diferencia de éste, es movido por otros objetivos. EL cracker busca violar la seguridad de un sistema informático buscando el beneficio personal o para hacer daño.

El término cracker procede de "criminal hacker", y fue creado para ser diferenciado del hacker. Se denomina así a aquella persona con comportamientos compulsivos que alardea de su capacidad para burlar sistemas electrónicos e informáticos.

Características

1. Busca el bien personal o el daño ajeno
2. Compulsivo
3. Afán de reconocimiento de sus capacidades
4. Elevado conocimiento técnico

Phreakers

Se caracterizan por poseer amplios conocimientos en el área de telefonía terrestre y móvil, incluso más que los propios técnicos de las compañías telefónicas. Su objetivo se centra en romper la seguridad de las centrales telefónicas y superar retos intelectuales, de complejidad creciente, relacionados con incidencias de seguridad o fallas en los sistemas telefónicos, que les permitan obtener privilegios no accesibles de forma legal.

Los phreakers son la vieja escuela del hacking. Algunos phreakers conocidos son Steve Wozniak y Steve Jobs (fundadores de Apple), el Capitán Crunch ("la caja azul") y Kevin Mitnick.

Características

1. Amplio conocimiento en telecomunicaciones
2. Buscan violar la seguridad de los sistemas de comunicaciones
3. Precursores del movimiento hacking

Lammer

A este grupo pertenecen aquellas personas deseosas de alcanzar el nivel de un hacker, pero su poca formación y sus escasos conocimientos le impiden alcanzar su sueño. Su trabajo se reduce a ejecutar programas creados por otros, a bajar de forma indiscriminada cualquier tipo de script publicado en la red, etc.

Éste tipo de sujetos es muy numeroso en la red. Sus mas frecuentes ataques se caracterizan por bombardear el correo electrónico (spam), emplear de forma habitual programas sniffers para monitorizar la red, interceptar contraseñas, enviar mensajes con direcciones falsas amenazando sistemas (lo cual en muchas ocasiones no suele ser cierto). Emplean en ocasiones "backdoors" (puertas traseras) o virus con el fin de molestar y sin dimensionar las consecuencias de sus actos.

Características

1. Busca la satisfacción personal o el daño ajeno
2. Desea alcanzar el nivel del hacker
3. Ejecuta programas creados por otros
4. Pocos conocimientos

Bucaneros / Piratas

Son los comerciantes de la red. Suelen poseer un amplio conocimiento de negocios. Su objetivo está centrado en comercializar productos piratas bajo el ánimo de lucrarse en corto tiempo y con el mínimo esfuerzo.

Características

1. Comerciantes
2. Conocimientos de negocios

Newbie

Es el típico aprendiz de hacker.

Características

1. Novatos
2. Cacharrean en la red
3. "Inofensivos"

Script Kiddie

También llamados "Clickers" ó "Wannabes", son usuarios de Internet sin conocimientos sobre hacking o cracking, aunque aficionados a estos temas. No los comprenden realmente, simplemente son internautas que se limitan a recopilar información de la red y buscar programas que luego ejecutan sin mas llegando a veces incluso a infectarse a si mismos.

Características

1. Simples usuarios sin conocimientos
2. Usan programas de hacking o cracking sin comprenderlos

2.3. Malware

Se trata de programas que se introducen en nuestros sistemas de formas muy diversas con el fin de producir efectos no deseados y nocivos. Una vez que un programa de éstas características se haya introducido en un ordenador, se colocará en lugares donde el usuario pueda ejecutarlos de manera no intencionada.

Existen malwares para todos los gustos y colores. Podemos citar algunos tipos como los troyanos, keyloggers, spywares, adwares, rootkits, gusanos, downloaders, etc.

Características:

1. Pueden tener numerosas formas
2. Producen efectos diferentes según su tipo de intención

La mejor forma de defensa contra los malwares es no ejecutar nada desconocido y por supuesto contar con un buen antivirus actualizado.

2.4. Métodos de defensas

Una vez conocidos los peligros a los que nos enfrentamos, necesitamos medios para protegernos contra ellos.

