CURSO

INTRODUCCIÓN A LA SEGURIDAD EN INTERNET Y APLICACIONES

Prólogo

En la actualidad, la fusión de casi cualquier actividad con las computadoras tiene una gran gama de aplicaciones, que se extiende aún más por el uso de internet o redes amplias. Todos los sectores de la sociedad, organismos públicos o privados, centros de educación, compañías, hogares, etc., se conectan entre sí o conforman una red.

Por lo que los problemas de seguridad en las redes son comunes y uno de los temas de mayor relevancia de hoy en día en la computación es la seguridad e integridad de la información.

En este curso de Introducción a la Seguridad en Internet y Aplicaciones se abordan aspectos fundamentales y actuales, tales como: Programación Segura, Criptología, Firewall y Seguridad en Internet, para que el desarrollador de software adquiera los conocimientos indispensables que le permitan implementar seguridad en sus actividades cotidianas.

Agradecimiento y Reconocimiento

Después de una ardua tarea de investigación se ha logrado la creación de una obra vasta en conocimiento en el desarrollo de las Tecnologías de la Información y Comunicación.

La presente obra no hubiera sido posible sin la valiosa aportación de destacados autores y especialistas en la materia. Es por ello que a manera de reconocimiento queremos agradecer su participación:

INTRODUCCIÓN A LA SEGURIDAD EN INTERNET Y APLICACIONES

M. en C. Gerardo Contreras Vega
Universidad Veracruzana

M. en C. Carlos A. Ochoa Rivera
Universidad Veracruzana

Ing. Adolfo de Jesús Solís Muñiz
Universidad Autónoma de Chiapas

Introducción

El presente curso te ofrece el estudio, análisis y aplicación de escenarios de seguridad a través de diversas alternativas para cada caso particular.

Abordaremos siete unidades con conceptos y temas los cuales son: Políticas y Procedimientos de Seguridad, Programación Segura, Criptografía, Código Dañino. Firewalls, Seguridad en Internet y Redes Privadas Virtuales (VPN).

Utilizaremos documentos, libros, publicaciones y páginas de Internet para su lectura y consulta, además de actividades como foros de discusión, exámenes en línea, análisis de casos y prácticas de laboratorio, que desarrollarás durante el curso de manera individual y en equipos, contando con el apoyo de los profesores tutores, quienes resolverán tus dudas administrativas y académicas vía email y foros.

Esperamos que tengas una participación proactiva y entusiasta, cumpliendo en los tiempos establecidos y en las formas solicitadas para concluir el curso con éxito. Y esperamos, al término del curso que tengas las bases suficientes para afrontar y proponer esquemas de seguridad en redes de comunicaciones.

Intención Educativa

¿Confías en tu Red o en la Internet?

¡Tú puedes ayudar a que la red sea segura!

Al tomar el curso de seguridad podrás desarrollar habilidades con el fin de proponer soluciones o alternativas de protección competitivas, utilizando para ello el autoaprendizaje y la retroalimentación de conocimientos, y asumiendo actitudes de honestidad, responsabilidad, confidencialidad y empatía. **"Estudia la materia de seguridad y siéntete seguro"**

Objetivos Generales

Al terminar el curso, el alumno será capaz de:

• Analizar diferentes escenarios de seguridad y propondrá alternativas para la prevención y resolución de problemas de seguridad en redes y sistemas.

• Identificar los diferentes conceptos de seguridad y tipos de ataques en una red de computadoras.

• Desarrollar y utilizarár herramientas contra ataques en una red de computadoras.

Metodología

Se utilizarán distintos métodos de aprendizaje para cubrir los contenidos. A continuación se describirán dichos métodos. Adicionalmente, en las actividades de aprendizaje se encontrarán las indicaciones específicas en relación con la técnica utilizada en cada una de ellas.

Aprendizaje Basado en Problemas (ABP)

La técnica de Aprendizaje Basado en Problemas (ABP, ó del inglés "PBL-Problem Based Learning") es una técnica didáctica constructivista, la cual consta de los siguientes pasos:

1. El docente formará grupos de alumnos que trabajarán en equipo para resolver el problema del escenario propuesto. Se nombrará un secretario por equipo, el cual organizará el funcionamiento del mismo y se encargará de enviar los resultados de las actividades realizadas al profesor.

2. El docente presentará el material del escenario, asignará roles y unificará criterios.

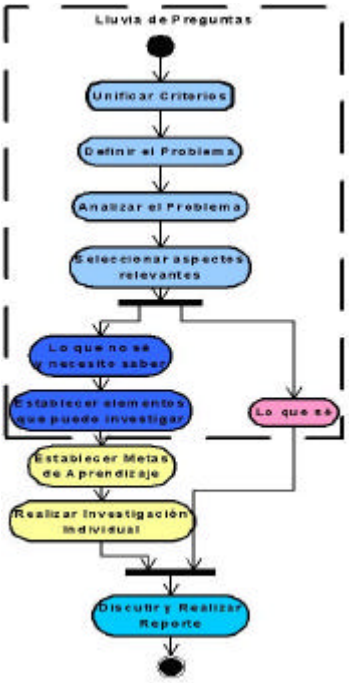
3. Los estudiantes elaborarán una serie de preguntas respecto al escenario; posteriormente, procurarán resumir el problema del escenario planteándolo (de ser posible) en forma de pregunta. El docente verificará que tal pregunta sea la apropiada.

4. Una vez establecida la pregunta principal y las preguntas particulares, los estudiantes analizarán y seleccionarán en equipo las preguntas más relevantes a investigar.

5. La primera tarea de cada estudiante será la entrega de un reporte individual correspondiente a una investigación hecha por él mismo.

6. Posteriormente los integrantes discutirán en equipo los resultados de su investigación para establecer en común las respuestas a las preguntas planteadas.

7. En la segunda y última tarea, cada integrante entregará un reporte individual, sin embargo, este deberá reflejar los aspectos puestos en común en la discusión en grupo. Adicionalmente, en algunos casos de programación, se solicitarán la entrega de programas.



Método de Casos

El método de casos difiere de los sistemas de enseñanza tradicionales porque exige que el alumno tome parte activa en el análisis de los problemas y en la toma de decisiones para la solución a situaciones reales muy específicas.

Un aspecto muy importante del método de casos, es el aprender de la experiencia de otros casos similares para la toma de decisiones a la hora de solucionar un problema. La solución de un mayor número de casos incrementará la habilidad para identificar problemas y formular soluciones a problemas reales.

En el método de casos, el proceso que se siga para tomar decisiones y las decisiones mismas, sustentadas en un análisis adecuado, son la clave. Este tipo de ejercicios nos permite aprender a comunicar criterios, defender hechos y opiniones en debates.

Los pasos a seguir en el método de casos se muestran en el siguiente diagrama:

Comprobación de Lectura

La técnica de comprobación de lectura tiene como finalidad fomentar en el alumno la habilidad de leer, analizar y comprender. La comprensión se comprobará al final de cada lección, ya que se presenta una evaluación por medio de preguntas muy puntuales acerca de la lectura.

Los materiales que se utilizarán en este método son una recopilación de diferentes autores de un tema, para homogenizar los conceptos e ideas referentes al tema.

La técnica de comprobación de lectura es una de las más empleadas en los procesos de enseñanza-aprendizaje y tiene como finalidad conformar conceptos e ideas propias al alumno, por lo que no pretende que se memoricen los temas tratados.

Fuentes de Información

Cresson Wood Charles, *Information Security Policies Made Easy*, México 1997, UNAM

Barbara L. Dijker, *A Guide to Developing Computing Policy Documents - Series: Short topics in system administration N° 2*, 1996, USENIX Association for SAGE (System Administration Guild)

RFC (Request for Comments) **1281, 1244** – Site Security Handbook, Internet Engineering Task Force (IETF), 1991.

McMillan Rob, *Site Security Policy Development*, Australia 1995, Information Technology Services - Griffith University

Práctical UNIX & Internet Security – Garfinkel & Spafford, O'Reilly

Vega Calderón César, *Consideraciones para la elaboración de políticas de seguridad en cómputo para una organización*, 1998, Día Internacional de la Seguridad en Cómputo

Internet Security Policy: A Technical Guide - NIST Special Públication 800-XX - Barbara Guttman, Robert Bagwill, 1997

Stephen L. Arnold, Ph.D., Security Policies for the Internet

CAF "Academic Computing Policy Statements" Archive (ejemplos de políticas seguidas en diversos sitios) –
The Electronic Frontier Foundation (EFF) - 1999

Carling, M; Degler, Stephen; Administración de Sistemas Linux, Guía Avanzada, Madrid 1999, Prentice Hall Iberia.

Anónimo; Linux Máxima Seguridad; Madrid 2000, Pearson Educación, S. A.

Siyan, Karanjit; Hare, Chris; Firewalls y la seguridad en Internet; Segunda Edición, 1997, Prentice-Hall Hiberoamericana, S.A.
Bibliografía (2)

Garfinkel, Simson; Spafford, Gene; Seguridad y comercio en el Web; 1999, McGrawHill & O'Reilly.

McClure, Stuart; Scambray, Joel; Hacking Expose, Network Security Secres & Solutions; Tercera Edición, 2002; Osborne/McGraw-Hill.

Forma de Evaluación

El alumno deberá cumplir con el 100% de asistencia y deberá completar todos los ejercicios y tareas descritos en cada una de las unidades.

La forma de evaluación de este curso es por medio de las rúbricas descritas para cada una de las unidades, siendo 3 el puntaje más alto que se pueda alcanzar, los criterios que se toman para evaluar cada uno de los ejercicios vienen descritos en las tablas de las rúbricas.

Contenido

1. Políticas y Procedimientos de Seguridad

1.1. Seguridad Física

Niveles de seguridad

De acuerdo con las norma de seguridad establecidas por el departamento de la defensa de Estados Unidos, basados en el libro naranja[1], se usan varios niveles para proteger de ataques el hardware, el software y la información almacenada. Estos niveles se refieren a diferentes tipos de seguridad física, autentificación de usuario, confiabilidad del software del sistema operativo y aplicaciones de usuario. Estos estándares también imponen límites a los sistemas que puedan conectarse a su equipo.

Nivel D1

El nivel D1 es la forma más baja de seguridad. Esta norma establece que el sistema entero no es confiable. No se dispone de protección para el hardware; el sistema operativo se compromete fácilmente y no existe autentificación respecto de los usuarios y sus derechos a tener acceso a la información almacenada en la computadora. Los sistemas operativos no distinguen entre los usuarios y no tienen definido ningún método para determinar quién está en el teclado. Asimismo, no tienen ningún control con respecto a la información a la que se pueda tener acceso en las unidades de disco duro de la computadora.

Nivel C1

El nivel C tiene dos subniveles de seguridad: el C1 y el C2. El nivel C1, Sistema de Protección de Seguridad Discrecional, se refiere a la seguridad disponible en un sistema Unix típico. Existe cierto nivel de protección para el hardware, ya que éste no puede comprometerse fácilmente. Los usuarios deben identificarse ante el sistema mediante su login y contraseña. Se emplea esta combinación para determinar los derechos de acceso a programas e información que tiene cada usuario. Estos derechos de acceso son los permisos de archivos y de directorio.

[1] El libro Naranja se adopto comó estándar del Departamento de la Defensa en 1985, y ha constituido el método básico para evaluar la seguridad de sistemas operativos multiusuario en mainframes y mini, así como bases de datos y redes.

Los controles de acceso discrecional permiten al dueño del archivo o directorio, así como al administrador del sistema, evitar que ciertas personas o grupos tengan acceso a dichos programas o información. La mayoría de las tareas cotidianas de administración del sistema son realizadas por la cuenta raíz (root), y también es común que dicha cuenta la conozca más de una persona en la organización, lo cual resulta difícil identificar a dichas personas que la usan.

Nivel C2

El subnivel C2 esta diseñado con las funciones del nivel C1, más algunas características adicionales que crean un ambiente de acceso controlado. Este ambiente tiene la capacidad de restringir aún más el que los usuarios ejecuten ciertos comandos o tengan acceso a ciertos archivos, con base no sólo en los permisos, sino también en los niveles de autorización. Este nivel de seguridad requiere que se audite el sistema, lo cual implica registrar una auditoría por cada acción que ocurra en el sistema.

La auditoría se utiliza para llevar registros de todas las acciones relacionadas con la seguridad, como pueden ser las actividades efectuadas por el administrador del sistema. La auditoria requiere de autentificación adicional, lo que implica que la auditoria utilice recursos adicionales del procesador y del subsistema de disco.

Nivel B1

El nivel de seguridad B consta de tres niveles. El nivel B1, llamado Protección de Seguridad Etiquetada, es el primer nivel con soporte para seguridad de multinivel, como el secreto y el ultra secreto. En este nivel se establece que el dueño del archivo no puede modificar los permisos de un objeto que esté bajo control de acceso obligatorio.

Nivel B2

El nivel B2, conocido como Protección Estructurada, requiere que todos los objetos estén etiquetados, los dispositivos como discos, cintas y terminales, pueden tener asignado uno o varios niveles de seguridad. Este es el primer nivel en el que se aborda el problema de la comunicación de un objeto con otro que se encuentra en un nivel de seguridad inferior.

Nivel B3

El nivel B3, llamado de Dominios de Seguridad, refuerza los dominios con la instalación de hardware. Por ejemplo, se utiliza hardware de manejo de memoria

para proteger el dominio de seguridad contra accesos no autorizados y modificaciones de objetos en diferentes dominios de seguridad. Este nivel requiere también que la terminal del usuario esté conectada al sistema a través de una ruta de acceso confiable.

Nivel A

Este nivel es conocido como Diseño Verificado, constituye actualmente el nivel de seguridad validada más alto. Cuenta con un proceso estricto de diseño, control y verificación. Para alcanzar este nivel de seguridad, deben incluirse todos los componentes de los niveles inferiores; el diseño debe verificarse matemáticamente, y debe realizarse un análisis de los canales abiertos y de distribución confiable. La distribución confiable significa que el hardware y el software hayan estado protegidos durante su traslado para evitar violaciones de los sistemas de seguridad.

Planeación de Seguridad en Redes

Es importante tener una política de seguridad de red bien concebida y efectiva que pueda proteger la inversión y los recursos de información de su compañía. Vale la pena implementar una política de seguridad si los recursos y la información que su organización tiene en sus redes merecen protegerse. La mayoría de las organizaciones tienen en sus redes información delicada y secretos importantes; esto debe protegerse del vandalismo del mismo modo que otros bienes valiosos como la propiedad corporativa y los edificios de oficinas.

La mayoría de los diseñadores de redes por lo general empiezan a implementar soluciones de firewall antes de que se haya identificado un problema particular de seguridad de red. Quizá una de las razones de esto es que idear una política de seguridad de red efectiva significa plantear preguntas difíciles acerca de los tipos de servicios de intercedes y recursos cuyo acceso se permitirá a los usuarios, y cuáles tendrán que restringirse debido a los riesgos de seguridad.

Si actualmente sus usuarios tienen acceso irrestricto a la red, puede ser difícil aplicar una política que limite ese acceso. También debe tomar en cuenta que la política de seguridad que usted debe usar es tal, que no disminuirá la capacidad de su organización. Una política de red que impide que los usuarios cumplan efectivamente con sus tareas, puede traer consecuencias indeseables: los usuarios de la red quizá encuentren la forma de eludir la política de seguridad, lo que la vuelve inefectiva. Una política de seguridad en redes efectiva es algo que todos los usuarios y administradores de redes pueden aceptar y están dispuestos a aplicar.

Política de Seguridad del Sitio

Una organización puede tener muchos sitios, y cada uno contar con sus propias redes. Si la organización es grande, es muy probable que los sitios tengan diferente administración de red, con metas y objetivos diferentes. Si esos sitios no están conectados a través de una red interna, cada uno de ellos puede tener sus propias políticas de seguridad de red. Sin embargo, si los sitios están conectados mediante una red interna, la política de red debe abarcar todos los objetivos de los sitios interconectados.

En general, un sitio es cualquier parte de una organización que posee computadoras y recursos relacionados con redes. Algunos, no todos de esos recursos son los siguientes:

- **Estaciones de trabajo.**
- **Computadoras hosts y servidores.**
- **Dispositivos de interconexión: gateways, routers, bridges, repetidores.**
- **Servidores de terminal.**
- **Software para conexión de red y de aplicaciones.**
- **Cables de red.**
- **La información de archivos y bases de datos.**

La política de seguridad del sitio debe tomar en cuenta la protección de esos recursos. Debido a que el sitio está conectado a otras redes, la política de seguridad del sitio debe considerar las necesidades y requerimientos de seguridad de todas las redes interconectadas. Éste es un punto importante ya que es posible idear una política de seguridad que salvaguarde sus intereses pero que sea dañina para los de otros.

1.2. Políticas

Políticas de Seguridad

El término política de seguridad se suele definir como el conjunto de requisitos definidos por los responsables directos o indirectos de un sistema que indica en términos generales qué está y qué no está permitido en el área de seguridad durante la operación general de dicho sistema. Al tratarse de `términos generales', aplicables a situaciones o recursos muy diversos, suele ser necesario refinar los requisitos de la política para convertirlos en indicaciones precisas de qué es lo permitido y lo denegado en cierta parte de la operación del sistema, lo que se denomina política de aplicación específica.

Al elaborar las políticas que reflejan la seguridad en su red pueden adoptarse dos posturas principales. Estas declaraciones fundamentales constituyen la base de todas las demás políticas de seguridad y regulan los procedimientos puestos en marcha para implementarlas.

Una política de seguridad puede ser prohibitiva, si todo lo que no está expresamente permitido está denegado, es decir, si una organización proporciona una serie de servicios bien determinados y documentados, y cualquier otra cosa esta prohibida, o permisiva, si todo lo que no está expresamente prohibido está permitido. Evidentemente la primera aproximación es mucho mejor que la segunda de cara a mantener la seguridad de un sistema; en este caso la política contemplaría todas las actividades que se pueden realizar en los sistemas, y el resto serían consideradas ilegales.

Cualquier política ha de contemplar seis elementos claves en la seguridad de un sistema informático:

Disponibilidad

Es necesario garantizar que los recursos del sistema se encontrarán disponibles cuando se necesitan, especialmente la información crítica.

Utilidad

Los recursos del sistema y la información manejada en el mismo han de ser útil para alguna función.

Integridad

La información del sistema ha de estar disponible tal y como se almacenó por un agente autorizado.

Autenticidad

El sistema ha de ser capaz de verificar la identidad de sus usuarios, y los usuarios la del sistema.

Confidencialidad

La información sólo ha de estar disponible para agentes autorizados, especialmente su propietario.

Posesión

Los propietarios de un sistema han de ser capaces de controlarlo en todo momento; perder este control en favor de un usuario malicioso compromete la seguridad del sistema hacia el resto de usuarios.

Para cubrir de forma adecuada los seis elementos anteriores, con el objetivo permanente de garantizar la seguridad corporativa, una política se suele dividir en puntos más concretos a veces llamados normativas. El estándar ISO 17799 define las siguientes líneas de actuación:

- **Seguridad organizacional**

  Aspectos relativos a la gestión de la seguridad dentro de la organización (cooperación con elementos externos, soporte externo, estructura del área de seguridad, etc).

- **Clasificación y control de activos**

  Inventario de activos y definición de sus mecanismos de control, así como etiquetado y clasificación de la información corporativa.

- **Seguridad del personal**

  Formación en materias de seguridad, cláusulas de confidencialidad, reporte de incidentes, monitorización de personal, etc.

- **Seguridad física y del entorno**

  Bajo este punto se engloban aspectos relativos a la seguridad física de los recintos donde se encuentran los diferentes recursos de la organización y de los sistemas en sí, así como la definición de controles genéricos de seguridad.

- **Gestión de comunicaciones y operaciones**

  Este es uno de los puntos más interesantes desde un punto de vista estrictamente técnico, ya que engloba aspectos de la seguridad relativos a la operación de los sistemas y telecomunicaciones, como los controles de red, la protección frente a malware, la gestión de copias de seguridad o el intercambio de software dentro de la organización.

- **Controles de acceso**

Definición y gestión de puntos de control 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 las aplicaciones, cifrado de datos, control de *software*, etc.

- **Gestión de continuidad de negocio**

Definición de planes de continuidad, análisis de impacto, simulacros de catástrofes, etc.

Requisitos Legales

Evidentemente, una política ha de cumplir con la normativa vigente en el país donde se aplica, si una organización se extiende a lo largo de diferentes países, su política tiene que ser coherente con la normativa del más restrictivo de ellos. En este apartado de la política se establecen las relaciones con cada ley: derechos de propiedad intelectual, tratamiento de datos de carácter personal, exportación de cifrado, etc., junto a todos los aspectos relacionados con registros de eventos en los recursos (logs) y su mantenimiento.

Definir una política de seguridad de red significa elaborar procedimientos y planes que salvaguarden los recursos de la red contra pérdida y daño. Uno de los enfoques posibles para elaborar dicha política es examinar lo siguiente:

- **¿Qué recursos está tratando de proteger?**
- **¿De quienes necesita proteger los recursos?**
- **¿Qué tan posibles son las amenazas?**
- **¿Qué tan importante es el recurso?**
- **¿Qué medidas puede implementar para proteger sus bienes de forma económica y oportuna?**

Se debe examinar periódicamente su política de seguridad de red para ver si han cambiado los objetivos y las circunstancias de la red. A continuación se presenta una hoja de trabajo donde se puede llevar de forma ordenada nuestras ideas conforme a los lineamientos de seguridad a seguir.

La columna “Número de recursos de red” es un número de red de identificación interna de los recursos que van a ser protegidos.

La columna “Nombre del recurso de red” es la descripción en lenguaje común de los recursos. La importancia del recurso puede estar en una escala numérica del 0 al 10, o en expresiones de lenguaje natural como bajo, medio, alto, etc.

La columna “Tipo de usuario del que hay que protegerse el recurso” puede tener designaciones como interno, externo, invitado o nombres de grupos como usuarios de contabilidad, asistentes corporativos, etc.

La columna “Flexibilidad de una amenaza” puede estar en una escala numérica del 0 al 10, o en expresiones de lenguaje natural como baja, media, alta, etc.

La columna “Medidas que se implementarán para proteger el recurso de red” puede tener valores tales como “permisos de sistema operativo” para archivos y directorios; “pistas/alertas de auditoria” para servicios de red; “routers de selección” y “firewalls” para hosts y dispositivos de conectividad de red; y cualquier otra descripción del tipo de control de seguridad.

Hoja de trabajo para desarrollar un planteamiento de seguridad

Recursos de la fuente		Importancia del recurso	Tipo de usuario del que hay que proteger al recurso	Posibilidad de amenaza	Medidas que se implementarán para proteger al recurso de la red
Número	**Nombre**				

En general, el costo de proteger las redes de una amenaza debe ser menor que el de recuperación en caso de que se viera afectado por una amenaza de seguridad. Si no se tiene el suficiente conocimiento de lo que está protegiendo y de las fuentes de la amenaza, puede ser difícil alcanzar un nivel aceptable de seguridad.

Es importante hacer que en el diseño de la política de seguridad participe la gente adecuada, como grupos que incluyan a quienes están implicados en el control de auditoria, grupos de sistemas de información de universidades y organizaciones que manejan la seguridad física, de modo que sea más fácil la cooperación y aceptación de la política de seguridad de red.

1.3. Procedimientos

Análisis de Riesgos

En un entorno informático existen una serie de recursos (humanos, técnicos, de infraestructura, etc.) que están expuestos a diferentes tipos de riesgos: los `normales', aquellos comunes a cualquier entorno, y los excepcionales, originados por situaciones concretas que afectan o pueden afectar a parte de una organización o a toda la misma, como la inestabilidad política en un país o una región sensible a temblores. Para tratar de minimizar los efectos de un problema de seguridad se realiza lo que denominamos un análisis de riesgos, término que hace referencia al proceso necesario para responder a tres cuestiones básicas sobre nuestra seguridad:

- **¿Qué queremos proteger?**
- **¿Contra quién o de qué lo queremos proteger?**
- **¿Cómo lo queremos proteger?**

En la práctica existen dos aproximaciones para responder a estas preguntas, una cuantitativa y otra cualitativa. La primera de ellas es con diferencia la menos usada, ya que en muchos casos implica cálculos complejos o datos difíciles de estimar. Se basa en dos parámetros fundamentales: la probabilidad de que un suceso ocurra y una estimación del coste o las pérdidas en caso de que así sea; el producto de ambos términos es lo que se denomina coste anual estimado (EAC, *Estimated Annual Cost*), y aunque teóricamente es posible conocer el riesgo de cualquier evento (el EAC) y tomar decisiones en función de estos datos, en la práctica la inexactitud en la estimación o en el cálculo de parámetros hace difícil y poco realista esta aproximación.

El segundo método de análisis de riesgos es el cualitativo, de uso muy difundido en la actualidad especialmente entre las nuevas `consultoras' de seguridad (aquellas más especializadas en seguridad lógica, cortafuegos, *tests* de penetración y similares). Es mucho más sencillo e intuitivo que el anterior, ya que

ahora no entran en juego probabilidades exactas sino simplemente una estimación de pérdidas potenciales. Para ello se interrelacionan cuatro elementos principales: las amenazas, por definición siempre presentes en cualquier sistema, las vulnerabilidades, que potencian el efecto de las amenazas, el impacto asociado a una amenaza, que indica los daños sobre un activo por la materialización de dicha amenaza, y los controles o salvaguardas, contramedidas para minimizar las vulnerabilidades (controles preventivos) o el impacto (controles correctivos). Por ejemplo, una amenaza sería un hacker que queramos o no (no depende de nosotros) va a tratar de modificar nuestra página *Web* principal, el impacto sería una medida del daño que causaría si lo lograra, una vulnerabilidad sería una configuración incorrecta del servidor que ofrece las páginas, y un control la reconfiguración de dicho servidor o el incremento de su nivel de parcheo. Con estos cuatro elementos podemos obtener un indicador cualitativo del nivel de riesgo asociado a un activo determinado dentro de la organización, visto como la probabilidad de que una amenaza se materialice sobre un activo y produzca un determinado impacto.

Tras obtener mediante cualquier mecanismo los indicadores de riesgo en nuestra organización llega la hora de evaluarlos para tomar decisiones organizativas acerca de la gestión de nuestra seguridad y sus prioridades. Tenemos por una parte el *riesgo calculado*, resultante de nuestro análisis, y este riesgo calculado se ha de comparar con un cierto umbral (*umbral de riesgo*) determinado por la política de seguridad de nuestra organización; el umbral de riesgo puede ser o bien un número o bien una etiqueta de riesgo (por ejemplo, nivel de amenaza alto, impacto alto, vulnerabilidad grave, etc.), y cualquier riesgo calculado superior al umbral ha de implicar una decisión de reducción de riesgo. Si por el contrario el calculado es menor que el umbral, se habla de *riesgo residual*, y el mismo se considera asumible (no hay porqué tomar medidas para reducirlo). El concepto de asumible es diferente al de *riesgo asumido*, que denota aquellos riesgos calculados superiores al umbral pero sobre los que por cualquier razón (política, económica, etc.) se decide no tomar medidas de reducción; evidentemente, siempre hemos de huir de esta situación.

Una vez conocidos y evaluados de cualquier forma los riesgos a los que nos enfrentamos podremos definir las políticas e implementar las soluciones prácticas para minimizar sus efectos.

Identificación de Recursos

Debemos identificar todos los recursos cuya integridad pueda ser amenazada de cualquier forma, los cuales pueden ser los siguientes:

Hardware

Procesadores, tarjetas, teclados, terminales, estaciones de trabajo, ordenadores personales, impresoras, unidades de disco, líneas de comunicación, servidores, ruteadores, etc.

Software

Códigos fuente y objeto, utilidades, programas de diagnóstico, sistemas operativos, programas de comunicación, etc.

Información

En ejecución, almacenada en línea, almacenada fuera de línea, en comunicación, bases de datos, etc.

Personas

Usuarios, operadores, administradores, etc.

Accesorios

Papel, cintas, toners, etc.

Aparte del recurso debemos de considerar la visión intangible de cada uno de estos recursos (por ejemplo la capacidad para seguir trabajando sin ese recurso).

Es difícil generar estos aspectos intangibles de los recursos, ya que es algo que va a depender de cada organización, su funcionamiento, sus seguros, sus normas, etc. No obstante, siempre hemos de tener en cuenta algunos aspectos comunes: privacidad de los usuarios, imagen pública de la organización, reputación, satisfacción del personal y de los clientes, capacidad de procesamiento ante un fallo, etc.

Con los recursos correctamente identificados se debe generar una lista final, que ya incluirá todo lo que necesitamos proteger en nuestra organización.

Identificación de Amenazas

Una vez conocemos los recursos que debemos proteger es la hora de identificar las vulnerabilidades y amenazas contra ellos. Una vulnerabilidad es cualquier situación que pueda desembocar en un problema de seguridad, y una amenaza es la acción específica que aprovecha una vulnerabilidad para crear un problema de

seguridad; entre ambas existe una estrecha relación: sin vulnerabilidades no hay amenazas, y sin amenazas no hay vulnerabilidades.

Se pueden dividir las amenazas que existen sobre los sistemas informáticos en tres grandes grupos, en función del ámbito o la forma en que se pueden producir:

- **Desastres del entorno**

  Dentro de este grupo se incluyen todos los posibles problemas relacionados con la ubicación del entorno de trabajo informático o de la propia organización, así como con las personas que de una u otra forma están relacionadas con el mismo. Por ejemplo, se han de tener en cuenta desastres naturales (terremotos, inundaciones...), desastres producidos por elementos cercanos, como los cortes de fluido eléctrico, y peligros relacionados con operadores, programadores o usuarios del sistema.

- **Amenazas en el sistema**

  Bajo esta denominación se contemplan todas las vulnerabilidades de los equipos y su software que pueden acarrear amenazas a la seguridad, como fallos en el sistema operativo, medidas de protección que éste ofrece, fallos en los programas, copias de seguridad, etc.

- **Amenazas en la red**

  Cada día es menos común que una máquina trabaje aislada de todas las demás; se tiende a comunicar equipos mediante redes locales, intranets o la propia Internet, y esta interconexión acarrea nuevas - y peligrosas - amenazas a la seguridad de los equipos, peligros que hasta el momento de la conexión no se suelen tener en cuenta. Por ejemplo, es necesario analizar aspectos relativos al cifrado de los datos en tránsito por la red, a proteger una red local del resto de Internet, o a instalar sistemas de autenticación de usuarios remotos que necesitan acceder a ciertos recursos internos de la organización (como un investigador que conecta desde su casa a través de un módem).

Algo importante a la hora de analizar las amenazas a las que se enfrentan nuestros sistemas es analizar los potenciales tipos de atacantes que pueden intentar violar nuestra seguridad. Es algo normal que a la hora de hablar de atacantes todo el mundo piense en crackers, en piratas informáticos mal llamados hackers. No obstante, esto no es más que el fruto de la repercusión que en todos los medios tienen estos individuos y sus acciones; en realidad, la inmensa mayoría de problemas de seguridad vienen dados por atacantes internos a la organización afectada. En organismos universitarios, estos atacantes suelen ser los propios estudiantes (rara vez el personal), así como piratas externos a la entidad que

aprovechan la habitualmente mala protección de los sistemas universitarios para acceder a ellos y conseguir así cierto status social dentro de un grupo de piratas.

Los conocimientos de estas personas en materias de sistemas operativos, redes o seguridad informática suelen ser muy limitados, y sus actividades no suelen entrañar muchos riesgos a no ser que se utilicen nuestros equipos para atacar a otras organizaciones, en cuyo caso a los posibles problemas legales hay que sumar la mala imagen que nuestras organizaciones adquieren.

No siempre hemos de contemplar a las amenazas como actos intencionados contra nuestro sistema: muchos de los problemas pueden ser ocasionados por accidentes, desde un operador que derrama una taza de café sobre una terminal hasta un usuario que tropieza con el cable de alimentación de un servidor y lo desconecta de la línea eléctrica, pasando por temas como el borrado accidental de datos o los errores de programación. Por supuesto, tampoco tenemos que reducirnos a los accesos no autorizados al sistema: un usuario de nuestras máquinas puede intentar conseguir privilegios que no le corresponden, una persona externa a la organización puede lanzar un ataque de negación de servicio contra la misma sin necesidad de conocer ni siquiera un login y una contraseña, etc.

1.4. Normatividad

Medidas de Protección

Tras identificar todos los recursos que deseamos proteger, así como las posibles vulnerabilidades y amenazas a que nos exponemos y los potenciales atacantes que pueden intentar violar nuestra seguridad, también ver cómo proteger nuestros sistemas, sin ofrecer aún implementaciones concretas para protegerlos. Esto implica en primer lugar cuantificar los daños que cada posible vulnerabilidad puede causar teniendo en cuenta las posibilidades de que una amenaza se pueda convertir en realidad. Este cálculo puede realizarse partiendo de hechos sucedidos con anterioridad en nuestra organización, aunque por desgracia en muchos lugares no se suelen registrar los incidentes suscitados. En este caso, y también a la hora de evaluar los daños sobre recursos intangibles, existen diversas aproximaciones como el método Delphi, que básicamente consiste en preguntar a una serie de especialistas de la organización sobre el daño y las pérdidas que cierto problema puede causar; no obstante, la experiencia del administrador en materias de seguridad suele tener aquí la última palabra a la hora de evaluar los impactos de cada amenaza.

La clasificación de riesgos suele realizarse en base al nivel de importancia del daño causado y a la probabilidad aproximada de que ese daño se convierta en realidad; se trata principalmente de no gastar más dinero en una implementación para proteger un recurso de lo que vale dicho recurso o de lo que nos costaría recuperarnos de un daño en él o de su pérdida total. Por ejemplo, podemos seguir

un análisis similar en algunos aspectos al problema de la mochila: llamamos al riesgo de perder un recurso *i* (a la probabilidad de que se produzca un ataque), y le asignamos un valor de 0 a 10 (valores más altos implican más probabilidad); de la misma forma, definimos también de 0 a 10 la importancia de cada recurso, siendo 10 la importancia más alta. La evaluación del riesgo es entonces el producto de ambos valores, llamado peso o riesgo evaluado de un recurso, y medido en dinero perdido por unidad de tiempo (generalmente, por año).

De esta forma podemos utilizar hojas de trabajo en las que, para cada recurso, se muestre su nombre y el número asignado, así como los tres valores anteriores. Evidentemente, los recursos que presenten un riesgo evaluado mayor serán los que más medidas de protección deben poseer, ya que esto significa que es probable que sean atacados, y que además el ataque puede causar pérdidas importantes. Es especialmente importante un grupo de riesgos denominados inaceptables, aquellos cuyo peso supera un cierto umbral; se trata de problemas que no nos podemos permitir en nuestros sistemas, por lo que su prevención es crucial para que todo funcione correctamente.

A continuación se muestra una hoja de trabajo que puede usarse para registrar los cálculos mencionados con anterioridad.

La columna “Número de recursos de red” es un número de red de identificación interna del recurso.

La columna “Nombre del recurso de red” es una descripción en lenguaje común de los recursos.

La columna “Riesgo de los recursos de red (Ri)” puede estar en una escala numérica del 0 al 10, o en expresiones de lenguaje natural como bajo, medio, alto, etc.

La columna “Peso (importancia) del recurso (Wi)” puede estar en una escala numérica del 0 al 10, o en expresiones de lenguaje natural como bajo, medio, alto, etc. Si se usan valores numéricos en las columnas de riesgo y peso, se puede calcular el valor de la columna “Riesgo evaluado (Ri*Wi), como el producto de los valores de riesgo y peso.

El riesgo general de los recursos de la red se puede calcular mediante la siguiente fórmula:

WR= (R1*W1 + R2*W2+...+Rn*Wn) / (W1+W2+...+Wn)

Hoja de trabajo para el análisis de riesgo de seguridad de la red

Recursos de la red	Riesgo de los recursos de la red (Ri)	Peso (importancia) del recurso (Wi)	Riesgo evaluado (Ri * Wi)
Número	Nombre		

Una vez que conocemos el riesgo evaluado de cada recurso es necesario efectuar lo que se llama el análisis de costes y beneficios. Básicamente consiste en comparar el coste asociado a cada problema, con el coste de prevenir dicho problema. El cálculo de este último no suele ser complejo si conocemos las posibles medidas de prevención que tenemos a nuestra disposición: por ejemplo, para saber lo que nos cuesta prevenir los efectos de un incendio en la sala de operaciones, no tenemos más que consultar los precios de sistemas de extinción de fuego, o para saber lo que nos cuesta proteger nuestra red sólo hemos de ver los precios de productos como routers que bloqueen paquetes o cortafuegos completos. No sólo hemos de tener en cuenta el coste de cierta protección, sino también lo que nos puede suponer su implementación y su mantenimiento; en muchos casos existen soluciones gratuitas para prevenir ciertas amenazas, pero estas soluciones tienen un coste asociado relativo a la dificultad de hacerlas funcionar correctamente de una forma continua en el tiempo, por ejemplo dedicando a un empleado a su implementación y mantenimiento.

Cuando ya hemos realizado este análisis no tenemos más que presentar nuestras cuentas a los responsables de la organización (o adecuarlas al presupuesto que un departamento destina a materias de seguridad), siempre teniendo en cuenta que el gasto de proteger un recurso ante una amenaza ha de ser inferior al gasto que se produciría si la amenaza se convirtiera en realidad. Hemos de tener siempre presente que los riesgos se pueden minimizar, pero nunca eliminarlos completamente, por lo que será recomendable planificar no sólo la prevención ante de un problema sino también la recuperación si el mismo se produce; se suele hablar de medidas preventivas (aquellas que se toman para prevenir un problema) y medidas correctivas (aquellas que se toman cuando el daño se produce, para minimizar sus efectos).

Estrategias de Respuesta

Cada vez que se viola la política de seguridad, el sistema está sujeto a amenazas. Si no se producen cambios en la seguridad de la red cuando ésta sea violada, entonces debe modificarse la política de seguridad para eliminar aquellos elementos que no sean seguros.

¿Qué hacer cuando nuestra política de seguridad ha sido violada?

La respuesta a esta pregunta depende completamente del tipo de violación que se haya producido, de su gravedad, de quién la haya provocado, de su intención. Si se trata de accidentes o de problemas poco importantes suele ser suficiente con una llamada de atención verbal o una advertencia; si ha sido un hecho provocado, quizás es conveniente emprender acciones algo más convincentes, como la clausura de las cuentas de forma temporal o pequeñas sanciones administrativas. En el caso de problemas graves que hayan sido intencionados interesará emprender acciones más duras, como cargos legales o sanciones administrativas firmes (por ejemplo, la expulsión de una universidad).

Una gran limitación que nos va a afectar mucho es la situación de la persona o personas causantes de la violación con respecto a la organización que la ha sufrido. En estos casos se

suele diferenciar entre usuarios internos o locales, que son aquellos pertenecientes a la propia organización, y externos, los que no están relacionados directamente con la misma; las diferencias entre ellos son los límites de red, los administrativos, los legales o los políticos. Evidentemente es mucho más fácil buscar responsabilidades ante una violación de la seguridad entre los usuarios internos, ya sea contra la propia organización o contra otra, pero utilizando los recursos de la nuestra; cuando estos casos se dan en redes de I+D, generalmente ni siquiera es necesario llevar el caso ante la justicia, basta con la aplicación de ciertas normas sobre el usuario problemático (desde una sanción hasta la expulsión o despido de la organización).

Existen dos estrategias de respuesta ante un incidente de seguridad:

- Proteger.
- Proceder.

Perseguir y Procesar

La primera de estas estrategias, proteger y proceder, se suele aplicar cuando la organización es muy vulnerable o el nivel de los atacantes es elevado; la filosofía es proteger de manera inmediata la red y los sistemas y restaurar su estado normal, de forma que los usuarios puedan seguir trabajando normalmente. Seguramente será necesario interferir de forma activa las acciones del intruso para evitar más accesos, y analizar el daño causado. La principal desventaja de esta estrategia es que el atacante se da cuenta rápidamente de que ha sido descubierto, y puede emprender acciones para ser identificado, lo que incluso conduce al borrado de *logs* o de sistemas de ficheros completos; incluso puede cambiar su estrategia de ataque a un nuevo método, y seguir comprometiendo al sistema. Sin embargo, esta estrategia también presenta una parte positiva: el bajo nivel de conocimientos de los atacantes en sistemas habituales hace que en muchas ocasiones se limiten a abandonar su ataque y dedicarse a probar suerte con otros sistemas menos protegidos en otras organizaciones.

La segunda estrategia de respuesta, perseguir y procesar, adopta la filosofía de permitir al atacante proseguir sus actividades, pero de forma controlada y observada por los administradores, de la forma más discreta posible. Con esto, se intentan guardar pruebas para ser utilizadas en la segunda parte de la estrategia, la de acusación y procesamiento del atacante (ya sea ante la justicia o ante los responsables de la organización, si se trata de usuarios internos). Evidentemente corremos el peligro de que el intruso descubra su monitorización y destruya completamente el sistema, así como que nuestros resultados no se tengan en cuenta ante un tribunal debido a las artimañas legales que algunos abogados aprovechan; la parte positiva de esta estrategia es, aparte de la recolección de pruebas, que permite a los responsables conocer las actividades del atacante, qué vulnerabilidades de nuestra organización ha aprovechado para atacarla, cómo se comporta una vez dentro, etc. De esta forma podemos aprovechar el ataque para reforzar los puntos débiles de nuestros sistemas.

A nadie se le escapan los enormes peligros que entraña el permitir a un atacante proseguir con sus actividades dentro de las máquinas; por muy controladas que estén, en cualquier momento casi nada puede evitar que la persona se sienta vigilada, se ponga nerviosa y destruya completamente nuestros datos. Una forma de monitorizar sus actividades sin comprometer excesivamente nuestra integridad es mediante un proceso denominado jailing o encarcelamiento: la idea es construir un sistema que simule al real, pero donde no se encuentren datos importantes, y que permita observar al atacante sin poner en peligro los sistemas reales. Para ello se utiliza una máquina, denominada **sistema de sacrificio**, que es donde el atacante realmente trabaja, y un segundo sistema, denominado **de observación**, conectado al anterior y que permite analizar todo lo que esa persona está llevando a cabo. De esta forma conseguimos que el atacante piense que su intrusión ha tenido éxito y continué con ella mientras lo monitorizamos y recopilamos pruebas para presentar en una posible demanda o acusación. Si deseamos construir una cárcel es necesario que dispongamos de unos conocimientos medios o elevados de programación de sistemas; utilidades como `chroot()` nos pueden ser de gran ayuda, así como software de simulación como Deception Tookit *(*DTK*)*, que simula el éxito de un ataque ante el pirata que lo lanza, pero que realmente nos está informando del intento de violación producido.

