

CC51C Comunicación de Datos Seguridad

1 Introducción

Como definición general, podemos decir que un computador es seguro cuando se comporta de la manera que se espera. Cualquier alteración en su comportamiento respecto del esperado puede tener múltiples causas, que van desde un error humano, de software o hardware, hasta manipulación por terceros no autorizados. En un sentido más específico, se deben tener en cuenta los siguientes aspectos del comportamiento de un sistema:

- Confidencialidad: hay datos que no debieran poder ser leídos ni copiados por personas que no estén explícitamente autorizadas para ello.
- Integridad de datos: evitar que información (ya sean datos o bien programas) sean alterados o borrados.
- Disponibilidad: protección de los servicios para que los usuarios autorizados puedan acceder realmente al servicio en el momento que lo necesitan.
- Consistencia: que el sistema se comporte de la forma que se espera. Básicamente se refiere a la integridad del sistema en su totalidad.
- Control: regular el acceso a los recursos del sistema. Una vez definidos los accesos autorizados para cada usuario, estos accesos deben ser controlados de alguna manera.
- Auditoría: registro de eventos del sistema, para que sea posible determinar por qué y/o cuándo sucedió algún evento. Puede incluso permitir el “deshacer” ciertas operaciones.

Al hablar de información, se está refiriendo a la definición más amplia, que incluye todas las formas que pueda tomar, ej: datos y/o programas almacenados en discos duros, cintas de respaldo, hojas de papel, grabaciones de audio en cualquier medio, conversaciones telefónicas o de pasillo, por nombrar algunas.

2 Aspectos específicos

- IP spoofing
- Denial of Service (DoS)
- Distributed Denial of Service (DDoS)
- Firewalls

2.1 Tunneling

- protocolos no ruteables
- IPv6 over IPv4
- Multicast
- VLANs

3 Firewalling

Un firewall es un router con algunos aspectos adicionales, que permiten limitar según algunas reglas qué paquetes son ruteados y cómo.

Tipos de firewalls:

1. proxy based firewalls
2. packet filtering
3. NAT (Network Address Translation)
4. de tipo "Big Brother"(omnipotentes)

3.1 Proxy based firewalls

Un proxy es un servidor intermediario para algún servicio. Un cliente se conecta al proxy y le pide acceder algún recurso en un servidor. El proxy abre una conexión con el servidor y le entrega los resultados al cliente. Si el protocolo involucra más que una consulta y su resultado, el proxy continúa interactuando entre cliente y servidor. En muchas ocasiones, el proxy además puede mantener un cache de las respuestas que recibe del servidor y entrega al cliente. Así, al recibir un nuevo requerimiento que es idéntico a otro anterior, puede enviar los resultados sin necesidad de conectarse nuevamente al servidor, ahorrando así ancho de banda.

Los proxies son dependientes del servicio, ya que deben manejar el protocolo específico. No es posible construir un proxy que entienda a priori los protocolos conocidos, tanto actuales como futuros.

Los firewalls basados en proxy consideran en general una red privada (ej: 10.0.0.0), la cual no es accesible desde la Internet. La conectividad desde la red privada hacia Internet se provee mediante un servidor proxy que sí está conectado a la red. Sin embargo, ese servidor no actúa como router (nivel red), sino que solamente actúa a nivel de aplicación (o de sesión, si vemos el modelo ISO/OSI).

Existe también el concepto de proxying transparente, en donde un router redirige todos los paquetes que van a un puerto específico (por ejemplo el 80, que corresponde a HTTP) a un servidor proxy. Ese servidor proxy se hace pasar por el servidor, sin que el cliente se entere de que en realidad no está hablando directamente con el servidor.

3.2 Packet filtering firewalls

Un packet filtering firewall es básicamente un router, que puede decidir si rutear un paquete o no en base a ciertas reglas. En general no tiene mucho sentido aplicar reglas conociendo solamente los paquetes de red. Por eso que la mayoría de los routers que pueden actuar como filtros pueden además manejar el nivel transporte.

Es posible con reglas simples evitar que se realicen conexiones desde una red a otra, pero permitir conexiones en sentido contrario. O permitir conexiones a puertos específicos y a otros no.

3.3 NA[P]T (Network Address [Port] Translation)

La técnica de NAT se puede dividir en varias más específicas. Básicamente, un router que implementa NAT tiene la capacidad de mapear direcciones IP y/o puertos TCP (o UDP). La definición de ese mapeo de direcciones/puertos puede hacerse en forma estática o dinámica, según el uso que se le da a la traducción.