Por un lado, limitando el tráfico entre nuestra red y las redes externas permitiendo solo aquel que se considere seguro, o al menos que esté justificado, evitaremos un gran número de ataques posibles. Para conseguir este fin contamos con el filtrado de paquetes (firewalls) y los servidores proxy.

Por otro, si decidimos permitir el acceso a nuestras máquinas desde el exterior tendremos que asegurarnos de que los intentos de conexión proceden de quienes dicen proceder. No podemos fiarnos de los passwords débiles puesto que por medio de un ataque de fuerza bruta podrían obtenerse. Los métodos de autenticación serán los encargados de solucionar este problema.

Por último, si creemos que nuestra red puede ser objeto de un ataque hijacking o sniffing necesitamos alguna técnica para impedirlos. En este caso necesitaremos encriptar la conexión.

Así pues, podemos diferenciar tres frentes de defensa:

1. **Protección perimetral**: Comprende las estrategias para proteger los recursos de conectada a una red.
    a. Firewalls (Filtrado de paquetes, proxys…)
    b. Sistemas de detección de intrusiones (IDS, IPS, Honeypots,…)
    c. VPN´s

2. **Autenticación**: Es el proceso seguido por una entidad para probar su identidad ante otra. Distinguimos dos tipos de autenticación por tanto, la del usuario a una máquina durante la secuencia del login inicial, y de máquina a máquina durante la operación.

    Los passwords tradicionales son demasiado débiles para usarlos sobre una red, y por tanto se usan passwords no reusables. Estos cambian cada vez que se usan, y por tanto no son sensibles al sniffing. El método de autenticación por dirección IP del host (o bien su nombre DNS) es susceptible de ser atacado mediante spoofing con relativa facilidad y por tanto se usan técnicas de criptografía, contando con un "Centro de Distribución de Claves" (KDC) para la distribución y verificación de las mismas. El KDC mas conocido es KERBEROS.

3. **Criptografía**: Mediante el uso de la criptografía se intenta proteger la información a base de codificarla.

Analizar todos los aspectos de la red que necesiten protección y conseguir que cada zona esté protegida por sus respectivas medidas de seguridad es el objetivo que perseguimos. Para que una red sea segura, todos sus elementos deben serlo. Si un atacante consigue eludir tales defensas, en poco tiempo podrá causar pérdidas astronómicas (sin hablar de los atacantes internos).

Dentro del alcance de esta asignatura únicamente se tratará el primer frente de defensa, es decir, nos centraremos en estudiar los métodos de **seguridad perimetral**.

2.5. ANEXO: Protocolos de interés

Familia IP

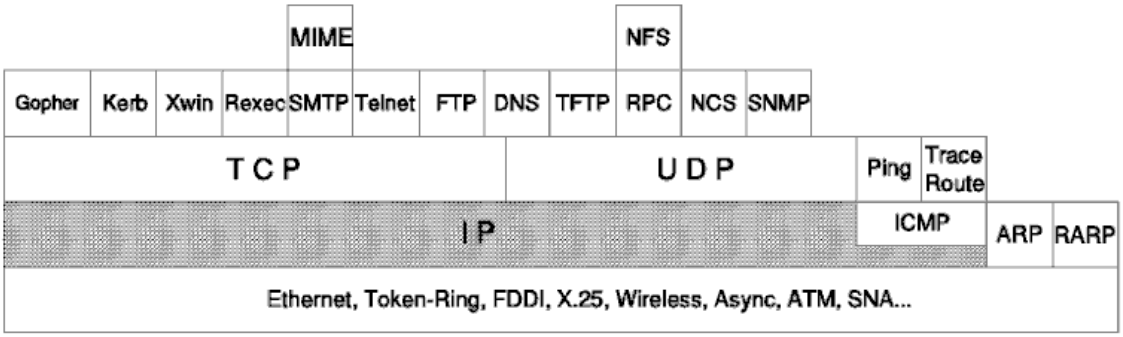

Fig. 2.10: Protocolos de la familia IP

Cabecera IP

Los mensajes IP pueden ser de 3 tipos:
- Generados por una aplicación que tenga transporte TCP
- Generados por una aplicación que tenga transportes UDP
- Generados por protocolos del propio nivel de red como ICMP ó IGRP.