1.5 Respaldos

Es común que la gente pierda grandes cantidades de datos porque no cuentan con un sistema de respaldo. Es por ello que los respaldos son una parte esencial de la seguridad y protección de los datos.

Existen muchas razones por las cuales las personas no realizan respaldos:

- Desconocen su importancia, sólo saben de ellas cuando los datos los han perdido.
- No consideran los respaldos como una tarea cotidiana que se deba realizar.
- Consideran tiempo perdido aquel espacio dedicado a la creación de respaldos.

Por lo general no se piensa que una computadora pueda fallar en su aspecto de software o de hardware. La recuperación de una falla en disco duro puede ser un evento difícil, aún cuando existe un respaldo, y de no existirlo la situación es mucho peor. El costo para recuperar los datos perdidos es muy alto, aún para los pequeños negocios esto puede significar una pérdida de miles de pesos.

Para asegurar que los respaldos están trabajando adecuadamente y que estamos protegidos sin invertir mucho tiempo, debemos determinar que archivos se quieren respaldar y que tan seguido se debe de llevar acabo el respaldo. Algunos archivos necesitaran respaldarse más seguido que otros.

Riesgos

Es probable que en algún momento nos enfrentemos a la necesidad de acceder a un respaldo de datos o programas a causa de una falla. La mayoría de la gente piensa en una falla de hardware para que exista perdida de la información. Los medios de almacenamiento pueden fallar (lo hacen) y generan así la pérdida de datos parcial o total. Existen otras maneras en las que los datos corren riesgo, algunas de ellas son:

- Error del software: La operación incorrecta de una aplicación puede llegar a eliminar archivos que contengan datos de nuestro interés.

- Corrupción del sistema de archivos: La presencia de bugs en drivers relacionados con el medio de almacenamiento o el subsistema asociado a este último pueden generar problemas en la lectura y/o escritura de datos desde y hacia el medio ocasionando pérdidas de información.

- Error de Hardware: Los dispositivos de almacenamiento de mayor uso en la actualidad están formados por componentes electrónicos y mecánicos. Ambas clases de componentes pueden fallar. Las fallas pueden ser de todo tipo.

- Borrado accidental: Ejemplos de este caso son: La eliminación de archivos, la eliminación de una partición o unidad lógica en forma accidental, el formateo accidental del medio, etc. Es una de las principales causas de pérdida de datos.

- Infección por virus: Los virus y el ingreso de usuarios con malas intenciones a los datos pueden causar grandes pérdidas de información.

- Desastre Natural: No nos olvidemos de la posibilidad de pérdida física del medio de almacenamiento que contiene los datos, ya sea por un siniestro que ocasione daños materiales por una pérdida accidental.

- Robo de equipo: El acceso no controlado a los equipos puede ocasionar robo parcial o total del equipo, lo cuál también significa pérdida total de la información.

Respaldo de la Información

Considerando los posibles escenarios que se nos pueden presentar de pérdida de la información, debemos considerar algunos aspectos para realizar el resguardo de los datos los cuales son:

- El resguardo debe realizarse en un medio de almacenamiento diferente al que estamos respaldando, ya que de nada nos sirve respaldar la información en un

mismo disco si puede presentarse algún error de hardware como los que se mencionaron anteriormente.

- Los respaldos de la información no deben guardarse en el mismo lugar donde se encuentran los datos, ya que de nada nos sirve guardarlos ahí si ocurre un siniestro o desastre natural y destruye el almacén original de datos junto con los respaldos.
- Los respaldos de los datos deben almacenarse de forma clara y ordenada, con la finalidad de facilitar la recuperación en caso de algún siniestro.

- Cuando se realice el respaldo de los datos se debe verificar que los datos almacenados son los correctos y se puede acceder a ellos.

- Debe almacenarse el medio de respaldo en un lugar seco y fresco libre de humedad, así como también no exponerlo al calor ni a los campos electromagnéticos.

Medios de Respaldo

La selección del medio destino de un respaldo depende de algunos aspectos a considerar que a continuación describimos:

- **Volumen de la información a respaldar**: Es uno de los principales factores que debemos considerar al seleccionar el medio destino. Por ejemplo: Si tenemos que realizar un respaldo de 4 GB, el medio adecuado que debemos de utilizar deberá ser uno que se acerque en capacidad a los datos que deseamos respaldar, como discos Jazz o cintas magnéticas.

- **Rendimiento**: Debemos considerar el tiempo de acceso a los datos y la velocidad de transferencia tanto para la escritura como para la lectura en el medio de almacenamiento destino.

- **Importancia de la información**: Si la información que estamos respaldando es crítica y debe estar disponible siempre, es necesario contar con un medio de respaldo que permita su recuperación en poco tiempo, no importa el costo del dispositivo o medio de almacenamiento.

- **Costo por MB del medio**: En la mayoría de los casos una solución óptima puede ser demasiado costosa, por lo cual habrá que buscar el mejor balance del costo/beneficio.

Teniendo en cuenta estas consideraciones, deberemos seleccionar el medio destino. A continuación, mencionaremos algunas opciones:

- Diskettes
  Discos Zip o de alta capacidad (120 MB)

- Discos Jazz
- CD-RW
- Discos Magnéticos-ópticos/Re-escribibles.
- Cartuchos de Cinta.
- Discos Rígidos.
- Flash-RAM

Tipos de Respaldos

Podemos crear diferentes tipos de respaldos sobre datos con los que contamos, para entender mejor los diferentes tipos de respaldos, es necesario tomar en cuenta los siguientes aspectos:

- Capacidad: Volumen de información que se respaldará.
- Automatización: Grado mediante el cual se puede realizar el respaldo sin que exista intervención del usuario.
- Expansión: Grado con el cuál se pueda incrementar el tamaño del respaldo.
- Confiabilidad: Que tan confiable es la información respaldada.
- Simplicidad: Facilidad de uso del método de respaldo.
- Universalidad: Que tan común es el hardware y software requerido.
- Rendimiento: Tiempo que toma llevar a cabo el respaldo.

La clasificación de tipos de respaldos se puede llevar mediante 2 formas:

1. Respaldo de Información.
2. Duplicados de Información en línea (Implementación RAID).

Es común que en algunos casos, solamente necesitemos efectuar un respaldo de nuestra información personal, pero en otras situaciones si es necesario que definamos una estrategia para realizar el respaldo de la información. Una de las dudas que nos pueden surgir al hacer la planeación de la estrategia al respecto es la frecuencia con la que se hacen los respaldos y cuál es la información a respaldar. Tomando en cuenta estos dos factores, surgen tres diferentes subtipos de respaldo que podemos combinar en nuestra estrategia:

- **Completo**. Se crea una copia de todas las carpetas y archivos que seleccionemos. Es ideal para crear la primera copia de todo el contenido de una unidad o bien de sus archivos de datos solamente.

Ventajas:
- Todos los archivos seleccionados forman parte del respaldo.
- Para restaurar uno o más archivos, se los toma directamente de este respaldo.

Desventajas:
- Cada vez que se realiza el respaldo se tiene que hacer sobre el conjunto total de datos a respaldar, lo que ocasiona que requiera demasiado espacio de almacenamiento.

- **Incremental**: Esta clase de respaldo, como su nombre lo indica, solamente genera una copia de respaldo con todos aquellos archivos que hayan sido modificados (o aparentemente haberlo sido debido a cambios en su fecha de modificación) os es hayan creado desde el último respaldo realizado, ya sea este último incremental o completo. Si se utiliza por primera vez en una unidad en vez de un respaldo completo, se comportará como este último, pero en los respaldos siguientes, irá copiando solamente lo nuevo o lo modificado.

  Ventajas:
  - Es mucho más rápido que el uso de sucesivos respaldos completos.
  - Requiere menor cantidad de espacio en el medio destino que sucesivos respaldos completos.
  - Se pueden ir manteniendo diferentes versiones de los mismos archivos en cada uno de los respaldos incrementales, con lo que se podría restaurar la versión deseada.

  Desventajas:
  - Se pueden estar copiando archivos cuyo contenido no haya sido modificado, ya que compara las fechas de modificación, y se puede haber guardado sin que se hayan efectuado cambios en su contenido.
  - Para restaurar determinados archivos o inclusive, todos, es necesario tener todos los medios de los respaldos incrementales que se hayan efectuado desde el último respaldo completo o primer respaldo incremental.
  - Como consecuencia de esta búsqueda por varios respaldos, la restauración de unos pocos archivos toma mucho más tiempo.

- **Diferenciales**: Es similar al incremental, la única diferencia es que compara el contenido de los archivos a la hora de determinar cuáles se modificaron de manera tal que solamente copia aquéllos que hayan cambiado realmente y no se deja engañar por las fechas de modificación de los mismos.

Ventajas:
- Todas las del backup incremental, pero requieren aún menor espacio en el medio de destino.

Desventajas:
- Todas las del backup incremental, menos la última.
- No todas las herramientas de respaldo dan soporte a esta clase.

Estos tipos de respaldo se pueden combinar entre sí para definir una estrategia tomando en cuenta factores como la frecuencia con la que se debe realizar el respaldo y volumen de datos a respaldar.

Respaldo	Archivos en Respaldo	Ventajas	Desventajas
Completo ("Full")	Todos	Con este respaldo, únicamente, es posible recuperar toda la información	Tiempo de ejecución
Incremental	Archivos con modificaciones. (Aquellos que hayan cambiado desde el último Respaldo completo)	Velocidad	Requiere del último Respaldo completo y de todos los respaldos de Incremento que se siguieron para recuperar el sistema
Diferencial	Archivos con modificaciones. (Aquellos que hayan cambiado desde el último Respaldo completo)	Sólo requiere del último Respaldo Completo y del último Respaldo Diferncial.	Ocupa mayor espacio en disco comparado con respaldos de Incremento.

Duplicado de Información en Línea (RAID)

RAID (Redundant Array of Inexpensive Disk o Matriz Redundante de Discos de Bajo Costo):
El método RAID es una combinación de discos duros para formar una unidad lógica de discos duros en la que se almacenan los datos de forma conjunta. Es un conjunto de 2 o más discos duros que operan como grupo y logran ofrecer una forma más avanzada de respaldo ya que:

- Es posible mantener copias en línea (redundancia).
- Agiliza las operaciones del Sistema (Base de Datos).

- El sistema es capaz de recuperar información sin intervención de un Administrador.

Existen varias configuraciones de Tipo RAID, las más utilizadas son las siguientes:

- **RAID-0**: En esta configuración cada archivo es fraccionado ("Striped") y sus fracciones son colocadas en diferentes discos. No existe redundancia de datos. Los datos se desglosan en bloques que son distribuidos entre todas las unidades de discos existentes en ese arreglo. Gracias a este sistema se consiguen tasas de transferencias de lectura y escritura muy elevadas que corresponden a multiplicar la velocidad de cada disco por el número de discos usados. Este tipo de implementación sólo agiliza el proceso de lectura de archivos, pero en ningún momento proporciona algún tipo de respaldo (redundancia).

- **RAID-1**: Este nivel usa un sistema de distribución de bloques con un disco de comprobación. Se duplica cada disco del conjunto, aumentando así la fiabilidad y la disponibilidad. La velocidad de acceso para lectura es muy rápido ya que el controlador selecciona el disco con el mejor tiempo de búsqueda. En orden ascendente, este es el primer tipo de RAID que otorga cierto nivel de respaldo; cada vez que se vaya a guardar un archivo en el sistema éste se copiara integro a DOS discos (en línea), es por esto que RAID-1 también es llamado en espejo. Además de proporcionar un respaldo en caliente ("hot") en dado caso de fallar algún disco del grupo, RAID-1 también agiliza la lectura de archivos (si se encuentran ocupadas las cabezas de un disco "I/O") ya que otro archivo puede ser leído del otro disco y no requiere esperar a finalizar el "I/O" del primer disco.

- **RAID-3**: Reduce la cantidad de información de paridad requerida, ya que una sola unidad de discos detecta el error, mientras que los controladores incorporados en los discos determinan que unidad ha causado el error. Esta configuración al igual que RAID-0 divide la información de todos los archivos ("Striping") en varios discos, pero ofrece un nivel de respaldo que RAID-0 no ofrece. En RAID-0 si falla un disco del grupo, la información no puede ser recuperada fácilmente, ya que cada disco del grupo contiene una parte del archivo, sin embargo RAID-3 opera con un disco llamado "de paridad". Este "disco de paridad" guarda fragmentos de los archivos necesarios para recuperar toda la información, con esto, es posible reproducir el archivo que se perdió a partir de esta información de paridad.

- **RAID-5**: El problema que presenta RAID-3 es que el "disco de paridad" es un punto crítico en el sistema; ¿Qué ocurre si falla el disco de paridad? En lugar de una unidad de paridad, los datos codificados se distribuyen por todo el conjunto de discos. Para resolver el problema de paridad, RAID-5, no solo distribuye todos los archivos en un grupo de discos ("Striping"), sino también la información de paridad es guardada en todos los discos del sistema.

1.6 Auditorias

¿Qué es un sitio?

Cualquier organización (militar, gubernamental, comercial, académica, etc.) que posea recursos relativos a redes y computadoras.

¿Qué es un usuario?

Cualquier persona que hace uso de alguno de los recursos de cómputo con los que cuenta una organización.

¿Qué es un administrador?

El responsable de mantener en operación continúa los recursos de cómputo con los que cuenta un sitio

¿Qué es seguridad en cómputo?

Un conjunto de recursos destinados a lograr que los activos de una organización sean *confidenciales*, *íntegros*, *consistentes* y *disponibles* a sus usuarios, *autentificados* por mecanismos de *control de acceso* y sujetos a ***auditoria***.

- ***Confidencial***: la información debe ser leída por su propietario o por alguien **explícitamente** autorizado para hacerlo.

- ***Íntegro***: la información no debe ser borrada ni modificada por alguien que carezca de autorización para hacerlo.
- ***Consistente***: el sistema, al igual que los datos, debe comportarse como uno espera que lo haga.

- ***Disponible***: la información debe estar siempre disponible en el lugar y cantidad de tiempo requeridos.
- ***Autentificado***: únicamente deben ingresar al sistema personas autorizadas, siempre y cuando comprueben que son usuarios legítimos.

- ***Control de acceso***: debe conocerse en todo momento quién entra al sistema y de dónde procede.

- ***Auditoria***: deben conocerse en cada momento las actividades de los usuarios dentro del sistema.

¿Qué es un incidente?

Un evento que pone en riesgo la seguridad de un sistema de cómputo.

¿Qué es un ataque?

Un incidente cuyo objetivo es causar daño a un sistema, robar información del mismo, o utilizar sus recursos de forma no autorizada.

¿Qué es un firewall?

Un dispositivo de hardware y software que actúa como una barrera protectora entre una red privada y el mundo exterior; se usa para proteger el acceso a los recursos internos desde el exterior, así como para controlar los recursos externos que son accedidos desde la red privada.

¿Qué son las herramientas de seguridad?

Programas que nos permiten incrementar la fiabilidad de un sistema de cómputo. Existe una gran variedad de ellas, casi todas de libre distribución. Algunos ejemplos de herramientas de seguridad a considerar para implementar un esquema de seguridad son:

- Para el manejo de contraseñas: anipasswd, passwd+, crack, John The Ripper, S/Key.
- Para el manejo de autenticación: Kerberos, SecureRPC.
- Para el monitoreo de redes: Satan, ISS.
- Para auditoría interna: COPS, Tiger, Tripwire.
- Para control de acceso: TCP-Wrapper, PortSentry.

¿Qué es un incidente?

Un evento que pone en riesgo la seguridad de un sistema de cómputo.

¿Qué es un ataque?

Un incidente cuyo objetivo es causar daño a un sistema, robar información del mismo, o utilizar sus recursos de forma no autorizada.

¿Qué es un firewall?

Un dispositivo de hardware y software que actúa como una barrera protectora entre una red privada y el mundo exterior; se usa para proteger el acceso a los recursos internos desde el exterior, así como para controlar los recursos externos que son accedidos desde la red privada.

¿Qué son las herramientas de seguridad?

Programas que nos permiten incrementar la fiabilidad de un sistema de cómputo. Existe una gran variedad de ellas, casi todas de libre distribución. Algunos ejemplos de herramientas de seguridad a considerar para implementar un esquema de seguridad son:

- Para el manejo de contraseñas: anipasswd, passwd+, crack, John The Ripper, S/Key.
- Para el manejo de autenticación: Kerberos, SecureRPC.
- Para el monitoreo de redes: Satan, ISS.
- Para auditoría interna: COPS, Tiger, Tripwire.
- Para control de acceso: TCP-Wrapper, PortSentry.
- Mantener las computadoras alejadas del fuego, humo, polvo y temperaturas extremas.
- Colocarlas fuera del alcance de rayos, vibraciones, insectos, ruido eléctrico (balastras, equipo industrial, etc.), agua, etc.
- Mantener las computadoras alejadas de comida y bebida.
- No desatender las sesiones de trabajo activas.

Políticas y Procedimientos de Seguridad

Políticas de seguridad son los documentos que describen, principalmente, la forma adecuada de uso de los recursos de un sistema de cómputo, las responsabilidades y derechos que tanto usuarios como administradores tienen y qué hacer ante un incidente de seguridad.

Mientras las políticas indican el “qué”, los procedimientos indican el “cómo”. Los procedimientos son los que nos permiten llevar a cabo las políticas. Ejemplos que requieren la creación de un procedimiento son:

- Otorgar una cuenta.
- Dar de alta a un usuario.
- Conectar una computadora a la red.
- Localizar una computadora.
- Actualizar el sistema operativo.
- Instalar software localmente o vía red.
- Actualizar software crítico.
- Exportar sistemas de archivos.
- Respaldar y restaurar información.
- Manejar un incidente de seguridad.

Para que esto sirva de algo, las políticas deben ser:

- Apoyadas por los directivos.
- Únicas.

- Claras (explícitas).
- Concisas (breves).
- Bien estructurado.
- Servir de referencia.
- Escritas.
- Revisadas por abogados.
- Dadas a conocer.
- Entendidas por los usuarios.
- Firmadas por los usuarios.
- Mantenerse actualizadas.

Las políticas son parte fundamental de cualquier esquema de seguridad eficiente. Como administradores, nos aminoran los riesgos, y nos permiten actuar de manera rápida y acertada en caso de haber una emergencia de cómputo. Como usuarios, nos indican la manera adecuada de usar un sistema, indicando lo que puede hacerse y lo que debe evitarse en un sistema de cómputo, contribuyendo a que no seamos "malos vecinos" de la red sin saberlo. El tener un esquema de políticas facilita grandemente la introducción de nuevo personal, teniendo ya una base escrita y clara para capacitación; dan una imagen profesional a la organización y facilitan una auditoría.

Los principales puntos que deben contener las políticas de seguridad son:

- Ámbito de aplicación.
- Análisis de riesgos.
- Enunciados de políticas.
- Sanciones.
- Sección de uso ético de los recursos de cómputo.
- Sección de procedimientos para el manejo de incidentes.

Al diseñar un esquema de políticas de seguridad, conviene que dividamos nuestro trabajo en varias diferentes políticas específicas a un campo - cuentas, contraseñas, control de acceso, uso adecuado, respaldos, correo electrónico, contabilidad del sistema, seguridad física, personal, etc.

- ***Políticas de cuentas***: Establecen qué es una cuenta de usuario de un sistema de cómputo, cómo está conformada, a quién puede serle otorgada, quién es el encargado de asignarlas, cómo deben ser creadas y comunicadas. **Ejemplos**:

    - Las cuentas deben ser otorgadas exclusivamente a usuarios legítimos.

    - Una cuenta deberá estar conformada por un nombre de usuario y su respectiva contraseña.

  - El nombre de usuario de una cuenta deberá estar conformado por la primera letra de su nombre y su apellido paterno.
- ***Políticas de contraseñas***: Son una de las políticas más importantes, ya que por lo general, las contraseñas constituyen la primera y tal vez única manera de autenticación y, por tanto, la única línea de defensa contra ataques. Éstas establecen quién asignará la contraseña, qué longitud debe tener, a qué formato deberá apegarse, cómo será comunicada, etc. **Ejemplos**:

  - La longitud de una contraseña deberá siempre ser verificada de manera automática al ser construida por el usuario. Todas las contraseñas deberán contar con al menos siete caracteres.

  - Todas las contraseñas elegidas por los usuarios deben ser difíciles de adivinar. No deben ser utilizadas palabras que aparezcan en el diccionario, secuencias conocidas de caracteres, datos personales ni acrónimos.

  - Está prohibido que los usuarios construyan contraseñas compuestas de algunos caracteres constantes y otros que cambien de manera predecible y sean fáciles de adivinar.

  - Los usuarios no deben construir contraseñas idénticas o muy parecidas a contraseñas anteriores.

- ***Políticas de control de acceso***: Especifican cómo deben los usuarios acceder al sistema, desde dónde y de qué manera deben autentificarse. **Ejemplos**:

  - Todos los usuarios deberán acceder al sistema utilizando algún programa que permita una comunicación segura y encriptada.

  - Está prohibido acceder al sistema con una cuenta diferente de la propia, aún con la autorización del dueño de dicha cuenta.

  - Si un usuario está fuera del sitio de trabajo, debe conectarse a una máquina pública del sitio y, únicamente desde ésta, hacer la conexión a la computadora deseada.

  - Al momento de ingresar al sistema, cada usuario deberá ser notificado de la fecha, hora y dirección desde la que se conectó al sistema por última vez, lo cual permitirá detectar fácilmente el uso no autorizado del sistema.

- ***Políticas de uso adecuado***: Especifican lo que se considera un uso adecuado o inadecuado del sistema por parte de los usuarios, así como lo que está permitido y lo que está prohibido dentro del sistema de cómputo.

Antes de diseñar el esquema de políticas de uso adecuado, conviene hacer las siguientes preguntas:

- ¿Se permite irrumpir en cuentas ajenas?
- ¿Se permite adivinar contraseñas?
- ¿Se permite interrumpir el servicio?
- ¿Puede leerse un archivo ajeno cuyos permisos ante el sistema incluyen el de lectura para todos?
- ¿Puede modificarse un archivo ajeno cuyos permisos ante el sistema incluyen el de escritura para todos?
- ¿Pueden los usuarios compartir sus cuentas?
- ¿Puede copiarse el software que no lo permita en su licencia?
- ¿Puede obtenerse una "licencia para hackear"?

La respuesta a **todas** estas preguntas debe ser **negativa**.

Existen dos enfoques: ***permisivo*** (todo lo que no esté explícitamente prohibido está permitido) y ***paranoico*** (todo lo que no esté explícitamente permitido está prohibido). Cuál de estas elegir dependerá del tipo de organización y el nivel de seguridad que esta requiera.

Tras haber aclarado estos últimos puntos, podemos proceder a algunos ejemplos:

- Está terminantemente prohibido ejecutar programas que intenten adivinar las contraseñas alojadas en las tablas de usuarios de máquinas locales o remotas.

- La cuenta de un usuario es personal e intransferible, por lo cual no se permite que este comparta su cuenta ni su contraseña con persona alguna, aún si ésta acredita la confianza del usuario.

- Está estrictamente prohibido hacer uso de herramientas propias de delincuentes informáticos, tales como programas que rastrean vulnerabilidades en sistemas de cómputo propio o ajeno.

- Está estrictamente prohibido hacer uso de programas que explotan alguna vulnerabilidad de un sistema para proporcionar privilegios no otorgados explícitamente por el administrador.

- No se permite bajo ninguna circunstancia el uso de cualquiera de las computadoras con propósitos de ocio o lucro.

- ***Políticas de respaldos***: Especifican qué información debe respaldarse, con qué periodicidad, qué medios de respaldo utilizar, cómo deberá ser restaurada la información, dónde deberán almacenarse los respaldos, etc.**Ejemplos:**

- El administrador del sistema es el responsable de realizar respaldos de la información periódicamente Cada treinta días deberá efectuarse un respaldo completo del sistema y cada día deberán ser respaldados todos los archivos que fueron modificados o creados.
- La información respaldada deberá ser almacenada en un lugar seguro y distante del sitio de trabajo.

- Deberá mantenerse siempre una versión reciente impresa de los archivos más importantes del sistema.

- En el momento en que la información respaldada deje de ser útil a la organización, dicha información deberá ser borrada antes de deshacerse del medio.

- ***Políticas de correo electrónico***: Establece tanto el uso adecuado como inadecuado del servicio de correo electrónico, los derechos y obligaciones que el usuario debe hacer valer y cumplir al respecto. **Ejemplos:**

  - El usuario es la única persona autorizada para leer su propio correo, a menos que él mismo autorice explícitamente a otra persona para hacerlo, o bien, que su cuenta esté involucrada en un incidente de seguridad de cómputo.

  - Está estrictamente prohibido usar la cuenta de correo electrónico proporcionada por la organización para propósitos ajenos a sus actividades laborales.

  - No se permite el uso de la cuenta de correo electrónico para subscribirse a listas electrónicas de discusión de interés personal El usuario deberá limitarse a estar subscrito a las listas indicadas y aprobadas por la organización.

- ***Políticas de contabilidad del sistema*:** Establecen los lineamientos bajo los cuales pueden ser monitoreadas las actividades de los usuarios del sistema de cómputo, así como la manera en que debe manejarse la contabilidad del sistema y el propósito de la misma. **Ejemplos**:

  - Deberán ser registrados en bitácoras todos los comandos emitidos por todos los usuarios del sistema, para propósitos de contabilidad.

  - Cada semana deberá hacerse el corte de contabilidad del sistema, cifrándose y respaldándose la información generada en un dispositivo de almacenamiento permanente.

Análisis de Riesgos

Un análisis de riesgo es el resultado de responder a las siguientes preguntas:

- ***¿Qué quiero proteger?*** Mis recursos: Personal, información, hardware, software, documentación, consumibles, etc.
- ***¿De quién necesito protegerlo?*** De cualquiera que constituya una amenaza, en cualquiera de estos rubros:
  - ***Acceso no autorizado*:** Utilizar recurso de cómputo sin previa autorización.
  - ***Daño a la información*:** Modificación o eliminación de la información en el sistema.
  - ***Robo de información***: Acceso a cierta información sin previa autorización.
  - ***Divulgación de la información***: Públicar detalles del sistema, como podrían ser las contraseñas, secretos, investigaciones, etc.
  - ***Negación del servicio***: Obligar al sistema a negar recursos a usuarios legítimos.
- ***¿Qué tantos recursos estoy dispuesto a invertir?***
- ***¿Cómo puedo/debo protegerlo?***

Estos dos últimos puntos, obviamente, dependerán por completo de cada caso, por lo cual no los desarrollamos.

Metodología del Desarrollo

Un esquema de políticas de seguridad debe llevar ciertos pasos, para garantizar su funcionalidad y permanencia en la institución. Nuestra propuesta es seguir los pasos que detallamos a continuación:

- ***Preparación***: la recopilación de todo tipo de material relacionado con cuestiones de seguridad en la organización: Manuales de procedimientos, planes de contingencia, cartas compromiso, etc.
- ***Redacción***: escribir las políticas de una manera clara, concisa y estructurada. Requiere de la labor de un equipo en el que participen abogados, directivos, usuarios y administradores.

- ***Edición*****:** reproducir las políticas de manera formal para ser sometidas a revisión y aprobación.

- ***Aprobación***: probablemente, la parte más difícil del proceso, puesto que es común que la gente afectada por las políticas se muestre renuente a aceptarlas. En esta etapa es fundamental contar con el apoyo de los directivos.
- ***Difusión*****:** dar a conocer las políticas a todo el personal de la organización mediante proyecciones de video, páginas Web, correo electrónico, cartas compromiso, memos, *banners*, etc.

- ***Revisión***: las políticas son sometidas a revisión por un comité, que discutirá los comentarios emitidos por las personas involucradas.

- ***Aplicación*****:** es peor tener políticas y no aplicarlas que carecer de ellas. Una política que no puede implementarse o hacerse cumplir, no tiene ninguna utilidad. Debe predicarse con el ejemplo.

- ***Actualización*****:** en el momento requerido, las políticas deberán ser revisadas y actualizadas, respondiendo a los cambios en las circunstancias. El momento ideal es justo después de que ocurra un incidente de seguridad.

Antes de comenzar a desarrollar las políticas de seguridad, tenemos que preguntarnos:

- *¿Quién debe poder usar los recursos?* - Sólo personal autorizado: Estudiantes, profesores, usuarios externos, investigadores, etc.

- *¿Qué constituye un uso adecuado de los recursos?* - ¿Qué actividades están permitidas? ¿Qué actividades están prohibidas?

- *¿Quién debe poder proporcionar acceso al sistema?* - Si no se tiene control sobre quién está dando acceso al sistema, tampoco se podrá tener control sobre quién lo utiliza.

- *¿Quién debe tener privilegios de administrador?* - Debe utilizarse el principio del mínimo privilegio: Proporcionar sólo los privilegios suficientes para ejecutar las tareas necesarias.

- *¿Cuáles son los derechos y responsabilidades de los usuarios?* - ¿Existen restricciones en cuanto al consumo de recursos? ¿Cuáles son? ¿Qué constituye un abuso en términos de desempeño del sistema? ¿Qué requerimientos deben cumplir las contraseñas de los usuarios? ¿Debe el usuario hacerse responsable de sus propios respaldos de información? ¿Puede el usuario divulgar información propietaria? ¿Qué tan privado es el correo electrónico de los usuarios? ¿Pueden los usuarios subscribirse a *cualquier* lista de discusión o grupo de noticias? ¿Puede un usuario falsificar correo?

¿Puede un usuario intentar adivinar contraseñas de sistemas, sean remotos o locales? ¿Puede un usuario llevar a cabo actividades ilegales (fraude, difamación, etc.)?

- *¿Cuáles son los derechos y responsabilidades de los administradores?* - ¿Pueden monitorear o leer los archivos de los usuarios? ¿Tienen derecho a examinar el tráfico de una máquina en específico? ¿Tienen derecho a examinar el tráfico de toda la red? ¿A qué grado pueden hacer uso de sus privilegios? ¿Qué tanto deberán respetar la privacía de los usuarios? ¿Cómo deben resguardar su contraseña? ¿Cómo deben manejar información sensitiva?

- *¿Cómo debe manejarse la información sensitiva?* - Debe evitarse que los usuarios almacenen información valiosa en sistemas poco seguros.

Ahora, es necesario tomar en cuenta diferentes escenarios, para contemplarlos a todos al crear nuestras políticas. Tomemos los siguientes ejemplos:

Tarde o temprano, todas las políticas serán violadas. ¿Qué puede llevar a que una política sea violada?

- Negligencia.
- Error accidental.
- Desconocimiento de la misma.
- Falta de entendimiento de la misma.

¿Qué debemos hacer si una política es violada?

- Investigar quién llevó a cabo esta violación.
- Investigar cómo y por qué ocurrió esta violación.
- Aplicar una acción correctiva (disciplinaria).

¿Qué sucede si un usuario local viola las políticas de un sitio remoto?

- Debe haber acciones a seguir bien definidas con respecto a los usuarios locales.
- Debe estarse bien protegido en contra de posibles acciones desde el sitio remoto.

¿Cómo reaccionar ante un incidente de seguridad? Hay dos estrategias básicas:

Proteger y Perseguir.

- Su principal objetivo es proteger y preservar los servicios del sitio, y restablecerlos lo más rápido posible.

- Se realizan acciones drásticas, tales como dar de baja los servicios, desconectar el sistema de la red, apagarlo, etc.

- Lo utilizamos cuando:

  - Los activos están bien protegidos.
  - Se correo un gran riesgo debido a la intrusión.
  - No existe la posibilidad o disposición para enjuiciar.
  - Se desconoce la base del intruso.
  - Los usuarios son poco sofisticados y su trabajo es vulnerable.
  - Los recursos de los usuarios son minados.

Perseguir y enjuiciar.

- Su objetivo principal es permitir que los intrusos continúen con sus actividades en el sistema hasta que pueda identificarse a los responsables.

  - Lo utilizamos cuando:

    - Los recursos están bien protegidos.
    - Se dispone de respaldos confiables.
    - El riesgo para los activos es mayor que el daño de esta y futuras intrusiones.
    - El ataque proviene de un sitio con el que guardamos cierta relación, y ocurre con cierta frecuencia e intensidad.
    - El sitio posee cierta atracción para los intrusos.
    - El sitio está dispuesto a correr el riesgo a que se exponen los activos al permitir que el ataque continúe.
    - Puede controlarse el acceso al intruso.
    - Se cuenta con herramientas de seguridad confiables.
    - El personal técnico conoce a profundidad el sistema operativo y sus utilerías.
    - Existe disposición para la persecución por parte de los directivos.
    - Existen leyes al respecto.
    - En el sitio existe alguien que conozca sobre cuestiones legales.

Caso de Estudio

El Área de Seguridad en Cómputo (ASC) forma parte de la Dirección de Cómputo para la Investigación (DCI) de la Dirección General de Servicios de Cómputo Académico (DGSCA) de la Universidad Nacional Autónoma de México (UNAM). Fue fundada por Diego Zambroni en 1995, a raíz de un incidente ocurrido en 1993, que comprometió la seguridad de la recién adquirida supercomputadora CRAY-YMP 4/464.

El ASC tiene como propósitos:

- Atender incidentes de seguridad ocurridos en el dominio .unam.mx.
- Promover la cultura de la seguridad en cómputo a nivel nacional e internacional.
- Evaluar productos de seguridad comerciales y académicos.
- Emitir recomendaciones concernientes al mejoramiento de la seguridad en los sistemas de cómputo.
- Administrar la seguridad de las supercomputadoras de la UNAM.
- Administrar la seguridad de las máquinas del Departamento de Supercómputo y Visualización de la DGSCA.

El ASC brinda los siguientes servicios:

- Listas de discusión.
- FTP anónimo.
- Información en HTML accesible por Web.
- Correo electrónico para reportes de incidentes y solicitud de información.
- Emisión de boletines de seguridad.
- Tutoriales en línea vía Web.
- Organización de seminarios GASU.
- Organización de DISC (avalado por la ACM).

Hasta la fecha, *no existen políticas de seguridad como tales implementadas en el ASC*, pero se está trabajando en ello. Hasta el momento, pasos que se han tomado al respecto son:

- Se cuenta con el apoyo directivo.
- El ASC como tal es un Equipo de Respuesta a Incidentes.
- Se tienen implementadas una gran variedad de herramientas de seguridad.
- Se tienen ciertos controles de seguridad física.
- Se están evaluando diversos firewalls.

La capacitación, educación y concientización están a la orden del día, con seminarios internos y seminarios por videoconferencia desde la Universidad de Purdue.

Se está trabajando en una propuesta de transición de ASC a UNAM-CERT.

¿Qué falta por hacer?

- Redactar las políticas formales de seguridad.
- Aprobarlas.
- Darlas a conocer.
- Revisarlas.
- Implementarlas.

Ejercicio 1

Identificación de vulnerabilidades en la seguridad.

Tema: Políticas y Procedimientos de Seguridad

Calificación máxima: 100

Introducción

En cualquier lugar donde el aspecto informático esté involucrado, existe la necesidad de contar con ciertas políticas y mecanismos que nos garanticen la seguridad de la información que se maneja. En la mayoría de los casos, los problemas de seguridad informática son ocasionados por errores humanos que pasan por alto situaciones o intentan reducir los procedimientos.

Estos problemas de seguridad abarcan todo el ámbito de usuarios de los sistemas, por lo que resulta sumamente importante conocer las políticas de seguridad para auditarlas y garantizar que se apliquen correctamente.

Instrucciones:

1. Ve y analiza detalladamente el video "escenario de muestra" que se presenta a continuación:

2. Del escenario que acabas de visualizar, identifica los principales problemas que observaste.

3. Llévalos al foro correspondiente, y definan como grupo qué es lo que necesitan investigar para resolver los problemas que identificaron (metas de aprendizaje). Antes de iniciar la

investigación individual para elaborar el reporte correspondiente, deberás contar con el visto bueno del tutor.

4. Realiza un reporte individual con las respuestas del punto anterior que incluya tu propuesta de solución al problema.

5. Una vez terminado tu reporte, colócalo en la sección de tareas (recuerda que debes entregarlo a más tardar en la fecha límite establecida).

6. Puedes apoyarte de la lectura del capitulo 1 para la elaboración de tu reporte.

NOTA: Puedes recurrir a material actualizado adicional que encuentres en Internet o impreso que te sea de utilidad.

Guía Tutorial del Desarrollo del Ejercicio 1

Identificación de Vulnerabilidades en la Seguridad.

1. **Materia:** Seguridad en Redes e Internet
2. **Tema:** Políticas y Procedimientos de Seguridad
3. **Objetivo:** El alumno será capaz de identificar los riesgos en la seguridad asociados con la administración del acceso a los recursos y del flujo de datos en la red.
4. **Escenario:** Video de Acceso ilegal a instalaciones y a información confidencial
5. **Posibles términos que deban clarificarse:**
    - Niveles de Seguridad.
    - Controles Biométricos.
    - Seguridad Física.
    - Políticas y procedimientos de seguridad.
6. **Definición del problema esperada:**

- ¿Qué aspectos de seguridad física vulnerables se identificaron?
- ¿Qué aspectos de seguridad en el sistema vulnerables se identificaron?

7. Preguntas esperadas en el análisis del problema:

- ¿Cómo logro acceder físicamente al equipo?
- ¿Qué sistema de seguridad opera en el lugar?
- ¿Qué políticas de acceso se tienen implementadas en el lugar?
- ¿Qué sistema operativo tiene el equipo?
- ¿Cómo logro ingresar al sistema informático?
- ¿Que políticas de claves tienen implementada en el lugar para acceder al sistema?
- ¿Cómo logro entrar y salir sin que se dieran cuenta?
- ¿Cómo fue posible que no dejara rastro de su acceso al sistema?
- ¿A quién ha sido asignada la responsabilidad de cuidar la seguridad?

8. Metas de aprendizaje:

- ¿Qué políticas de seguridad física existen?
- ¿Qué amenazas se tienen con la información y los recursos?
- ¿Cuáles son las medidas que se deben tomar cuando un sistema de seguridad ha sido violado?
- ¿Quiénes son las personas involucradas en la seguridad?
- ¿Cuáles son los mecanismos de prevención de problemas de seguridad?
- ¿Qué tan importante es contar con un esquema de auditoria de seguridad?

Ejercicio 2

Tema: Políticas y Procedimientos de Seguridad

Práctica 2: Detección de vulnerabilidades y diseño de políticas y procedimientos de seguridad.

En cualquier lugar, donde el aspecto informático esté involucrado, existe la necesidad de contar con ciertas políticas y mecanismos que nos garanticen la seguridad de la información que se maneja. En la mayoría de los casos, los problemas de seguridad informática son ocasionados por errores humanos que pasan por alto situaciones o intentan reducir los procedimientos, estos problemas de seguridad abarcan todo el ámbito de usuarios de los sistemas, es por esto resulta sumamente importante contar con políticas de seguridad diseñadas de manera formal para garantizar la seguridad de la información y los recursos al aplicarlas correctamente.

Objetivo: El Alumno será capaz de identificar las vulnerabilidades de seguridad de una empresa y propondrá políticas y procedimientos sólidos para garantizar la seguridad de los activos en un escenario real.

Instrucciones:

1. Analice un escenario real accesible para ti, para realizar las siguientes tareas:

   - Identificar los activos clave con los que cuenta la empresa.
   - Identificar las posibles amenazas de seguridad con las que cuenta la empresa.
   - Determinar los riesgos que representa la mayor amenaza.
   - Identificar la fuente y la probabilidad de los riesgos a la seguridad.
   - Identificar los puntos vulnerables en su infraestructura de seguridad actual.
   - Proponga al menos 10 políticas y procedimientos de seguridad sólidas para la protección de los activos.
   - Proponga al menos 10 medidas efectivas para proteger los activos.
   - Explique la forma en la que se pueden integrar las medidas de seguridad propuestas a la administración diaria y a las prácticas de soporte.
   - Discute en el foro correspondiente de forma grupal los puntos importantes que obtuvo de esta actividad de campo.
   - Antes de realizar tu propuesta, pide el visto bueno del tutor correspondiente.
   - Realiza tu propuesta individual y colócala en la sección de tareas (recuerda que debes entregar tu reporte antes de la fecha límite.

Como recurso adicional puedes consultar los siguientes documentos electrónicos y sitios de interés:

Páginas Electrónicas de Referencia

NOTA: Puedes recurrir a material actualizado adicional que encuentres en Internet o impreso que te sea de utilidad.

- http://www.symantec.com/region/mx/enterprisesecurity/content/expert/LAM_1165.html
- http://h30095.www3.hp.com/servicios/infraestructura_it/it_segu_plane_plane_segu.html
- http://www.g2security.com/servicios_politicas.cfm

Páginas de Referencia Electrónica

http://www.cert.org.mx

http://www.linuxsecurity.org

http://www.kernel.org

http://www.securityfocus.com

http://www.rootshell.com

http://www.nsa.gov

Fuentes de Referencia Bibliográfica

Administración de Sistemas Linux, Guía Avanzada

Carling, M; Degler, Stephen, Madrid 1999,

Prentice Hall Iberia.

Linux Máxima Seguridad

Anónimo, Madrid 2000

Pearson Educación, S. A.

Firewalls y la seguridad en Internet,

Siyan, Karanjit; Hare, Chris; 1997,

Prentice-Hall Hiberoamericana, S.A.

Seguridad y Comercio en el Web

Spafford, Gene; Garfinkel, Simson; 1999,

McGrawHill & O'Reilly.

Hacking Expose, Network Security Secres & Solutions

McClure, Stuart; Scambray, Joel Tercera Edición, 2002,

Osborne/McGraw-Hill.