En estricto rigor, el proxy transparente también es una traducción de direcciones o NAT.

Ejemplos de aplicación: - Uso de un host en una red interna privada para proveer servicios hacia Internet: se asigna una dirección real que llega al router que implementa NAT, y éste lo hace llegar al host cambiando la dirección a nivel IP. También cambia los paquetes salientes para que se vean como provenientes de la dirección real, y el cliente externo no se da cuenta de estas traducciones.

- Multiplexión de direcciones IP: se tienen por ejemplo 500 computadores, y se desea conectarlos a Internet, pero se dispone solo de una red clase C, de 253 hosts como máximo (+ broadcast, número de red y router = 256 hosts). En general, si no hay más del 50% usando recursos al mismo tiempo, se pueden asignar direcciones dinámicamente mientras se usen, y se reciclan cuando ya no exista ninguna conexión activa.

- Multiplexión de conexiones (también llamado masquerading o NAPT): el principio es el mismo de la multiplexión de direcciones IP, pero en vez de mapear direcciones, se mapean conexiones (Network Address Port Translation). Esto se usa en el caso de contar por ejemplo con una sola dirección IP, y querer realizar conexiones desde múltiples direcciones internas (privadas). Es comparable al esquema de proxy, pero a nivel de red (IP). Por eso, no es dependiente del servicio, ya que solamente se preocupa de las conexiones. Claro que hay excepciones. Porque hay protocolos que usan más de una conexión, y en particular esa conexión puede originarse desde el servidor hacia el cliente. Un ejemplo concreto de esto es el servicio de ftp, donde la conexión de control la inicia el cliente, pero al pedir una transferencia de archivos, es el servidor quien inicia la conexión. Como esa conexión va a llegar al router, y no existe un mapeo definido, no va a funcionar. Para solucionar esto, es necesario monitorear el contenido de los paquetes, y anticipar esas conexiones, creando los mapeos necesarios e incluso modificando el contenido de los paquetes para que todo funcione. Y eso es claramente dependiente del servicio.

- Pool de servidores o load balancing: muchas veces es deseable contar con varios servidores que prestan un mismo servicio, para poder soportar la carga de trabajo. Una

forma de lograr esto es usando una red privada, y NAT para que las conexiones entrantes se vayan distribuyendo entre los servidores disponibles. Esto, sumado a algún algoritmo o protocolo que permita distribuir la carga entre los servidores, es completamente transparente para el cliente. Además permite que la caída o mantención de servidores específicos pasen desapercibidos, siempre y cuando el NAT no redirija paquetes hacia ese servidor. Obviamente este esquema sirve cuando el cuello de botella es la capacidad de procesamiento y no el ancho de banda de la red.

- Load balancing de la red: se trata de aplicar el mismo esquema del pool de servidores para el caso en que la red es el cuello de botella. Se tiene redundancia de conexiones (ej: un router está conectado a varios proveedores), y por lo tanto existen varios caminos entre dos routers que usan NAT. Entonces, en vez de enviar una conexión completa a través de uno de los caminos (la solución por defecto), pueden usar dos direcciones distintas para llegar al mismo host. Pero como son paquetes de una misma conexión, el host de origen no puede ir cambiando la dirección de destino. Entonces son los routers los cuales van eligiendo la dirección (y por ende el camino), y el último router antes de llegar al host vuelve la dirección de destino a la original, para que la reciba correctamente el receptor.

3.4 Firewalls omnipotentes ("Big Brother")

Los firewalls tipo "Big Brother" son los que además de las funcionalidades como filtrado, traducción de direcciones y proxying permiten monitorear el flujo de información y actuar en base al contenido. Estas funcionalidades permiten por ejemplo detectar transmisiones de virus, de información corporativa secreta, ofensas e injurias contra el jefe, entre otras maravillas. Algunos ejemplos más útiles son: - monitorear conexiones ftp para detectar cuándo el servidor se conectará al cliente para enviarle un archivo, y permitir pasar esa conexión. Luego de la transmisión se deniegan nuevamente los permisos. - detectar conexiones maliciosas en base al contenido (intentos reiterados de adivinar passwords, scanning sistemático de servicios, etc), y alertar sobre el hecho.