Todos tienen algo en común, cuentan con una cabecera IP:

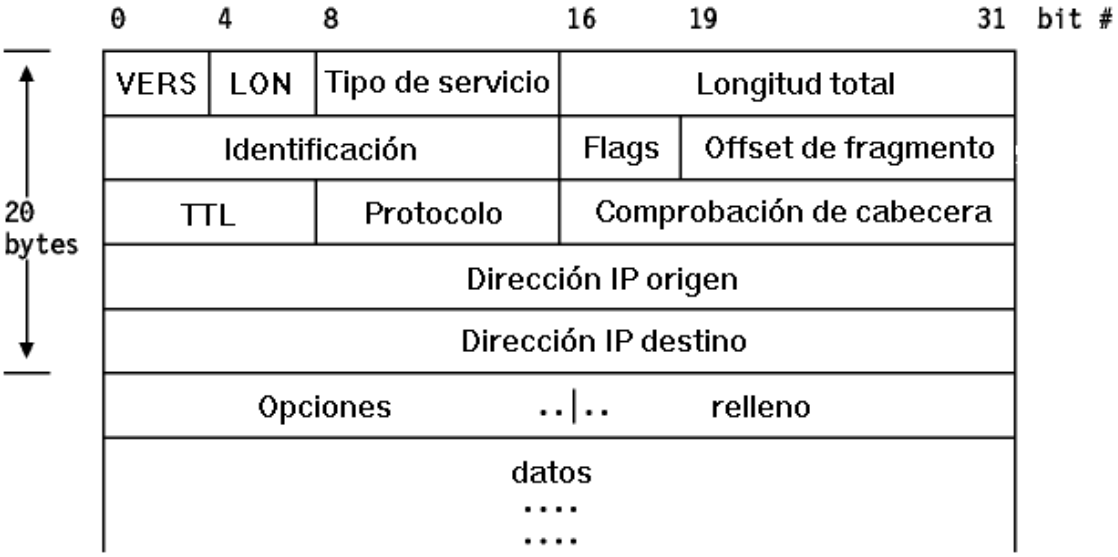

Fig. 2.11: Datagrama IP

Sería interesante familiarizarnos con esta figura (o incluso fotografiarla mentalmente) pues es primordial para entender muchos ataques y técnicas de defensa.

De todos los campos, los que más nos interesan son:
- Dirección origen.
- Dirección destino.
- El número de protocolo, que indica de qué protocolo es la siguiente cabecera. Ej: 1 para ICMP, ó 6 para TCP.
- Información de fragmentación del mensaje. Si está fragmentado o no, y, si lo está, qué número de fragmento es.