2. Programación Segura

1.5. Técnicas de Programación

El principal problema de la programación lo constituyen los programas que tiene *el bit SUID activo;* si un programa sin este bit activo tiene un fallo, lo normal es que ese fallo solamente afecte a quien lo ejecuta. Al tratarse de un error de programación, algo no intencionado, su primera consecuencia será el mal funcionamiento de ese programa. Este esquema cambia radicalmente cuando el programa está *con el bit SUID activo:* en este caso, el error puede comprometer tanto a quien lo ejecuta como a su propietario, y como ese propietario es por norma general el *root* automáticamente se compromete a todo el sistema. Para la codificación segura de este tipo de programas, proporciona unas técnicas básicas de programación:

- **Máximas restricciones a la hora de elegir el UID y el GID**

Una medida de seguridad básica que todo administrador ha de seguir es realizar todas las tareas con el mínimo privilegio que estas requieran; así, a nadie se le ocurre (o se le debería ocurrir) conectar a IRC o aprender a manejar una aplicación genérica bajo la identidad de *root*. Esto es directamente aplicable a la hora de programar: cuando se crea un programa *con el bit SUID activo* se le ha de asignar tanto el UID como el GID menos peligroso para el sistema. Por ejemplo, si un programa servidor se limita a mostrar un mensaje en pantalla y además escucha en un puerto por encima de 1024, no necesita para nada estar *con el bit SUID activo* a nombre de *root* (realmente, es poco probable que ni siquiera necesite estar *con el bit SUID activo)*; si pensamos en un posible error en dicho programa que permita a un atacante obtener un *shell* vemos claramente que cuanto menos privilegio tenga el proceso, son menores las consecuencias del error.

- ***Reset* de los UIDs y GIDs efectivos antes de llamar a '*exec()*'**

Uno de los grandes problemas de los programas *con el* ***bit SUID activo*** es la ejecución de otros programas de manera inesperada; por ejemplo, si el usuario introduce ciertos datos desde teclado, datos que se han de pasar como argumento a otra aplicación, nada nos asegura *a* ***priori*** que esos datos sean correctos o coherentes. Por tanto, es evidente que se debe de limpiar el UID y el GID efectivos antes de invocar a **exec()**, de forma que cualquier ejecución inesperada se realice con el mínimo privilegio necesario; esto también es aplicable a funciones que indirectamente realicen el **exec()**, como **system()** o **popen()**.

- Es necesario cerrar todos los descriptores de archivo, excepto los estrictamente necesarios, antes de llamar a **exec()**.

- Los descriptores de archivo son un parámetro que los procesos Unix heredan de sus padres; de esta forma, si un programa *con el* ***bit SUID activo*** está leyendo un archivo,

cualquier proceso hijo tendrá acceso a ese archivo a no ser que explícitamente se cierre su descriptor antes de ejecutar el **exec()**.

La forma más fácil de prevenir este problema es activando un **flag** que indique al sistema que ha de cerrar cierto descriptor cada vez que se invoque a **exec()**; esto se consigue mediante las llamadas **fcntl()** e **ioctl()**.

- Hay que asegurarse de que **chroot()** realmente limita.

Los enlaces duros entre directorios son algo que el núcleo de muchos sistemas Unix no permiten debido a que genera bucles en el sistema de archivos, algo que crea problemas a determinadas aplicaciones; por ejemplo, Linux no permite crear estos enlaces, pero Solaris o Minix sí. En estos últimos, en los clones de Unix que permiten ***hard links*** entre directorios, la llamada **chroot()** puede perder su funcionalidad: estos enlaces pueden seguirse aunque no se limiten al entorno con el directorio raíz restringido. Es necesario asegurarse de que no hay directorios enlazados a ninguno de los contenidos en el entorno **chroot()** (podemos verlo con la opción `-l' de la orden ls, que muestra el número de enlaces de cada archivo).

- Comprobaciones del entorno en que se ejecutará el programa

En Unix todo proceso hereda una serie de variables de sus progenitores, como el umask, los descriptores de archivos, o ciertas variables de entorno (*$PATH*, *$IFS*...); para una ejecución segura, es necesario controlar todos y cada uno de estos elementos que afectan al entorno de un proceso. Especialmente críticas son las funciones que dependen del *shell* para ejecutar un programa, como **system()** o **execvp():** en estos casos es muy difícil asegurar que el ***shell*** va a ejecutar la tarea prevista y no otra.

Por ejemplo, imaginemos el siguiente código:

```
#include <stdlib.h>

main(){
  system("ls");
}
```

A primera vista, este programa se va a limitar a mostrar un listado del directorio actual; no obstante, si un usuario modifica su $PATH de forma que el directorio `.' ocupe el primer lugar, se ejecutará ./ls en lugar de /bin/ls. Si el programa ./ls fuera una copia del shell, y el código anterior estuviera con el bit SUID activo por el root, cualquier usuario podría obtener privilegios de administrador.

Quizás alguien puede pensar que el problema se soluciona si se indica la ruta completa (/bin/ls) en lugar de únicamente el nombre del ejecutable; evidentemente, esto arreglaría el fallo

anterior, pero seguirían siendo factibles multitud de ataques contra el programa. Desde la modificación del *$IFS* (como veremos más adelante) hasta la ejecución en entornos restringidos, existen muchísimas técnicas que hacen muy difícil que un programa con estas características pueda ser considerado seguro.

- **Nunca** *poner el bit SUID activo en shellscripts*. Aunque en muchos sistemas Unix la activación del bit *SUID en activo* en *shellscripts* no tiene ningún efecto, muchos otros aún permiten que los usuarios (especialmente el **root**) creen procesos interpretados y *el bit SUID activo.* La potencia de los intérpretes de órdenes de Unix hace casi imposible controlar que estos programas no realicen acciones no deseadas, violando la seguridad del sistema, por lo que bajo ningún concepto se deben de utilizar procesos por lotes para realizar acciones privilegiadas.

- No utilizar **creat()** para bloquear.Una forma de crear un fichero de bloqueo es invocar a **creat()** con un modo que no permita la escritura del archivo (habitualmente el 000), de forma que si otro usuario tratara de hacer lo mismo, su llamada a **creat()** fallaría. Esta aproximación, que a primera vista parece completamente válida, no lo es tanto si la analizamos con detalle: en cualquier sistema Unix, la protección que proporcionan los permisos de un fichero sólo es aplicable si quien trata de acceder a él no es el *root*. Si esto es así, es decir, si el UID efectivo del usuario que está accediendo al archivo es 0, los permisos son ignorados completamente y el acceso está permitido; de esta forma, el *root* puede sobreescribir archivos sin que le importen sus bits *rwx*, lo que implica que si uno de los procesos que compiten por el recurso bloqueado está con el bit SUID activo a nombre del administrador, el esquema de bloqueo anterior se viene abajo.

  Para poder bloquear recursos en un programa *con el bit SUID activo* se utiliza la llamada **link(),** ya que si se intenta crear un enlace a un fichero que ya existe **link()** falla aunque el proceso que lo invoque sea propiedad del *root* (y aunque el fichero sobre el que se realice no le pertenezca).También es posible utilizar la llamada al sistema **flock()** de Unix, aunque esto no se recomienda por motivos de portabilidad entre estos.

- **Capturar todas las señales**

  Un problema que puede comprometer la seguridad del sistema Unix es el volcado de la imagen en memoria de un proceso cuando éste recibe ciertas señales (el clásico *core dump*). Esto puede provocar el sobreflujo de información sensible que el programa estaba leyendo: por ejemplo, en versiones del programa login de algunas versiones antiguas de Unix, se podía leer parte de **/etc/shadow** enviando al proceso la señal SIGTERM y consultando el archivo de overflow.

  No obstante, este problema no resulta tan grave como otro también relacionado con los *core dump*: cuando un programa *con el bit SUID activo* muestra su imagen el archivo resultante tiene el mismo UID que el UID real del proceso. Esto puede permitir a un usuario obtener un archivo con permiso de escritura pero que pertenezca a otro usuario

(por ejemplo, el *root*): evidentemente esto es muy perjudicial, por lo que parece claro que en un programa *con el bit SUID activo* necesitamos capturar todas las señales que Unix nos permita (recordemos que SIGKILL no puede capturarse ni ignorarse, por norma general).

- **Hay que asegurarse de que las verificaciones realmente verifican**

Otra norma básica a la hora de escribir aplicaciones *setuid* es la desconfianza de cualquier elemento externo al programa; hemos de verificar siempre que las entradas (teclado, archivo...) son correctas, ya no en su formato sino más bien en su origen, de quién proviene un archivo del que nuestro programa lee sus datos, de una fuente fiable o de un atacante que por cualquier método, no nos importa cuál, ha conseguido reemplazar ese archivo por otro que él ha creado.

- **Cuidado con las recuperaciones y detecciones de errores**

Ante cualquier situación inesperada y por lo general, poco habitual, incluso forzada por un atacante un programa *con el bit SUID activo* debe detenerse sin más; nada de intentar recuperarse del error, detenerse sin más. Esto, que quizás rompe muchos de los esquemas clásicos sobre programación robusta, tiene una explicación sencilla, cuando un programa detecta una situación inesperada, a menudo el programador asume condiciones sobre el error (o sobre su causa) que no tienen por qué cumplirse, lo que suele desembocar en un problema más grave que la propia situación inesperada. Para cada posible problema que un programa encuentre (entradas muy largas, carácteres erróneos o de control, formatos de datos erróneos...) es necesario que el programador se plantee qué es lo que su código debe hacer, y ante la mínima duda detener el programa.

- **Cuidado con las operaciones de entrada/salida**

La entrada/salida entre el proceso y el resto del sistema constituye otro de los problemas comunes en programas *con el bit SUID activo*, especialmente a la hora de trabajar con archivos; las condiciones de carrera aquí son algo demasiado frecuente: el ejemplo clásico se produce cuando un programa *con el bit SUID activo* ha de escribir en un archivo propiedad del usuario que ejecuta el programa (**no** de su propietario). En esta situación lo habitual es que el proceso cree el archivo, realize sobre él un chown() al rUID y al rGID del proceso (es decir, a los identificadores de quién está ejecutando el programa), y posteriormente escriba en el archivo; el esqueleto del código sería el siguiente:

```
fd=open("fichero",O_CREAT);
fchown(fd,getuid(),getgid());
write(fd,buff,strlen(buff));
```

Pero, qué sucede si el programa se interrumpe tras realizar el **open()** pero antes de invocar a **fchown()**, y además el **umask** del usuario es 0?. El proceso habrá dejado un archivo que

pertenece al propietario del programa (generalmente el **root**) y que tiene permiso de escritura. La forma más efectiva de solucionar el problema consiste en que el proceso engendre un hijo mediante **fork()**, hijo que asignará a sus eUID y eGID los valores de su rUID y rGID (los identificadores del usuario que lo ha ejecutado, no de su propietario). El padre podrá enviar datos a su hijo mediante **pipe()**, datos que el hijo escribirá en el fichero correspondiente: así el archivo en ningún momento tendrá por qué pertenecer al usuario propietario del programa, con lo que evitamos la condición expuesta anteriormente.

Sin embargo, el estilo correcto de programación no siempre es la solución a los problemas de seguridad del código; existen llamadas a sistema o funciones de librería que son un clásico a la hora de hablar de ***bugs*** en nuestro ***software***. Como norma, tras cualquier llamada se ha de comprobar su valor de retorno y manejar los posibles errores que tenga asociados, con la evidente excepción de las llamadas que están diseñadas para sobrescribir el espacio de memoria de un proceso (la familia **exec(),** por ejemplo) o las que hacen que el programa finalice (típicamente, **exit()**) . Algunas de las llamadas consideradas más peligrosas (bien porque no realizan las comprobaciones necesarias, bien porque pueden recibir datos del usuario) son las siguientes:

- **system():** Esta es la llamada que cualquier programa *con el bit SUID activo* debe evitar a toda costa. Si aparece en un código destinado a ejecutarse con privilegios, significa casi con toda certeza un grave problema de seguridad; en algunas ocasiones su peligrosidad es obvia (por ejemplo si leemos datos tecleados por el usuario y a continuación hacemos un **system()** de esos datos, ese usuario no tendría más que teclear **/bin/bash** para conseguir los privilegios del propietario del programa), pero en otras no lo es tanto: imaginemos un código que invoque a **system()** de una forma similar a la siguiente:

```
#include <stdio.h>
#include <stdlib.h>

main(){
system("/bin/ls");
}
```

  El programa anterior se limitaría a realizar un listado del directorio desde el que lo ejecutemos. Al menos en teoría, ya que podemos comprobar que no es difícil "engañar" a **system():** no tenemos más que modificar la variable de entorno *$IFS* (*Internal Field Separator*) del *shell* desde el que ejecutemos el programa para conseguir que este código ejecute realmente lo que nosotros le indiquemos. Esta variable delimita las palabras (o símbolos) en una línea de órdenes, y por defecto suele estar inicializada a *Espacio*, *Tabulador*, y *Nueva Línea* (los separadores habituales de palabras); pero, qué sucede si le indicamos al *shell* que el nuevo carácter separador va a ser la barra, **`/'?**. Muy sencillo: ejecutar **`/bin/ls'** será equivalente a ejecutar **`bin ls'**, es decir, una posible orden denominada `bin' que

recibe como parámetro **`ls'**. Por ejemplo, bajo SunOS - en la mayoría de las versiones de Unix -, y utilizando **sh** (no **bash**) podemos hacer que **`bin'** sea un programa de nuestra elección, como **`id'**:

```
$ cp /bin/id bin
$ ejemplo
bin  ejemplo.c  ejemplo
$ IFS=/
$ export IFS
$ ejemplo
uid=672(toni) gid=10(staff)
$
```

Como podemos ver, acabamos de ejecutar un programa arbitrario; si en lugar de `id' hubiéramos escogido un intérprete de órdenes, como **`bash' o `sh'**, habríamos ejecutado ese *shell*. Y si el programa anterior estuviera *setudiado*, ese *shell* se habría ejecutado con los privilegios del propietario del archivo (si imaginamos que fuera root, podemos hacernos una idea de las implicaciones de seguridad que esto representa).

- **exec(), popen():** Similares a la anterior; es preferible utilizar execv() o execl(), pero si han de recibir parámetros del usuario sigue siendo necesaria una estricta comprobación de los mismos.

- **setuid(), setgid()...:** Los programas de usuario no deberían utilizar estas llamadas, ya que no han de tener privilegios innecesarios.

- **strcpy(), strcat(), sprintf(), vsprintf()...:** Estas funciones no comprueban la longitud de las cadenas con las que trabajan, por lo que son una gran fuente de *buffer overflows*. Se han de sustituir por llamadas equivalentes que sí realicen comprobación de límites (**strncpy(), strncat()...**) y, si no es posible, realizar dichas comprobaciones manualmente.

- **getenv():** Otra excelente fuente de desbordamientos de *buffer*; además, el mal uso de la información leída puede ser peligroso, es importante recordar que el usuario generalmente puede modificar el valor de las variables de entorno. Por ejemplo, qué sucedería si ejecutamos desde un programa una orden como **`cd $HOME'**, y resulta que esta variable de entorno no corresponde a un nombre de directorio sino que es de la forma **`/;rm -rf /'?**. Si algo parecido se hace desde un programa que se ejecute con privilegios en el sistema, podemos imaginarnos las consecuencias.

- **gets(), scanf(), fscanf(), getpass(), realpath(), getopt()...:** Estas funciones no realizan las comprobaciones adecuadas de los datos introducidos, por lo que pueden desbordar en algunos casos el *buffer* destino o un *buffer* estático interno al sistema.

Es preferible el uso de **read()** o **fgets()** siempre que sea posible (incluso para leer una contraseña, haciendo por supuesto que no se escriba en pantalla), y si no lo es al menos realizar manualmente comprobaciones de longitud de los datos leídos.

- **gethostbyname(), gethostbyaddr():** Seguramente ver las amenazas que provienen del uso de estas llamadas no es tan inmediato como ver las del resto; generalmente hablamos de desbordamiento de *buffers*, de comprobaciones de límites de datos introducidos por el usuario, es impotante pensar en datos que un atacante nos introduce directamente desde teclado o desde un archivo, pero cuyo valor puede forzar incluso desde sistemas que ni siquiera son el nuestro. Por ejemplo, todos tendemos a asumir como ciertas las informaciones que un servidor DNS más o menos fiables, por ejemplo alguno de nuestra propia organización nos brinda. Imaginemos un programa como el siguiente (se han omitido las comprobaciones de errores habituales por cuestiones de claridad):

```
#include <stdio.h>
#include <stdlib.h>
#include <netdb.h>
#include <unistd.h>
#include <arpa/inet.h>

int main(int argc, char **argv){
struct in_addr *dir=(struct in_addr *)malloc(sizeof(struct in_addr));
struct hostent *máquina=(struct hostent *)malloc(sizeof(struct \
      hostent));
char *orden=(char *)malloc(30);
dir->s_addr=inet_addr(*++argv);
máquina=gethostbyaddr((char *)dir,sizeof(struct in_addr),AF_INET);
sprintf(orden,"finger @%s\n",máquina->h_name);
system(orden);
return(0);
}
```

Este código recibe como argumento una dirección IP, obtiene su nombre vía **/etc/hosts** o DNS, y ejecuta un finger sobre dicho nombre; aparte de otros posibles problemas de seguridad (por ejemplo, seríamos capaces de procesar **cualquier** información que devuelva el **finger?**, qué sucede con la llamada a **system()?)**, nada extraño ha de suceder si el nombre de máquina devuelto al programa es `normal':

```
luisa:~/tmp$ ./ejemplo 192.168.0.1
[rosita]
No one logged on.
luisa:~/tmp$
```

Pero, qué pasaría si en lugar de devolver un nombre `normal' (como **`rosita'**) se devuelve un nombre algo más elaborado, como **`rosita;ls'**? Podemos verlo:

```
luisa:~/tmp$ ./ejemplo 192.168.0.1
[rosita;ls]
No one logged on.
ejemplo  ejemplo.c
luisa:~/tmp$
```

Exactamente: se ha ejecutado la orden **`finger @rosita;ls'** (esto es, un **`finger'** a la máquina seguido de un **`ls'**). Podemos imaginar los efectos que tendría el uso de este programa si sustituimos el inocente **`ls'** por un **`rm -rf $HOME'**. Un atacante que consiga controlar un servidor DNS (algo no muy complicado) podría introducir datos maliciosos en nuestra máquina sin ningún problema. Para evitar esta situación debemos hacer una doble búsqueda inversa y además no hacer ninguna suposición sobre la corrección o el formato de los datos recibidos; en nuestro código debemos insertar las comprobaciones necesarias para asegurarnos de que la información que recibimos no nos va a causar problemas.

- **syslog():** hemos de tener la precaución de utilizar una versión de esta función de librería que compruebe la longitud de sus argumentos; si no lo hacemos y esa longitud sobrepasa un cierto límite (generalmente, 1024 *bytes*) podemos causar un desbordamiento en los *buffers* de nuestro sistema de *log*, dejándolo inutilizable.

- **realloc():** ningún programa - privilegiado o no, que maneje datos sensibles (por ejemplo, contraseñas, correo electrónico y especialmente aplicaciones criptográficas) debe utilizar esta llamada; **realloc()** se suele utilizar para aumentar dinámicamente la cantidad de memoria reservada para un apuntador. Lo habitual es que la nueva zona de memoria sea contigua a la que ya estaba reservada, pero si esto no es posible **realloc()** copia la zona antigua a una nueva ubicación donde pueda añadirle el espacio especificado. ¿Cuál es el problema? La zona de memoria antigua se libera (perdemos el apuntador a ella) pero no se pone a cero, con lo que sus contenidos permanecen inalterados hasta que un nuevo proceso reserva esa zona; accediendo a bajo nivel a la memoria (por ejemplo, leyendo **/proc/kcore o /dev/kmem**) sería posible para un atacante tener acceso a esa información.

  Realmente, **malloc()** tampoco pone a cero la memoria reservada, por lo que a primera vista puede parecer que cualquier proceso de usuario (no un acceso a bajo nivel, sino un simple **malloc()** en un programa) podría permitir la lectura del antiguo contenido de la zona de memoria reservada. Esto es falso si se trata de nueva memoria que el núcleo reserva para el proceso invocador: en ese caso, la memoria es limpiada por el propio *kernel* del operativo, que invoca a **kmalloc()** (en el caso de Linux, en otras versiones de Unix el nombre puede variar aunque la idea sea la misma) para hacer la reserva. Lo que sí es posible es que si liberamos una zona de memoria (por ejemplo con **free()**) y a continuación la volvemos a reservar, en el mismo proceso, podamos acceder a su

contenido: esa zona no es "hueva" (es decir, el núcleo no la ha reservado de nuevo), sino que ya pertenecía al proceso. De cualquier forma, si vamos a liberar una zona en la que está almacenada información sensible, lo mejor en cualquier caso es ponerla a cero manualmente, por ejempb mediante **bzero()** o **memset()**.

- **open():** El sistema de archivos puede modificarse durante la ejecución de un programa de formas que en ocasiones ni siquiera imaginamos; por ejemplo, en Unix se ha de evitar escribir siguiendo enlaces de archivos inesperados (un archivo que cambia entre una llamada a **lstat()** para comprobar si existe y una llamada a **open()** para abrirlo en caso positivo, como hemos visto antes). No obstante, no hay ninguna forma de realizar esta operación atómicamente sin llegar a mecanismos de entrada/salida de muy bajo nivel; Peter Gutmann propone el siguiente código para asegurarnos de que estamos realizando un **open()** sobre el archivo que realmente queremos abrir, y no sobre otro que un atacante nos ha puesto en su lugar:

```
struct stat lstatInfo;
char *mode="rb+";
int fd;

if(lstat(fileName,&lstatInfo)==-1)
   {
   if(errno!=ENOENT) return( -1 );
   if((fd=open(fileName,O_CREAT|O_EXCL|O_RDWR,0600))==-1) return(-1);
   mode="wb";
   }
else
   {
   struct stat fstatInfo;
   if((fd=open(fileName,O_RDWR))==-1) return(-1);
   if(fstat(fd,&fstatInfo)==-1 || \
      lstatInfo.st_mode!=fstatInfo.st_mode || \
      lstatInfo.st_ino!=fstatInfo.st_ino || \
      lstatInfo.st_dev!=fstatInfo.st_dev)
      {
      close(fd);
      return(-1);
      }
   if(fstatInfo.st_nlink>1||!S_ISREG(lstatInfo.st_mode))
      {
      close(fd);
      return(-1);
      }
#ifdef NO_FTRUNCATE
   close(fd);
```

```
    if((fd=open(fileName,O_CREAT|O_TRUNC|O_RDWR))==-1) return( -1 );
    mode="wb";
#else
    ftruncate(fd,0);
#endif /* NO_FTRUNCATE */
    }
stream->filePtr=fdopen(fd,mode);
if(stream->filePtr==NULL)
    {
    close(fd);
    unlink(fileName);
    return(-1);  /* Internal error, should never happen */
    }
}
```

Como podemos ver, algo tan elemental como una llamada a **open()** se ha convertido en todo el código anterior si queremos garantizar unas mínimas medidas de seguridad; esto nos puede dar una idea de hasta que punto la programación `segura' puede complicarse.

No obstante, en muchas ocasiones es preferible toda la complicación y parafernalia anteriores para realizar un simple **open()** a que esa llamada se convierta en un fallo de seguridad en nuestro sistema. No hay ningún programa que se pueda considerar perfecto o libre de errores (como se cita en el capítulo 23 de [GS96], una rutina de una librería puede tener un fallo o un rayo gamma puede alterar un *bit* de memoria para hacer que nuestro programa se comporte de forma inesperada), pero cualquier medida que nos ayude a minimizar las posibilidades de problemas es siempre positivo.

1.6. Manejo de Excepciones

Una excepción es un evento inesperado, generalmente con proporciones de calamidad. El buen programador desea que su programa sea robusto, de forma que ante estos eventos reaccione adecuadamente; si no es posible corregir los problemas que se producen después de una falla, es por lo menos deseable que el programa tenga una degradación suave, evitando de esta manera un estruendoso fin de ejecución. En lo posible se trata siempre de sobreponerse de la mejor manera a la catástrofe. La acción a realizar suele consistir en una acción correctiva, o la generación de un informe acerca del error producido.

Ejemplos de lenguajes con gestión de excepciones son Java, Net, C++, Objective-C, Ada, Eiffel, y Ocaml.

Existen muchos tipos de excepciones. Por ejemplo, cuando en tiempo de ejecución se usa un índice de vector que está fuera de rango se produce una excepción. También sucede cuando se hace una división por cero, o cuando el resultado de una expresión es demasiado pequeño (desbordamiento y bajo rebase). Los principales tipos de excepción son los siguientes:

- Errores en tiempo de ejecución, como división por cero, sobre rebase (overflow), bajo rebase (underflow), o intentar leer de un archivo que no existe.
- Excepciones generadas por las bibliotecas de programas, que aprovechan el mecanismo de excepciones para simplificar el uso de la biblioteca.
- Excepciones generadas por el programador, porque mediante la generación y manejo de excepciones es posible construir programas más robustos o retornar de invocaciones que están profundamente anidades.

El manejo de las excepciones permite independizar el código de uso de un recurso del manejo de los errores, lo cual no retrasaría la ejecución del programa porque el código de acceso al recurso se ejecuta siempre, pero el código de manejo de errores se ejecuta sólo cuando este sucede.

¿Como se implementa?

Los lenguajes con gestión de excepciones incorporan en sus bibliotecas la capacidad de detectar y notificar errores.

Cuando un error es detectado se implementan las siguientes acciones:

- Se interrumpe la ejecución del código en curso.
- Se crea un objeto excepción que contiene información del problema. Esta acción es conocida como "lanzar una excepción". Existe un mecanismo estandar de gestión de error.
- Si hay un manejador de excepciones en el contexto actual le transfiere el control. En caso contrario, pasa la referencia del objeto excepción al contexto anterior en la pila de llamadas.
- Si no hay ningún manejador capaz de gestionar la excepción, el hilo que la generó es terminado.

Por ejemplo, si intentamos leer un archivo inexistente usando la clase **FileReader** del lenguaje Java, la implementación de la propia clase detectará el problema, y lanzará una excepción de tipo **FileNotFoundException**.

¿Por qué es importante usar excepciones?

El código de tratamiento del error está separado del resto del programa. Esto aumenta la legibilidad y permite centrarse en cada tipo de código.

Un mismo manejador puede gestionar las excepciones de varios ámbitos inferiores. Esto reduce la cantidad de código necesaria para gestionar los errores.

Existe un mecanismo estandar de gestión de error. Lenguajes anteriores como C empleaban el retorno de valores especiales en cada método para señalar condiciones anomalas. Esto implicaba que a cada llamada debía seguirle un código de gestión de errores que distorsionaban la legibilidad del código.

Buenas Prácticas

Tratamiento de excepciones:

1. Cada capa debe lanzar una excepción al nivel de abstracción que le corresponda. Por ejemplo, la capa de persistencia debe lanzar PersistenciaException y no SQLException. Observa que si no usamos excepciones genericas, estamos acoplando una capa a los detalles de implementación de la otra.

2. Si no hay recuperación posible de una excepción no se recomienda recuperarla. En este caso permite su propagación hasta que llegue como error al usuario, o alguna capa pueda realizar una acción correctiva.
3. Considera envolver excepciones checked con excepciones no checked.Consulta Checked vs unchecked acerca de cuando debemos usar una u otra.

4. No hagas log si vas a relanzar la excepción. Si el sistema de gestión de excepciones de la aplicación está bien definido, será más productivo y claro dejar que lo haga por nosotros.

5. No captures excepciones y las dejes sin tratar. Si omites el error, obligas a pasar el depurador para encontrarlos.

6. Nunca pierdas los stacktraces de las excepciones que envuelvas. El stacktrace es información esencial para descubrir el problema. Hay entornos especiales (RMI, EJB, aplicaciones web) que pueden requerir medidas especificas. En JDKs anteriores a 1.4 debemos definir nuestras propias excepciones encadenadas. Permite que las excepciones se propaguen entre capas.

7. Declara las RuntimeExceptions en el throws del método. Es más legible dejar explícito su uso, y ayuda a las herramientas que manipulan la clase para exportarla como servicio, aplicar aop, generar código, etc.

8. Evitar que los threads terminen sin avisar debido a excepciones. Usa Thread.UncaughtExceptionHandler pendiente: esto puede usarse para capturar excepciones en los GUI.

9. Usa una excepción adecuada de la API o define la tuya propia. Cada excepción debe reflejar un tipo de error. Al crear excepciones personalizadas pordemos añadirles información acerca del contexto del error.

10. Evita que finally falle. Cuando se produce una excepción en el try, y luego otra en el finally, es esta última la que se propaga. La anterior queda oculta.

11. En un bloque finally hay poco código, prestale atención.

12. Evita usar excepciones para control de flujo. La creación del objeto excepción es costosa porque se crea con una copia de todas las llamadas de la pila. Otros lenguajes como Ocaml se limitan a resetear el puntero de pila a un valor de una tabla de consulta.

13. Considera el uso de gestores de excepción personalizados. Frameworks como struts permiten activar handlers determinados para cada excepción.

14. Haz configurable el tratamiento de excepciones. El tratamiento debe variar para cada contexto. Por ejemplo, es adecuado que una aplicación web adjunte un stacktrace en la propia página si el usuario pertenece a los grupos users, developers.

Hay dos tipos de excepciones:

- Checked: El programa no compila si no son capturadas. Son las subclases de **Exception**.

- Unchecked: El programa compila aunque no sean capturadas. Son las subclases de **RuntimeException**.

Cuando usar checked:

- Cuando el fallo es recuperable.

- Cuando queremos obligar al programador a considerar la posibilidad de un fallo.

Esto puede depender del dominio del problema. Por ejemplo:

- Log4j se traga las excepciones para no interferir en el programa.

- En un programa crítico (el controlador del Mars Rover) deberían predominar las checked.

SQLExecutor solo lanza unchecked. Ver SQLExecutor en TSS, developer.com

En los métodos públicos de una librería. Hay quien prefiere RuntimeException entre capas.

Opino que es una mala idea porque:

- Que una librería lance RuntimeException puede resultar en un fallo inesperado para el cliente.
- No es responsabilidad de la librería decidir si el fallo es recuperable o no.
- Aunque documentemos con throws en los métodos, la realidad es que los IDEs no advierten de ello (no tienen porque), y no tenemos el javadoc al lado para tener en cuenta que el método puede lanzar unchecked.

Cuando usar unchecked:

Por comodidad dentro de una librería. No todas las excepciones son irrecuperables ni están causadas por errores de programación. Es perfectamente valido lanzar o relanzar como unchecked dentro de una aplicación que esta preparada para tratar con la situación. Observa sin embargo, que relanzar como unchecked puede causar falsos resultados en los test unitarios, y (para el que le importe) está desaconsejado por SUN.

Cuando el fallo es irrecuperable y deseamos que la excepción termine con la operación en curso. En este caso no deseamos capturar.

¿Que pasa cuando usamos checked?

- Estamos informando a los programadores que fallos pueden producirse y sus significados.

- Los programadores perezosos optarán por:

    - añadir un **throws Exception** en cada método.
    - añadir un **throws loquesea** con tal de que compile.
    - tragarse la excepción con un catch vacío.

- Al considerar cada excepción posible, aumenta la calidad del código y disminuye la productividad. James Goslin escribió en su weblog:

"One of the design principles behind Java is that I don't care much about how long it takes to slap together something that kind of works. The real measure is how long it takes to write something solid."

Que este debate es cuestión de gustos lo atestigua el hecho de que Net solo tenga excepciones unchecked. De hecho Java es el primer lenguaje en lanzar excepciones checked:

Niveles de abstracción

Una buena práctica de la gestión de excepciones, es que cada capa debe lanzar una excepción al nivel de abstracción que le corresponda. Lo cual implica que entre capas solo deben lanzarse excepciones genericas.

Por ejemplo, la capa de persistencia debe lanzar PersistenciaException y no SQLException. De lo contrario estaremos acoplando el código a una implementación de persistencia concreta (JDBC). Si mañana cambiamos JDBC por JDO, tendremos que sustituir SQLException por JDOException a lo largo del código. El código sería:

```
try {
  // acceso a base de datos
} catch(SQLException e) {
  throw new PersistenciaException(e);
}
```

Observa que los métodos de la biblioteca o capa, accesibles por los clientes, deben capturar todas las excepciones posibles, para así poder relanzarlas dentro de una generica.

Un argumento en contra de todo lo anterior, es que:

- **Las excepciones deben explicar que ha pasado, y no de donde vienen**. Una excepción sirve para informar de un error, ¿qué importa donde se haya producido? Es decir, el nombre de la excepción debe ser indicativo de un error concreto. Esto a su vez, implica que una librería solo debe lanzar excepciones unchecked, pues de otro modo sería engorroso tener que capturar la variedad excepciones que puede lanzar la biblioteca. En realidad no importa que el cliente no las capture porque normalmente los fallos son irrecuperables para el cliente. Esto es así porque el cliente no está en el contexto de la librería, y por tanto no necesita liberar recursos ante un fallo (es tarea de la librería), no es capaz de modificar las condiciones para que la petición tenga éxito.

Por tanto, basta con hacerla unchecked y dejar que se propague por las capas hasta un manejador central de excepciones que informe del error.

No es necesario usar checked porque el cliente no podrá recuperarse del error.

Usar checked no genericas acopla el código. Si aun así quiere realizar alguna acción puede:

- Capturar la generica de la biblioteca y realizar alguna acción concreta. Esto es poco frecuente, lo normal es asumir el fallo y dejar que la excepción se propague.
- Capturar **Exception** y realizar alguna acción dependiendo de una configuración de manejadores de excepciones. Esto es lo más común. Suele hacerse tras haberse propagado la excepción hasta la última capa.

Página Electrónica de Referencia

Excepciones

- http://www.1x4x9.info/files/excepciones/html/online-chunked/ar01s02.html

1.7. Buffer Overflow

De los ataques más peligrosos destacan los que se basan en un recurso conocido como "desbordamiento de búfer" (*buffer overflow*), mismos que han hecho presa a equipos de cómputo con virus como *Sasser*, *MyDoom*, *Blaster*, etc. y que llegan inocentemente como archivos adjuntos o con extensiones aparentemente inofensivas (como txt, dat, mp3, entre otras).

En términos simples, es cuando un programa (generalmente un servidor o "demonio") recibe una entrada mayor a la que espera, sobreescribiendo, por tanto, áreas críticas de memoria. Esto genera que se ejecute cierto código, generalmente proporcionando al usuario acceso al sistema como root.

Si el desbordamiento de búfer se realiza de forma intencional, difícilmente podríamos pensar que el programador quiera dejar una bendición en el equipo de cómputo atacado. Por lo general se ejecuta código diverso, normalmente orientado a dañar de una u otra forma la información, la funcionalidad del sistema operativo o vulnerabilidad del sistema entero.

Existen algunos medios para evitar que el código maligno se ejecute sin permiso alguno; uno de los más efectivos es identificar las áreas de memoria para asegurarse que lo que se ejecute sea efectivamente propio. En la memoria sólo se almacenan dos tipos de elementos: datos (algo que puede leerse y escribirse) o código ejecutable (algo que sólo puede realizar tareas). Esto reduce la cantidad de trabajo que debe realizarse para minimizar el riesgo, pero requiere de generar nuevas formas de almacenamiento de elementos en la memoria.

Aprovecharse de páginas Web (código HTML) con diferentes huecos de seguridad. Una de las maneras comunes de entrar al sistema –o por lo menos inhabilitarlo– es realizando algunos trucos en las formas HTML. Por ejemplo, es común que el programador que definió una forma HTML no se asegure que la información que el usuario ha escrito para llenar en un campo, no incluya caracteres especiales o esté fuera de especificaciones. ¿La consecuencia? Un atacante puede introducir información inválida y ocasionar que se caiga el servidor (debido a un buffer overflow). En algunos casos la explotación de este tipo de huecos puede incluso darle acceso privilegiado (root o administrador) al atacante.

1.8. Programación WEB

La programación avarca el manejo de los siguientes programas:

- CGI's
- php
- asp
- perl

De los cuales cada uno tiene su forma en la cual es segura hacer este tipo de programación.

Programación en Asp Segura

Introducción

Comenzamos en HTMLWeb una nueva sección, la de programación de páginas web, y la iniciamos con éste manual de introdución, que llega hasta nosotros gracias a la colaboración de elguruprogramador.com.

En éste manual vamos a ver algunas de las caracteristicas más importantes de esta tecnología, y para comenzar por buen camino hay que entender algunos conceptos básicos:

- La sigla ASP significa Active Server Pages.
- Microsoft es el creador.
- Se ejecuta en el servidor donde esta alojada la página web.
- El cliente (navegador del usuario) no nota diferencias con una página normal (HTML).
- Por medio de ASP se puede tener acceso a bases de datos.
- Para su implementación se utiliza lenguaje script como VBScript o JScript.
- Se puede utilizar HTML y ASP en una misma página.
- Es totalmente ampliable gracias a que es compatible con la tecnología ActiveX.

¿Cómo funciona?

Aqui está paso a paso como funciona una página ASP.

1. Un usuario por medio del navegador solicita una página ASP.
2. Esta solicitud llega al servidor el cual tiene alojada la página pedida.
3. Este servidor procesa la página ASP y devuelve codigo HTML.
4. El usuario visualiza la página en su navegador.

Noten que para el usuario no existe diferencia en ASP y HTML, por que a su navegador siempre llega codigo HTML puro, el único que requiere un trabajo extra es el servidor, el cual tiene que procesar el codigo ASP y tranformarlo en HTML para su posterior envío al cliente.

¿Cómo utilizarlo?

Para utilizarlo nesecitamos un servidor que procese nuestras páginas ASP, para ello existe el Internet Information Sever bajo Windows NT o 2000 y bajo Windows 95/98 esta el Personal Web Server, que viene el CD de instalación de Windows 98 o en el CD del Visual Studio.

Validación de datos

Cuando un usuario visita nuestras páginas y cumplimenta un formulario en ellas, nosotros deseamos que los datos que ha introducido cumplan una serie de requisitos: que no se dejen ciertos campos vacíos, que los valores introducidos en campos numéricos sean efectivamente numéricos, que no sobrepasen un cierto valor o extensión.

Es cierto que normalmente este tipo de validaciones se realizan en el navegador cliente antes del envío del formulario, soliéndose usar para ello JavaScript (compatible con la mayoría de los navegadores). Pero también es cierto que existen poderosos motivos para realizar una posterior comprobación de datos en el servidor antes del proceso de los mismos.

Una de estas razones es la seguridad. De todos es sabido las vulnerabilidades que presentan muchos lenguajes de script en servidor, por las que una persona con intenciones malignas puede enviar en un campo de formulario cadenas de datos especialmente diseñadas para permitirle ejecutar instrucciones propias de código en el servidor, con lo que puede llegar a tener acceso a información privada alojada en el mismo, introducir virus o troyanos e incluso hacerse con el control total del servidor.

Otra utilidad importante de las funciones de manejo de cadenas y números es la conversión de unos formatos y tipos de datos a otros, con lo que podemos por ejemplo solucionar algunos de los aspectos de presentación y operación que se nos presenta a la hora de trabajar con euros. No vamos a tratar en esta ocasión aspectos como las operaciones matemáticas con números (dedicaremos un capítulo a ello en su momento); solamente el manejo de cadenas y números en su forma general.

Programacion en PHP

PHP es uno de los lenguajes de lado servidor más extendidos en la web. Nacido en 1994, se trata de un lenguaje de creación relativamente creciente que ha tenido una gran aceptación en la comunidad de webmasters debido sobre todo a la potencia y simplicidad que lo caracterizan.

PHP nos permite colocal pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas programados íntegramente en un lenguaje distinto al HTML. Por otra parte, y es aquí donde reside su mayor interés con respecto a los lenguajes pensados para los CGI, PHP ofrece un sinfín de funciones para la explotación de bases de datos de una manera llana, sin complicaciones.

Podríamos efectuar la quizás odiosa comparación de decir que PHP y ASP son lenguajes parecidos en cuanto a potencia y dificultad si bien su sintaxis puede diferir sensiblemente. Algunas diferencias principales pueden, no obstante, mencionarse:

- PHP, aunque multiplataforma, ha sido concebido inicialmente para entornos UNIX y es en este sistema operativo donde se pueden aprovechar mejor sus prestaciones. ASP, siendo una tecnología Microsoft, esta orientado hacia sistemas Windows, especialmente NT.

  Las tareas fundamentales que puede realizar directamente el lenguaje son definidas en PHP como funciones mientras que ASP invoca más frecuentemente los objetos. Por supuesto, esto no es más que una simple cuestión de forma ya que ambos lenguajes soportan igualmente ambos procedimientos.

- ASP realiza números as tareas sirviéndose de componentes (objetos) que deben ser comprados (o programados) por el servidor a determinadas empresas especializadas. PHP presenta una filosofía totalmente diferente y, con un espíritu más generoso, es progresivamente construido por colaboradores desinteresados que implementan nuevas funciones en nuevas versiones del lenguaje.

Los puntos importantes que se de de tener para hacer una programacion segura en PHP:

- Verificar el contenido de las variables, sobre todo cuando se trate de incluir archivos.
- No confiar en la variables globales
- No permitir la “incorporación” de archivos
- Utilizar la extensiónn *.php* para todo archivo, no usar *.inc* ni *.class*
- Almacenar los archicos de configuración sensibles en directorios fuera de la aplicación
- Evitar los servidores compartidos
- Validad tipo de información que se solicita

Programación Segura en Perl

- Verificar que perl y sus librerías solo puedan ser modificadas por root.
- Tener cuidado al utilizar la sentencia *eval.*
- Establecer en cada programa la variable *PATH.*
- Utiliza la emulación de perl para scripts con setuid.

1.8.1. Programación en MS .Net

Una explicacion breve de lo que es la tecnologia .NET se puede describir como sigue. La tecnología .Net está construida sobre un sistema operativo (e.j: Windows .NET Server) y .NET frameworks que proporcionan una serie de capacidades y funcionalidad básica (.NET

Framework); puedes añadir más posibilidades añadiendo servidores .NET en la cima de la base anterior (Microsoft SQL Server, BizTalk Server, Exchange Server, etc.). Ejecutándose a través de estos servidores están los servicios .NET (.NET Services), incluyendo servicios de identificación (Passport) y otros muchos más (agrupados en la iniciativa HailStorm, cuyo fututo es cuando menos incierto). Los datos y las "experiencias de usuario", o maneras para los usuarios de aprovechar las posibilidades de los servidores y de los clientes, están disponibles usando una variedad de dispositivos de la "era .NET", desde pc's de sobremesa a portátiles a PDAs a teléfonos móviles. Finalmente, usted puede crear aplicaciones o realizar otras tareas de desarrollo en cualquiera de los niveles anteriores haciendo uso de Visual Studio .NET.

Seguridad en .Net

Los ataques comunes pueden ser:

- Ataques Organizativos.
- Atacantes.
- Ataques automatizados.
- Denegación de servicio (DoS).
- Virus, caballos de Troya y gusanos.
- Infracciones accidentales de la seguridad.

Acontinuacion se citan algunas de las consecuencias si se tiene una seguridad insuficiente en la programacion .Net:

- Robo de propiedad intelectual.
- Tiempo de inactividad del sistema.
- Pérdida de productividad.
- Daño a la reputación de la empresa.
- Pérdida de confianza de los clientes.
- Graves pérdidas financieras por pérdida de ingresos.

Para hacer una mejor programacion en .Net se recomienda:

- Adoptar el modelado de amenaza, con él se tendrán identificadas las vulnerabilidades de seguridad.umenta el conocimiento de la arquitectura de las aplicaciones.

- Entrenar al equipo de desarrollo, con esta recomendación se evitarán defectos comunes en la seguridad y se tendrá una correcta aplicación de las tecnologías. El equipo de desarrollo debe de utilizar y aplicar las siguientes medidas de seguridad:

    - Cifrado.
    - Hashing.

- Firmas digitales.
- Certificados digitales.
- Comunicación segura.
- Autenticación.
- Autorización.
- Servidores de seguridad.
- Auditoría.
- Service Pack y actualizaciones.

- Revisión de código, para proteger al código que permite el acceso a la red. Se ejecuta de forma predeterminda, utiliza protocolos no autentificados y se ejecuta con protocolos elevados.

- Usar herramientas, con ello se tendrán pruebas más coherentes para detectar la vulnerabilidad.

Ejercicio 1

A continuación se emplea un ejercicio interactivo en el formato flash , siguiendo las instrucciones que se te indican:

Simulación: Como actua un ataque de Inyección de SQL

Ejercicio 2

A continuación se emplea un ejercicio interactivo en el formato flash , siguiendo las instrucciones que se te indican:

Simulación: Como actua un ataque de Inyección de SQL

Páginas de Referencia Electrónica

Fundamentos de Seguridad de las Aplicaciones:

- http://es.tldp.org/Manuales-LuCAS/doc-unixsec/unixsec-html/node1.html
- http://www.microsof.com/spanish/msdn/comunidad/uni.net/Sec1/default.asp
- http://www.microsoft.com/spanish/msdn/spain/eventos/talleres/descarga.asp

2. Criptografía

2.1. Introducción

La criptgrafía es la ciencia de escribir o leer mensajes codificados, es la base para disponer mecanismos de autenticación, integridad y confidencialidad. La autenticación establece la identidad del emisor y receptor de la información. La integridad asegura que los datos no han sido alterados y la confidencilidad asegura que nadie, excepto el emisor y el receptor puedan entender los datos.

El hombre ha utilizado la criptografía desde hace miles de años, las primeras evidencias de esto se dan en el antiguo Egipto. Cuando un egipcio importante moría, su cuerpo era momificado y era enterrado con pergaminos que portaban oraciones del "Libro de los Muertos". En estos pergaminos los sacerdotes escribía contraseñas secretas que le permitían al fallecido comprar su entrada al cielo (Referenca timeline de la criptografía http://world.std.com/~cme/html/timeline.html). La palaba criptogafía proviene de las palabras krypto (escondido) y graphia (escritura) lo que la criptografía es la ciencia de escribir de forma secreta. En la criptografía se crea mensajes que sólo pueden ser leídos por las personas.

La criptografía antigua era elemental; por lo general consistía en una mezcla del tipo anagrama, donde los caracteres sólo eran cambiados de lugar. Por ejemplo la palabra egipcio puede transponerse a oegicpi.

Los mecanismos de criptografía usan algoritmos (funciones matemáticas) y un valor secreto conocido como llave. Los algoritmos son conocidos y se encuentran disponibles para cualquier persona, la parte que permanece secreta es la llave y esta es la que provee la seguridad requerida. La llave es análoga a la combinación de un candado, el concepto de seguridad de un candado se conoce, pero no su combinación para abrirlo. Entre más números tiene una combinación, más difícil es adivinar dicha combinación, de igual forma para la criptografía, entre más bits tenga la llave más difícil será encontrarla. Entre mayor sea el rango de posibles llaves más difícil será encontrarla en un ataque de fuerza bruta. En un ataque de fuerza bruta, se aplican todas las combinaciones posibles de una llave a un algoritmo hasta encontrar el mensaje cifrado, la siguiente tabla muestra el número de llaves que deben probarse hasta agotar todas las posibilidades, dada la longitud de una llave:

Combinaciones de fuerza bruta

	Longitud de Combinación	
40	2^{40} =	1,099,511,627,77
56	2^{56} =	7.205759403793 x 10
64	2^{64} =	1.844674407371 x 1
112	2^{112} =	5.192296858535 x 1033
128	2^{128} =	3.402823669209 x 1038

Entre mayor sea la longitud de una llave, mayor será la dificultad para romperla, pero también será mucho más caro, computacionalmente hablando, el proceso de cifrado y descifrado. La meta es hacer que el costo de romper la llave sea mayor al costo de la información que cifra dicha llave.
Existen 3 tipos de funciones criptográficas que permiten la autenticación, integridad y confidencialidad: Encriptación de llave simétrica, encriptación de llave asimétrica y funciones de hash unidireccionales (se hablará de cada una de ellas más adelante).

Perspectiva Histórica de Políticas Internacionales

A nivel internacional existen diferentes controles o leyes respecto a la exportación e importación de la criptografía. El comité coordinado para el control multilateral de exportación (COCOM, The Coordinating Committee for Multilateral Export Controls) fue una organización internacional que proporcionaba controles de exportación para productos y datos técnicos para los países socios y sus destinos, provee acuerdos para controlar la exportación y manejo de tecnologías como lo son supercomputadoras, chips DSP, criptografía, láser, entre otros. COCOM fue un esfuerzo generalizado de los países occidentales, durante la “guerra fría”, para prevenir el flujo de tecnología en países comunistas. Los miembros de COCOM eran Australia, Bélgica, Canadá, Dinamarca, Francia, Alemania, Grecia, Italia, Japón, Luxemburgo, Holanda, Noruega, Portugal, España, Turquía, Reino Unido, y los Estados Unidos.

Una de las principales reglas de COCOM fue la de prevenir que la criptografía fuese exportada a países “peligrosos” (tales como Libia, Irak, Irán, Corea del Norte). La exportación a otros países por lo general se permitía, aunque en algunos estados se requiere de una licencia para su uso.

En 1991, COCOM adoptó la General Software Note (GSN), la cual permitía exportar software criptográfico entre sus miembros, pero los Estados Unidos, Francia y el Reino Unido crearon leyes por separado. En 1994 COCOM desapareció y algunos países mantienen sus reglas de exportación de la criptografía.

En 1995, 28 países decidieron seguir el trabajo de COCOM y se formó el acuerdo de Wassenaar sobre control de exportaciones para el uso de armas convencionales, mercancías y

tecnologías de uso dual (Wassenaar Arrangement on Export Controls for Conventional Arms and Dual-Use Goods and Technologies) (http://www.wassenaar.org/).

Finalmente en 1996, 31 países firmaron este acuerdo. Dicho acuerdo considera a la criptografía como una mercancía de uso dual (puede ser utilizada para propósitos militares y civiles).

La Organización para la Cooperación Económica y Desarrollo (OECD, Organization for Economic Cooperation and Development) desarrolló la guía para políticas sobre criptografía, esta guía incluye los ocho principios básicos sobre criptografía (el documento completo se encuentra en http://www.oecd.org/). Estas guías son recomendaciones no obligatorias para los gobiernos y no forman parte de las leyes internacionales, estas guías proveen los principios básicos que los países deben tomar en cuenta cuando se desarrollan políticas de criptografía a nivel nacional, los ocho principios básicos son:

1. Los métodos criptográficos deben ser confiables para poder tener confiabilidad en el uso de la información y de los sistemas de comunicación.
2. Los usuarios deben tener derecho a escoger cualquier método de criptografía, bajo las leyes aplicables.
3. Los métodos de criptografía deben ser desarrollados en respuesta a las necesidades, demandas y responsabilidades de los individuos, negocios y gobiernos.
4. Los estándares técnicos, criterios y protocolos para la criptografía deben ser desarrollados y promovidos a nivel nacional e internacional.
5. Los derechos fundamentales de la privacidad de los individuos, incluyendo el secreto de comunicación y protección de datos personales, deben ser respetados en las políticas nacionales de criptografía y en la implementación y uso de métodos criptográficos.
6. Las políticas nacionales sobre criptografía deben permitir acceso legal a texto plano y claves criptográficas de los datos cifrados. Dichas políticas deben respetar los otros principios contenidos en esta guía al mayor grado posible.
7. Si es establecido por contrato o por legislación, las responsabilidades de individuos y entidades que ofrecen servicios de criptografía y aquellos que mantienen o acceden a llaves criptográficas deben ser claramente establecidos.
8. El gobierno debe cooperar para coordinar las políticas de criptografía, como parte de ese esfuerzo, el gobierno debe remover o evitar obstáculos injustificados en nombre de las políticas de la criptografía.

Páginas Electrónicas de Referencia

Anónimo; Linux Máxima Seguridad; Madrid 2000, Pearson Educación, S.A.
Garfimkel, Simson; Spafford, Gene; Seguridad y comercio en el web; 1999, McGrawHill & O'Reilly.

http://world.std.com/~cme/html/timeline.html
http://cwis.kub.nl/~frw/people/koops/lawsurvy.htm
http://www.gilc.org/crypto/crypto-survey.html.
http://www.oecd.org/
http://www.wassenaar.org/

2.2. Criptografía de Llave Pivada.

Encriptación de llave simétrica

También se le conoce como encriptación de llave secreta, utiliza una llave común y el mismo algoritmo para para cifrar y descifrar un mensaje. La figura 1 muestra a dos usuarios, Bob y Alice, que desean comunicarse de manera segura. Ambos se ponen de acuerdo en el algoritmo de cifrado a utilizar. También ambos deben conocer la llave secreta para el cifrado y descifrado.

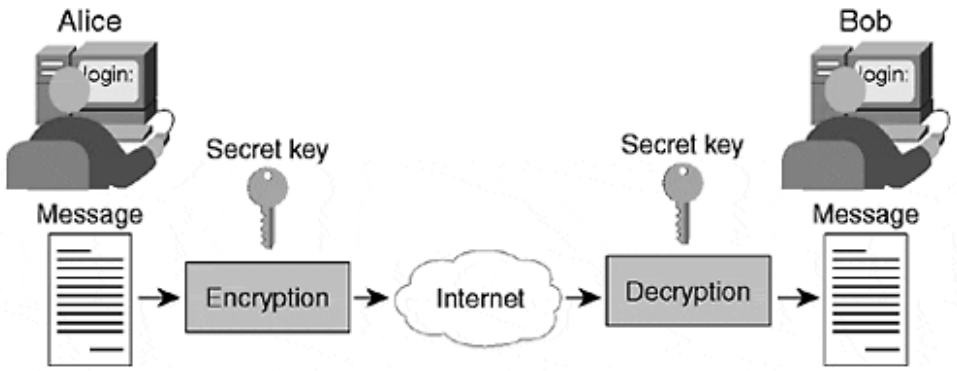


Figura1. Ejemplo de encriptación de llave simétrica.

Un ejemplo de un algoritmo simple de llave privada es el algoritmo de cifrado de César (Caesar Cipher), el cual remplaza cada letra del mensaje original con una letra del alfabeto n lugares más allá. El algoritmo cambia las letras a la derecha o a la izquierda dependiendo (dependiendo si se cifra o descifra). La figura 2 muestra la comunicación entre Alice y Bob utilizando el algoritmo de Cesar y utilizando una llave, n, igual a tres letras. Por ejemplo la letra A se cambia por la letra D. Los pasos de este algoritmo son los siguientes:

- Alice y Bob se ponen de acuerdo en utilizar el algoritmo de César para comunicarse y eligen una llave n=3 como su llave secreta.
- Alice utiliza el algoritmo para cifrar un mensaje confidencial para Bob y se lo envía por correo electrónico.
- Cuándo Bob recibe el correo, descifra el mensaje y lee el mensaje original.

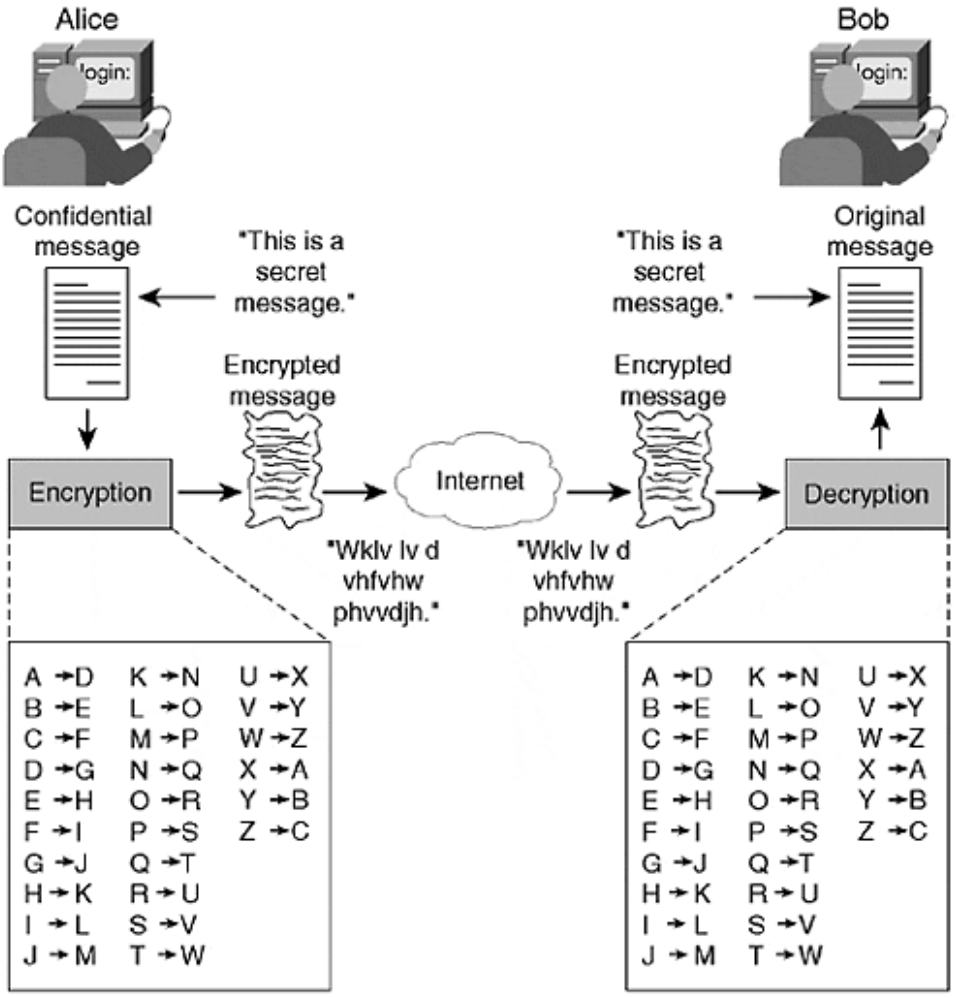


Figura 2. Cifrado utilizando el algoritmo de César

Cualquier persona que intercepte el mensaje sin conocer la llave secreta no podrá leer el mensaje. Sin embargo si alguien intercepta el mensaje cifrado y conoce el algoritmo, podría

hacer un ataque de fuerza bruta para descifrar el mensaje. Si el alfabeto tiene 28 letras tendría que probar 27 llaves diferentes para encontrar la correcta.

Algunos algoritmos de llave secreta trabajan con mensajes en bloques de 64 bits, de esta manera es necesario dividir los mensajes en bloques de 64 bits y después unirlos, lo que permite una protección adicional a los datos.

Existen 4 modos comunes que definen un método de combinar texto plano, la llave secreta y el texto cifrado para generar la cadena del texto cifrado que es transmitida al receptor. Los cuatro modos son:

- Electronic CodeBook (ECB).
- Cipher Block Chaining (CBC).
- Cipher FeedBack (CFB).
- Output FeedBack (OFB).

El mecanismo ECB codifica cada bloque de 64 bits de manera independiente, pero utiliza la misma llave. Esto se ve como una debilidad que puede ser fácilmente explotable para una persona que desea cambiar la información.

Los siguientes tres algoritmos (CBC, CFB y OFB) poseen propiedades inherentes que agregan un elemento de aleatoriedad al mensaje cifrado. Si se envía el mismo bloque de texto plano utilizando alguno de esos algoritmos, se obtienen diferentes bloques de texto cifrado cada vez.

Esto es logrado utilizando diferentes llaves de cifrado o un vector de inicialización (IV). Un IV es un bloque cifrado de datos aleatorios utilizados como los primeros 64 bits del bloque para iniciar el proceso de encadenamiento. Si alguien escucha la información que pasa, tendrá siempre información diferente, por lo que prácticamente no tendrá información relevante.

A continuación se listan algunos algoritmos de llave simétrica:

- ***DES*** *(Data Encryption Standard, Estándar de Encriptación de Datos)*: fue adoptado como estándar del gobierno de los Estados Unidos en 1977 y como estándar ANSI en 1981. DES es un algoritmo de bloque que utiliza una llave de 56 bits y tiene varios modos de operación, de acuerdo con el propósito que se utilice. DES es considerado como un algoritmo de cifrado fuerte (puede encontrarse mas información de DES en http://www.itl.nist.gov/fipspubs/fip46-2.htm y http://encyclopedia.thefreedictionary.com/Data%20Encryption%20Standard).

- ***DESX*:** Consiste en una modificación al algoritmo DES que mejoran la seguridad del algoritmo, haciendo casi imposible descifrar la llave.

- ***Triple-DES*:** Proporciona una forma de, por lo menos, duplicar la seguridad de DES, mediante el uso del algoritmo DES tres veces con tres diferentes llaves (utilizar simplemente DES dos veces con dos llaves diferentes no es fiable ya que no mejora la seguridad al grado que se podría esperar, debido a un tipo teórico de ataque conocido como "encontrarlo a la mitad", en el cual un atacante intenta, de modo simultáneo, encriptar el texto plano con una sola operación DES y descifrar el mensaje cifrado con otra operación DES, hasta que se encuentra una correspondencia a la mitad). Este algoritmo se emplea por lo general en instituciones financieras y gobierno como una alternativa a DES (más información sobre Triple-DES en http://csrc.nist.gov/cryptval/des.htm).

- ***Blowfish*:** Es un algoritmo de bloque rápido, compacto y sencillo, inventado por Bruce Schneier. Permite tener una llave de longitud variable de hasta 448 bits, y está optimizado para ejecutarse en procesadores de 32 y 64 bits. El algoritmo se ha puesto en el dominio público (más información sobre Blowfish puede encontrarse en http://www.schneier.com/blowfish.html).

- ***IDEA*** *(International Data Encryption Algorithm, Algoritmo Internacional de Encriptación de datos)*: fue desarrollado en Zurich, Suiza, por James L. Massey y Xuejia Lai, y públicado en 1990. Utiliza una llave de 128 bits que es muy segura. Este algoritmo se utiliza dentro de PGP (http://web.mit.edu/network/pgp.html, www.pgp.com) para cifrar y descifrar mensajes vía correo electrónico. El principal problema con IDEA es una serie de patentes de software que existen sobre el algoritmo, lo que ha obstaculizado su utilización (más información http://en.wikipedia.org/wiki/International_Data_Encryption_Algorithm).

- ***RC2*** *(Rivest Cipher 2)*: Ronals Rivest desarrolló originalmente este algoritmo de bloque y RSA lo mantuvo como secreto comercial. Después se divulgó en un mensaje anónimo en Usenet en 1986. RC2 se vende con una implementación que permite llaves entre 1 y 2048 bits, por lo general la llave se limita a 40 bits en el software que se vende fuera de los Estados Unidos (mayor información de este algoritmo puede ser encontrada en http://theory.lcs.mit.edu/~rivest/crypto-security.html).

- ***RC4*** *(Rivest Cipher 4):* Es un algoritmo de flujo fue desarrollado por Ronals Rivest y RSA Data Security lo mantuvo como secreto comercial. Al igual que RC2 fue divulgado de manera anónima por Usenet en 1994. Permite llaves de entre 1 y 2028 bits (mayor información de este algoritmo puede ser encontrada en http://theory.lcs.mit.edu/~rivest/crypto-security.html).

- ***RC5*** *(Rivest Cipher 5):* Este algoritmo también fue desarrollado por Ronals y públicado en 1994. Permite que la longitud de la llave, el tamaño del bloque de datos y el número de pasadas de encriptación las defina el usuario.

Ataques contra los algoritmos simétricos de encriptación

Si se utiliza la criptografía para proteger información, se debe presuponer que quienes no se desean que accedan a la información registrarán los datos encriptados e intentarán desencriptarlos por la fuerza. Para ser útil, un sistema criptográfico debe ser resistente a este tipo de ataque directo.

Lo ataques contra la información encriptada se clasifican en tres categorías principales:

- Ataques de búsqueda de llaves (fuerza bruta).
- Criptoanálisis.
- Ataques basados en el sistema.

Ataques de búsqueda de llaves (fuerza bruta)

La forma más fácil de romper un código es probar todas las llaves posibles, una tras otra (suponiendo que el violador de códigos puede reconocer el resultado de utilizar la llave correcta). Casi todos los intentos fallaran, pero al final uno tendrá éxito y permitirá al violador acceder al sistema o desencriptar el texto cifrado. Estos *ataques,* se conocen como *ataques de búsqueda de llaves o ataques de fuerza bruta.*

No hay modo de protegerse contra un ataque de búsqueda de llaves, ya que no hay forma de evitar que un atacante intente desencriptar el mensaje probando todas las llaves posibles.

Los ataques de búsqueda de llaves no son muy efectivos. Algunas veces ni siquiera son posibles: en muchas ocasiones simplemente existen demasiadas llaves que probar y no hay tiempo para probarlas todas. Por otra parte, muchos ataques de búsqueda de llaves se simplifican considerablemente porque la mayoría de los usuarios eligen llaves basadas en claves de acceso pequeñas con caracteres legibles.

Si se considera el algoritmo de encriptación RC4, utilizado comúnmente por los navegadores para encriptar la información enviada a través del World Wide Web, RC4 puede emplearse con cualquier longitud de llave entre 1 y 2048 bits, pero, en general, se utiliza con llaves secretas de 40 o de 128 bits.

Con una llave de 40 bits, existen 2^{40} (1.1×10^{12}) llaves posibles que pueden utilizarse. Con una computadora que se puede adquirir en cualquier tienda especializada, capaz de probar un millón de llaves por segundo, se pueden probar todas las llaves en menos de 13 días. Carl

Ellison señala que en 1994, un ingeniero que tenía 20,000 dolares en partes construyo un motor de búsqueda RC4 que podía procesar 150 millones de llaves por segundo. En 1997, se rompió un código de 40 bits en 3.5 horas. Es evidente que una llave de 40 bits puede ser blanco de un ataque de búsqueda de llaves.

Por otra parte, una llave de 128 bits es muy resistente a un ataque de búsqueda de llaves, pues permite 2^{128} (3.4×10^{38}) llaves posibles. Aun si existiera una computadora que pudiera probar mil millones de llaves por segundo y si se contara con mil millones de semejantes computadoras, tomaría 1013 años probar todas las llaves RC4 de 128 bits posibles. Este lapso es aproximadamente mil veces mayor que la edad del universo, que se estima en 1.8×10^{10} años.

A partir de este sencillo análisis, parece ser que RC4 con una llave de 128 bits debe ser suficiente para la mayoría de las necesidades criptográficas -ahora y siempre-. Por desgracia, hay varios factores que hacen que esta solución no sea técnica, legal o políticamente conveniente para muchas aplicaciones.

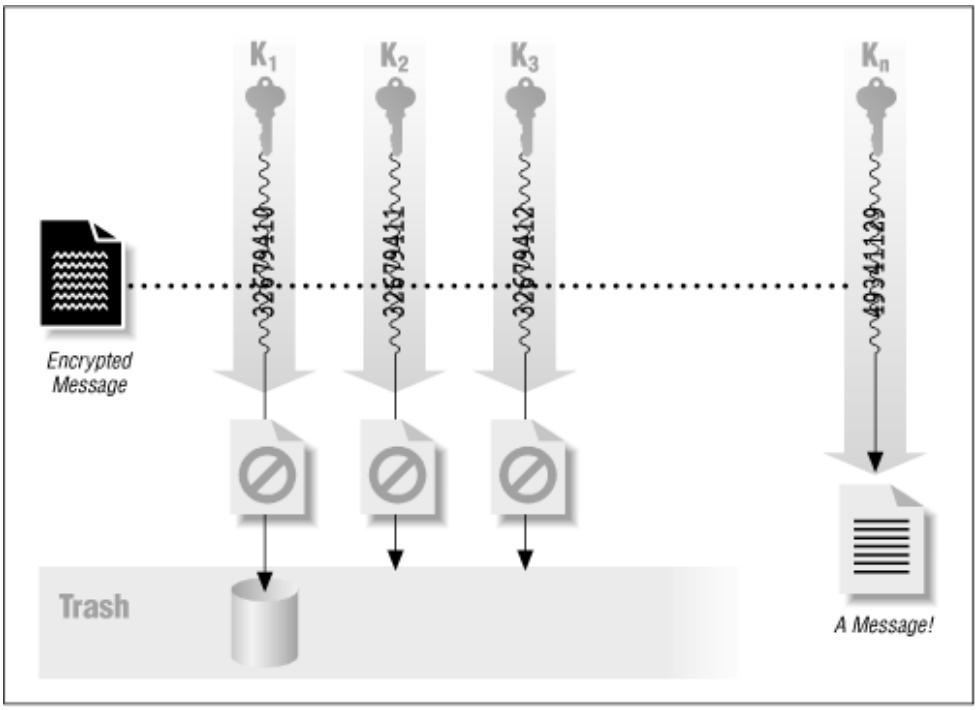


Figura 3. Ataque de fuerza bruta

Criptoanálisis

Si la longitud de la llave fuera el único factor determinante para la seguridad de un código, cualquiera que estuviera interesado en intercambiar mensajes secretos simplemente utilizaría códigos de llaves de 128 bits y todos los criptoanálisis (personas que rompen códigos) tendrían

que buscar nuevos trabajos. La criptografía sería una rama resuelta de las matemáticas, al igual que la adición simple.

Lo que mantiene vigente el interés en la criptografía es el hecho de que la mayoría de los algoritmos de encriptación no satisfacen nuestras expectativas. Los ataques de búsqueda de llaves rara vez son necesarios para acceder al contenido de un mensaje encriptado. Más bien, es posible vencer a la mayoría de los algoritmos de encriptación mediante una combinación de matemáticas complejas y poder de cómputo. El resultado es que muchos mensajes encriptados pueden descifrarse sin conocer la llave. En ocasiones un criptoanalista hábil puede descifrar texto encriptado sin conocer siquiera el algoritmo de encriptación.
Un ataque criptoanalítico puede tener dos metas. El criptoanalista pude tener texto cifrado y quiere descubrir el texto llano, o tener el texto cifrado y buscar la llave utilizada para encriptarlo (estas metas son similares, pero no son exactamente idénticas). Los siguientes ataques se utilizan por lo común cuando se conoce el algoritmo de encriptación, y pueden lanzarse contra el tráfico del WWW.

Ataque de texto llano conocido

En este tipo de ataque, el criptoanalista tiene un bloque de texto llano y el bloque correspondiente de texto cifrado. Aunque esto parezca poco probable, en realidad es muy común, utilizar la criptografía para proteger correo electrónico (con encabezados estándar al principio del mensaje) o discos duros (con estructuras conocidas en lugares predeterminados del disco). La meta de un ataque de texto llano conocido es determinar la llave criptográfica (y posiblemente el algoritmo), los cuales pueden emplearse entonces para desencriptar otros mensajes.

Ataque de texto llano elegido

En este tipo de ataque, el criptoanalista hace que el sujeto del ataque encripte (sin saberlo) bloques selectos de datos, creando un resultado que el criptoanalista puede entonces analizar. Llevar a cabo un ataque de texto llano elegido es más fácil de lo que podría parecer (por ejemplo, el sujeto del ataque puede ser una conexión de radio que encripta y retransmite mensajes recibidos por teléfono). La meta de un ataque de texto llano elegido es determinar la llave criptográfica, la cual puede entonces ocuparse para desencriptar otros mensajes.

Criptoanálisis diferencial

En este ataque, que es una variante del ataque de texto llano elegido, se encriptan muchos textos ligeramente diferentes entre si y se comparan los resultados.

Análisis diferencial de fallas

Este ataque se lleva a cabo contra sistemas criptográficos integrados al hardware. Se somete al dispositivo a factores ambientales (calor, fuerza, radiación) para que cometan errores

durante la encriptación o desencriptación. Estas fallas pueden analizarse para, posiblemente, obtener a partir de ellas el estado interno del dispositivo, incluyendo la llave o algoritmo de encriptación.

La única forma confiable de determinar si un algoritmo es resistente o no, es públicarlo y esperar que alguien encuentre una debilidad. Este proceso de revisión todavía no es perfecto, pero es mejor que ninguna revisión. No se debe confiar en las personas que dicen haber desarrollado un nuevo algoritmo de encriptación pero no pueden divulgarlo porque pondría en peligro su seguridad. Si el algoritmo se utiliza para almacenar información valiosa, un atacante comprará (o robará) una copia del programa que lo implemente, lo desensamblará, y averiguará como funciona. Como en el caso de RC2 y RC4, ¡el atacante puede incluso públicar el algoritmo obtenido mediante la ingeniería inversa! La verdadera seguridad criptográfica yace en la apertura y la revisión pública.

Ataques basados en el sistema

Otra forma de romper un código es atacar el sistema criptográfico que utiliza el algoritmo criptográfico, sin atacar el algoritmo en si.

Uno de los casos mas espectaculares de ataques basados en en el sistema fue el algoritmo de encriptación de vídeo VC-1, utilizado para encriptar las primeras transmisiones de televisión vía satélite. Durante años, los piratas de vídeo vendían decodificadores que interceptaban la transmisión de las llaves y las utilizaban para desencriptar las transmisiones.

El algoritmo de encriptación VC-1 era seguro, pero el sistema en su conjunto era débil (este caso también demuestra que, cuando se trata de mucho dinero, algunas personas probablemente encontraran las fallas en un sistema débil de encriptación y las explotarán).

Muchos de los ataques iniciales contra la instrumentación de SSL de Netscape fueron en realidad ataques contra la implementación de Navigator, no el protocolo SSL en si. En un ataque públicado, los investigadores Wagner y Goldberg, de Berkeley, descubrieron que el generador de números aleatorios de Navigator de Netscape no era en realidad aleatorio. Era posible que los atacantes monitorizaran de cerca la computadora en la que se ejecutaba Navigator, predijeran la configuración inicial del generador de números aleatorios y con ello determinaran la llave aleatoria por un método sencillo. En otro ataque, los investigadores descubrieron que podían modificar fácilmente el programa, de forma que no se ejecutara el generador de números aleatorios, eliminando de raíz la necesidad de adivinar la llave

2.3. Criptografía de Llave Pública

La existencia de la criptografía de llave publica, también conocida como **encriptación asimétrica,** fue presentada por primera vez en forma impresa en el otoño de 1975 por Whitfield Diffie y Martin Hellman. Ellos, que en ese tiempo estaban en la Universidad de Stanford, redactaron un documento en el que presuponían la existencia de una técnica de encriptación en la cual la información encriptada con una llave podría ser desencriptada solo con otra llave aparentemente no relacionada con la primera. Robert Merkle, es ese tiempo estudiante graduado de Berkeley, tenia ideas similares, pero debido a los caprichos del sistema académico de publicación, los textos de Merkle no se publicaron sino hasta que la idea de la encriptación de llave publica era muy conocida.

Se pueden utilizar los mismos algoritmos, o diferentes pero complementarios para cifrar y descifrar los datos, pero se requieren dos diferentes, pero relacionadas, llaves: una llave privada y una llave pública. Por ejemplo, si Alice y Bob quieren comunicarse utilizando un algoritmo de llave pública, necesitan tener, cada uno, su propia llave pública y su propia llave privada. Cuando se comuniquen uno con el otro de manera segura, Alice y Bob utilizan diferentes llaves para cifrar y descifrar los datos. Esto puede observarse en la figura 1.

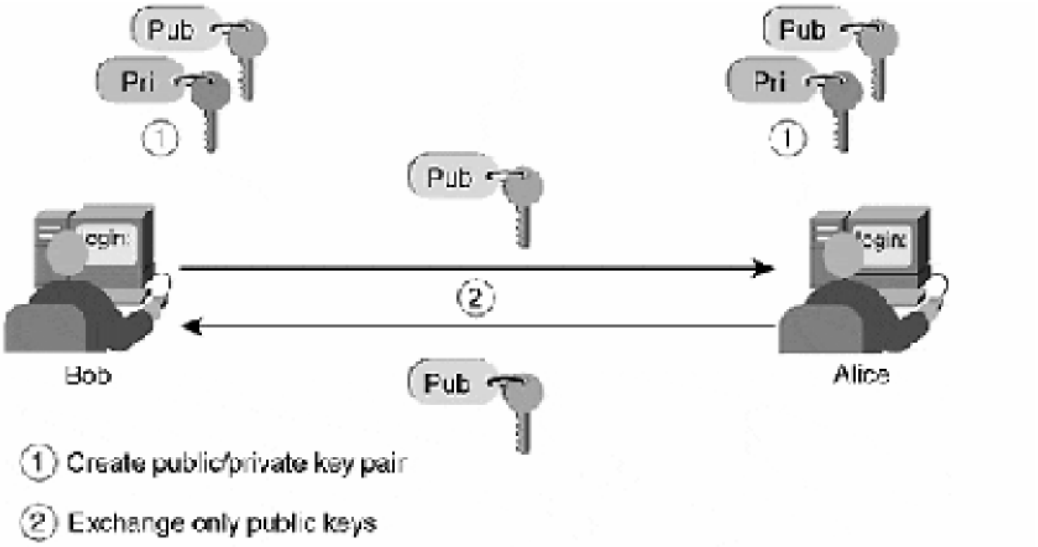


Figura 1. Ejemplo de comunicación de llave pública entre Alice y Bob

Los siguientes pasos son necesarios para que Alice y Bob intercambien datos de manera confidencial:

1. Tanto Alice como Bob deben crear su llave pública y privada.
2. Alice y Bob intercambian sus llaves públicas.
3. Alice escribe un mensaje a Bob y utiliza la llave pública de Bob para cifrar el mensaje y lo envía a través de Internet.

4. Alice utiliza su llave privada para descifrar el el proceso anterior Bob utiliza su llave privada para descifrar el mensaje y lee el mensaje.
5. Bob escribe su respuesta para Alice y cifra su mensaje utilizando su llave pública. Envía su mensaje por Internet.
6. Alice utiliza su llave privada para descifrar el mensaje.

El proceso anterior se muestra en la figura 2:

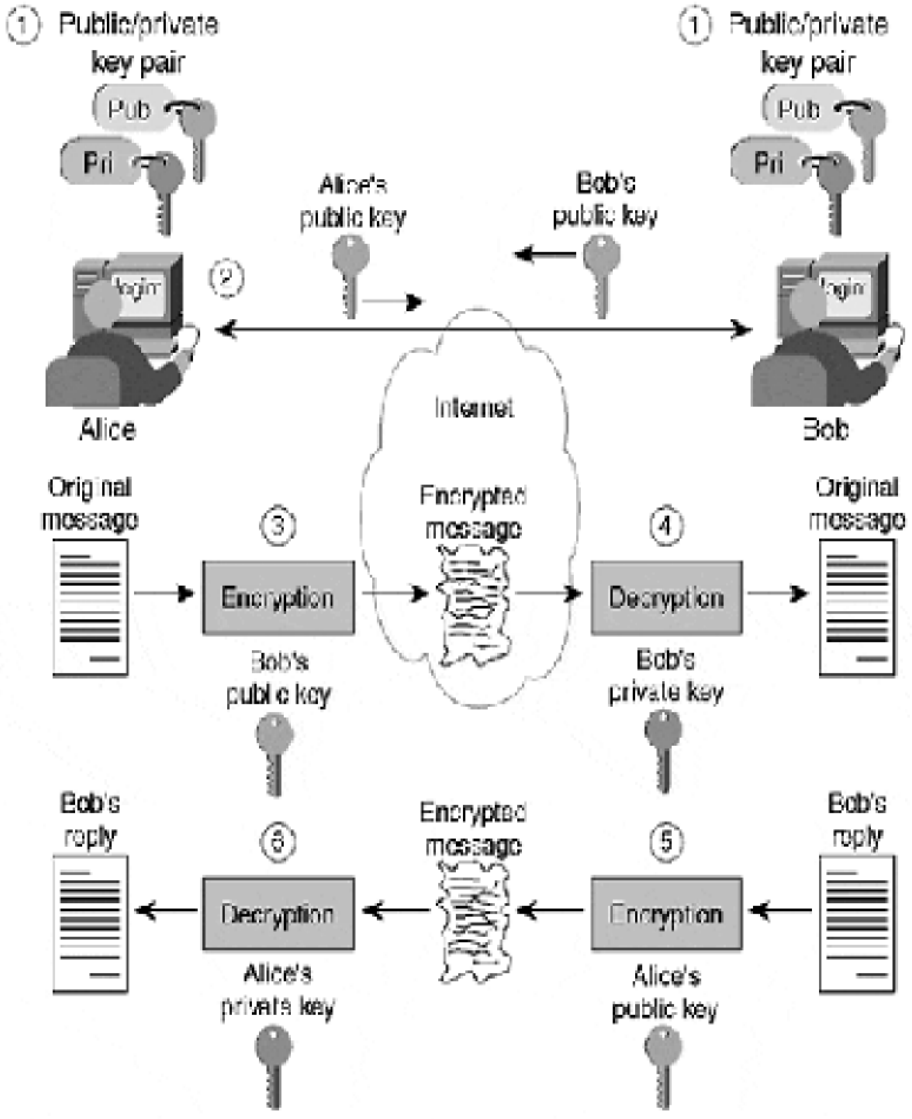


Figura 2. Comunicación entre Alice y Bob utilizando un algoritmo de llave secreta.

La confidencialidad de los datos esta asegurada ya que Alice al enviar el mensaje inicial ya que solo Bob puede descifrar el mensaje con su llave privada. La integridad de los datos también se conserva ya que para modificar el mensaje, un atacante, necesitaría la lleve privada de Bob.

De igual forma la integridad y confidencialidad se asegura al regresar el mensaje de regreso ya que el mensaje solo puede descifrarse con la llave privada de Alice.

Algo importante de resaltar es el hecho de que una tercera persona podría pretender ser Alice, ya que las llaves públicas por lo general esta disponibles a cualquier individuo, por lo que es muy importante el verificar que fue la persona que mando el mensaje original quien dice ser. La figura 3 muestra como con la criptografía de llave pública se resuelve este problema, al proveer de autenticación de envío y no-repudio.

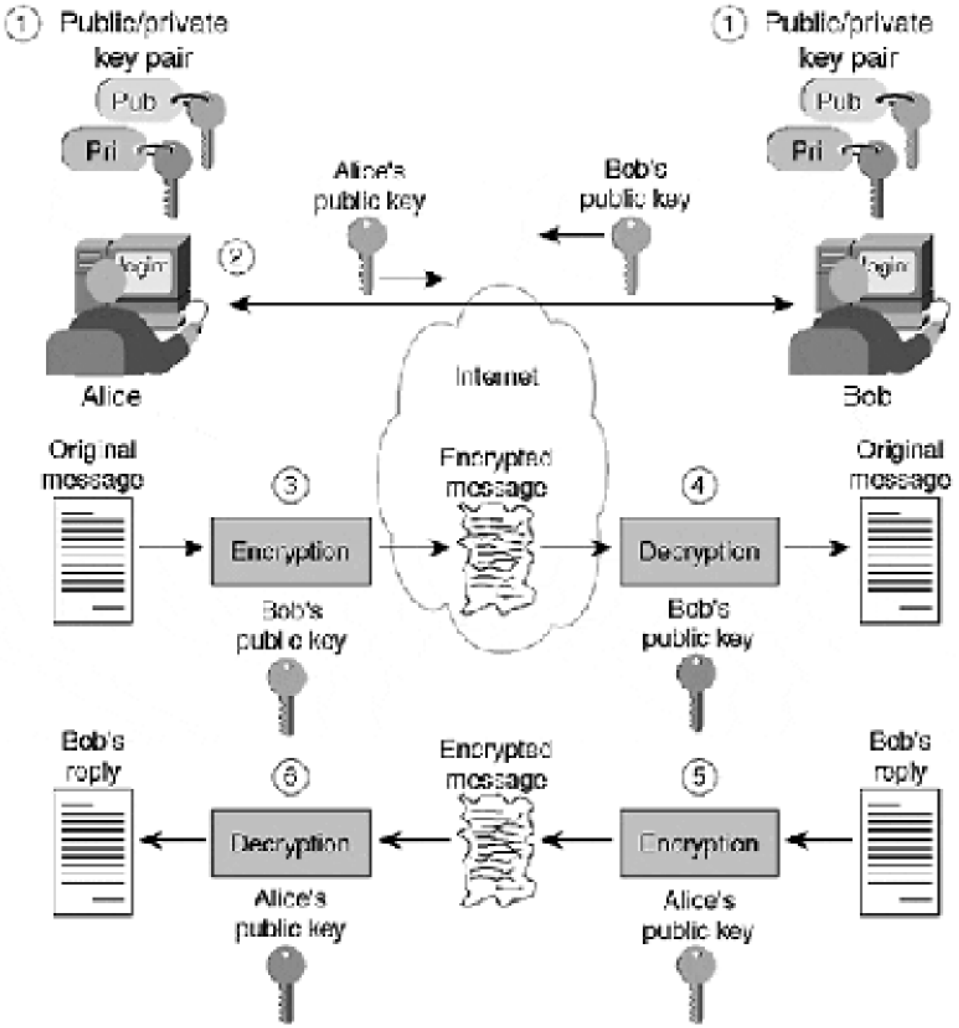


Figura 3. Autenticación del emisor y no-repudio.