Cabecera TCP

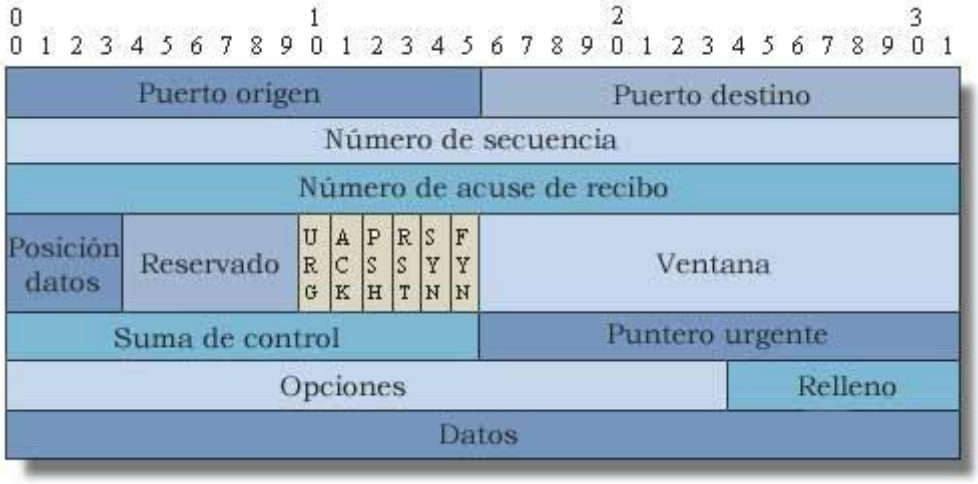

Fig. 2.12: Cabecera TCP

Source Port: Número de puerto origen (16 bits).
Destination Port: Número de puerto destino (16 bits).
Sequence Number: Número de secuencia (de paquetes fragmentados).
Acknowledge Number: ACK=1 contiene el número de secuencia del siguiente dato que el transmisor espera recibir.
Data Offset (HLEN): Longitud de la cabecera.

1. Expresado en palabras de 32 bits.
2. Indica el comienzo de los datos.
3. Múltiplo de 32.

Reserved: Campo reservado para usos futuros
Flags:

4. **URG:** Indica datos urgentes.
5. **ACK:** Valida el campo *Acknowledgment*.
6. **PSH:** Fuerza el envío de buffer del emisor.
    - PSH=1 : Se manda inmediatamente la información del buffer en uno o mas segmentos. El receptor los toma y los procesa de inmediato.
    - PSH=0 : El envío y recepción se hacen según la conveniencia del transmisor y el receptor.
7. **RST:** Conexión abortada.
8. **SYN:** Sincroniza números de secuencia.
    - SYN=0 : Número de secuencia del primer octeto de datos en cada segmento.
    - SYN=1 : Número de secuencia inicial (ISN). Primer octeto de datos es ISN +1.
    - Se usa para reordenar segmentos que llegan desordenados.
9. **FIN:** Indica fin de la conexión.

Checksum: Suma de comprobaciones.
Urgent Pointer: Puntero urgente.
Options / Padding: Opciones y relleno.
Datos

La información que más nos interesa de la cabecera TCP, desde el punto de vista de la seguridad es:

- El número de puerto origen del mensaje, que identifica un proceso en el sistema origen.
- El número de puerto destino, que identifica un proceso en el sistema destino.
- El número de secuencia, indicando el número de bytes enviados.
- El número de ACK que indica el último byte recibido.
- Una serie de bits de opciones de sesión TCP. Ej: SYN: Flag usado en los dos primeros pasos de la creación de una sesión

TCP Three Way handshake

Mecanismo de conexión TCP en tres etapas. Por cada petición de conexión, entre un cliente y un servidor, el servidor crea una tabla que lleva la cuenta de sesiones semi-establecidas de TCP (una tabla que lleva la cuenta de sesiones "a medio hacer"). Las tres etapas son:

1. Enviar un paquete TCP con el flag SYN activado.
2. Recibir un SYN-ACK
3. Enviar un ACK

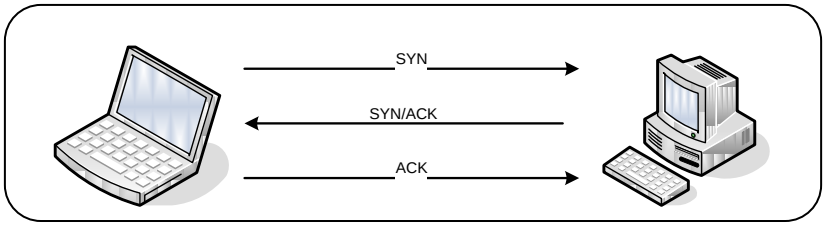

Fig. 2.13: Three Way Handshake

Hasta que nos se completan los tres pasos no se da por establecida la conexión.

Protocolo ARP

Implementa el mecanismo de resolución de una dirección IP a una dirección MAC. El hardware de red se comunica intercambiando tramas Ethernet en la capa de enlace de datos.

Cuando se manda un paquete IP, la máquina emisora necesita conocer la dirección MAC de la receptora. Para obtenerla, se envía una petición broadcast ARP a toda la red. Esta petición preguntará, "¿Cuál es la dirección MAC asociada a esta dirección IP?". La máquina que tenga la dirección IP en cuestión responderá a través de un paquete ARP, proporcionando a la máquina emisora la dirección MAC solicitada. A partir de aquí, la máquina origen conocerá cual es la dirección MAC que corresponde a la dirección IP a la que se quiere enviar paquetes. Esta correspondencia permanecerá durante algún tiempo en la caché (para evitar hacer una nueva petición cada vez que se envía un paquete).