Cómo puede verse, en el cifrado asimétrico la llave privada es la más importante y que debe permanecer secreta. Si esta llave se compromete alguien podría hacerse pasar por ti.

El Mecanismo para generar el par de llaves (la llave pública y la llave privada) es complejo y el resultado son dos números aleatorios muy grandes. Dado que estos números deben preservar la unicidad de cada par de llaves pública y privada, deben aplicarse criterios matemáticos rigurosos para la generación de estos números y este proceso conlleva un trabajo de procesador intensivo. Los criterios matemáticos aseguran que las llaves generadas no son débiles.

La encriptación de llave pública no es tan difundida para datos confidenciales debido a sus restricciones de rendimiento, ya que el proceso de cifrado y descifrado es suele ser lento.

Este tipo de cifrado se utiliza en aplicaciones referentes a firmas digitales y administración de llaves.

Desde que aparecieron estos algoritmos se han desarrollado varios sistemas de encriptación. Por desgracia, ha habido mucho menos desarrollo en los algoritmos de llave pública que en los de llaves simétricas. La razón tiene que ver con la forma en que se diseñan estos algoritmos. Los buenos algoritmos de llaves simétricas simplemente revuelven su entrada con base en la llave de entrada; el desarrollo de un nuevo algoritmo de llaves simétricas solo requiere idear nuevas formas de hacer esa revoltura en forma confiable. Los algoritmos de llave pública por lo general se basan en la teoría numérica. El desarrollo de nuevos algoritmos de llave pública requiere la identificación de nuevos problemas matemáticos con propiedades específicas.
A continuación, se resumen los sistemas de llave pública que se utilizan comúnmente en la actualidad:

Intercambio de llaves Diffie-Hellman
Sistema para intercambio de llaves criptográficas entre partes activas. Diffie-Hellman no es en realidad un método de encriptación y desencriptación sino un método para desarrollar e intercambiar una llave privada compartida mediante un canal de comunicación público. A fin de cuentas, ambas partes acuerdan utilizar algunos valores numéricos comunes, y luego cada una de ellas crea una llave. Se intercambian transformaciones matemáticas de las llaves. Cada parte puede calcular entonces una tercera llave de sesión, la cual no puede ser fácilmente derivada por un atacante que conozca ambos valores intercambiados.

RSA
RSA es un sistema criptográfico de llave pública bien conocido, desarrollado por Baldona Rivest, Ada Shamir y Leonard Adleman, entonces profesores del Instituto Tecnológico de Massachussets. RSA puede ocuparse para encriptar información y como fundamento de un sistema de firmas digitales. Las firmas digitales pueden utilizarse para probar la autoría y autenticidad de información digital. La llave puede tener cualquier longitud, dependiendo de la implementación que se utilice.

ElGamal
Llamado así en honor a su creador, Taher ElGamal, este es un sistema de encriptación de llave publica basado en el protocolo de intercambio de llaves Diffie-Hellman. En forma similar a RSA, ElGamal puede emplearse tanto para encriptación como para firmas digitales.

DSS
El Estándar de Firmas Digitales (DSS, Digital Signature Standard) fue desarrollado por la Agencia de Seguridad Nacional de Estados Unidos (NSA, National Security Agency) y adoptado como Estándar de Procesamiento de Información Federal (FIPS) por el Instituto Nacional de Estándares y Tecnología (NIST). DSS se basa en el algoritmo de Firmas Digitales (DSA). Aunque DSA permite utilizar llaves de cualquier longitud, bajo el DSS de FIPS solo se permiten llaves de 512 y 1024 bits. DSS especifica que solo puede ocuparse para firmas digitales, aunque es posible utilizar implementaciones de DSA también para encriptación.

Ataques contra Algoritmos de llave pública

En teoría, los algoritmos de llave pública son mas fáciles de atacar que los de llave simétrica, pues puede presuponerse que el atacante tiene una copia de la llave publica utilizada para encriptar el mensaje. La labor del atacante se simplifica aun mas, ya que el mensaje tal vez indica que algoritmo de llave publica se uso para encriptarlo.
Los ataques contra algoritmos de llave pública se dividen en general en dos categorías:

- Ataques de factorización
- Ataques algorítmicos

Los ataques de factorización son el tipo mas popular de ataque contra mensajes encriptados mediante llave publica porque son los mas fáciles de comprender. Estos ataques intentan derivar una llave secreta a partir de su llave pública correspondiente.

En el caso del sistema de llave publica RSA, este ataque puede instrumentarse mediante la factorización de un número asociado con la llave pública. Con otros sistemas de llave publica, este ataque necesita resolver otros tipos de problemas matemáticos de alta dificultad.

Hoy en día, la fortaleza del popular algoritmo RSA depende de la dificultad de factor izar números grandes. El problema de este tipo de factorización ha llamado la atención de los matemáticos durante siglos y con seguridad seguir siendo materia de interés. Se han encontrado algunos métodos eficientes para factorizar clases muy pequeñas de números con propiedades especiales, pero el problema en general de la factorización de números aun se considera "difícil". No obstante, no se ha mostrado prueba alguna de que la factorización de números sea realmente "difícil" desde un punto de vista computacional, por lo que puede llegar un momento en que sea necesario descartar RSA a favor de algún nuevo algoritmo.

2.4. Firmas Digitales

Las firmas digitales son uno de los elementos claves para el desarrollo de transacciones financieras en línea, correo electrónico y de negocios. Una firma digital es un identificador electrónico que utiliza criptografía para asegurar la integridad, autenticidad y no repudio de la información. Los requerimientos legales de una firma o de otro método basado en papel de autenticación es por lo general un obstáculo para el uso de tecnologías electrónicas. En muchas partes del mundo se han hecho esfuerzos por legislar leyes para eliminar la necesidad de firmas escritas.

Las firmas digitales son análogas a las firmas manuscritas. Estas últimas están basadas en la forma física en que la persona firma su nombre. Pero pueden ser fácilmente falsificadas.

Una firma digital es una precisa forma matemática de adjuntar la identidad de una persona a un mensaje. Son mucho más difíciles de falsificar que las firmas escritas, y el mensaje firmado no puede ser modificado sin invalidar la firma.

Las firmas digitales se basan en la criptografía de clave pública. Este tipo de sistemas criptográficos utiliza dos "claves". Al tomar un mensaje y aplicarle una de las claves en el proceso de cifrado, se obtendrá un mensaje distorsionado o "cifrado" (o, en el contexto actual, "firmado"). Aplicándole la otra clave al mensaje distorsionado en un proceso de descifrado y se obtendrá el mensaje original.

Una de esas claves es "*pública*". Todo el mundo conoce esa clave --o puede obtenerla, como si fuera un número de teléfono. La otra clave es "*privada*". Solo tú conoces tu clave privada. Al firmar (cifrar) algo con tu clave privada, le estas poniendo tu sello personal. Nadie más puede hacerlo, ya que ellos no conocen tu clave privada.

Tratando el mensaje

Por razones prácticas, cuando se utilizan las firmas digitales la gente no firma (cifra) realmente el mensaje original con su clave privada. Primero se produce una versión condensada del mensaje. Esta versión condensada a menudo se llama un "resumen del mensaje" (*digest*). Otros términos equivalentes son *'hash* del mensaje"o "huella digital".

Lo que se firma, entones, es el resumen del mensaje. El usuario firma el resumen del mensaje con su clave privada. Cualquiera puede descifrar el resumen del mensaje utilizando la clave pública del Firmante.

Existen diferentes algoritmos para resumir mensajes. Más adelante veremos ejemplos de cuatro de ellos: MD5, RIPEM128, SHA-1 y RIPEM168.

Así es como funciona el proceso de firma. Supongamos que tenemos un emisor y un receptor de un mensaje. El receptor desea verificar que el mensaje no haya sido alterado, así como la identidad del remitente.

Emisor

- S1. El emisor envía un mensaje.
- S2. El emisor calcula un resumen (*hash*) del mensaje original.
- S3. El emisor firma este resumen del mensaje con su clave privada, y envía el resumen firmado junto con el mensaje original.

Receptor

- R1. El receptor recibe el mensaje, así como el resumen firmado.
- R2. El receptor también calcula por separado un resumen del mensaje (*hash*) para el mensaje recibido.
- R3. El receptor utiliza la clave pública del remitente para descifrar el resumen firmado del mensaje que recibió, y compara el resultado con el resumen calculado en el paso de progresión R2. Si el resumen del mensaje calculado es igual que el valor descifrado del resumen firmado, entonces el receptor ha verificado que el mensaje no se ha alterado. Además, puesto que la clave pública del remitente fue utilizada para el descifrado, el receptor puede estar seguro de la identidad del remitente.

Numerando

Para poder realizar una firma digital, es necesario primero convertir el mensaje en un Número. Este Número es entregado a la función de *Hash*, que produce el resumen del mensaje. Esta función convierte un número grande (el mensaje) en un número pequeño (el resumen).

Para que esto funcione, no debería ser sencillo encontrar dos mensajes que produjeran el mismo resumen. Si se pudiera hacer, podrías cambiar el mensaje correspondiente a una firma, como aquel banco que cambió páginas internas del contrato.
El número pequeño del resumen suele tener una longitud de 128 bits (MD5, RIPEM128), o de 160 bits (SHA-1, RIPEM160). Cada bit puede ser tanto un "0" como un "1". Por lo tanto existen 2 elevado a 128 posibles resúmenes de 128 bits de largo, o 2 elevado a 160 resúmenes de 160 bits.

Ahora bien, el proceso de obtención del resumen a partir del mensaje debe ser determinístico. Debe ser repetible. El mismo mensaje siempre debe dar el mismo resumen. Sino, el proceso de verificación no funcionaría.

Pero, al mismo tiempo, la salida de la función de *hash* debe parecer aleatoria. Debería resultar imposible obtener el mensaje a partir del resumen. De otra manera, alguien podría obtener varios mensajes que tendrían el mismo resumen.

Para que una función de *hash* sea buena, debe ser una función unidireccional. Debe funcionar en un sentido, pero no en el contrario. Además debe ser muy difícil encontrar dos mensajes diferentes que produzcan el mismo resumen.

Cuando ya dispone del resumen del mensaje, el número pequeño, debe firmarlo (cifrarlo). Esto también involucra una transformación matemática. Una forma muy común de cifrar un número (por ejemplo el algoritmo RSA) consiste en elevar ese número a una potencia, y después tomar el resultado de la exponenciación modulo n, donde n es un entero grande. (Módulo n significa dividir el número por n, y tomar el resto de la división. Por ejemplo, 11 modulo 5 es igual a 1, porque el cociente es 2 y el resto es 1.).

Ejemplo de Mensaje

Suponiendo que el mensaje que debe ser firmado es la siguiente carta del Gran Banco del Caribe al Senador Hand N. Till

```
**********
Senator Hand N. Till
Washington, DC

Dear Senator:

This letter is to inform you we have
opened account no. 338907021 on your
```

```
behalf, and deposted therein a
legislative incentive payment of
$1 million.  For withdrawl, you will
need to use your password BJRGUD7693.

Yours very truly,

Arnold C. Creole
Vice President
Greater Caribbean Bank
```

El mensaje empieza por la "S" de "Senator" y termina pon la "k" de "Bank". Cada línea excepto la última acaba con un retorno de carro (CR) y un salto de línea (LF), que indica el fin de la actual línea y el principio de la siguiente.

Destacar que hay una falta de ortografía en la carta: "withdrawl" en lugar de "withdrawal". Pero el mensaje ha sido firmado tan pronto se ha acabado de escribir, si se corrige la falta de ortografía, la firma no se correspondería con la carta.

Se verán ahora los pasos seguidos para firmar esta carta ("mensaje"). Primero, se quiere convertir el mensaje en un número. Existen distintos métodos. Pero la forma más simple consiste en utilizar la codificación ASCII de los caracteres que forman el mensaje. Cada código ASCII es un número binario. Después de esta conversión, se pueden concatenar todos los números binarios para obtener un gran número.

En la primera frase, la traducción es la siguiente:

Carácter	Valor Binario	Valor Hex	Valor Decimal
S	1010011	53	83
e	1100101	65	101
n	1101110	6E	110
a	1100001	61	97
t	1110100	74	116
o	1101111	6F	111
r	1110010	72	114
	0100000	20	32
H	1001000	48	72
a	1100001	61	97
n	1101110	6E	110
d	1100100	64	100
	0100000	20	32
N	1001110	4E	78
.	0101110	2E	46
	0100000	20	32
T	1010100	54	84
i	1101001	69	105
l	1101100	6C	108
l	1101100	6C	108
CR	0001101	0D	13
LF	0001010	0A	10

La representación binaria de todo el mensaje para el Senador Till es:

```
1010011  1100101  1101110  1100001  1110100  1101111 1110010
0100000  1001000  1100001  1101110  1100100  0100000 1001110
0101110  0100000  1010100  1101001  1101100  1101100 0001101
0001010  1010111  1100001  1110011  1101000  1101001 1101110
1100111  1110100  1101111  1101110  0101100  0100000 1000100
1000011  0001101  0001010  0001101  0001010  1000100 1100101
1100001  1110010  0100000  1010011  1100101  1101110 1100001
1110100  1101111  1110010  0111010  0001101  0001010 0001101
0001010  1010100  1101000  1101001  1110011  0100000 1101100
1100101  1110100  1110100  1100101  1110010  0100000 1101001
1110011  0100000  1110100  1101111  0100000  1101001 1101110
1100110  1101111  1110010  1101101  0100000  1111001 1101111
1110101  0100000  1110111  1100101  0100000  1101000 1100001
1110110  1100101  0001101  0001010  1101111  1110000 1100101
1101110  1100101  1100100  0100000  1100001  1100011 1100011
1101111  1110101  1101110  1110100  0100000  1101110 1101111
0101110  0100000  0110011  0110011  0111000  0111001 0110000
0110111  0110000  0110010  0110001  0100000  1101111 1101110
0100000  1111001  1101111  1110101  1110010  0001101 0001010
1100010  1100101  1101000  1100001  1101100  1100110 0101100
0100000  1100001  1101110  1100100  0100000  1100100 1100101
1110000  1101111  1110011  1110100  1100101  1100100 0100000
1110100  1101000  1100101  1110010  1100101  1101001 1101110
0100000  1100001  0001101  0001010  1101100  1100101 1100111
1101001  1110011  1101100  1100001  1110100  1101001 1110110
1100101  0100000  1101001  1101110  1100011  1100101 1101110
1110100  1101001  1110110  1100101  0100000  1110000 1100001
1111001  1101101  1100101  1101110  1110100  0100000 1101111
1100110  0001101  0001010  0100100  0110001  0100000 1101101
1101001  1101100  1101100  1101001  1101111  1101110 0101110
0100000  0100000  1000110  1101111  1110010  0100000 1110111
1101001  1110100  1101000  1100100  1110010  1100001 1110111
1101100  0101100  0100000  1111001  1101111  1110101 0100000
1110111  1101001  1101100  1101100  0001101  0001010 1101110
1100101  1100101  1100100  0100000  1110100  1101111 0100000
1110101  1110011  1100101  0100000  1111001  1101111 1110101
1110010  0100000  1110000  1100001  1110011  1110011 1110111
1101111  1110010  1100100  0100000  1000010  1001010 1010010
1000111  1010101  1000100  0110111  0110110  0111001 0110011
0101110  0001101  0001010  0001101  0001010  1011001 1101111
1110101  1110010  1110011  0100000  1110110  1100101 1110010
1111001  0100000  1110100  1110010  1110101  1101100 1111001
0101100  0001101  0001010  0001101  0001010  1000001 1110010
1101110  1101111  1101100  1100100  0100000  1000011 0101110
0100000  1000011  1110010  1100101  1101111  1101100 1100101
0001101  0001010  1010110  1101001  1100011  1100101 0100000
1010000  1110010  1100101  1110011  1101001  1100100 1100101
1101110  1110100  0001101  0001010  1000111  1110010 1100101
1100001  1110100  1100101  1110010  0100000  1000011 1100001
1110010  1101001  1100010  1100010  1100101  1100001 1101110
0100000  1000010  1100001  1101110  1101011
```

Notar que los espacios que hay entre los números binarios no tienen ningún tipo de significado. Sólo son para que resulte más fácil su lectura. El número realmente es una larga tira de ceros y unos, todos seguidos, sin espacios.

Las cadenas de ceros y unos ocupan mucho espacio, por eso normalmente la gente no utiliza representación binaria. Los números hexadecimales (base 16) necesitan una cuarta parte de los carácteres que precisa la representación en binario. Por este motivo escribiremos el número correspondiente al mensaje en representación hexadecimal en lugar de en binario. Remarcar que la notación hexadecimal utiliza los numerales del 0 al 9 y las letras de la "a" a la "f", donde "a" equivale al decimal 10,"b" al decimal 11, y así hasta llegar a la "f", que equivale al decimal 15. (Es muy común que los números hexadecimales se escriban utilizando las letras en mayúsculas de la "A" a la "F". Tienen el mismo significado que las letras en minúsculas, esto es, los números decimales del 10 al 15.).

Así, el mismo mensaje al Senador Till, escrito en notación hexadecimal, es:

```
53 65 6e 61 74 6f 72 20 48 61 6e 64 20 4e 2e 20 54 69 6c 6c 0d 0a
57 61 73 68 69 6e 67 74 6f 6e 2c 20 44 43 0d 0a 0d 0a 44 65 61 72
20 53 65 6e 61 74 6f 72 3a 0d 0a 0d 0a 54 68 69 73 20 6c 65 74 74
65 72 20 69 73 20 74 6f 20 69 6e 66 6f 72 6d 20 79 6f 75 20 77 65
20 68 61 76 65 0d 0a 6f 70 65 6e 65 64 20 61 63 6f 75 6e 74 20
6e 6f 2e 20 33 33 38 39 30 37 30 32 31 20 6f 6e 20 79 6f 75 72 0d
0a 62 65 68 61 6c 66 2c 20 61 6e 64 20 64 65 70 6f 73 74 65 64 20
74 68 65 72 65 69 6e 20 61 0d 0a 6c 65 67 69 73 6c 61 74 69 76 65
20 69 6e 63 65 6e 74 69 76 65 20 70 61 79 6d 65 6e 74 20 6f 66 0d
0a 24 31 20 6d 69 6c 6c 69 6f 6e 2e 20 20 46 6f 72 20 77 69 74 68
64 72 61 77 6c 2c 20 79 6f 75 20 77 69 6c 6c 0d 0a 6e 65 65 64 20
74 6f 20 75 73 65 20 79 6f 75 72 20 70 61 73 73 77 6f 72 64 20 42
4a 52 47 55 44 37 36 39 33 2e 0d 0a 0d 0a 59 6f 75 72 73 20 76 65
72 79 20 74 72 75 6c 79 2c 0d 0a 0d 0a 41 72 6e 6f 6c 64 20 43 2e
20 43 72 65 6f 6c 65 0d 0a 56 69 63 65 20 50 72 65 73 69 64 65 6e
74 0d 0a 47 72 65 61 74 65 72 20 43 61 72 69 62 62 65 61 6e 20 42
61 6e 6b
```

El siguiente paso consiste en firmar este mensaje, de tal manera que el Senador Till sabrá que vino del Gran Banco del Caribe, y no de, por ejemplo, *Louis Freeh's Henchcreatures*. Esto significa que se debe utilizar una función de *hash.*

Aquí se presentan cuatro resúmenes producidos por cuatro funciones de *hash* distintas.

Función de *hash*	**Resumen del mensaje**
MD5	5670E64BF6CEBB46 31A25CF6990F82C0
RIPEM128	B4BB17FD0E09091A 2DF095F0B9647B41
SHA-1	1A01B56EB33FA84A 39EEDDD927977726 38331E94
RIPEM160	ABA54F46348F56D1 E492AE09A472D143 9D64E0F1

Claro se mostrará que pasa con los resúmenes si realizamos pequeñas alteraciones al mensaje original. Primero, se corrige el fallo de ortografía.Con esta corrección, ahora el mensaje es:

```
**********

Senator Hand N. Till
Washington, DC

Dear Senator:

This letter is to inform you we have
opened account no. 338907021 on your
behalf, and deposted therein a
legislative incentive payment of
$1 million.  For withdrawal, you will
need to use your password BJRGUD7693.

Yours very truly,

Arnold C. Creole
Vice President
Greater Caribbean Bank
**********
```

Con este cambio, los resúmenes del mensaje son totalmente diferentes, como se puede ver en la siguiente tabla:

Función de *hash*	**Resumen del mensaje**
MD5	8BDF43C9BC320AE8 874E9EED73DDCF55
RIPEM128	5BF83DAE28ACBF49 BD9EDB6D26DE1EE9
SHA-1	22257C5870B7CB00 E6B5AABA45913C24 DEAB6F7D
RIPEM160	713FE23D55F95ACD B5D744C0B616774B 1DDAA4E7

Como segundo ejemplo, se queda "withdrawl" tal y como estaba, pero se cambia en número de cuenta, de tal manera que el último dígito sea un "2" en lugar de un "1". De esta manera el mensaje quedaría como:

```
**********

Senator Hand N. Till
Washington, DC

Dear Senator:

This letter is to inform you we have
opened account no. 338907022 on your
behalf, and deposted therein a
legislative incentive payment of
$1 million.  For withdrawl, you will
need to use your password BJRGUD7693.

Yours very truly,

Arnold C. Creole
Vice President
Greater Caribbean Bank
**********
```

A continuación se muestran los resúmenes del nuevo mensaje con el dígito alterado:

Función de *hash*	**Resumen del mensaje**
MD5	0742CD5D4EA8B857 E57352C6B21CE7FB
RIPEM128	9BE546E061E64BD3 3659E49569774A25
SHA-1	B55F080E45CF15D9 3542A9F4C7CED8B3 48EF1E17
RIPEM160	C97428911B8C925E 990839FE2BB5B9E9 BD0EC4EF

.

El Mensaje Firmado

:

Con estos ejemplos se ha demostrado que si se altera el mensaje original, el valor de la función *hash* también cambiará. Por tanto, el resumen del mensaje puede utilizarse para comprobar la integridad del mensaje original

Para nuestra firma digital, utilizaremos la función de *hash* SHA-1

hash = 1A01B56EB33FA84A 39EEDDD927977726 38331E94

El siguiente paso consiste en cifrar el resumen del mensaje con la clave privada del firmante. Esto significa que necesitamos algún método de cifrado, con un par de claves (pública/privada). La clave privada se usará para firmar, mientras que la clave pública se utilizará para la verificación.

Se utilizará el sistema RSA. Este sistema firma valores elevándolos al exponente de la clave privada d, tomando el resultado módulo n. Supongamos que n y d tienen los siguientes valores.

```
n =

AF8207E25BF6A83E 17F5980E23AB498F 5B77E84821905716
EEDBD4F621EDE141 D38117147AD98E8B 549BC35B0C689475
D710013D83AE1945 FF405E2D4EC54A56 CB88BBB220C31F12
63978A307A36F0C2 316CA3CB921271B3 7B550728D560A884
4E1740C31DFCA3BC 0FB80D34AAF986D7 02FD633BD26F16AA
8A3C329E2D1E970D

 d =

0F2591B89F67322D E9B3706408000861 2EEBB248475D45A6
DD066BE2B21AED8D D8CB134AD92F5D75 F8DF5884CB155B7A
B00CD98E8D86C0F7 A187D498E46B7276 D6881D1502BA90C5
5EE073BB1565A969 681C3CE6660A8744 B6D54C6659299E1A
B8424F41DE1B277C D22DC1D81DEA69B2 8F99A4C9C852A82C
6123EDAAB94D0CA1
```

Donde n es un número de 1024 bits, que es lo mismo que 128 bytes o 256 números hexadecimales. Igualmente con d, excepto que d tiene cuatro ceros al principio, por lo que realmente solo tiene 1020 bits de longitud.
Lo que se desea hacer es elevar el *hash* al exponente d, y después tomar el resultado módulo n.

Esta es la idea. Sin embargo, el procedimiento de firma del RSA especifica que el valor *hash* 1A01B56EB33FA84A 39EEDDD927977726 38331E94 debe ser rellenado (*padded*) para que tenga la misma longitud que n. Entonces, como el *hash* tiene 160 bits, los bits de relleno --exactamente, 864 bits-- se añaden delante para que tenga 1024 bits de longitud.

Todo esto esta detallado en el RSA's Public Key Cryptography Standard No. 1 (PKCS#1). Explicar los motivos de este procedimiento se sale de los objetivos de este artículo. Pero diremos que añadir bits de relleno, en principio, no supone ningún riesgo (dependiendo de como lo hagamos). Siempre es posible cifrar el *hash* con el relleno, y más tarde eliminar ese relleno una vez que se haya descifrado el criptograma.

Según el PKCS#1 debemos añadir 00 delante del *hash*, quedando:

00 1A01B56EB33FA84A 39EEDDD927977726 38331E94

Esta es la cola del número de relleno. El principio de ese número es 00 02, seguido de tantos números hexadecimales obtenidos aleatoriamente como sean necesarios para obtener un total de 256 dígitos hexadecimales (1024 bits)
De esta manera, el número debería ser algo como:

00 02 . . . números aleatorios . . 00 1A01B56EB33FA84A 39EEDDD927977726 38331E94

Este es uno de los conjuntos de reglas disponibles. Por supuesto, no todo el mundo sigue las mismas reglas. Yo estoy haciendo los cálculos con el lenguaje de programación Java, usando las bibliotecas de clases de Cryptix. Cryptix rellena el *hash* usando una variante del PKCS#1, obteniendo el siguiente entero

```
entero con relleno =

01FFFFFFFFFFFFFF FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF FFFFFF0030213009
06052B0E03021A05 0004141A01B56EB3 3FA84A39EEDDD927
97772638331E94
```

Podemos ver que el relleno empieza por 01, al que le siguen un montón de FFFFFF's, después 00, un número misterioso "30213009 06052B0E03021A05 000414", y finalmente el *hash* 1A01B56EB33FA84A 39EEDDD927977726 38331E94

Tomando el entero con relleno, lo elevamos a la potencia d, y reducimos el resultado módulo n. Esto nos da la firma digital firm. Si hacemos los cálculos, obtenemos:

```
firm =

149522ECA960A6B4 A46F1546B6D5F74B C3570CD7DD981EA1
0B506B346FB159BE 7F7BAA26F6A8A143 090B4D0A944AE4D7
96C17A4587267B05 A991D76EDE989583 9E47C19054CDB818
5BD21EE36BAC9803 CE005383A1083AB5 79411AB26BE28631
1BF17D021002B6D5 2EE82CEB2FC554A8 BDE5874D82B20B9F
21EBD65F5FD93102
```

Esta es la firma digital que se enviará junto a la carta al Senador Till para que pueda verificar su autenticidad, así como la identidad del firmante

Verificando la Firma

Cuando el Senador Till recibe el mensaje, calcula su propio *hash* (llamémosle hash*) a partir del mensaje recibido. Debería obtener el valor:

```
hash* = 1A01B56EB33FA84A 39EEDDD927977726 38331E94
```

Si el mensaje no ha sido alterado. Seguidamente compara hash* con el valor descifrado de firm para ver si son iguales.
Para descifrar firm, obtiene la clave pública del Gran Banco del Caribe (e, n), y ve que e tiene un valor (en hexadecimal) **e =** 010001

Pequeño y dulce. Pero no muy inteligente, ya que muchas organizaciones aparte del Gran Banco del Caribe pueden estar usando el mismo valor de e. (Sin embargo, lo que la hace única es la combinación de e con el módulo n. Siempre que n sea único, no tendremos porque preocuparnos de e.).
Así que el Senador Till toma el valor firm, lo eleva a e y se queda con el módulo n del resultado. Esto produce el número (confíen en mí, he hecho los cálculos):

```
01FFFFFFFFFFFFFF FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF FFFFFF0030213009
06052B0E03021A05 0004141A01B56EB3 3FA84A39EEDDD927
97772638331E94
```

Una vez que el Senador Till quita el relleno, él (o, más bien, su software) ve que el final es igual que el valor hash*, lo nos asegura la validez de la firma.
La figura 1 muestra el proceso de verificación de una firma digital.

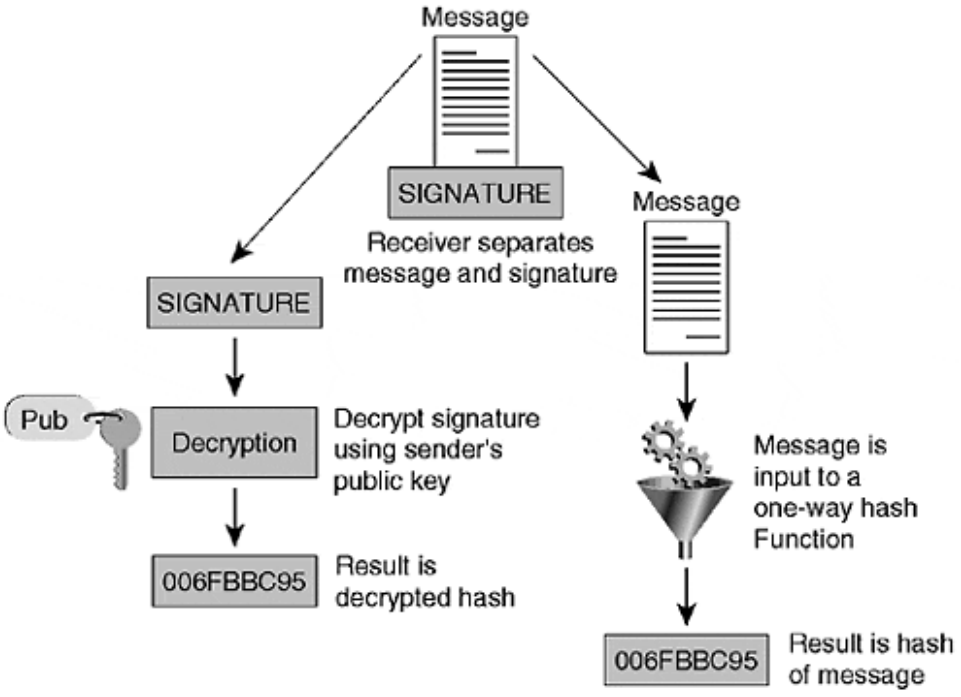


Figura 1. Proceso de verificación de una firma digital

Certificados Digitales

Un certificado digital en un mensaje firmado que es utilizado para atestiguar la validez de una llave pública o entidad. Los certificados requieren de un formato común y están basados en el estándard ITU-T X.509. La figura 2 muestra un ejemplo de un certificado digital basado en el estándard X.509.

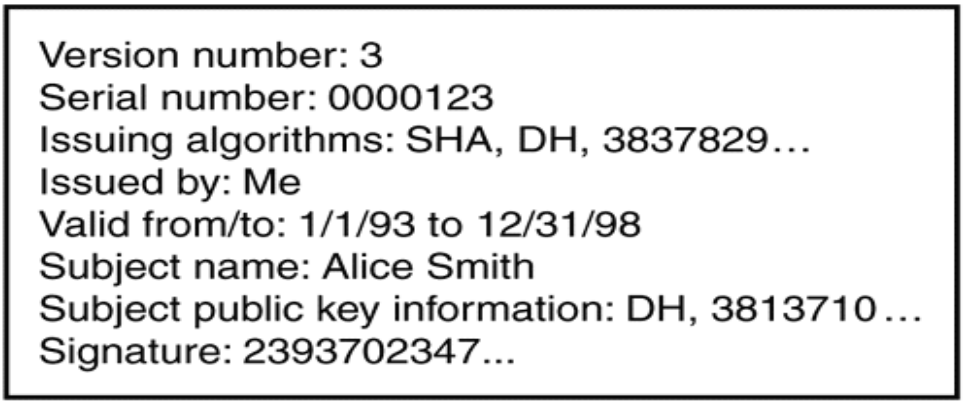


Figura 2. Certificado digital

Los certificado digitales son una forma de proveer validez a una llave pública de una entidad y a futuro se ve como el mecanismo para proveer autenticación a las grandes compañías. Aunque se ha trabajo mucho en certificados digitales aun falta mucho por hacer, por ejemplo la determinación que validación de certificados, su administración y una interpretación semántica consistente.

Autoridades Certificadoras

Una autoridad certificadora (CA, certification authority) es una organización que emite certificados de llave pública, conceptualmente, parecen tarjetas bibliográficas con firma criptográfica. Los certificados, firmados con las llaves públicas de la autoridad certificadora contienen el nombre de una persona, su llave pública, número de serie y otra información. El certificado comprueba que una llave pública específica es propiedad de un individuo u organización en particular.

Existen muchas formas en que las autoridades certificadoras pueden ofrecer su servicio:

CA interna

Una organización puede operar una CA para certificar a sus propios empleados, sus puestos y sus niveles de autoridad. Tal jerarquía de certificación podría emplearse para controlar el acceso a los recursos o al flujo de información internos. Por ejemplo,

cada empleado de una organización podría crear una llave y recibir un certificado para ella. Además, dicho certificado sería enviado a los sistemas a los que deba tener acceso. Las computadoras en toda la organización podrían entonces decidir si otorgan o no acceso a un empleado basados en la certificación de su llave. De esta forma, la empresa evita la necesidad de distribuir una lista de control de acceso y archivos de claves de acceso a todas sus computadoras.

CA externa de empleados

Una empresa podría contratar a una compañía externa para que le dé servicios de certificación para sus empleados, de la misma forma en que podría contratar a un laboratorio fotográfico para crear tarjetas de identificación.

CA externa de clientes

Una empresa podría contratar a una compañía externa para operar una autoridad certificadora para sus clientes actuales o potenciales. Al confiar en las prácticas de certificación de una compañía externa, la empresa se ahorraría el costo de crear sus propios procedimientos.

CA confiable de terceros

Una compañía o un gobierno pueden operar una CA que relacione llaves públicas con los nombres legales de individuos y empresas. Esa CA puede utilizarse para permitir a personas sin relación anterior establecer su identidad y participar en transacciones legales.

Para utilizar los certificados emitidos por una CA es necesario tener una copia de su llave pública. Hoy en día, estas llaves se incluyen en programas como navegadores y sistemas operativos. El usuario puede agregar otras llaves públicas de autoridades certificadoras manualmente.
Es evidente que las CA que no distribuyen sus llaves públicas están en desventaja.

Revocación

Además de emitir certificados, las CA necesitan alguna forma de revocarlos, por lo siguiente:

- La llave privada del tenedor puede haber sido interceptada o violada.
- La CA puede descubrir que ha emitido el certificado a la persona o entidad incorrecta.
- El certificado puede haber sido emitido para dar acceso a un servicio específico y el

individuo puede haber perdido su autorización para utilizarlo.
- Los sistemas de la CA pueden haber sido violados e forma que alguien tenga la capacidad de emitir certificados falsos.

Se ha propuesto una forma de manejar las revocaciones: una lista de revocación de certificados (CRL, *certificate revocation list)*. Una CRL es una lista de todos los certificados revocados por la CA y que aún no expiran por otras razones. Idealmente, una CA emite una CRL a intervalos regulares. Además de listar los certificados revocados, la CRL específica durante cuánto tiempo es válida y dónde obtener la siguiente CRL.

En teoría, las CRL son interesantes: permiten a las computadoras que no están conectadas a una red determinar si un certificado es válido o si se ha revocado. No obstante, en la práctica tienen varios problemas:

- Tienden a crecer con rapidez.
- Existe un periodo en que un certificado aparente ser válido, sin serlo, entre el momento en que se revoca y en el que se distribuye la nueva CRL.
- La información contenida en las CRL puede emplearse para analizar el tráfico.

En vez de CRL, la mayoría de las CA tal vez utilizará verificación en tiempo real mediante bases de datos en línea conectadas a una red; digamos Internet. Tales sistemas eliminan limpiamente los problemas de las CRL, aunque necesitan una red confiable y que este disponible.

Una alternativa sugerida por Carl Ellison de Cybercash, es utilizar certificados con lapsos de expiración muy breves –de uno a dos minutos--. En efecto, esto requiere que la persona que utilice el certificado se comunique con la CA antes de cada transacción, lo cual en algunos casos, puede ser más eficiente que obligar al receptor de un certificado a verificarlo con la CA.

Documento de Prácticas de Certificación (CPS)

¿Que significa tener una llave certificada? La respuesta a esta pregunta depende de quién hace la certificación y qué políticas sigue. Una CA interna en una compañía de las Fortune 500 puede certificar que la persona cuya llave está firmada es un empleado activo. Por otra parte, el hipotético firmador de llaves anónimas Cyberpunk podría firmar cualquier llave enviada a una dirección de correo electrónico específico.

El documento de prácticas de certificación (CPS, *certification practices statement)* es un documento – valga la tautología- legal públicado por las CA que describe sus políticas y procedimientos para la emisión y revocación de certificados digitales. Responde a la pregunta ¿que significa el que esta organización firme una llave?

Los CPS están diseñados para que los lean humanos, no máquinas. Es posible que en un futuro los términos y condiciones de las CA se estandaricen lo suficiente para que haya programas que procesen de manera automática los CPS. Una empresa puede estar dispuesta a aceptar una certificación de una CA que garantice ciertas políticas mínimas de certificación y que puede asumir cierta responsabilidad en caso de que no se cumplan tales políticas -y siempre y cuando la CA esté respaldada por una afianzadora adecuada-. Alternativamente, las leyes o el mercado podrán fomentar que las autoridades certificadoras adopten políticas estándares a lo largo de la industria, del mismo modo que los emisores de tarjetas de crédito han adoptado políticas más o menos estándares pero se distinguen entre sí mediante tasas de interés, comisiones y servicios adicionales.
En la siguiente figura (Figura 3) muestra como Bob puede obtener la llave pública de Alice a través de una entidad certificadora:

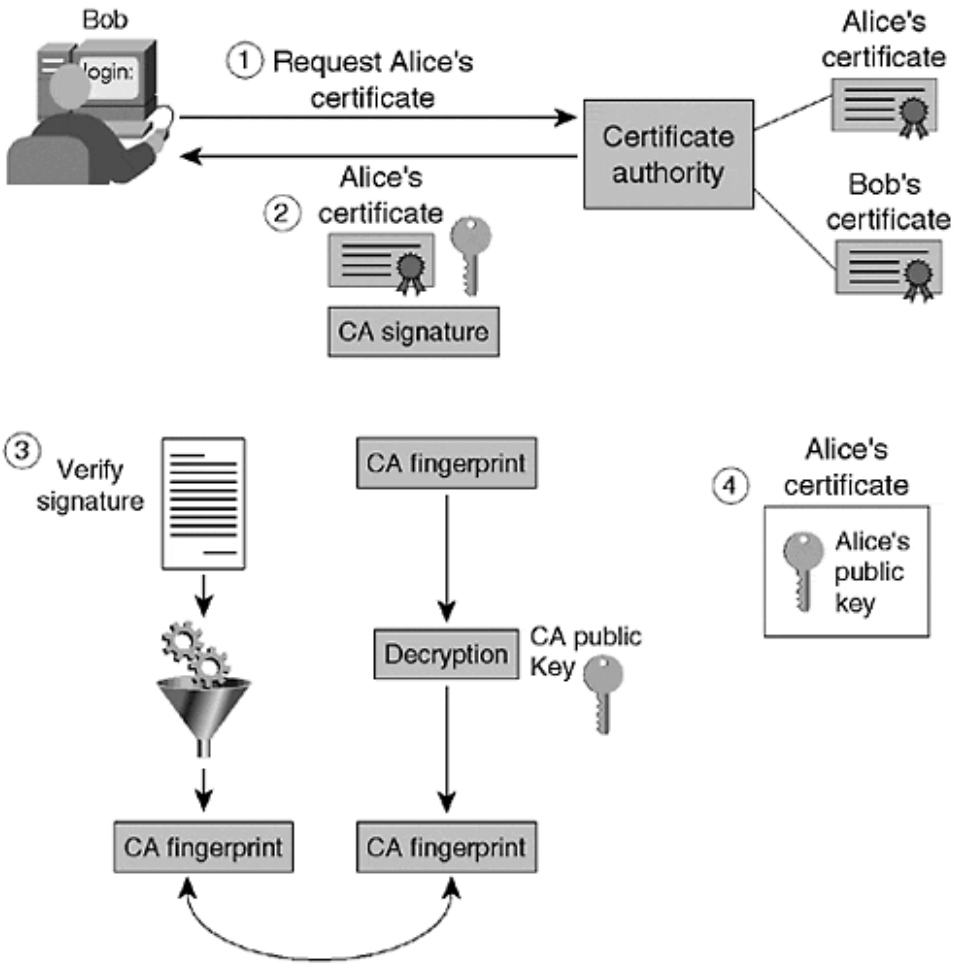


Figura 3. Ejemplo del uso de una CA

Los pasos que se siguen son:

- Bob solicita la firma digital de Alice a la CA.
- La CA envía el certificado de Alice, el cual es firmado por la llave privada de la CA.
- Bob recibe el certificado y verifica la firma de la CA.
- Dado que el certificado de Alice contiene su llave pública, Bob tiene una versión "notarizada" de la llave pública de Alice.

2.5. Funciones de Hash

Una función de Hash toma un mensaje de longitud variable como entrada y la salida es un mensaje de longitud fija. Este mensaje de longitud fija es llamado el "hash", o el resumen (digest) del mensaje del mensaje original de entrada. Para que un algoritmo sea considerado criptográficamente conveniente, es decir, que sea seguro, para una función hash, debe mostrar las siguientes propiedades:

- Debe ser consistente, es decir, la misma entrada siempre debe crear la misma salida.
- Debe ser aleatoria, o al menos dar la apariencia de aleatoriedad para prevenir que pueda adivinarse el mensaje original.
- Debe ser único, es decir, debe ser imposible encontrar dos mensajes que produzcan el mismo resumen
- De ser unidireccional, es decir, si se tiene la salida, debe se extremadamente difícil, mas no imposible, comprobar la entrada del mensaje.

Las funciones hash unidireccionales son las más utilizadas para proveer una huella digital "fingerprint" de un mensaje o archivo, al igual que una huella digital de una persona, una huella hash es única y provee integridad y autenticidad del mensaje.