Carlos J. Mateos Orozco
Alejandro Carrasco Muñoz

Capítulo 4

Seguridad perimetral

4.1. Introducción a la seguridad perimetral

La seguridad perimetral comprende uno de los principales pilares para la defensa de redes. Se basa en el establecimiento de recursos de protección de los diferentes perímetros que conforman una red. La seguridad perimetral permite definir diferentes niveles de seguridad, permitiendo el acceso a determinados usuarios, internos o externos, a ciertos servicios y recursos, y denegando cualquier tipo de acceso a otros.

En base al tipo de elementos presentes en cada perímetro de red, la criticidad de éstos y el uso que se quiera dar a la red, se definirán diferentes niveles de protección. Por ejemplo, Las medidas de protección implementadas sobre una red de servidores de bases de datos serán diferentes a las aplicadas a los servidores de publicación, visibles desde Internet, y diferentes a los de una red de ofimática.

Básicamente se define como el conjunto de elementos y sistemas, encargados de:

- Proteger perímetros físicos o lógicos.
- Controlar el tráfico que cursa entre perímetros.
- Monitorizar el tráfico entre perímetros.
- Detectar y bloquear tentativas de intrusión.
- Proporcionar en cada perímetro un único punto de interconexión.
- Ocultar detalles de la red como nombres, topología, tipos de dispositivos de interconexión, etc.
- Implementar políticas de seguridad.

El perímetro protegido se denomina **perímetro de seguridad**. Debe estar claramente separado de la red externa o zona de riesgo. Éste suele ser propiedad de la misma organización que lo forma.

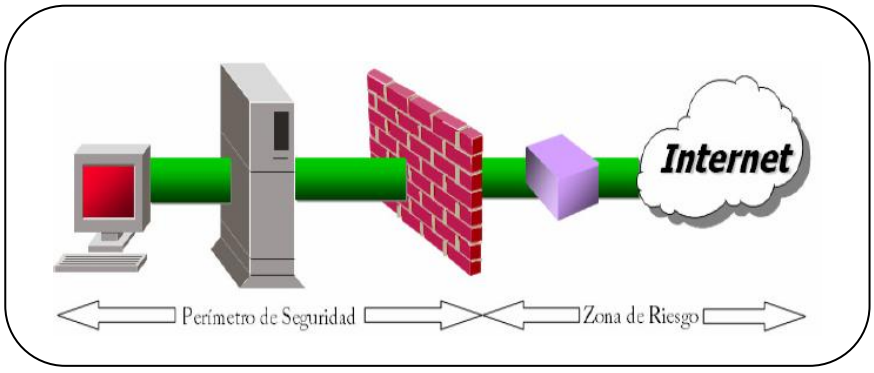

Fig. 4.1: Perímetro de seguridad

Los perímetros de red pueden incluir conexiones a:

- Redes de servidores
- Internet
- Publicaciones (visibles desde Internet)
- Usuarios remotos
- Sedes remotas
- Redes inalámbricas
- Redes de socios