En la figura 1 se muestra a Alice y Bob utilizando una función de hash unidireccional para verificar que nadie ha modificado el contenido de un mensaje que fue transmitido.

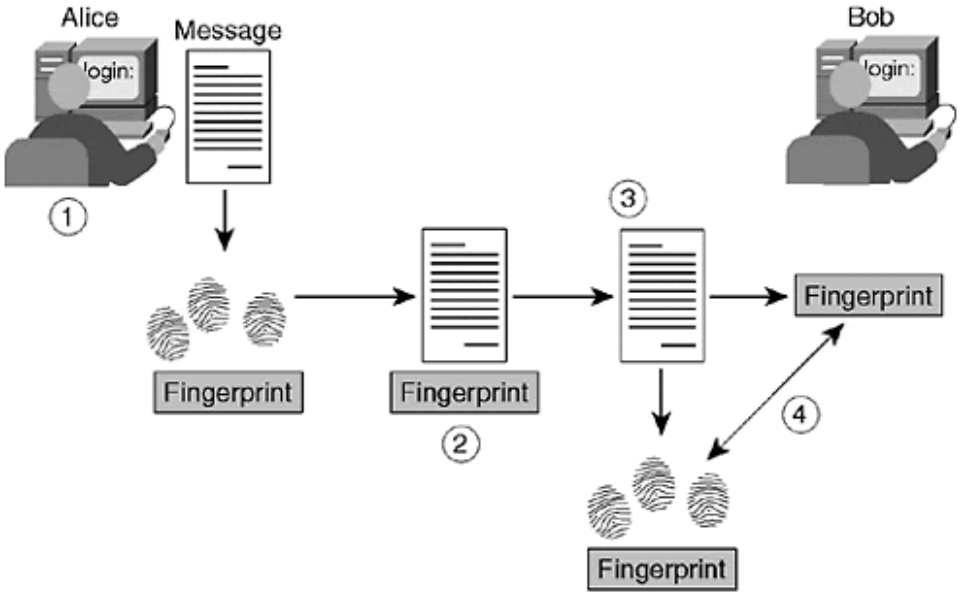


Figura 1. Utilización de una función de hash

Los siguientes pasos se llevan a cabo para mantener la integridad de los datos.

- Alice escribe un mensaje y utiliza como entrada una función de hash unidireccional.
- El resultado de la función de hash es agregada como la huella y se envía a Bob.
- Bob recibe el mensaje y la huella y utiliza el mensaje como entrada de la misma función unidireccional de hash que utilizó Alice.
- Si la huella resultante es igual a la que agregó Alice, significa que el mensaje no fue modificado.

El problema con las funciones de las funciones hash de huella digital es que puede se forzado y es sujeto a un ataque de hombre-a-la-mitad (man-in-the-middle attack). Este ataque se refiere a que una entidad que se encuentra escuchando en un canal que se presume es segura y que se hace pasar por el emisor o el receptor. Utilizar eficientemente una función de hash como huella y combinarla con tecnología de llave pública, da como resultado las firmas digitales.

Las principales funciones de hash que se utilizan son:
- Algoritmo Message Digest 4 (MD4).
- Algoritmo Message Digest 5 (MD5).
- Secure Hash Algorithm (SHA).

MD4 y MD5 fueron diseñados por Ron Rivest en el Massachusetts Institute of Technology (MIT).

SHA fue desarrollado por la National Institute o Standards and Technology (NIST).

Las dos funciones de hash mas utilizadas son MD5 y SHA. MD5 procesa su entrada en cloques de 512 bits y produce un mensaje de 160 bits. SHA consume más recursos de procesados y suele ser más lento que MD5.

Puede encontrarse más información sobre las funciones de hash y los algoritmos en las siguientes direcciones web:

- http://www.itl.nist.gov/fipspubs/fip180-1.htm
- http://www.secure-hash-algorithm-md5-sha-1.co.uk
- http://www.w3.org/PICS/DSig/SHA1_1_0.html
- http://www.faqs.org/rfcs/rfc1321.html
- http://userpages.umbc.edu/~mabzug1/cs/md5/md5.html
- http://www.faqs.org/rfcs/rfc1320.html
- http://www.rsasecurity.com/rsalabs/node.asp?id=2253

Ejercicio 1

Después de leer los documentos del capitulo 3 resuelva la siguiente práctica.
Utilización de el programa “Jhon the ripper” para comprobar la vulnerabilidad de utilizar contraseñas simples

Objetivo: El alumno demuestra la importancia de utilizar contraseñas fuertes y no basadas en diccionarios.

Instrucciones:

- Descargar e instalar el programa “john the ripper” (http://www.openwall.com/john/).

- Copia el texto que se muestra a continuación, el cual es un archivo llamado shadow.

```
root:$1$F3JXGnhv$M8nYgSYegFCpGiOmf9BIG.:12500:0:99999:7:::
daemon:*:12500:0:99999:7:::
bin:*:12500:0:99999:7:::
sys:*:12500:0:99999:7:::
sync:*:12500:0:99999:7:::
games:*:12500:0:99999:7:::
man:*:12500:0:99999:7:::
lp:*:12500:0:99999:7:::
```

```
mail:*:12500:0:99999:7:::
news:*:12500:0:99999:7:::
uucp:*:12500:0:99999:7:::
proxy:*:12500:0:99999:7:::
postgres:*:12500:0:99999:7:::
www-data:*:12500:0:99999:7:::
backup:*:12500:0:99999:7:::
operator:*:12500:0:99999:7:::
list:*:12500:0:99999:7:::
irc:*:12500:0:99999:7:::
gnats:*:12500:0:99999:7:::
nobody:*:12500:0:99999:7:::
pato:$1$022b6yvG$ltFbbbV.KOHTJRknsDPsA1:12720:0:99999:7:::
identd:!:12500:0:99999:7:::
sshd:!:12500:0:99999:7:::
gdm:!:12675:0:99999:7:::
smg:$1$AK1Ls5yh$av3gwt31Po2XU2.6mGMLE/:12676:0:99999:7:::
mysql:!:12676:0:99999:7:::
hommer:$1$EHRKvsIa$lQhhMtOJ1tcKARTkzqSvi.:12720:0:99999:7:::
bart:$1$WUgBw8bu$qpdxf3QyUPqyoUn7cFQ331:12720:0:99999:7:::
lizza:$1$W/GUtyqD$Iscf47aGKQRGZAQ6kptmh0:12720:0:99999:7:::
magui:$1$fN9X0P3m$.jvMvosYypBBVdfikhU2P0:12720:0:99999:7:::
```

Archivo “shadow”

- Utilizar el programa “john th ripper” con el archivo descargado.
- Contestar las siguientes preguntas:
    - ¿Cuántas contraseñas descubrió?
    - ¿Qué características tienen las cuentas y contraseñas descubiertas?
    - ¿Qué recomendaciones puedes hacer para poner una contraseña?

Productos esperados:

- Reporte de contraseñas encontradas.
- Recomendaciones de uso de contraseñas.

Ejercicio 2

Cifrado y descifrado de un mensaje utlizando algoritmo de llaves simétricas

Objetivo: El alumno comprueba el uso de una herramienta de cifrado asimétrico para cifrar y descifrar un mensaje.

Instrucciones:

1. Utilizando algún programa de cifrado simétrico generar su par de llaves (pública y privada, puedes utilizar gnupg http://www.gnupg.org).
2. Utilizando la llave pública de tu instructor envíale por correo electrónico un mensaje cifrado y tu llave publica (sin cifrar).
3. Recibirás, por email, un mensaje cifrado el cual deberás descifrar y mandar descifrado a tu instructor.

Productos esperados:

- Llave publica del alumno.
- Mensaje cifrado del alumno.
- Mensaje original del maestro sin cifrar.

Se observa en la animación interactiva y se sigue las instrucciones que en ella se encuentran.

Desafio Criptográfico

Fuentes de Referencia Bibliográfica

Anónimo; Linux Máxima Seguridad; Madrid 2000, Pearson Edu

Garfimkel, Simson; Spafford, Gene; Seguridad y comercio en el web; 1999, McGrawHi & O'Reilly.

Anónimo; Linux Máxima Seguridad; Madrid 2000, Pearson Ed Garfimkel, Simson;

Spafford, Gene; Seguridad y comercio en el web; 1999, McGrawHi & O'Reilly.

Páginas de Referencia Electrónica

http://world.std.com/~cme/html/timeline.html

http://cwis.kub.nl/~frw/people/koo

http://www.gilc.org/crypto/crypto

http://www.oecd.org/

http://www.wassenaar.org/

http://www.itl.nist.gov/fipspubs/fip46-2.htm

http://encyclopedia.thefreedictionary.com/Data%

http://csrc.nist.gov/cryptval/des.htm

http://www.schneier.com/blowfish.html

http://web.mit.edu/network/pgp.html

http://www.pgp.com

http://en.wikipedia.org/wiki/International_Data_E

http://theory.lcs.mit.edu/~rivest/crypto-securit

http://theory.lcs.mit.edu/~rivest/crypto-securit

3. Código Dañino

Introducción

Estamos acostumbrados todos los días a tomar ciertas precauciones o actitudes, que consideramos lógicas o de sentido común, para "proteger" nuestra misma existencia: mirar a los dos lados antes de cruzar una calle, correr las cortinas de las ventanas para proteger nuestra intimidad, no comer productos alimenticios que consideramos dudosos, poner un antirrobo al coche. Un sin fin de pequeñas acciones que van dirigidas a procurarnos una mayor seguridad.

La incorporación de las nuevas tecnologías en nuestras vidas nos encamina a un nuevo mundo virtual fantástico, pero también nos plantean nuevas situaciones que debemos conocer y afrontar.

Uno de los temas que tendemos a olvidar, o a dejar un poco de lado, al incorporarnos a Internet es precisamente la seguridad en Internet.

Ciertamente parece que no le damos demasiada importancia a dicho tema, la mayoría de las veces por simple desconocimiento. Nos empezamos a preocupar solo cuando nos ocurre algún problema, algunas veces con muy poco remedio y a algunas otras solo lo consideramos como un pequeño fastidio.

Las diversas situaciones en las que nos podemos ver envueltos por no tener prevista una mínima seguridad en nuestra conexión pueden llegar a ser realmente preocupantes, incluso aunque no las percibamos en ese momento. Por ejemplo tal vez nos hemos infectado con un virus y pensemos que bueno, como ultima medida formatear el disco duro y listo. Tal vez ni siquiera nos enteremos de que estamos infectados hasta que ya es muy tarde y mientras tanto seamos una plataforma de expansión del virus. También hemos oído hablar de los hackers y piensas, bueno, pues si entran tampoco van a ver nada. Seguramente no pensaremos que ese hacker nos va a utilizar como escudo y plataforma para ataques a sitios "más importantes". Nos va a utilizar para protegerse él, dejándonos en un serio aprieto. Tal vez pensemos que mientras navegamos por Internet nada pueden saber de nosotros si no se lo contamos.

Muchos "tal vez" que nos deben de hacer reflexionar y tomarnos en consideración el tema de la seguridad en Internet.

En esta unidad vamos a intentar introducirnos en la seguridad informática aplicada a la privacidad de nuestra información y a la protección de nuestro sistema.

3.1. Integridad de archivos

Objetivos

Distinguirás los parámetros para mantener la integridad de archivos de cómputo, mediante la utilización de software verificador de archivos, buscando con esto mantener la estabilidad de los medios de almacenamiento de datos

4.1.1 Vulnerabilidad

Algunos programas poseen "agujeros" que pueden facilitar la infección de nuestro ordenador.

Como en el mito griego del famoso héroe Aquiles, una vulnerabilidad representa un punto a través del cual es posible vencer la seguridad de un ordenador. Una vulnerabilidad es un fallo en la programación de una aplicación cualquiera, y que puede ser aprovechado para llevar a cabo una intrusión en el ordenador que tenga instalado dicho programa.

Generalmente, dicho fallo de programación se refiere a operaciones que provocan un funcionamiento anormal de la aplicación. Esta situación anómala puede ser producida artificialmente por una persona maliciosa para poder introducirse en un ordenador sin el consentimiento del usuario. En ocasiones, es suficiente con abrir un documento creado "artesanalmente" con ese fin específico.

Esto le permitirá al usuario malicioso realizar un gran abanico de acciones en el ordenador vulnerable, desde ejecutar ficheros hasta borrarlos, introducir virus, acceder a información, etc.

Aunque son más conocidas las vulnerabilidades asociadas a sistemas operativos, navegadores de Internet y programas de correo electrónico, cualquier programa puede presentar vulnerabilidades: procesadores de textos, bases de datos, aplicaciones de reproducción de archivos de sonido, etc.

Una vulnerabilidad no representa un peligro inmediato para el ordenador. Sin embargo, es una vía de entrada potencial para otras amenazas, tales como virus, gusanos, troyanos y backdoors, que sí pueden tener efectos destructivos.

Por ello, es altamente recomendable estar informado acerca de las vulnerabilidades descubiertas en los programas instalados y aplicar los parches de seguridad más recientes proporcionados por la empresa fabricante, accesibles a través del sitio web de la misma.

Algunos ejemplos de gusanos que aprovechan vulnerabilidades para llevar a cabo sus acciones son: Blaster, Bugbear.B, Klez.I y Nachi.A.

3.2. Integridad del sistema

Objetivos

Distinguirás los parámetros para mantener la integridad del sistema, aplicando utilerías y mejores prácticas para la seguridad en sistemas de redes.

4.2.1 ¿Qué tan visible es mi computadora?

Para la mayoría de nosotros, Internet es un misterio. No sabemos cómo funciona y no sabemos que tan visibles somos cuando nos movemos por este campo virtual. Sin embargo, nos damos cuenta que un cierto porcentaje de personas entienden la tecnología de Internet muy bien y no todos tienen las mejores intenciones. Esto es suficiente para que el usuario promedio se sienta un poco incómodo. Después de todo, muchos de nosotros estamos en Internet todo el tiempo, intercambiando información con los amigos y socios, investigando, leyendo las noticias. Naturalmente que todos queremos saber un poco más sobre el peligro cotidiano a que estamos expuestos. ¿Quién no se ha hecho las siguientes preguntas?: ¿Qué aspectos de mi computadora son visibles para los demás? ¿Qué partes podemos acceder o manipular? ¿Están mis mensajes de correo electrónico y mensajes instantáneos protegidos de personas entrometidas? ¿Puede cualquier persona ver lo que hago mientras estoy en la Web? Por supuestos que las respuestas a estas preguntas varían dependiendo de su situación. Afortunadamente existen algunas respuestas generales que junto al poco conocimiento y protección, hacen que usted pueda crear una zona cómoda en la Web.

Lo fundamental son los puertos y las direcciones IP

Cuando usted está conectado a Internet, otras personas pueden ver cosas básicas de su computadora. Si no pueden, no tendrían motivos para estar en Internet. El hecho es que usted debe darse a conocer para hacer uso de Internet. Por lo tanto, cierto grado de exposición al peligro es inherente a la vida en Internet. Entonces, ¿qué es exactamente estar visible? Para los principiantes: sus puertos y dirección IP. Los puertos son pasajes de información entre su computadora e Internet. Todos los puertos tienen un número para una función específica. Por ejemplo, la mayor parte del tráfico en la Web pasa a través del puerto número 80. Los correos electrónicos viajan por el puerto 25. Personas mal intencionadas pueden explorar sus puertos para ver si están abiertos y si algunos de los más interesantes están disponibles, pueden usarlos para atacar su computadora. Por su parte, su dirección IP es un Identificador numérico único para su computadora. Es parecido a un número telefónico o dirección de correo

electrónico y sin él, usted no podría recibir nada por Internet. También es una forma práctica para que los hackers rastreen su equipo.

A pesar de la visibilidad de estos detalles básicos, es posible proteger su computadora de los ataques. Soluciones de seguridad como Norton Personal Firewall protegen su computadora y los puertos abiertos de los hackers. También, permiten que pase el tráfico autorizado a su equipo. Si no tiene un firewall, podría por lo menos evitar utilizar una dirección IP estática. Con una dirección IP estática, su computadora tiene el mismo identificador cada vez que navega en Internet, lo cual hace que sea una presa fácil. Por el contrario, si su dirección IP cambia cada vez que se conecta, usted se hace un poco más escurridizo. Verifique con su proveedor de Internet para garantizar que su dirección IP se genere dinámicamente.

Exposición al peligro por el uso compartido de archivos de Windows

Si usted ejecuta los sistemas operativos Windows®, otros usuarios de Internet pueden ver su información NetBIOS. NetBIOS es un programa utilizado para las operaciones y comunicación básicas de las redes, como el uso compartido de archivos e impresoras. Si comparte archivos localmente a través del NetBIOS y también está conectado a Internet, los hackers pueden ver, descargar o borrar sus archivos. La buena noticia es que es muy fácil impedir los ataques si reconfigura las especificaciones de su NetBIOS.

Revelación de sus hábitos de navegación

Hasta cierto punto otras personas pueden ver sus movimientos en la Web. Como lo sabe su navegador registra el historial de navegación Es una función práctica cuando usted desea volver a un sitio que visitó recientemente. Lo que no sabe es que su navegador también comunica su historial más inmediato, como la dirección URL de la página de la que acaba de salir, al siguiente sitio Web. Los propietarios de los sitios Web pueden usar esta información para estudiar sus hábitos de navegación, especialmente si usted regresa a sus sitios con frecuencia. Afortunadamente, hay una solución fácil para este problema. Su navegador no puede pasar a la dirección de una página de referencia a menos que de un clic en el enlace. Por tanto, si no quiere que los dueños del sitio Web sepan donde ha estado, no de clic en el enlace de ese sitio. A cambio, escriba la dirección URL del sitio en la barra de direcciones del navegador. También es posible reconfigurar algunos navegadores para bloquear la transmisión de direcciones de referencia.

Los programas spyware son incluso más invasivos. Residen en su computadora y recolectan información sobre sus hábitos informáticos. Luego envían su perfil por Internet al editor del programa, quien puede usar la información para enviarle anuncios publicitarios o puede vender su perfil a terceros. La mayoría de spyware termina en su computadora sin su conocimiento. Sin embargo, si usted ejecuta Norton Internet Security™, podrá identificar cualquier programa, como spyware, que intente acceder a

Internet desde su equipo. Una vez usted lo identifica y decide que no lo quiere en su computadora, simplemente puede eliminarlo y quedar tranquilo.

Los mensajes electrónicos y la mensajería instantánea

Desafortunadamente sus mensajes electrónicos no son muy seguros. Desde que salen de su computadora hasta que llegan a su destino, muchas personas diferentes, desde hackers de servidores de correo hasta administradores de servidores de correo, pueden leer su correo. De igual forma, sus mensajes instantáneos viajan por Internet desprotegidos. Son objetivos precisos para que los curiosos los intercepten y lean.

Para disminuir las posibles violaciones a la privacidad, use un servicio de correo de confianza y asegúrese de que éste emplea protección firewall poderosa en sus servidores de correo. Usted también debe cambiar con frecuencia las contraseñas de mensajería instantánea y correo electrónico especialmente si usa correo electrónico en la Web como Yahoo o Hotmail. Si su correo electrónico contiene información muy confidencial, podría usar la tecnología de encriptación de claves. Aunque la encriptación requiere un pequeño esfuerzo, es la mejor forma de garantizar que únicamente los destinatarios a quienes van dirigidos sus correos electrónicos puedan leerlos. En cuanto a la mensajería instantánea, su discreción es la mejor protección. Si tiene algo importante que decir, espere a hacerlo por otro medio más seguro.

Los virus y otros programas maliciosos

Algunos programas maliciosos pueden hacer que todo su sistema sea visible a los hackers. Por ejemplo, algunos caballos de Troya ponen un mecanismo a control remoto en su computadora, permitiendo acceso total a su equipo. Una vez que un hacker tiene el control, puede usar su equipo para atacar otros o puede ver, alterar o borrar los contenidos de su computadora. Los programas maliciosos pueden acceder a su sistema de muchas maneras: por correo electrónico, por medio de los sitos Web o incluso a través de disquetes. Para protegerse contra esta clase de peligros, instale un programa antivirus de confianza como Norton AntiVirus™.

Haga una evaluación total

Aunque usted puede solucionar cada uno de estos problemas de visibilidad individualmente, un sistema completo de seguridad es la mejor manera de evaluar el nivel de riesgo real de su sistema. Symantec Security Check, una herramienta gratuita de evaluación, evalúa muchos puntos posibles de vulnerabilidad, como los puertos, la configuración del NetBIOS y la susceptibilidad a virus y caballos de Troya. Una vez que Security Check ha determinado sus niveles de riesgo, tiene la ventaja de ofrecer sugerencias para proteger su sistema. Después de que Security Check hace una revisión total y que toma las medidas adecuadas para proteger su sistema, el misterio

relacionado con la visibilidad en Internet desaparecerá y usted descubrirá que Internet es una morada mucho más cómoda.

4.2.2 Protección del Equipo

A medida que Internet se ha vuelto cada vez más popular y cada vez es más fácil conseguir conexiones de alta velocidad para usarlo, el peligro que supone el asalto de un intruso a los equipos domésticos ha aumentado. Resulta tentador pensar que, como individuo o pequeña empresa, y al no ser una compañía ni una institución grande ni muy aparente, se estará a salvo ya que nadie conocerá su equipo en Internet.

Desgraciadamente, la llegada de herramientas automáticas que buscan los equipos activos y los prueban para encontrar sus vulnerabilidades implica que no puede depender de la invisibilidad para protegerse. Menos mal que puede poner en práctica muchos métodos para establecer esta protección.

Desarrollo de buenos hábitos

La primera medida de seguridad y la más importante es desarrollar buenos métodos y prácticas para usar el equipo. Podría parecer algo molesto, pero no lo es. Puesto que *usted* es quien controla en última instancia los programas, páginas Web y otro contenido que se cargan y se muestran en su equipo, puede hacer mucho por protegerse sin necesidad de disponer de software adicional ni de realizar cambios de configuración. La seguridad comienza en usted mismo y no dentro de su equipo. A continuación se enumeran algunos hábitos que puede poner en práctica:

Use las herramientas adecuadas. Hay varias herramientas de seguridad muy útiles que todo usuario doméstico debería usar, como por ejemplo un buen detector de virus. Consiga las herramientas adecuadas y utilícelas para ayudarle a proteger su información. (Y no olvide tampoco realizar una copia de seguridad de su sistema con regularidad.)

Acción: si dispone de un detector de virus, compruebe si existen actualizaciones para defenderse de nuevos tipos de virus. Si no dispone de ninguno, consígalo.

Mantenga su sistema actualizado: use Windows Update para comprobar periódicamente si hay nuevas actualizaciones de seguridad. Cuando vea que hay alguna actualización disponible, instálela lo antes posible. No es mala idea incluso suscribirse a los boletines de seguridad de Microsoft (en inglés); aunque tienden a ser bastante técnicos, constituyen una forma estupenda de mantenerse al corriente de determinadas cosas.

Agudice su paranoia: no quiero decir con eso que deba empezar a mirar al exterior por si aparecen helicópteros negros, sino que tiene que aprender a controlar los documentos adjuntos del correo electrónico que abre y qué sitios Web visita. En particular, si recibe un documento adjunto de correo inesperado de alguien que no conoce, **no lo abra**. En relación a esto, si recibe un documento adjunto que no espera de alguien desconocido, merece la pena comprobarlo dos veces para asegurarse de que pretendían enviárselo a usted. Puede resultar algo pesado, pero le compensará la primera vez que alguien a quien conoce se contagie con un virus que usted haya podido evitar.

Use un método de defensa de zona: las zonas de seguridad contribuyen a su protección, ya que permiten controlar qué código se ejecuta y en qué lugar. Compruebe que tiene la configuración de zona apropiada.

Refuerzo de la seguridad de Internet Explorer

Internet Explorer ofrece zonas de seguridad (en inglés) que le permiten controlar el momento en que IE descargará y ejecutará contenido de los sitios Web. Estas zonas son sólo agrupaciones de sitios; puede aplicar una configuración de seguridad en cada una de modo que todos los sitios de dicha zona estén sujetos a las mismas restricciones. A cada zona se le aplica un nivel de seguridad (bajo, medio, alto o personalizado); cuanto mayor es, menos cosas se pueden hacer. Para los usuarios domésticos hay tres zonas interesantes:

- **La Zona Sitios de Confianza**, donde el usuario coloca explícitamente los sitios que sabe que son de confianza, como Microsoft.com.

- **La Zona Sitios Restringidos**, donde el usuario pone los sitios en los que *no* confía de forma explícita. Los sitios de la zona Sitios restringidos son, como su nombre indica, restringidos, de forma que los controles y las secuencias de comandos que aparecen en sus páginas no se ejecutarán en el equipo local.

- **La Zona Internet**, donde se agrupan todos los sitios Web que no están en ninguna otra zona.

Por ejemplo, puede elegir permitir automáticamente la descarga y ejecución de los controles ActiveX de sitios de Internet concretos. Al decidir cuidadosamente en qué sitios confía y qué grado de riesgo está dispuesto a afrontar personalmente, puede personalizar la configuración de las zonas de seguridad de su equipo de modo que sólo se permita la carga del código que desee ejecutar.

Acción: compruebe minuciosamente la configuración de las zonas de seguridad de la instalación de Internet Explorer (consulte esta guía si necesita ayuda). Para la mayor parte de los usuarios, deseará que la zona Internet se configure con la seguridad de nivel medio; esto es así de forma predeterminada, pero merece la pena volver a comprobarlo, en especial si su equipo lo utilizan otras personas.

Refuerzo de la seguridad de Outlook y Outlook Express

Outlook y Outlook Express ofrecen numerosas características de seguridad para ayudar a impedir que el código peligroso pueda molestarle. Sin embargo, la característica más importante sigue siendo ¡no ejecutar documentos adjuntos desconocidos! Las características integradas que puede usar variarán en función de la versión de Outlook u Outlook Express que utilice.

Dado que Outlook y Outlook Express son clientes de correo con muchísimas características, permiten enviar y recibir correo HTML con formato muy variado. Por desgracia, al igual que con las páginas Web, los intrusos pueden incrustar código HTML o secuencias de comandos peligrosas en los mensajes HTML para que se ejecuten cuando se lea el mensaje o se haga clic en los vínculos que éste contenga.

Por suerte, tanto Outlook como Outlook Express permiten el uso de las zonas de seguridad de Internet Explorer, con lo que puede protegerse frente al contenido peligroso del correo que provenga de orígenes de confianza si configura sus zonas de forma apropiada.

Si utiliza Outlook Express

Si usa Outlook Express, puede hacer dos cosas, principalmente:

- Si usa una versión de Outlook Express anterior a Outlook Express 6.0, actualice el sistema a la versión actual.

- Repase las características de seguridad de Outlook Express 6.0 para que sepa cómo funcionan.

Acción: aumente la seguridad de Outlook Express activando su capacidad para bloquear ciertos tipos de documentos adjuntos. Cuando lo haga, Outlook Express comprobará por duplicado cada documento adjunto en relación a la lista de tipos de archivos registrados como inseguros o para "confirmar antes de abrir". Haga lo siguiente:

- Inicie Outlook Express.

- En el menú Herramientas, elija el comando Opciones.
- Cuando aparezca el cuadro de diálogo Opciones, haga clic en la ficha Seguridad.
- Busque la casilla de verificación "No permitir que se guarden o abran archivos adjuntos que puedan contener un virus". Actívela. Vea la Figura 1.
- Haga clic en el botón Aceptar.

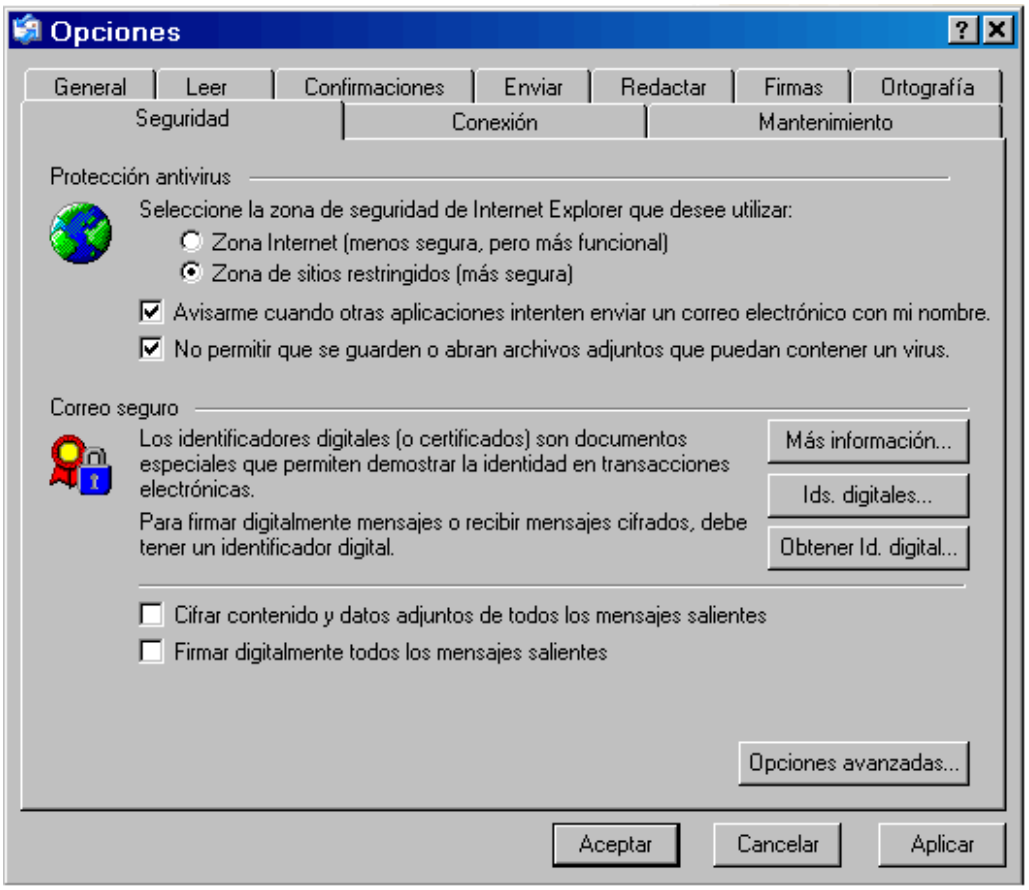


Figura 1. Outlook Express puede bloquear los archivos adjuntos peligrosos si le indica que lo haga.

Si utiliza Outlook

Outlook es el cliente de correo electrónico estrella de Microsoft; por lo tanto, dispone de algunas características de seguridad muy eficaces y flexibles, incluso cuando se ejecuta en el modo Internet. Estas características limitan los programas externos que pueden enviar correo, los tipos de acciones que Outlook puede realizar cuando llegan mensajes HTML o JavaScript, y los tipos de archivos que se pueden enviar y recibir como archivos adjuntos. A continuación se enumeran las acciones que puede realizar:

- Compruebe que Outlook está configurado para ejecutarse en la Zona de sitios restringidos. De este modo, puede recibir correo con formato HTML, pero su equipo está protegido frente a mensajes HTML peligrosos.
- Si usa Outlook 98, consiga la actualización de seguridad de Outlook 98.
- Si ejecuta Outlook 2000, consiga Office 2000 Service Release 1a y, después, instale la actualización de seguridad de Outlook 2000.
- Si usa Outlook 2002, relájese; ya incluye estas características de seguridad.
- Si es responsable de la administración de su propio equipo, no olvide comprobar con regularidad si hay actualizaciones. Puede encontrarlas en:
  http://office.microsoft.com/officeupdate/default.aspx?CTT=6&Origin=EC790020113082
- (A continuación, haga clic en "Actualizaciones de productos"). Las actualizaciones de Windows se pueden encontrar en:
  http://v4.windowsupdate.microsoft.com/es/default.asp

4.2.3 Herramientas de seguridad esenciales para los usuarios de oficinas domésticas

Normalmente, cuando se disponía de un equipo en casa, el mayor riesgo al que se debía hacer frente era la pérdida de datos provocada por un incendio, una avería hardware u otra catástrofe de índole similar. Aunque esos riesgos no han desaparecido, el uso generalizado de la conectividad permanente de alta velocidad a Internet nos ha expuesto a otras amenazas nuevas, al tiempo que han intensificado algunas de las antiguas. El aspecto favorable es que, con las herramientas adecuadas, usted puede hacer mucho por proteger su equipo doméstico frente a ataques y virus maliciosos, y también frente a otros problemas.

Algunas de estas herramientas vienen incluidas con varias versiones de Windows. Otras provienen de otros proveedores, como Symantec o McAfee, por ejemplo. No es necesariamente importante que use una marca concreta de herramientas; lo es más que disponga de las apropiadas, independiente de quién las fabrique.

Máximas más importantes

La mejor defensa es un buen ataque. Desde el punto de vista de la seguridad, eso significa que la acción más valiosa que puede emprender es tomar medidas proactivas para mantener el software de su sistema actualizado; dado que los intrusos malintencionados identifican nuevas formas de aprovechar puntos débiles, debe tener siempre actualizado el sistema para que su equipo esté protegido.

Acción: si utiliza Windows XP o Windows 2000, vaya a la página Microsoft Baseline Security Advisor (en inglés). MBSA es una aplicación basada en el Web que identifica qué revisiones de seguridad debería tener y las compara con las que *tiene*. Siga las recomendaciones de MBSA y comenzará con una sólida configuración de base.

Hay varias herramientas de seguridad integradas en Windows y en Internet Explorer, como son Windows Update, el Servidor de seguridad de conexión a Internet (ICF, *Internet Connection Firewall*) de Windows XP y otras diversas relacionadas con la privacidad. Aprender a usarlas le ayudará a conseguir la máxima seguridad con el menor costo.

Recursos de información

De acuerdo, los recursos de información técnica (sitios Web, libros y artículos como éste) pueden no parecer herramientas de la misma categoría que otras mencionadas aquí. Sin embargo, son igual de importantes porque cuanto más sepa acerca de los principios de seguridad, mejor equipado estará para tomar una buena decisión en relación a la configuración de seguridad de su equipo.

Windows XP Expert Zone

Windows XP Expert Zone es la comunidad oficial en línea para los entusiastas de Microsoft Windows XP. Se compone de artículos que publican semanalmente los expertos del sector acerca de cómo sacar provecho de las características y la funcionalidad de Windows XP.

Lista de comprobación de prácticas recomendadas

Microsoft ha proporcionado diversas listas de comprobación que pretenden servir de ayuda a los usuarios domésticos a la hora de identificar los pasos que deben seguir para proteger sus equipos en su domicilio.

Secure Microsoft Products

Junto con la lista de comprobación de prácticas recomendadas, Microsoft ha proporcionado sugerencias para que los usuarios domésticos sepan cómo usar y habilitar características de seguridad y privacidad en algunos de los productos de Microsoft más conocidos.

Fundamentos de privacidad

A medida que los usuarios y las corporaciones descubren lo eficaz que puede llegar a ser el Web tanto en la empresa como en el tiempo libre, trasladan a Internet las tareas que una vez se llevaron a cabo a través del teléfono o del correo. Con unos pocos clics de un explorador Web, una gran cantidad de información puede cambiar de manos. Microsoft ha proporcionado un sitio Web para dar a conocer a los usuarios domésticos el uso de diversos métodos que pueden emplear para evitar que su información privada caiga en las manos equivocadas

Servidores de seguridad

En el lenguaje de ingeniería, un servidor de seguridad es una barrera que aísla los componentes sensibles del peligro. Por ejemplo, esa pieza de metal que separa el compartimiento del pasajero de un automóvil del compartimiento correspondiente al motor es técnicamente un servidor de seguridad. Al igual que el de su homólogo de la automoción, el trabajo de un servidor de seguridad de red es mantener ciertos tipos de tráfico apartados del equipo y de los dispositivos de una red. Un servidor de seguridad bien diseñado e implementado correctamente puede mejorar en gran medida la seguridad de la red y reducir riesgos.

Un sistema servidor de seguridad puede incluir tanto componentes de software como de hardware. Hay diversas compañías que ofrecen soluciones de servidor de seguridad de software para equipos y redes domésticas. Además de los servidores de seguridad de terceros que puede instalar, Windows XP viene con un servidor de seguridad muy fácil de utilizar que se llama Servidor de seguridad de conexión a Internet, o ICF para abreviar. ICF de Windows XP proporciona una forma fácil de impedir que el tráfico exterior a la red tenga acceso a los elementos del equipo.

Recuerde que ICF se habilitará de forma predeterminada en las conexiones de red que cree mediante el Asistente para conexión nueva, de modo que no tiene que hacer nada especial para protegerlas. Si desea configurar su funcionamiento manualmente, Microsoft Technet contiene una gran variedad de artículos (algunos pueden estar en inglés) acerca de ICF de Windows XP, incluido uno en el que se proporcionan instrucciones paso a paso para habilitarlo. (También hay un artículo de Knowledge Base que proporciona una guía muy detallada, si lo prefiere.)

Acción: si utiliza Windows XP, active ICF en su conexión principal a Internet mediante estas sencillas instrucciones.

Detectores de virus

El término "virus" se usa para describir los programas informáticos autoduplicables que se propagan entre los archivos de un equipo e, incluso, entre equipos diferentes. Los virus suelen, en la mayoría de los casos, tener un fin malintencionado, como sobrescribir archivos o desperdiciar ancho de banda enviando copias de sí mismos a todos los destinatarios que haya en la libreta de direcciones del usuario.

La mejor forma de protegerse contra los virus es no contagiarse en el primer sitio que aparezca. Desde luego, en la práctica, esto resulta difícil, si no imposible, debido a que para estar completamente a salvo de los virus, un equipo tendría que permanecer aislado y no aceptar ninguna entrada externa que proviniera de Internet, discos, CD-ROM, etcétera. Su siguiente mejor opción es usar un paquete antivirus comercial de gran calidad. Estos detectores examinan los archivos, carpetas, mensajes de correo y páginas Web de un equipo en busca de los modelos distintivos de código de virus. Cuando el detector encuentra algo que parece un virus, pone en cuarentena al objeto sospechoso y advierte al usuario acerca de lo que ha encontrado.

Acción: si utiliza Outlook como cliente de correo, consiga e instale la Actualización de seguridad de Outlook (en inglés), que le ofrece una gran protección frente a los virus que se extienden a través del correo electrónico. Tenga en cuenta que la funcionalidad de esta actualización está integrada en Outlook 2002, que se incluye como parte de Office XP.

TechNet tiene una gran página de resumen que se actualiza continuamente para reflejar los virus que van apareciendo. Mientras la esté leyendo, reflexione sobre el hecho de que la mayor parte de los virus que porta el correo electrónico se difunden porque la gente no actúa como debe; por ejemplo, ejecuta los documentos adjuntos que provienen de orígenes desconocidos.

Herramientas de privacidad

En sentido estricto, las herramientas de privacidad no proporcionan ninguna seguridad adicional frente a intrusiones, virus u otros tipos de ataque. Sin embargo, contribuyen a proteger la información de un equipo para evitar que se revele de forma accidental o intencionada. Debería considerar qué grado de privacidad le conviene utilizar; para ello, puede leer la lista que ofrece Microsoft de fundamentos de la privacidad (en inglés) y decidir a continuación qué herramientas de seguridad de Microsoft o de otros proveedores le interesa usar.

Por ejemplo, supongamos que es médico. Ocasionalmente, conserva los historiales de los pacientes u otra información que debe ser confidencial en un equipo que utiliza en la oficina de su domicilio. Desea que estos registros no puedan leerlos otras personas, ni siquiera si alguien consiguiera robar el equipo. Si utiliza Windows 2000 Professional o Windows XP Professional, puede usar el Sistema de archivos de cifrado (EFS, *Encrypting File System*) para cifrar los datos seleccionados del disco duro de modo que nadie excepto usted o los usuarios que designe puedan leerlos, ni siquiera si se sustrae el equipo.
¿Y la privacidad en Internet? Internet Explorer 6.0 incluye un completo conjunto de características de protección de la privacidad, como son la capacidad de controlar cómo acepta su explorador los cookies y si mostrará o no las páginas de sitios que no tengan una directiva de privacidad claramente definida . Si le preocupa más que los fisgones puedan ver lo que hace en Internet, Internet Explorer dispone de herramientas que pueden encargarse de eso también.

3.3. Virus, Troyanos y Gusanos

4.3.1. Virus, gusanos y Caballos de Troya

Definición de Virus
Un virus informático, por definición, es un pequeño programa (o código) capaz de autoreproducirse. Sin embargo, algunos virus modernos también incluyen algo que los torna peligrosos: rutinas dañinas, también conocidas como bombas. La rutina dañina es la parte del virus que puede dificultarle la vida al usuario. Esta puede formatear su disco duro, dañar la FAT o destruir la Tabla de Partición del disco. También puede bloquear su sistema o crear algunos “efectos especiales” que a veces son interesantes de observar (como letras que caen en la pantalla). Recuperarse después de un ataque virósico no es una tarea trivial. Realmente consume mucho trabajo, costo y tiempo. Según un estudio hecho en 1994 por Dataquest, una de las principales organizaciones encuestadoras en el área de informática, el costo medio de recuperación a un ataque de virus en una red es de ¡US $15,000! Para empeorar la situación, la misma encuesta demostró que el 85% de las redes atacadas por virus vuelven a infectarse en un plazo

de no más de 30 días. Los virus informáticos han estado entre nosotros por muchos años, pero a fines de los 80's recibieron un reconocimiento general. Hoy en día, los usuarios de computadoras reconocen a los virus informáticos como una amenaza latente.

Otra definición de virus informático: es un pequeño programa creado para alterar la forma en que funciona un equipo sin el permiso o el conocimiento del usuario. Un virus debe presentar dos características:

Debe ser capaz de ejecutarse a sí mismo. A menudo coloca su propio código en la ruta de ejecución de otro programa.

Debe ser capaz de replicarse. Por ejemplo, puede reemplazar otros archivos ejecutables con una copia del archivo infectado. Los virus pueden infectar tanto equipos de escritorio como servidores de red.

Como ya se menciono, algunos virus están programados para atacar el equipo dañando programas, eliminando archivos o reformateando el disco duro. Otros no están creados para causar daño alguno, sino para replicarse y dar a conocer su presencia mediante la presentación de mensajes de texto, vídeo o sonido. Incluso estos virus benignos pueden crear problemas al usuario informático. Normalmente hacen uso de la memoria correspondiente a los programas legítimos. Como resultado, pueden provocar a menudo un comportamiento irregular en el equipo e incluso hacer que el sistema deje de responder. Además, muchos virus contienen errores que pueden ocasionar pérdidas de datos y bloqueos del sistema.