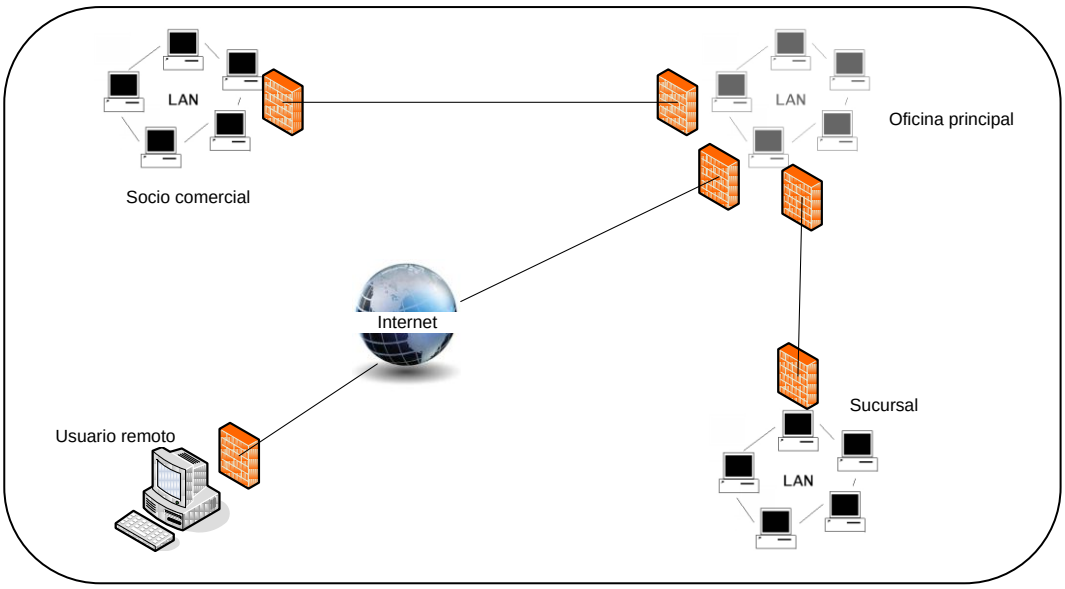

Fig. 4.2: Perímetros de red

4.2. Elementos de la seguridad perimetral

La frontera fortificada de nuestra red puede incluir lo siguiente:

- Routers
- Firewalls
- Proxy
- Sistemas de detección de intrusiones (IDS)
- Redes privadas virtuales (VPNs)
- Zonas desmilitarizadas (DMZ) y subredes controladas

Echemos un ligero vistazo a estos conceptos antes de entrar en profundidad.

Router

Dispositivo de nivel 3 que direcciona paquetes (datagramas) a través de la red basándose en la información de la capa de red. Se llama **router frontera** o gateway exterior al último router que controlamos antes del acceso a Internet.

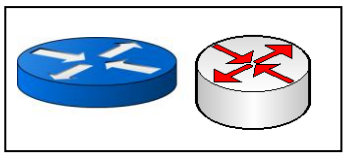

Fig. 4.3: Icono de un router

Firewalls

Son los dispositivos encargados de controlar el acceso entre los diferentes perímetros de la red. Éstos se configuran mediante reglas de filtrado para permitir el paso únicamente a paquetes autorizados, consiguiendo además, ser transparentes para los usuarios legítimos de la red.

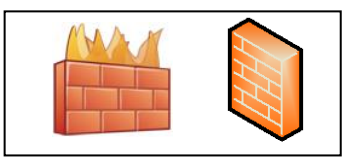

Fig. 4.4: Iconos de un firewall

Sistema de detección de intrusiones (IDS)
Un IDS es como un sistema de alarma antirrobo de la red. Está formado por un conjunto de sensores, localizados estratégicamente por la red interna que capturan todo el tráfico y buscan detectar amenazas y acciones hostiles.

Red privada virtual (VPN)
Sesión de red protegida establecida a través de canales no protegidos (ej: Internet). Una VPN permite a un usuario externo a la red interna participar en ésta como si estuviese conectado directamente a ella.

Zona desmilitarizada (DMZ)
En esta zona suelen situarse los servicios públicos. Su objetivo es separar los servidores que sirven aplicaciones visibles desde redes externas, de aquellos servidores internos.

4.3. Buenas prácticas

En esta sección se presentan los puntos en los que un administrador debe poner mayor desempeño a la hora de asegurar su red.

1. Diseñar la red teniendo en cuenta la seguridad

La seguridad de una red debe ser planificada desde la fase del diseño de la misma. Controlar y monitorizar son las premisas fundamentales sobre las que debe de girar todo nuestro diseño.

- **Controlar**: Hace referencia a la capacidad de decidir a qué tráfico le permitimos el acceso y a cual no. Esta función es implementada por los firewalls.

- **Monitorizar:** Es la capacidad para acceder al tráfico que entra y sale de una red y de una máquina en concreto. Nos permite crear estadísticas del uso de la red, localizar congestiones y por supuesto, detectar ataques contra nuestras máquinas.