Ciclo de vida de un Virus

Los virus de computadoras así como los biólogos, poseen ciclos de vida que aquí se describen:

Creación:

Esta se produce cuando una persona con conocimientos de assembler trabaja en su gestación por varias semanas y termina cuando un nuevo virus que se encuentra programado para reproducirse rápidamente y hacer daño en algún momento determinado especificado por el programador.

Gestación:

Describe el proceso por el cual el virus se aloja en un sector determinado con el propósito de futuras reproducciones. Usualmente esto es hecho en un programa muy usado ubicándolo luego en una BBS o distribuyendo copias a través de la oficina, escuela, etc.

Reproducción:

Los virus, por su propia naturaleza, se replican. Un virus con buen diseño se replicará por un largo tiempo antes de que sea activado.

Activación:

Los virus que poseen rutinas de destrucción de datos se activarán cuando ciertas condiciones sean dadas. Algunos virus se activan en una fecha determinada mientras que otros poseen una especie de conteo regresivo interno de tiempo. Pero, aunque ciertos virus no tengan rutinas de daño (con lo cual sólo se reproducen), deben de considerarse igualmente destructivos, ya que toman recursos del sistema.

Descubrimiento:

Esta fase de ciclo de vida del virus, no necesariamente tiene por que producirse luego de la activación, pero usualmente sucede de esta manera. Esto se produce cuando alguien de la noticia de un nuevo virus. Usualmente, pasa a manos de la NCSA (National Computer Security Association) en Washington DC y es documentado y distribuido a diseñadores de antivirus.

Asimilación:

Luego del descubrimiento, los diseñadores de software modifican sus productos para incluir la detección del nuevo virus.

Erradicación:

Si los suficientes diseñadores de antivirus son capaces de detectarlo y limpiarlo así como los suficientes usuarios adquieren el antivirus apropiado para combatirlo el virus pude estar cerca de ser extinguido. A pesar de que, ningún virus ha desaparecido por completo, algunos han cesado por largo tiempo de aparecer en la comunidad informática.

Tipos de metodologías Antivirus

Existen distintos tipos de métodos de detección, protección y limpieza de virus informáticos.

- Rastreo
- Chequeo de Integridad
- Monitoreo de comportamiento

Rastreo:

El rastreo de virus se basa en una lista de detección y localización de virus conocidos. Esta lista de detección llamada “pattern” es una única pieza de código que identifica a un virus. Es el indicador y detector de un virus en particular. Cuando un nuevo virus aparece, es extraído del archivo infectado, analizado y actualizado a la lista de detección. Luego el programa de rastreo se ejecuta y comienza a efectuar una comparación entre los archivos rastreados y una base de datos de reconocimiento de virus. Si se encuentra una coincidencia, se considera que un archivo se encuentra infectado. Si no son encontradas coincidencias el rastreo es concluido y el archivo no se considera infectado. Los rastreadores de calidad chequearán las áreas de archivo y sectores de arranque. Una dificultad con los rastreadores tradicionales de virus es que si un nuevo virus no ha sido analizado infecta su computadora y no hay manera de detectarlo. Por lo tanto, si un archivo infectado con un nuevo virus no se encuentra dentro de la base de datos de detección, el rastreador no será capaz de detectarlo. Sin embargo, el rastreo, es un gran método para detectar virus conocidos en archivo.

Chequeo de integridad:

El chequeo de integridad depende de la habilidad y de la continuidad de monitoreo y chequeo del estado de los ejecutables con respecto a algunos cambios. Se debe primero ejecutar un “checksum” para todos los ejecutables cada vez que un software es agregado o el sistema se cambiado. El “chequeador” de integridad guarda el estado de cada archivo de aplicación en el disco duro y luego “chequea” ese estado y ve que cambios fueron hechos. Si se detecta un cambio, el “chequeador” de integridad avisa de la existencia de un virus. El principal defecto de los “chequeadores” de integridad es que no previenen de la acción, esto significa que avisa que un cambio se produjo en la integridad de un archivo, pero ya es tarde para efectuar una acción. De hecho, los “chequeadores” de integridad no pueden identificar la causa del cambio que se produjo en un momento. En conclusión los “chequeadores” de integridad de archivos usados como herramienta única no pueden ofrecer una protección de virus efectiva.

Monitoreo del Comportamiento:

El monitoreo de comportamientos extraños producidos por virus está usualmente acompañado de la instalación de un programa residente en memoria. A este tipo de programas se los denomina TSR (terminate-stay-resident) por una razón, debe monitorear los pedidos que son pasados a las interrupciones del DOS. Un comportamiento de virus se diferncia con respecto a las otras aplicaciones por realizar determinadas operaciones extrañas o anormales, éstas son denominadas “actividad de virus”. Esta actividad generalmente requiere la escritura de un sector de arranque, abrir un archivo ejecutable para la escritura, o ubicarse en una posición residente en memoria. Basados en determinados acciones comunes correspondientes a virus una

cierta cantidad de reglas deben ser establecidas para discernir entre la actividad del virus y las actividades normales de las aplicaciones. El monitoreo basado en reglas es una muy efectiva manera de detectar todos los virus conocidos como no conocidos. Esto impide la infección de un archivo ejecutable, antes de que ésta se produzca pudiendo, de esta manera, prevenir el daño. Por consiguiente, se puede decir que este método es particularmente preventivo de: Virus (tanto conocidos como no conocidos), Caballo de Troya o Bombas lógicas.

Virus en Internet

Introducción

Dentro de la tipificación de virus de archivo se podría decir que, un virus informático, por definición, es cualquier programa (o código) que se reproduce incorporando una copia de sí mismo en otro archivo. Un virus es particularmente peligroso si no se posee una detección y protección contra ellos. Es típico que los usuarios no sepan que sus sistemas están siendo infectados hasta que ven resultados que pueden ir de lo irrelevante hasta lo catastrófico.

Los virus tienen dos partes: una para producir daños y otra para reproducirse. La rutina que tiene como objetivo dañar afectar al sistema de varias maneras: formatea una unidad, destruye la tabla de particiones o utiliza al mismo entre otras cosas.

La frecuencia de los virus informáticos y sus altos costos han motivado el desarrollo de adopción de protecciones de virus en los servidores de redes. De hecho actualmente existen más de 10,000 virus en todo el mundo y de acuerdo con encuestas realizadas, las infecciones en distintos países oscilan entre el 35% y el 85% anual.

Los virus se presentan tanto en los sistemas Apple como en las PCS IBM compatibles. Sin embargo son más comunes en los últimos ya que la arquitectura MS-DOS hace que la programación de virus sea relativamente simple.

Limpiar un virus luego de la infección es costoso. Dependiendo de la magnitud de la infección, se calcula una incidencia en costos entre los 2,000 y los 500, 000 dólares estadounidenses por pérdida de datos y de productividad.

Tipos de Virus

En la parte más básica, los virus se pueden clasificar en dos tipos: los de archivo y los de arranque. Los virus de archivo residen dentro de los archivos tipo .EXE o .COM. Estos toman el control del sistema cuando estos son ejecutados, o agregan una copia de ellos en estos archivos. Los virus de arranque residen en el sector del disco que es cargado a memoria cuando el sistema se inicializa, cargándose en memoria antes que todos los otros programas. Esto permite al virus controlar las interrupciones del DOS y

poder, de esta manera, infectar discos flexibles que sean insertados y leídos en la disketera.

Dentro de esta tipificación existe una sub-clasificación de virus: los mutantes y los clandestinos “stealth”. Los primeros se encargan de cambiar su código cuando migran de un archivo a otro con el fin de no ser reconocidos por los rastreadores de virus comunes y así continuar su infección. En cambio los clandestinos cuando se cargan en memoria, toman el control de comportamiento del sistema y muestran en pantalla datos que en realidad nos son verdaderos. Por ejemplo el virus XUXA 2.1 que no deja ver la diferencia de tamaño del archivo infectado comparado con el original al acceder a un directorio.

Recientemente, los virus de archivo sólo podían infectar los del tipo .EXE y .COM, pero se ha detectado la presencia de nuevos virus que atacan a los archivos de trabajo, es decir, los generados por procesadores de textos como el Word, o plantillas de cálculo como Exel. Estos virus evaden a gran cantidad de rastreadores y es muy común verlos adosados en un archivo correspondiente a un e-mail. Esto último hace que su distribución e infección sea mucho más rápida y eficiente.

Entrando en Internet

Antes de la creciente popularidad de Internet, la distribución e infección de virus en una computadora a otra era más lenta. Por ejemplo, el virus Michelangelo apareció en Asia en 1991, y no apareció en los Estados Unidos hasta 2 años después. En Internet, un solo archivo puede transmitir un virus desde San Pablo hasta Tokio en sólo un minuto y considerado que actualmente, el número de archivos transmitidos por año en Internet se encuentra en el orden de billón notamos la diferencia de tiempo de reproducción de un virus actualmente comparando con años anteriores.

Los virus nacidos en Internet generalmente no son interceptados por los antivirus que se encuentran en el servidor a causa de la conexión. Generalmente la conexión a Internet no pasa por el servidor de la red interna a causa de una incompatibilidad de protocolos y sistemas utilizados. Mientras que los típicos servidores de red operan bajo Netware usando el protocolo IPX, el 80% de los Getways utilizados para Internet opera bajo UNIX usando el protocolo TCP/IP.

Capacidad de detección de virus

Tomaremos como ejemplo las características de un protocolo comercial especifico el ServerProtect for NT de TrendMicro, para describir las funciones que realiza para la detección de virus en archivos y sistemas.

Detección de virus conocidos

Cada virus contiene un único código de identificación, pero diferentes tipos de virus pueden hacer más fácil o difícil la tarea de encontrarlo e identificarlo como un virus conocido dentro del archivo. Tomando como un ejemplo al ServerProtect for NT ofrece diferentes tipos de métodos de rastreo para asegurar una rápida y segura detección.

DeepScan- Alta velocidad de rastreo

Existen sólo ciertos lugares dentro del archivo donde el código del virus se aloja. La tecnología DeepScan de Trend sigue el flujo de ejecución del programa para detectar e identificar el virus. Esta misma tecnología detecta virus sin necesidad de rastrear el archivo por completo, lo que provocaría un consumo elevado de tiempo. En este proceso implica que el tamaño del archivo a rastrear sea independiente a la velocidad de rastreo.

Softmice- Decodificación de virus polimórficos

Dado que el código de los virus polimórficos está encriptado y dado que la apariencia del virus va cambiando a medida que se reproduce, el método de detección DeepScan no funcionará correctamente. Por esto, ServerProtect for NT utiliza otra tecnología llamada Softmice para la detección de este tipo de virus. Esta tecnología crea una “PC virtual” generando un nuevo ambiente en donde desencriptará el virus para poder luego detectarlo e identificarlo sin problemas.

Wildcard Virus Scanner- detecta nuevas formas de virus conocidos

Muchos creadores de virus generan nuevos virus apartir de existentes modificando partes del código. Estas modificaciones pueden engañar a los antivirus tradicionales, ServerProtect incorpora dentro de su tecnología de detección, Wildcard Virus Scaner, capaz de detectar modificaciones e identificar al nuevo virus.

Compressed File Scanner

Detecta virus en archivos comprimidos, el motor de rastreo de ServerProtect NT incluye rutinas de descompresión de archivos que le permite detectar virus dentro de archivos comprimidos. ServerProtect NT reconoce todos los formatos de compresión más utilizados.

Macro Scanner- detecta virus macro conocidos

Los virus macro se han popularizado rápidamente en este último tiempo desde que el primero fue detectado cerca de Agosto de 1995. El rastreador de virus macro de

ServerProtect está diseñado para detectar todos los virus macro conocidos desde el servidor NT.

Detección de virus desconocidos

A parte de detectar virus conocidos, un rastreador eficiente debe estar preparado para detectar la presencia de virus no conocidos en el sistema. ServerProtect también incorpora varios métodos de detección de virus no conocidos para cumplir este objetivo.

Behavior Monitor

Detecta virus desconocidos por comportamiento, por definición, un virus es un programa que intenta reproducirse. Para llevar a cabo ese cometido debe replicarse y hacer algunos cambios en el sistema sin el permiso y el consentimiento del usuario. Analizando el sistema de búsqueda de este tipo de comportamiento virósicos es como ServerProtect puede reportar la detección de un virus no conocido. De esta forma alerta al administrador para efectuar una investigación más profunda. Es importante señalar que estos comportamientos no son generados por el funcionamiento normal de un programa, sino por programas que incluyan código de virus.

Checksumming

Protege los archivos de cambios no esperados, ServerProtect for NT utiliza chequeos de integridad de archivos CRC para detectar cambios no esperados en archivos ejecutables del sistema. Esta tecnología combinada con la detección de virus no conocidos por comportamiento, crea un seguro mecanismo de protección.

MacroTrapTM

Detecta virus macro no conocidos, teniendo en cuenta la cantidad de virus macro que surgen día a día es importante contara con mecanismos especiales para detectar aquéllos que todavía no son conocidos por los sistemas antivirus tradicionales. MacroTrap utiliza una combinación de la tecnología OLE de Microsoft con un sistema capaz de detectar virus macro no conocidos por comportamiento. Esta tecnología funciona a partir de la detección de cambios y el análisis de macros en documentos de MS Word y planillas de cálculo MS Exel generados por algún virus macro no conocido.

En general la mayoría de las aplicaciones comerciales utiliza procedimientos similares para la detección de virus, de la eficiencia en sus algoritmos de búsqueda depende su efectividad.

Tipos de virus conocidos

Virus que infectan archivos

Este tipo de virus ataca a los archivos de programa. Normalmente infecta el código ejecutable, contenido en archivos .com y .exe, por ejemplo. También pueden infectar otros archivos cuando se ejecuta un programa infectado desde un disquete, una unidad de disco duro o una red. Muchos de estos virus están residentes en memoria. Una vez que la memoria se infecta, cualquier archivo ejecutable que no esté infectado pasará a estarlo. Algunos ejemplos conocidos de virus de este tipo son Jerusalén y Cascade. También llamados infectores parásitos, estos programas son generalmente, mas pequeños que 4kb. Los virus de archivo "viven" como unos parásitos dentro del archivo .EXE o .COM. Cuando el programa es ejecutado, el virus toma el control del sistema. Luego de tomar el control, usualmente buscará otro archivo .exe o .com y tratará de agregar una copia de sí mismo en esos archivos. Cuando este "ataca" otro archivo, puede también ubicarse al principio del código del archivo "víctima" (ponerse antes del archivo y agregar una instrucción de salto), agrangarse (ponerse al final del archivo y agregar una instrucción de salto al comienzo), o puede sobrescribir el código del mismo insertándose entre el código del archivo víctima. Por lo tanto, estos virus pueden expandirse rápidamente sin que el usuario lo note en la ejecución de sus programas. Así como la mayoría de los virus, los virus de archivo observarán el sistema esperando que una condición se cumpla. Esta condición puede ser tanto la fecha del sistema como el número de archivos en el disco, entre otras cosas. Cuando la condición está dada, el virus comienza su rutina de daño si es que la tiene.

Virus del sector de arranque

Estos virus infectan el área de sistema de un disco, es decir, el registro de arranque de los disquetes y los discos duros. Todos los disquetes y discos duros (incluidos los que sólo contienen datos) tienen un pequeño programa en el registro de arranque que se ejecuta cuando se inicia el equipo. Los virus del sector de arranque se copian en esta parte del disco y se activan cuando el usuario intenta iniciar el sistema desde el disco infectado. Estos virus están residentes en memoria por naturaleza. La mayoría se crearon para DOS, pero todos los equipos, independientemente del sistema operativo, son objetos potenciales para este tipo de virus. Para que se produzca la infección basta con intentar iniciar el equipo con un disquete infectado. Posteriormente, mientras el virus permanezca en memoria, todos los disquetes que no estén protegidos contra escritura quedarán infectados al acceder a ellos. Algunos ejemplos de virus del sector de arranque son Form, Disk Killer, Michelangelo y Stoned.

Virus del sector de arranque maestro

Los virus de arranque (también llamados usualmente de buteo) sólo pueden expandirse a través de disquettes. Si usted está familiarizado con el DOS, entonces sabrá que

cada vez que el sistema se inicializa, generalmente la unidad A será a la primera a la que se pedirá acceso, y luego la unidad C. Si usted tiene un disquete puesto en la unidad A, el sistema intentará inicializar el DOS desde ese disquete. Los virus de buteo "viven" en un sector de los discos que son cargados en memoria en tiempo de arranque del sistema. Por lo tanto, el virus de arranque se carga a memoria antes de que cualquier programa lo haga. Esta es la respuesta del porque no se debe arrancar el sistema con un disquete a menos que se esté completamente seguro que está limpio de virus. Una vez que el virus de buteo se carga en memoria, puede observar las interrupciones que realizan el DOS y reinfectar más disquetes así como son puestos y accedidos desde la disquetera. Este tipo de virus son difíciles de descubrir, aunque fáciles de detectar por los antivirus.
Estos virus están residentes en memoria e infectan los discos de la misma forma que los virus del sector de arranque. La diferencia entre ambos tipos de virus es el lugar en que se encuentra el código vírico. Los virus del sector de arranque maestro normalmente guardan una copia legítima del sector de arranque maestro en otra ubicación. Los equipos con Windows NT infectados por virus del sector de arranque o del sector de arranque maestro no podrán arrancar. Esto se debe a la diferencia en la forma en que el sistema operativo accede a la información de arranque, en comparación con Windows 95/98. Si el sistema con Windows NT está formateado con particiones FAT, normalmente se puede eliminar el virus arrancando desde DOS y utilizando un programa antivirus. Si la partición de arranque es NTFS, el sistema deberá recuperarse utilizando los tres discos de instalación de Windows NT. Algunos ejemplos de virus del sector de arranque maestro son NYB, AntiExe y Unashamed.

Virus múltiple

Estos virus infectan tanto los registros de arranque como los archivos de programa. Son especialmente difíciles de eliminar. Si se limpia el área de arranque, pero no los archivos, el área de arranque volverá a infectarse. Ocurre lo mismo a la inversa. Si el virus no se elimina del área de arranque, los archivos que hayan sido limpiados volverán a infectarse. Algunos ejemplos de virus múltiples son One_Half, Emperor, Anthrax y Tequila.

Virus de macro

El hecho de pensar que un virus pudiera atacar la valiosa información de una empresa en forma directa, almacenada en un vulnerable archivo de trabajo MS-Word o al esfuerzo de varias horas representado en un archivo MS-Exel, hasta ahora, no pasaba por la cabeza de los usuarios. Valiéndose de las macros automáticas introducidas en dichos programas y del hecho que éstas se ejecuten sin el consentimiento del usuario, nace un nuevo concepto en virus informáticos, los "Macro Virus". Los primeros que aparecieron fueron los conocidos como Word.Npad y WinWord. Concept (este último conocido como MS-Word AAAZAO y Prank). Actualmente, está apareciendo virus Macro que no sólo se reproduce, sino que también causan daño en los archivos del

sistema, tal es el caso de virus Tedius, un nuevo virus Macro que luego de infectar la computadora y en un momento determinado elimina subdirectorios y archivos de arranque del sistema dejando, de esta manera, inutilizable al mismo.
Estos virus infectan los archivos de datos. Son los más comunes y han costado a empresas importantes gran cantidad de tiempo y dinero para eliminarlos. Con la llegada de Visual Basic en Microsoft Office 97, se puede crear un virus de macro que no sólo infecte los archivos de datos, sino también otros archivos. Los virus de macro infectan archivos de Microsoft Office: Word, Exel, PowerPoint y Acces. Actualmente están surgiendo también nuevos derivados en otros programas. Todos estos virus utilizan el lenguaje de programación interno de otro programa, creado para permitir a los usuarios automatizar ciertas tareas dentro del programa. Debido a la facilidad con que se pueden crear estos virus, existen actualmente miles de ellos en circulación. Algunos ejemplos de virus de macros son W97M.Melissa, WM.NiceDay y W97M.Groov.

Virus “Stealth”

Los virus “Stealth” son variaciones de los virus de archivo. Cabe recordar que una forma simple de reconocer la infección de un virus de archivo es comparando el tamaño en bytes del archivo infectado con el limpio. Los virus “Stealth” no muestra esta diferencia de tamaño al usuario, guardando una copia de la información del archivo en el disco. Recuerde que una vez que el virus está residente en memoria puede hacer muchas cosas. Por lo tanto, cada vez que se ejecute un “DIR” (con el virus en memoria) se obtendrá una lista de los archivos y sus tamaños antes de que se produjera la infección, camuflando así su presencia.

Virus Polimórficos

Estos virus son también llamados “mutantes”. Los virus polimórficos trabajan de la siguiente manera: Se ocultan en un archivo y se cargan en memoria cuando el archivo infectado es ejecutado. Pero a diferencia de hacer una copia exacta de sí mismo cuando infecta otro archivo, modifica esa copia para verse diferente cada vez que infecta un nuevo archivo. Valiéndose de estos “motores de mutación”, los virus polimórficos pueden generar miles de copias diferentes de sí mismos. A causa de esto, los rastreadores convencionales han fallado en la detección de los mismos. Ciertamente, cada rastreador de virus creado antes de Enero de 1992 no será capaz de detectar virus polimórficos. De hecho, la mayoría de las herramientas de rastreo utilizadas actualmente todavía no pueden detectar estos virus. La tecnología de algunas herramientas (como TrendMicro) es capaz de detectar en forma consistente este tipo de virus polimórficos observando eventos característicos que los mismos deben realizar para sobrevivir y expandirse. Cualquier virus, sin importar sus características debe hacer ciertas cosas para sobrevivir. Por ejemplo, debe infectar otros archivos y residir en memoria.

Gusanos y caballos de Troya

¿Qué es un gusano?

Los gusanos son programas que se replican a sí mismo de sistema a sistema sin utilizar un archivo para hacerlo. En esto se diferencia de los virus, que necesitan extenderse mediante un archivo infectado. Aunque los gusanos generalmente se encuentran dentro de otros archivos, a menudo documentos de Word o Exel, existe una diferencia en la forma en que los gusanos y los virus utilizan el archivo que los alberga. Normalmente el gusano generará un documento que ya contendrá la macro del gusano dentro. Todo el documento viajará de un equipo a otro, de forma que el documento completo debe considerarse como gusano. PrettyPark.Worm es un buen ejemplo de gusano.

¿Qué es un caballo de Troya?

Los caballos de Troya son impostores, es decir, archivos que pretenden ser benignos pero que, de hecho, son perjudiciales. Una diferencia muy importante con respecto a los virus reales es que no se replican a sí mismos. Los caballos de Troya contienen código dañino que, cuando se activa, provoca pérdidas o incluso robo de datos. Para que un caballo de Troya se extienda es necesario dejarlo entrar en el sistema, por ejemplo abriendo un archivo adjunto de correo. Un ejemplo de caballo de Troya es PWSteal.Trojan.

¿Qué es una falsa alarma de virus?

Las falsas alarmas de virus son mensajes, casi siempre enviados por correo electrónico, que se asemejan a las cartas en cadena. Algunas de las expresiones utilizadas a menudo en estas falsas alarmas son: Si recibe un mensaje de correo electrónico titulado [nombre de la falsa alarma de virus], no lo abra. Bórrelo inmediatamente. Contiene el virus [Nombre de la falsa alarma]. Borrará el contenido del disco duro y [aquí se especifica un peligro extremo e improbable]. Nombre de una empresa famosa ha informado hoy de la existencia de este virus. Remita este aviso a todos sus conocidos.

Muchos de los avisos de falsas alarmas de virus no se alejan mucho de este patrón. Si no está seguro de si un aviso es legítimo o si se trata de una falsa alarma, puede encontrar más información en: http://www.symantec.com/avcenter/hoax.html (este recurso se encuentra en inglés).

¿Qué no es un virus?

Debido a la publicidad que han recibido los virus, es fácil culparlos de cualquier problema informático. Los siguientes problemas normalmente no están causados por virus ni otro tipo de código dañino:

Problemas de hardware. No existen virus que puedan dañar físicamente el hardware, como por ejemplo chips, placas o monitores. El equipo emite un pitido durante el inicio y no aparee nada en pantalla. Normalmente esto se debe a un problema de hardware durante el proceso de arranque. Consulte la documentación del equipo para comprobar el significado de los pitidos.

El equipo no registra 640 k de memoria convencional. Esto puede ser un síntoma de virus, pero no es definitivo. Algunos controladores de hardware, como los correspondientes al monitor o a la tarjeta SCSI, pueden hacer uso de parte de esta memoria. Consulte con el fabricante del equipo o con el proveedor de hardware para averiguar si es el caso.

Tiene dos programas antivirus instalados y uno de ellos informa de la existencia de un virus. Aunque podría tratarse de un virus, también puede ser que uno de los programas antivirus detecte la firma del otro programa en la memoria. Si desea obtener más información, consulte el documento ¿Debería ejecutar más de un programa antivirus a la vez?

Se está usando Microsoft Word y este programa le avisa de que un documento contiene macros. Esto no significa que la macro sea un virus. No es posible abrir un documento concreto. Esto no indica la presencia de un virus necesariamente. Intente abrir otro documento o una copia de respaldo del documento en cuestión. Si otros documentos se abren sin problemas, puede que el documento esté dañado.

Ha cambiado la etiqueta de un disco duro. Todos los discos pueden tener una etiqueta. Se puede asignar una etiqueta a un disco duro mediante el comando Label de DOS o desde Windows.

Otro problema especifico con Norton Antivirus. Cuando se ejecuta ScanDisk, la función Auto-Protect de NAV informa de la existencia de actividad vírica. A continuación se ofrecen dos posibles soluciones.

Desactivación de Auto-Protect

1. Inicie NAV y desactive temporalmente la función Auto-Protect.
2. Ejecute ScanDisk y deje que corrija los errores.
3. Vuelva a activar Auto-Protect.

Modificación de una opción de ScanDisk

1. Inicie ScanDisk y elija la opción Completa.
2. Haga clic en Opciones.
3. Desactive la opción “No realizar pruebas de escritura”.
4. Ejecute ScanDisk de nuevo.

4.3.2 Clasificación de Riesgos

TrendLabs, la red global de centros investigación antivirus y soporte a productos de Trend Micro, provee una rápida respuesta a cualquier brote de virus o solución de soporte urgente para algún cliente. Cuando un caso es recibido TrendLabs inmediatamente evalúa la amenaza y asigna una clasificación de riesgos de **Muy Bajo, Bajo, Medio o Alto**. Muchos factores contribuyen a cada clasificación de riesgos como son:

- La extensión en la propagación que cada código malicioso ha logrado o sus estadísticas de prevalecía en el mundo real.

- El daño potencial de cada código malicioso en un solo sistema o en una red de sistemas.

Lo particular del código malicioso, el uso extenso del sistema objetivo, y la habilidad del código malicioso para esparcirse por sí mismo a otros archivos, sistemas o a través de redes.

El criterio para determinar la clasificación de riesgo esta diseñado para mejorar la productividad corporativa, ligada a cómo Trend Micro se ha posicionado el mismo en el ambiente gateway, servidor, y escritorio. Esto además toma en consideración las características específicas que cada producto puede agregar para mejorar la seguridad y estabilidad en la empresa. Los clientes son la más alta prioridad para Trend Micro y recibirán todos los beneficios de la certificación ISO 9002 de Trend Micro.

La evaluación de riesgos de TrendLabs es además un sistema de alerta preventivo (EWS early warning system) respaldado por investigadores profesionales de antivirus y de seguridad. La información obtenida de sus bastos centros de servicio y unidades de negocio (Bus bussiness units) es evaluada, analizada y redistribuida con la solución para ayudar a los administradores de red gerentes a evaluar la vulnerabilidad de su sistema y advertirles en cómo asegurar sus redes durante el brote de virus.

Las amenazas recientemente descubiertas son anunciadas en el Centro de Información de Virus que además alberga la base de datos completa de la Enciclopedia de Virus.

Adicionalmente, el sitio despliega las estadísticas de los 10 mas comunes virus, reflejando resultados del Centro Mundial de rastreo de virus. La severidad de cada amenaza es evaluada bajo los niveles de evaluación de riesgos generales, que depende de la combinación de factores marcados en la sección de Factores de Evaluación.

Factores de Evaluación

El esparcimiento en tiempo real de *infecciones reportadas* es medido por los reportes que vienen del Centro de Rastreo Mundial de virus, y también de las unidades de negocio alrededor del mundo, las cuales reciben los reportes de amenazas actuales reportan y soportan la información en sus áreas. Los reportes de otros vendedores de la industria antivirus y medios de atención contribuyen a este factor.

"*Daño potencial y Peligro para los sistemas*", son derivados de las características del programa malicioso. Algunos programas maliciosos han sido conocidos por atacar sistemas operativos de archivos importantes dejando a los sistemas inestables o incapaces de reiniciar.

"*Distribución Potencial*", es también derivada de las características del programa malicioso. Los gusanos de red de rápida distribución han sido capaces de distribuirse a través de continentes en cuestión de minutos. Algunos programas maliciosos además usan numerosas técnicas de infección y de distribución – comúnmente referidas como "blended threats" amenazas mezcladas. El virus Nimda, por ejemplo, fue capaz de esparcirse vía correo electrónico, redes compartidas y sitios Web infectados y por el tráfico Web (http/puerto 80).

Conforme un Nuevo sistema es creado y mejorado con funcionalidad extra, normalmente sigue una prueba de concepto del código malicioso. Esta característica única así como la amplia implementación de un sistema operativo o software además influye la distribución Potencial de cada código malicioso. Muchos virus escrito en el pasado no corren o se distribuyen en un nuevo sistema operativo, o sistemas operativos que tienen instalados los últimos parches de seguridad. Para mas información por favor lea las Prácticas seguras de Cómputo de Trend Micro.

Esquema de Niveles de Evaluación de Riesgos

Tipo: Alto
Criterio:
Muchos reportes de infección provenientes de las unidades de negocio (BU) reportan el rápido esparcimiento del código malicioso en donde los gateways y servidores de correo electrónico pueden requerir ser parchados.

Se inicia un proceso de solución de una Alerta Roja en los primeros 45 minutos: Una liberación oficial de un patrón (OPR) es desarrollada con una notificación de su disponibilidad, cualquier otra notificación relevante es enviada y las herramientas de reparación y la información de reparación de vulnerabilidades es colocada en las páginas de descarga.

Tipo: Medio
Criterio:
Los reportes de infección son recibidos de varias Bus así como las llamadas de soporte confirman instancias dispersas. Una Liberación de patrón oficial (OPR) es automáticamente colocada para los servidores de desarrollo y puesta a disposición para una descarga.

En caso de que el código malicioso se difunda por correo electrónico, las reglas de filtrado de contenido, llamadas políticas de prevención de infección (OPP outbreak prevention policies), son enviadas para bloquear automáticamente los archivos anexos relacionados, en los servidores equipados con la funcionalidad del producto .

Tipo: Bajo
Criterio:
Reportes de una sola infección son recibidos y definido como virus diario, referido como una liberación de patrón controlado (CPR controlled pattern releases), se ponen a disposición para una descarga.
En algunas instancias en donde una prueba de concepto de código malicioso es manejada, o numerosas investigaciones de soporte son recibidas, TrendLabs puede incrementar el nivel de **Muy Bajo** a **Bajo** y enviar la correspondiente notificación conjuntamente con una liberación oficial de patrón (OPR).

Tipo: MuyBajo
Criterio:
Todos los otros casos procesados automáticamente reciben una clasificación de Muy Bajo Riesgo y los CPRs diarios se ponen a disposición para su descarga. Los virus Zoo usualmente tienen una clasificación de Riesgo de **Muy Bajo** .

4.3.3. Protección contra los ataques “Día Cero”

Introducción

El 11 de noviembre de 2003, Microsoft sacó al mercado un parche para una falla en el servicio de las estaciones de trabajo de Windows 2000 y XP. De acuerdo con Microsoft, una intrusión exitosa le daría al atacante control total sobre un equipo en peligro. Lo que sucedió después sorprendió a casi todos. Menos de 24 horas después de que Microsoft sacara el parche, dos miembros de lista de correo de seguridad BugTraq publicaron un código para aprovechar la vulnerabilidad.
Como ha informado eWEEK, las epidemias de gusanos de los últimos tres años han adoptado generalmente el siguiente patrón: un distribuidor o investigador revela la vulnerabilidad, e inmediatamente se produce el lanzamiento del parche. Luego viene la confusión para poner parches a los sistemas vulnerables antes de que alguien lance el código de la vulnerabilidad en la Web. Finalmente, el código se usa como base para un gusano, que luego invade Internet con diversos niveles de éxito. Es un patrón que fue establecido por el gusano Código Rojo en julio de 2001, y que ha permanecido básicamente igual a través de las apariciones de Nimda, Slammer, Blaster y muchos otros gusanos más. Sin embargo, lo que ha cambiado es el tiempo que transcurre desde que sale un parche hasta que aparece un gusano. Esta “ventana de amenazas de vulnerabilidades” continúa reduciéndose… y rápidamente.

Las amenazas “día cero” son inminentes

Symantec en sus Informes sobre amenazas a la seguridad de Internet ha comentado anteriormente sobre la reducción del tiempo que transcurre entre el descubrimiento de la vulnerabilidad y el ataque generalizado. En la más reciente edición del informe, publicado en marzo de 2004, Symantec sugiere que las amenazas “día cero” son ahora inminentes. Una amenaza combinada “día cero” puede tener como objetivo una vulnerabilidad antes que sea anunciada y que exista el parche. Si se produce el ataque, se puede causar un gran daño antes de que los usuarios puedan parchear con eficacia sus sistemas. Como señala el Informe sobre las amenazas:

“Es prácticamente seguro que hay todavía vulnerabilidades desconocidas, que se pueden aprovechar remotamente y que están ocultas en tecnologías muy utilizadas. Una amenaza combinada “día cero” puede tener como objetivo esta clase de vulnerabilidades. Si se produce un ataque “día cero”, probablemente no habrá parches disponibles durante muchos días. Incluso se tardaría más en la identificación de los medios de propagación de ese gusano que éste en poner en peligro todos los sistemas vulnerables. Es sorprendente que esto no haya sucedido aún”

En esta situación sin precedentes, donde una respuesta oportuna y eficaz es absolutamente vital, ¿qué medidas pueden tomar las empresas para proteger sus

sistemas y garantizar su continuidad? La pregunta se hace más urgente a la luz de un reciente informe de la firma de investigación Gartner Inc. ("La seguridad e xige un mayor impulso en el control de vulnerabilidades" de noviembre de 2003), que descubrió que, el año pasado, muchas compañías comprobaron que los firewall perimétricos, los software antivirus y los sistemas de detección de intrusos no bastan para protegerse de los ataques cibernéticos. Los resultados de Gartner permiten explicar por qué cada vez más las empresas van más allá y buscan ayuda para proteger sus sistemas de información crítica.

Contratación externa: Una tendencia cada vez mayor

Existen razones de peso para que las compañías opten por la contratación externa para las funciones de seguridad de TI. Por ejemplo, como ha señalado Gartner, el personal responsable de las funciones de seguridad de TI en la mayoría de las compañías también realiza otras actividades y gasta mucho tiempo en proyectos que no son de seguridad. Para una organización con recursos limitados, la responsabilidad adicional de administrar la seguridad es con frecuencia muy agotadora. Gartner ha concluido que los equipos de las compañías se esfuerzan por conocer y protegerse de las últimas amenazas a la seguridad porque se requiere un monitoreo constante de los sistemas, lo que unas pocas empresas pueden hacer las 24 horas del día, los 7 días de la semana. Lo fundamental es que muchas empresas se están dando cuenta de que mantener la vigilancia necesaria requiere inversiones en personal, sistemas de TI y entrenamiento, lo que simplemente es demasiado caro. Para estas organizaciones, los servicios de seguridad administrada pueden ayudar a reducir el costo total de propiedad al reducir los gastos de las empresas necesarios para monitorear y administrar con eficacia la tecnología de seguridad.

Gartner, en su análisis de la evolución del mercado de servicios de seguridad administrados en 2003 ("Cuadrante mágico para los proveedores de servicios de seguridad administrada (MSSP) estadounidenses" de noviembre de 2003), también mencionó los siguientes aspectos como los factores que han contribuido al incremento de la contratación externa:

- Las grandes empresas continuaron restringiendo la contratación de personal y presionaron para contratar externamente muchas funciones de TI que no eran consideradas competencias básicas.

- Muchas empresas que han instalado sensores de sistemas de detección de intrusos (IDS) están pensando implementar los servicios de seguridad administrada como una forma de controlar la avalancha de falsos positivos.

- Con los ataques de gusanos de alta visibilidad a principios del año 2003, muchas empresas recurrieron a los servicios administrados para notificar más rápidamente sobre las vulnerabilidades, ataques y soluciones.

Consideraciones importantes

La seguridad eficaz requiere tecnología adecuada, personal experimentado y procesos probados, además de una continua labor de inteligencia contra las amenazas. ¿Cómo todo ello determina la selección de un proveedor de servicios de seguridad administrada (MSSP)? Para empezar, el MSSP debe ofrecer gran variedad de servicios entre los que se encuentran:

- Servicios administrados y monitoreados de firewall.
- Sistemas administrados y monitoreados de detección de intrusos (IDS) para redes.
- Servicios administrados y monitoreados de appliances de seguridad integrada.
- ISD monitoreados para equipos.
- Servicios administrados de evaluación de vulnerabilidades en Internet.
- Servicios administrados de cumplimiento de las políticas de seguridad.
- Servicios administrados de protección antivirus.

Las empresas también deben asegurarse de que el MSSP seleccionado opere con centros de operación de seguridad (SOC) múltiples desde el que pueda monitorear y administrar globalmente los asuntos de seguridad permanentemente las 24 horas, todos los días. Los aspectos importantes que se deben tener en cuenta son si el MSSP tiene respaldo apropiado y si los SOC están dispersos geográficamente de forma que complementen la configuración de la empresa.

Además, un MSSP debe poder correlacionar, analizar e interpretar con precisión los grandes volúmenes de información sobre la seguridad de la red en tiempo real. Las empresas deben establecer que el MSSP seleccionado examine toda la información en lugar de hacer muestreos estadísticos de la misma.

Los informes suministrados por un MSSP deben ser lo suficientemente detallados como para respaldar las decisiones que mejoren los esfuerzos de seguridad y determinen la productividad de los servicios administrados. Los informes completos

tienen información recolectada de appliances administrados, respuestas recomendadas, cambios en los appliances realizados por el MSSP e información sobre las últimas amenazas. Además, las empresas están cumpliendo cada vez más con la legislación que establecerá estudios rigurosos de cumplimiento, como la Ley Sarbanes-Oxley.

Quizás lo más importante es que las empresas necesitan estar seguras de que recibirán alertas tempranas sobre las nuevas amenazas globales para que puedan actuar antes de que el daño se produzca y se propague. El MSSP debe recolectar y analizar la información de los cientos de miles de puntos alrededor del mundo para detectar las tendencias.

Finalmente, las empresas deben estar satisfechas de que los expertos en seguridad del MSSP tengan las certificaciones del sector o de los productos en una gran variedad de proveedores y tecnologías, y de que sus instalaciones, personal, procesos y procedimientos han sido auditados por terceros de confianza. Las compañías también deben determinar la rotación del personal de planta de los SOC para realizar las operaciones permanentemente las 24 horas del día, los siete días de la semana, con el fin de garantizar que el personal de planta más cualificado no se concentre en un solo turno.

4.4 Backdoors, Hoaxes y Jokes

4.4.1. Troyanos backdoor la nueva generación de vandalismo cibernético

Los **Troyanos Backdoor**, se han consolidado como la temible nueva generación de vandalismo cibernético, debido a su modalidad de incursión ilegal en servidores, estaciones de trabajo o PCs, pudiendo tomar el control de los sistemas comprometidos, con los consiguientes daños que ello implica, nada menos que a través de cualquiera de los **65535 puertos TCP/IP**.

Los Troyanos Backdoor no son esencialmente virus, sino "Herramientas de Control Remoto"

Aunque su existencia no es nueva, ya que desde hace mucho tiempo se comercializan software legal para controlar redes LAN en forma remota, como **PC-Anywhere** o **Remote-Anything** entre muchísimos otros, también se distribuyen herramientas "**Freeware**" que cumplen similares cometidos.

A pesar que diariamente se propagan virus contenidos en archivos infectados dentro de mensajes de correo, no son pocos los usuarios que motivados por el desconocimiento, la curiosidad, o por no tener actualizados sus sistemas operativos, navegadores,

software de correo o antivirus, permiten que esta modalidad todavía sea la más utilizada por los creadores de virus.

En el año 2002 se han reportado una alarmante cantidad de **Troyanos/Backdoor** y se estima que existen más de 5,000 troyanos creados desde 1997, los cuales pueden ser monitoreados y controlados a través de un software Cliente asociado. En muchos portales de hackers se distribuyen estos ilegales sistemas, incluyendo las potentes herramientas de "**barrido**" de puertos TCP/IP que permiten detectar las "puertas traseras" abiertas, mediante las cuales pueden ingresar uno de sus componentes.

Troyanos/Backdoor de acceso remoto

Tienen dos componentes principales: el programa Servidor, que se instala en el sistema de la victima y el programa Cliente que actúa en la computadora del atacante. Ambos programas establecen una relación Cliente/Servidor entre la PC infectada y la del atacante. Por medio de estos troyanos el atacante puede ejecutar remotamente en los sistemas infectados las mismas acciones que el administrador de un Servidor o usuarios de las PC involucradas.

Troyano/Backdoor Cliente

El Cliente se encuentra en el equipo del atacante y generalmente tiene una interfaz con opciones y desde las cuales puede ejecutar las funciones que se hayan programado para que interactúen con los sistemas de las víctimas.

Troyano/Backdoor Servidor

El Servidor que se instala en el sistema de la victima, es un programa que ocupa muy poco espacio y está asociado al Cliente, para poder recibir las instrucciones o través del mismo, ejecutar las funciones que el intruso esté facultado. Los troyanos/backdoor se pueden transmitir por diversos medios:

Mensajes de Correo

Son la forma más fácil de propagación por medio de un archivo anexado al mensaje y si el receptor comete el error de ejecutarlo, instalará el Servidor, permitiendo que el intruso pueda controlar el o los equipos infectados.