Nuestra infraestructura debe contemplar la instalación de cortafuegos para controlar el acceso a la red y de equipos con capacidad de monitorización (IDS´s) para vigilar las zonas críticas.

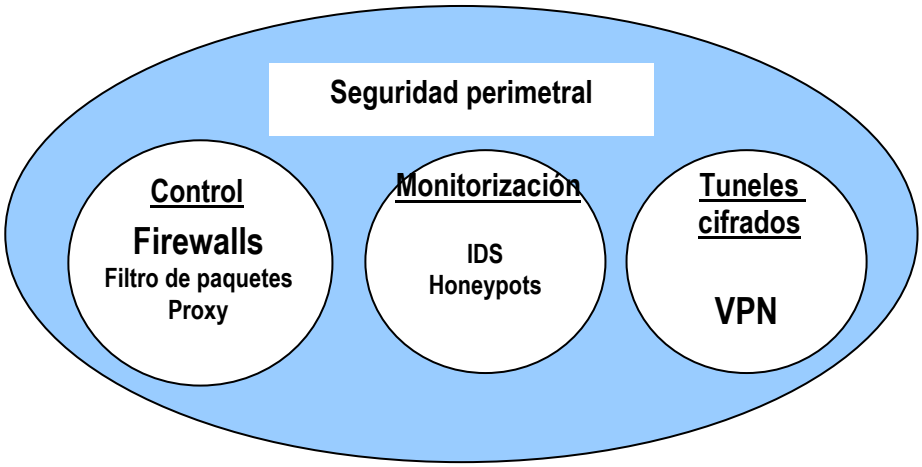

Fig. 4.5: Seguridad perimetral. Esquema

2. Separación de servicios y subredes

Para asegurar nuestra red es fundamental separar los servicios que ofrecemos según su funcionalidad y su sensibilidad. Por ejemplo, por un lado los equipos servidores y por otro los equipos de los clientes. Esta separación nos permite controlar el acceso de manera independiente a cada conjunto y tipo de servicios. Además así se garantiza que el posible compromiso de uno de los grupos no afectará al resto.

La primera separación lógica es aislar nuestra red corporativa de Internet y para ello debemos situar dispositivos cortafuegos en todos los puntos de acceso a las LAN. Lo restrictivo que sean éstos dependerá de la política de cada organización ó empresa, pero lo habitual suele ser restringir bastante el tráfico en la dirección Internet -> LAN (hacia dentro) y menos en la dirección opuesta o hacia fuera.

Otra separación habitual y necesaria es aislar la red interna de los servidores que ofrecen servicios a Internet (Web, FTP, SMTP…). Estas máquinas van a estar en contacto directo con el mundo exterior, lo cual implicará que exista un alto riesgo de que sean comprometidas. Si este es el caso, un atacante podrá tomar como plataforma, por ejemplo, un servidor Web, y desde ahí lanzar ataques al resto de nuestra red, pudiendo llegar a comprometer otros servidores importantes (bases de datos, directorio de usuarios, etc).

Se hace necesario un nivel adicional de seguridad. Debemos separar la red interna, desde un punto de vista lógico, de la zona donde van los servidores. Esta zona, como ya hemos visto, se conoce como DMZ (De-Militarized Zone) y se trata de una zona no segura.

Se deben de agregar tantas subdivisiones y capas de seguridad como sean necesarias. Establecer redes separadas dentro de la organización es importante para mantener el orden lógico sobre nuestra red permitir un mayor control sobre el acceso a los servicios que se ofrecen en cada una de ellas.

4.4. Objetivos

En este curso se pretende abordar la seguridad perimetral según sus dispositivos. En los siguientes capítulos se mostrarán con detalles estos elementos de protección:

1. **Dispositivos para el control** – Firewalls
    a. Filtrado de paquetes (capítulo 5)
    b. Proxys (capítulo 6)
2. **Dispositivos para monitorización** - Sistemas de detección de intrusiones
    a. IDS / IPS (capítulo 7)
    b. Honeypots
3. Túneles cifrados – VPN´s