Telnet

Funcionan en modo Cliente/Servidor y permite ejecutar comandos en el equipo infectado.

Telnet es un programa de emulación de terminal para las redes TCP/IP. Opera en una computadora y la conecta con un servidor de la red. A partir de ese instante, un usuario puede ingresar comandos a través del programa Telnet y cada instrucción será ejecutada como si la hubiera ingresado directamente en la consola del servidor o el equipo asignado.

Redes Compartidas
Este medio de contagio lo hemos explicado ampliamente en Virus en Peer to Peer. Ejemplo típico de intrusión a la red Kazaa de archivos compartidos a través de Telnet:

telnet <dirección-IP-víctima> 1214GET / HTTP/1.0
HTTP/1.1 200 OK
Content-Length: 2467
Accept-Ranges: bytes
Date: Tue, 08 Feb 2003 14:14:36 GMT
Server: KazaaClient Feb 08 2003 17:18:29
Connection: close
Last-Modified: Fri, 27 Dec 2002 14:14:36 GMT
X-Kazaa-Username: <Nonbre-usuario-víctima>
X-Kazaa-Network: MusicCity
X-Kazaa-IP: 200.44.XX.XXX:1214
X-Kazaa-SupernodeIP: 24.168.48.42:1214
Content-Type: text/html

Otros servicios de Internet (HTTP, FTP, ICQ, Chat, Mensajería Instantánea)

Es posible visitar una página Web en Internet, la misma que descargue automáticamente un troyano Backdoor Servidor y el sistema quedará infectado, bajo control del troyano Cliente. Del mismo modo podrá ocurrir en servidores **FTP**. Por lo general estos servidores, al ser reportados, serán deshabilitados por su ISP, en caso contrario el Proveedor de Servicios de Internet será merecedor a una sanción.

La popularidad del uso del **Chat** o de los servicios de Mensajería Instantánea, como **MSN Messenger**, **Yahoo Messenger**, **Netscape** o **AOL Messenger**, entre otros, han hecho posible la transmisión de virus, macro virus, gusanos, troyanos y backdoors, entre los usuarios conectados en una misma sesión.
Usuarios de una misma red local o por medio de diskettes.

El sabotaje interno, obviamente no podría ser descartado.

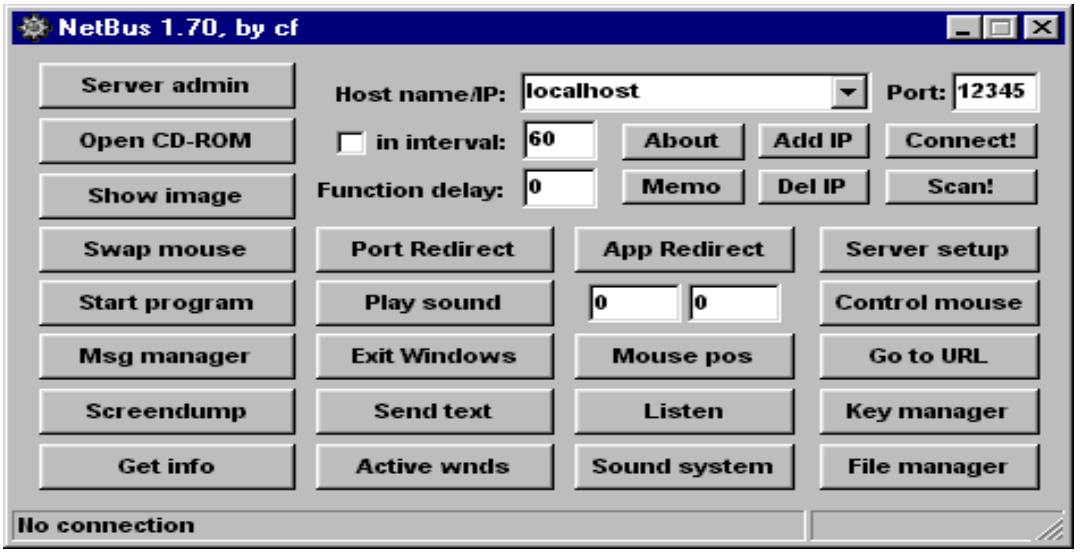
Recomendaciones:
Algunos aspectos importantes para disminuir el riesgo de intrusiones:

- Filtrado y protección de las comunicaciones (Firewall).
- Actualización de software (parches de seguridad).
- Monitoreo intensivo.
    - **No usar Claves de Acceso con nombres obvios o asociados al de los usuarios, como fechas de nacimiento, apelativos, etc.**
- Una estricta política de manejo y control de los usuarios en carpetas Compartidas.
- Cualquier acción preventiva, jamás estará de más.

Tres Troyanos/Backdoor famosos:

NetBus

Desarrollado por el programador sueco **Carl-Fredrik Neikter** en Marzo de 1998, quien alguna vez dijo: "NetBus fue creado para que la gente tenga algo de diversión con sus amigos. Yo espero que NetBus y programas similares como Back Orifice harán que la gente sea más consciente de los riesgos de seguridad de sus sistemas".
Desafortunadamente, el uso de estos programas fue más allá de ser aparentes simples travesuras:

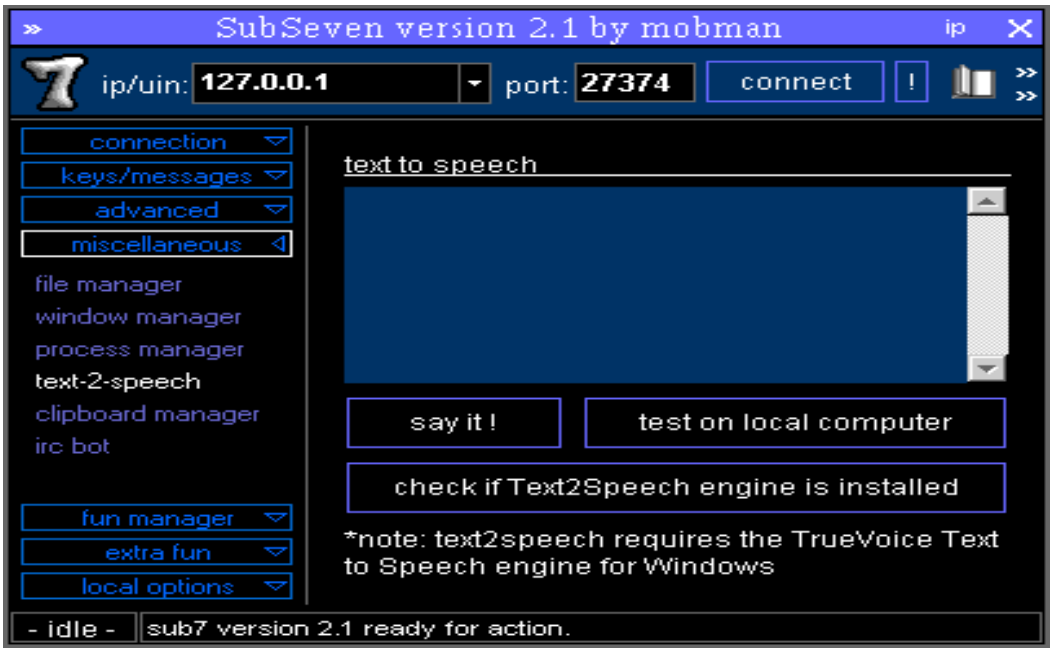


Back Orifice

El 03 de Agosto de 1998 el grupo de hackers conocidos como the **Cult of the Dead Cow** (el Culto a la Vaca Muerta) sorprendió al mundo con el lanzamiento de su popular y temido **Back Orifice**, que incluso fue distribuido con su código fuente y plug-ins, además de sus componentes compilados.

SubSeven

Desarrollado en Alemania por el grupo de hackers S7G o Sub7Germany:



Algunas capacidades de los Troyanos/backdoor:

- Extraer y enviar información del sistema al intruso.
- Descargar o enviar archivos de la computadora.
- Substraer o cambiar los password o archivos de passwords.
- Anular procesos en ejecución.
- Mostrar mensajes en la pantalla.
- Realizar acciones nocivas o dañinas: manipular el mouse, mostrar/ocultar la Barra de Tareas, abrir y cerrar la bandeja del CD, reiniciar la computadora, formatear el disco duro, entre otras acciones.

El sustento de esta gravísima amenaza es la existencia de **65535 Puertos TCP/IP** disponibles, a través de los cuales de encontrarse "abiertos", será posible ingresar un pequeño programa Servidor componente de cualquier sistema Troyano/Backdoor. Los puertos más empleados pueden verse en este URL:

http://www.simovits.com/sve/nyhetsarkiv/1999/nyheter9902.html

4.4.2. Amenazas y Malware

No solo los virus suponen una amenaza para nuestro ordenador. Hasta hace unos años, los virus constituían la principal amenaza para los equipos informáticos. Los virus son programas que se reproducen infectando otros ficheros o aplicaciones y realizan acciones perjudiciales para el usuario.

Con posterioridad aparecieron los gusanos, programas que no necesitan infectar otros ficheros para reproducirse, y que se propagan realizando copias de sí mismos, con el fin de colapsar las redes en las que se infiltran.

Y los troyanos y backdoors, aparentemente inofensivos, pero que buscan introducirse en el ordenador para capturar contraseñas y pulsaciones del teclado, permitir el acceso remoto a la información almacenada, etc.

Sin embargo, en los últimos tiempos, y debido principalmente a la generalización del uso de la informática y del acceso a Internet entre el gran público, han aparecido otras amenazas capaces de resultar muy dañinas y que obligan a redefinir el concepto de amenaza, también denominada como malware.

La palabra malware proviene de la composición de las palabras inglesas malicious software, es decir, programas maliciosos. Se entiende por malware cualquier programa, documento o mensaje que puede resultar perjudicial para un ordenador, tanto por pérdida de datos como por pérdida de productividad.

Así, aparte de los ya conocidos virus, gusanos, troyanos y backdoors cabe incluir dentro del malware:

- Dialer: tratan de establecer conexión telefónica con un número de tarificación especial.
- Joke: gasta una broma informática al usuario.
- Riesgo de Seguridad: son herramientas legales que pueden ser empleadas de forma malintencionada.

- Herramienta de Hacking: permite a los hackers realizar acciones peligrosas para las víctimas de los ataques.

- Vulnerabilidad: es un fallo en la programación de una aplicación, a través del cual se puede vencer la seguridad del ordenador y realizar intrusiones en el mismo.

- Programa Espía: recoge datos acerca de hábitos de uso de Internet y los envía a empresas de publicidad.

- Hoax: son mensajes de correo electrónico con advertencias sobre falsos virus.

- Spam: es el envío indiscriminado de mensajes de correo no solicitados, generalmente publicitarios.

Todos ellos configuran el panorama del malware en la actualidad.

Backdoors, Hoaxes y jokes

Backdoors

Un backdoor es un programa que se introduce en el ordenador de manera encubierta, aparentando ser inofensivo. Una vez es ejecutado, establece una "puerta trasera" a través de la cual es posible controlar el ordenador afectado. Esto permite realizar en las mismas acciones que pueden comprometer la confidencialidad del usuario o dificultar su trabajo.

Las acciones permitidas por los backdoors pueden resultar muy perjudiciales. Entre ellas se encuentran la eliminación de ficheros o la destrucción de la información del disco duro. Además, pueden capturar y reenviar datos confidenciales a una dirección externa o abrir puertos de comunicaciones, permitiendo que un posible intruso controle nuestro ordenador de forma remota.

Algunos ejemplos de backdoors son: Orifice2K.sfx, Bionet.318, Antilam y Subseven.213.

Virus falsos y bromas pesadas que pueden confundir al usuario.

Hoaxes Jokes

Existen ciertos tipos de mensajes o de software que a veces son confundidos con virus, pero que no lo son en ningún sentido. Es muy importante conocer las diferencias para no sufrir las consecuencias negativas de una confusión.

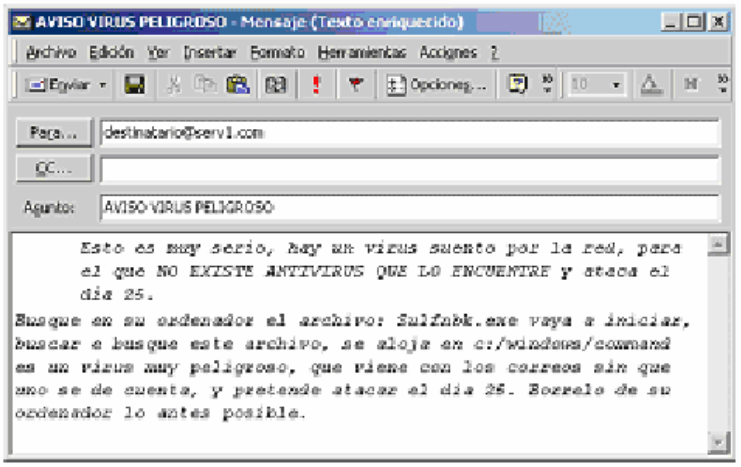
Hoaxes

Los hoaxes no son virus, sino mensajes de correo electrónico engañosos, que se difunden masivamente por Internet sembrando la alarma sobre supuestas infecciones víricas y amenazas contra los usuarios.

Los hoaxes tratan de ganarse la confianza de los usuarios aportando datos que parecen ciertos y proponiendo una serie de acciones a realizar para librarse de la supuesta infección.

Los hoax (mistificación, broma o engaño), son mensajes con falsas advertencias de virus, o de cualquier otro tipo de alerta o de cadena (incluso solidaria, o que involucra a nuestra propia salud), o de algún tipo de denuncia, distribuida por correo electrónico.

Si se recibe un hoax, no hay que hacer caso de sus advertencias e instrucciones: lo más aconsejable es borrarlo sin prestarle la más mínima atención y no reenviarlo a otras personas.



Mensaje de presentación del hoax SULFNBK.EXE.

Jokes

Un joke tampoco es un virus, sino un programa inofensivo que simula las acciones de un virus informático en nuestro ordenador. Su objetivo no es atacar, sino gastar una broma a los usuarios, haciéndoles creer que están infectados por un virus y que se están poniendo de manifiesto sus efectos.

Aunque su actividad llega a ser molesta, no producen realmente efectos dañinos.

Desde que el correo electrónico se extendió como medio de comunicación, los jokes comenzaron a llegar a las casillas de usuarios de todo el mundo. Normalmente

enviados por algún amigo para jugarnos una broma, algunas personas comenzaron a considerarlos molestos, y esto dio pie a que varias casas antivirus consideraran que aquellos jokes relacionados con virus debían ser incluidos en sus bases de firmas.

Dada la proliferación de estas bromas, incluso algunas comparativas de antivirus incluyeron este tipo de programas en sus muestras y, obviamente, aquellos productos que no detectaban Jokes por considerar que no hacen nada inofensivo, salían peor parados en sus ratios de detección frente a los que sí lo hacían.

Además de este inconveniente, muchas veces la detección de programas de broma trae confusión a los usuarios. Es normal que los soportes técnicos de las compañías antivirus reciban desesperados llamados de personas que ven que su antivirus detecta algo en su sistema, sin saber que al ser un Joke, es algo inofensivo.

Ante estos temas, muchas casas antivirus optaron de todas formas por incluir Jokes en sus bases para no tener problemas en las comparativas, lo cual tan sólo ayuda a acrecentar la confusión y transformar el antivirus en algo que no es: un detector de bromas.

Ejercicio 1. Verificación de Archivo

Instrucciones

1. Descarga el programa MD5summer, que sirve para verificar la integridad de archivos.

2. Instálalo en la computadora.

3. Selecciona 2 archivos de tu computadora, ejecuta MD5summer y escoge los archivos, genera el archivo de verificación de cada uno con la opcion “Create sums” ¿Qué sucedió?¿Que nombre tiene los archivos que creaste?.

4. Realiza el envió de ambos archivos a tu compañero predeterminado de tu clase en 2 emails, uno para cada uno, además adjunta el archivo generado por MD5.

5. Guarda en la misma carpeta los 4 archivos.

6. Con MD5summer comprueba la integridad de los envíos con la opción Verify sums. ¿Qué resultado obtuviste?
7. Ahora cambia el nombre de los archivos, y de nuevo intenta comprobar su integridad. ¿Qué sucedió ahora? ¿Por qué?

8. Ahora intercambia los nombres de los 2 los archivos por el original del otro ¿Cuál fue el resultado ahora? ¿Puedes explicarlos?.

9. ¿Qué tienes que hacer para que ambos archivos vuelvan a dar positivo en la verificación?

10. Elabora un reporte con tus resultados, con el formato de práctica siguiente:

    - Portada (1 cuartilla).
    - Introducción (¿Investiga para que sirve MD5, quien lo propone y como funciona? (máximo 3 cuartillas).
    - Desarrollo de la práctica.
    - Resultados.
    - Conclusiones (1 cuartilla).

11. Entrega la práctica en un archivo de Word, antes de las 24 hrs del dia XX de mes XX, con el nombre de Seg_Intpráctica411equipoXXgrupoXX.doc.

Ejercicio 2.

Instalación y prueba de SW antivirus por equipos

1. Descarga la versión de prueba e instale en la computadora o realice la búsqueda de virus en línea de 2 de los siguientes antivirus:

   http://www.mcafee.com/mx/
   Panda Antivirus Platinum 7.0 en http://www.pandasoftware.com/com/mx/
   Norton AntiVirus™ 2005 FREE trialware en http://nct.symantecstore.com/0001/
   Trend Micro en línea http://housecall.trendmicro.com/

2. Procede a la búsqueda de virus con dos de las herramientas antes mencionadas, toma el tiempo que se lleva en efectuar el proceso total.

3. Describe las características del al computadora analizada, tipo y velocidad de procesador, RAM, disco duro, sistema operativo, principales programas instalados y tipo de uso.

4. Copia el cuadro de resumen generado en una imagen (jpg, bmp, gif, etc) para incluirla en tu reporte, describe los resultados y su significado, si encontraste algún virus describe sus características y clasificación, así como la forma en que pudo lograr la infección.

5. Elabora un reporte con tus resultados, con el formato de práctica siguiente:

- Portada (1 cuartilla).
- Introducción (Describa características, ventajas y desventajas de los 2 antivirus. utilizados, incluya las variaciones de producto que existen(máximo 6 cuartillas).
- Desarrollo de la práctica.
- Resultados.
- Conclusiones (1 cuartilla).

6. Entregue la práctica en un archivo de Word, antes de las 24 hrs del dia XX de mes XX, con el nombre de SegInt_práctica431equipoXXgrupoXX.doc

Páginas Electrónicas de Referencia

• Panda Software. ***Amenazas y Malware.*** Disponible en
http://www.pandasoftware.es/virus_info/about_virus/keys1.htm
Consultado 30 de septiembre de 2004.

• Panda Software. ***Hoaxes y jokes.*** Disponible en
http://www.pandasoftware.es/virus_info/about_virus/keys3.htm
Consultado 30 de septiembre de 2004.

• Panda Software. ***Virus, gusanos, troyanos y backdoors***. Disponible en
http://www.pandasoftware.es/virus_info/about_virus/keys2.htm
Consultado 30 de septiembre de 2004.

• VsAntivirus. ¿Incluir o no incluir? Ésa es la cuestión (I. Sbampato).
Disponible en http://www.vsantivirus.com/sbam-incluir.htm
Consultado 30 de septiembre de 2004.

• VsAntivirus. ¿Qué es un HOAX? Disponible en
http://www.vsantivirus.com/hoaxes.htm
Consultado 30 de septiembre de 2004.

5. Firewalls

5.1 Conceptos

Firewalls

Muchos de los problemas de seguridad que aparecieron con la interconexión de redes en el surgimiento de Internet pueden ser remediados o atenuados mediante el uso de determinadas técnicas controles. Con un firewall podemos implementar un nivel de seguridad apropiado permitiendo al mismo tiempo el acceso a los vitales servicios de Internet. Un firewall es un sistema o un grupo de sistemas que implementan una política de control de acceso entre dos o más redes.

Podemos imaginarlo como compuesto por dos grandes módulos; uno destinado a bloquear los accesos y el otro a permitirlos. El firewall constituye la herramienta pero se desprende que debemos tener muy claro que tipo de control de acceso debemos implementar, y a su vez esto constituye un subconjunto de la política de seguridad de la compañía.

Un Firewall en Internet es un sistema o grupo de sistemas que impone una política de seguridad entre la organización de red privada y el Internet. El firewall determina cual de los servicios de red pueden ser accedidos dentro de esta por los que están fuera, es decir quien puede entrar para utilizar los recursos de red pertenecientes a la organización. Para que un firewall sea efectivo, todo tráfico de información a través del Internet deberá pasar a través del mismo donde podrá ser inspeccionada la información. El firewall podrá únicamente autorizar el paso del tráfico, y el mismo podrá ser inmune a la penetración. Desafortunadamente, este sistema no puede ofrecer protección alguna una vez que el agresor lo traspasa o permanece entorno a este.

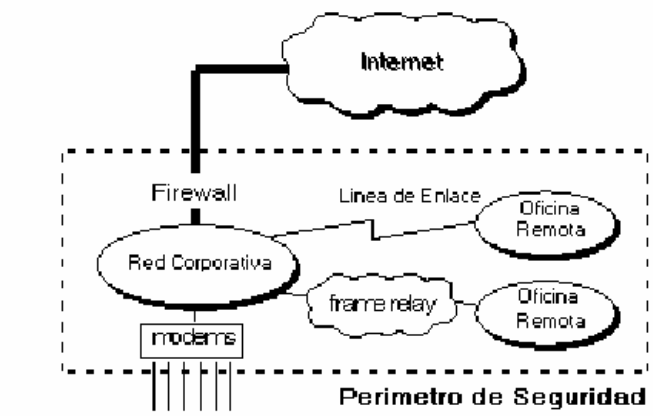


Ilustración 5.1.1 La Política De Seguridad Crea Un Perímetro De Defensa.

Esto es importante, ya que debemos de notar que un firewall de Internet no es justamente un ruteador, un servidor de defensa, o una combinación de elementos que proveen seguridad para la red. El firewall es parte de una política de seguridad completa que crea un perímetro de defensa diseñada para proteger las fuentes de información. Esta política de seguridad podrá incluir publicaciones con las guías de ayuda donde se informe a los usuarios de sus responsabilidades, normas de acceso a la red, política de servicios en la red, política de autenticidad en acceso remoto o local a usuarios propios de la red, normas de dial-in y dial-out, reglas de encriptación de datos y discos, normas de protección de virus, y entrenamiento. Todos los puntos potenciales de ataque en la red podrán ser protegidos con el mismo nivel de seguridad. Un firewall de Internet sin una política de seguridad comprensiva es como poner una puerta de acero en una tienda.

El firewall proporciona un único check-point que preserva a la intranet del ataque de intrusos que pudieran accederla. Nos permite monitorear la seguridad a través de sus alarmas y logs, los cuales deben ser revisados periódicamente pues debemos poder determinar hipotéticos intentos de acceso ya que el mismo firewall puede ser violado y una vez que esto sucede estamos sin protección. Como agregados a su función primordial los firewalls proveen dos funciones extras; el servicio de NAT-Network Address Translator que permite que un servidor de mi intranet se presente en Internet con un número de IP válido sin necesidad de reconfigurarlo, y por otro lado ofrece un punto de logeo y monitoreo del uso de Internet en cuanto a requerimientos para poder determinar anchos de bandas, problemas de saturación del vínculo, etc.

Como se dijo un firewall es parte de una política y no podemos dejarle librada toda la responsabilidad cuando tenemos accesos de tipo PPP por ejemplo, o un empleado divulga la información; este tipo de casos están denotando la existencia de back-doors evidentemente muy apetecibles como objetivo de cualquier ataque. Otro punto importante es que a pesar de que la información pase por un firewall este no nos puede garantizar 100% la ausencia de virus, pues su inmensa variedad hace imposible que se pueda analizar cada uno de los paquetes que pasan a través de él. Por último, si bien en la actualidad el tema está lo suficientemente pulido, existen aplicaciones que pueden estar mal diseñadas y que permiten que se puedan transmitir paquetes no deseados encapsulados dentro de los mensajes que trafican; este era el caso de viejas versiones del sendmail, por ejemplo. Por todo esto la utilización de un firewall no constituye la panacea para resolver todos los problemas de seguridad de Internet. Podemos citar algunos ejemplos de usos más comunes de firewalls. La protección ante la utilización de servicios vulnerables; expusimos la vulnerabilidad de algunas aplicaciones, entre las cuales podemos encontrar en la actualidad el NFS y NIS, por dar un ejemplo. Este tipo de servicios son vulnerables a los ataques; no podemos optar por deshabilitarlos pues son muy útiles en la intranet con lo cual con la utilización de un firewall estaríamos filtrando todos los accesos externos a este tipo de servicios. Por otro lado se puede administrar un control de acceso; esto se traduce en implementar políticas que permitan el acceso a algunos servidores y a otros no. Además la

utilización de un firewall nos permite perfeccionar el control de la seguridad en cuanto a su centralización, pues además de todo el subsistema de auditoria podríamos concentrar todos los add-on de software de seguridad en un punto central en lugar de implementarlo encada hosts (sumando a esto el mantenimiento que ello implica).

Otras soluciones de este tipo como por ejemplo Kerberos obligan a hacer actualizaciones hosts por hosts y si bien en algunos casos son las soluciones más adecuadas los firewalls tienden a simplificar esta tarea. Como una actividad secundaria podemos citar el hecho de que si todo el tráfico hacia Internet pasa por un firewall esto me permite a partir de su accounting determinar el grado de uso del vínculo de red y proyectar crecimiento; este último punto debe estar soportado por otras herramientas.

Beneficios de un firewall en Internet

Los firewalls en Internet administran los accesos posibles del Internet a la red privada. Sin un firewall, cada uno de los servidores propios del sistema se expone al ataque de otros servidores en el Internet. Esto significa que la seguridad en la red privada depende de la "Dureza" con que cada uno de los servidores cuenta y es únicamente seguro tanto como la seguridad en la fragilidad posible del sistema.

El firewall permite al administrador de la red definir un "choke point" (envudo), manteniendo al margen los usuarios no-autorizados (tal como hackers, crakers, vándalos, y espías) fuera de la red, prohibiendo potencialmente la entrada o salida al vulnerar los servicios de la red, y proporcionar la protección para varios tipos de ataques posibles. Uno de los beneficios clave de un firewall en Internet es que ayuda a simplificar los trabajos de administración, una vez que se consolida la seguridad en el sistema firewall, es mejor que distribuirla en cada uno de los servidores que integran nuestra red privada.

El firewall ofrece un punto donde la seguridad puede ser monitoreada y si aparece alguna actividad sospechosa, este generara una alarma ante la posibilidad de que ocurra un ataque, o suceda algún problema en el transito de los datos. Esto se podrá notar al acceder la organización al Internet, la pregunta general es "si" pero "cuando" ocurrirá el ataque. Esto es extremadamente importante para que el administrador audite y lleve una bitácora del tráfico significativo a través del firewall. También, si el administrador de la red toma el tiempo para responder una alarma y examina regularmente los registros de base. Esto es innecesario para el firewall, desde que el administrador de red desconoce si ha sido exitosamente atacado.
Concentra la seguridad Centraliza los accesos:

- Genera alarmas de seguridad.
- Traduce direcciones (NAT).
- Monitorea y registra el uso de Servicios de WWW y FTP.
- Internet.

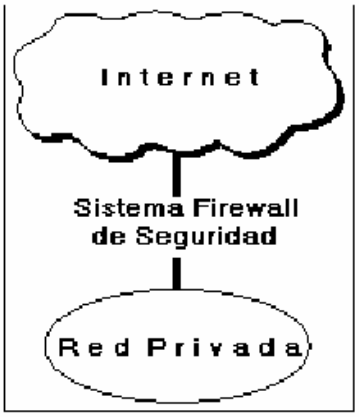


Ilustración 5.1.2 Beneficios de un Firewall de Internet.

Con el paso de algunos años, el Internet ha experimentado una crisis en las direcciones, logrando que el direccionamiento IP sea menos generoso en los recursos que proporciona. Por este medio se organizan las compañías conectadas al Internet, debido a esto hoy no es posible obtener suficientes registros de direcciones IP para responder a la población de usuarios en demanda de los servicios. Un firewall es un lugar lógico para desplegar un Traductor de Direcciones de Red (NAT - Network Address Translator) esto puede ayudar aliviando el espacio de direccionamiento acortando y eliminando lo necesario para re-enumerar cuando la organización cambie del Proveedor de Servicios de Internet (ISPs- Internet Service Providers) .Un firewall de Internet es el punto perfecto para auditar o registrar el uso del Internet. Esto permite al administrador de red justificar el gasto que implica la conexión al Internet, localizando con precisión los cuellos de botella potenciales del ancho de banda, y promueve el método de cargo a los departamentos dentro del modelo de finanzas de la organización.

Un firewall de Internet ofrece un punto de reunión para la organización. Si una de sus metas es proporcionar y entregar servicios información a consumidores, el firewall de Internet es ideal para desplegar servidores WWW y FTP.

Finalmente, el firewall puede presentar los problemas que genera un punto de falla simple. Enfatizando si este punto de falla se presenta en la conexión al Internet, aun así la red interna de la organización puede seguir operando - únicamente el acceso al Internet esta perdido.

La preocupación principal del administrador de red, son los múltiples accesos al Internet, que se pueden registrar con un monitor y un firewall en cada punto de acceso que posee la organización hacia el Internet. Estos dos puntos de acceso significan dos

puntos potenciales de ataque a la red interna que tendrán que ser monitoreados regularmente.

Limitaciones de un firewall

Un firewall no puede protegerse contra aquellos ataques que se efectúen fuera de su punto de operación. Por ejemplo, si existe una conexión dial-out sin restricciones que permita entrar a nuestra red protegida, el usuario puede hacer una conexión SLIP o PPP al Internet. Los usuarios con sentido común suelen "irritarse" cuando se requiere una autenticación adicional requerida por un Firewall Proxy server (FPS) lo cual se puede ser provocado por un sistema de seguridad circunvecino que esta incluido en una conexión directa SLIP o PPP del ISP.

Este tipo de conexiones derivan la seguridad provista por firewall construido cuidadosamente, creando una puerta de ataque. Los usuarios pueden estar consientes de que este tipo de conexiones no son permitidas como parte de integral de la arquitectura de la seguridad en la organización.

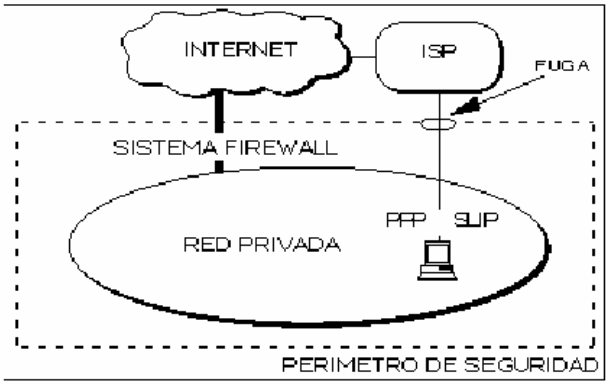


Ilustración 5.1.3 Conexión Circunvecina al Firewall de Internet.

El firewall no puede protegerse de las amenazas a que esta sometido por traidores o usuarios inconscientes. El firewall no puede prohibir que los traidores o espías corporativos copien datos sensitivos en disquetes o tarjetas PCMCIA y extraigan estas del edificio.

El firewall no puede proteger contra los ataques de la "Ingeniería Social", por ejemplo un Hacker que pretende ser un supervisor o un nuevo empleado despistado, persuade al menos sofisticado de los usuarios a que le permita usar su contraseña al servidor del corporativo o que le permita el acceso "temporal" a la red.

Para controlar estas situaciones, los empleados deberían ser educados acerca de los varios tipos de ataque social que pueden suceder, y a cambiar sus contraseñas si es necesario periódicamente. El firewall no puede protegerse contra los ataques posibles a la red interna por virus informativos a través de archivos y software. Obtenidos del Internet por sistemas operativos al momento de comprimir o descomprimir archivos binarios, el firewall de Internet no puede contar con un sistema preciso de SCAN para cada tipo de virus que se puedan presentar en los archivos que pasan a través de el.

La solución real esta en que la organización debe ser consciente en instalar software anti-viral en cada despacho para protegerse de los virus que llegan por medio de disquetes o cualquier otra fuente. Finalmente, el firewall de Internet no puede protegerse contra los ataques posibles en la transferencia de datos, estos ocurren cuando aparentemente datos inocuos son enviados o copiados a un servidor interno y son ejecutados despachando un ataque.

Por ejemplo, una transferencia de datos podría causar que un servidor modificara los archivos relacionados a la seguridad haciendo más fácil el acceso de un intruso al sistema. Como nosotros podemos ver, el desempeño de los servidores Proxy en un servidor de defensa es un excelente medio de prohibición a las conexiones directas por agentes externos y reduce las amenazas posibles por los ataques con transferencia de datos.

5.2 Tipos de Firewalls.

Conceptualmente hay dos tipos de firewalls, nivel de red y nivel de aplicación. Los firewalls de nivel de red toman sus acciones en función del origen, la dirección de destino y el port en cada paquete IP. Los modernos firewalls de este tipo se han sofisticado y mantienen información respecto del estado de las conexiones que están activas a través de él, etc. Este tipo de firewall tiende a ser muy rápidos y son transparentes al usuario.

Los firewalls de nivel de aplicación por lo general son hosts corriendo proxy servers, que no permiten el tráfico directo entre redes, manteniendo una elaborada auditoria y logeo del tráfico que pasa a través de él. Este tipo de firewall puede ser utilizado para realizar las tareas relativas al NAT, debido a que como las comunicaciones van de un lado hacia el otro se puede enmascarar la ubicación original.

Este tipo tiende a proveer una auditoria más detallada y un mayor grado de seguridad que los de nivel de red. Los routers de filtro de paquetes, que corresponden al primer grupo, realizan una decisión del tipo pasa no pasa para cada paquete que recibe. El router examina cada datagrama para determinar si se aplican sus reglas de filtrado. Las reglas de filtrado se basan en la información contenida en el header del paquete. Esta información consiste en el IP de origen, la IP de destino, el protocolo encapsulado

(TCP, UDP, ICMP), el port TCP/UDP de origen y de destino, etc. Toda esta información es controlada contra las reglas de filtrado definidas, pudiendo ser enrutada si existe una regla que lo permite, descartada si una regla así lo indica y si no existe regla comparable un parámetro previamente configurado determinará si el paquete pasa o no.

Dentro de este tipo están los que filtran en función del servicio involucrado. Esto es posible pues hay muchos servicios para los cuales están normalizados los ports en los que escuchan, por lo cual se pueden definir reglas que involucren el port, definiendo la aceptación o el rechazo.

Por otro lado frente a diferentes ataques que se fueron produciendo surgieron otros firewalls cuyas reglas son independientes del servicio; estas reglas exigen un análisis más detallado que involucra el ruteo, las opciones de IP, verificación de los fragmentos de desplazamiento y puntos por el estilo.

La mayoría de los firewalls implementados sobre Internet están desarrollados sobre el concepto de filtrado de paquetes. Este tipo de firewalls no son difíciles de configurar debido a que su software contiene una serie de reglas previamente configuradas y fundamentalmente son transparentes al usuario y no exigen instalar ningún software adicional en los hosts.

Por otro lado cuando se debe customizar de manera tal de adaptarlo a aplicaciones específicas de cada empresa la tarea se puede hacer algo compleja pues exige una figura de administrador que debe conocer los servicios de Internet, los distintos encabezados de los paquetes, los distintos valores que se espera encontrar en los campos a analizar. Si se requiere un filtrado complejo, las reglas pueden volverse demasiado largas con la consecuencia de una difícil administración y seguimiento.

Como dijimos los filtros a nivel de aplicación permiten aplicar un esquema de seguridad más estricto. En estos firewalls se instala un software específico para cada aplicación a controlar (un proxy server); de hecho si no se instala los servicios relativos a la aplicación las comunicaciones no podrán ser enrutadas, punto que no se convierte en trivial pues de esta forma estamos garantizando que todas aquellas nuevas aplicaciones desconocidas no podrán acceder a nuestra red. Otro ventaja que trae el uso de este tipo de firewall es que permite el filtrado del protocolo, por ejemplo se podría configurar el proxy server que atiende el FTP para que pueda aceptar conexiones pero denegar el uso del comando put asegurando de esta forma que no nos puedan escribir ningún archivo o que impida navegar por el FS; esto es lo que hay se conoce como un FTP anónimo.

Este tipo de configuración incrementa los costos de la plataforma sobre la cual funcionará el filtro. Algunos autores reconocen otro nivel de aplicación de firewall que

es nivel de circuito, que en realidad no procesa ni filtra el protocolo, sino que simplemente establece un circuito entre origen y destino.

Conceptualmente hay dos tipos de firewalls, nivel de red y nivel de aplicación

Los firewalls de nivel de red toman sus acciones en función del origen, la dirección de destino y el port en cada paquete IP. Los modernos firewalls de este tipo se han sofisticado y mantienen información respecto del estado de las conexiones que están activas a través de él, etc. Este tipo de firewall tiende a ser muy rápidos y son transparentes al usuario.

Los firewalls de nivel de aplicación por lo general son hosts corriendo proxy servers, que no permiten el tráfico directo entre redes, manteniendo una elaborada auditoria y logeo del tráfico que pasa a través de él. Este tipo de firewall puede ser utilizado para realizar las tareas relativas al NAT, debido a que como las comunicaciones van de un lado hacia el otro se puede enmascarar la ubicación original. Este tipo tiende a proveer una auditoria más detallada y un mayor grado de seguridad que los de nivel de red.

Los routers de filtro de paquetes, que corresponden al primer grupo, realizan una decisión del tipo pasa no pasa para cada paquete que recibe. El router examina cada datagrama para determinar si se aplican sus reglas de filtrado. Las reglas de filtrado se basan en la información contenida en el header del paquete. Esta información consiste en el IP de origen, la IP de destino, el protocolo encapsulado (TCP, UDP, ICMP), el port TCP/UDP de origen y de destino, etc. Toda esta información es controlada contra las reglas de filtrado definidas, pudiendo ser enrutada si existe una regla que lo permite, descartada si una regla así lo indica y si no existe regla comparable un parámetro previamente configurado determinará si el paquete pasa o no.

Dentro de este tipo están los que filtran en función del servicio involucrado. Esto es posible pues hay muchos servicios para los cuales están normalizados los ports en los que escuchan, por lo cual se pueden definir reglas que involucren el port, definiendo la aceptación o el rechazo.

Por otro lado frente a diferentes ataques que se fueron produciendo surgieron otros firewalls cuyas reglas son independientes del servicio; estas reglas exigen un análisis más detallado que involucra el ruteo, las opciones de IP, verificación de los fragmentos de desplazamiento y puntos por el estilo.

La mayoría de los firewalls implementados sobre Internet están desarrollados sobre el concepto de filtrado de paquetes. Este tipo de firewalls no son difíciles de configurar debido a que su software contiene una serie de reglas previamente configuradas y fundamentalmente son transparentes al usuario y no exigen instalar ningún software adicional en los hosts.

Por otro lado cuando se debe customizar de manera tal de adaptarlo a aplicaciones específicas de cada empresa la tarea se puede hacer algo compleja pues exige una figura de administrador que debe conocer los servicios de Internet, los distintos encabezados de los paquetes, los distintos valores que se espera encontrar en los campos a analizar. Si se requiere un filtrado complejo, las reglas pueden volverse demasiado largas con la consecuencia de una difícil administración y seguimiento.

Los filtros a nivel de aplicación permiten aplicar un esquema de seguridad más estricto. En estos firewalls se instala un software específico para cada aplicación a controlar (un proxy server); de hecho si no se instala los servicios relativos a la aplicación las comunicaciones no podrán ser enrutadas, punto que no se convierte en trivial pues de esta forma estamos garantizando que todas aquellas nuevas aplicaciones desconocidas no podrán acceder a nuestra red. Otro ventaja que trae el uso de este tipo de firewall es que permite el filtrado del protocolo, por ejemplo se podría configurar el proxy server que atiende el FTP para que pueda aceptar conexiones pero denegar el uso del comando put asegurando de esta forma que no nos puedan escribir ningún archivo o que impida navegar por el FS; esto es lo que hay se conoce como un FTP anónimo.

Este tipo de configuración incrementa los costos de la plataforma sobre la cual funcionará el filtro.

Algunos autores reconocen otro nivel de aplicación de firewall que es nivel de circuito, que en realidad no procesa ni filtra el protocolo, sino que simplemente establece un circuito entre origen y destino.

Las Fases

- **La fase I** consiste en presentar el funcionamiento del Firewall como sistema de seguridad de una red.
- **La fase II** comprende los componentes del sistema Firewall.
- **La fase III** relaciona las características y ventajas del Firewall.
- **La fase IV** es el diseño de decisión de un Firewall de Internet y el ultimo tema Limitaciones del Firewall.

FASE I
Funcionamiento del Firewall como sistema de seguridad de una red

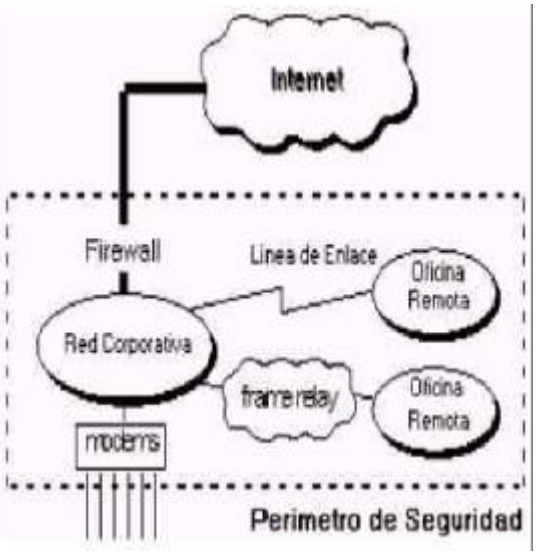


Figura 5.2.1

Un Firewall se conecta entre la red interna confiable y la red externa no confiable, (ver figura.1).Los Firewalls en Internet administran los accesos posibles del Internet a la red privada. Si no contamos con un Firewall, cada uno de los servidores de nuestro sistema se exponen al ataque de otros servidores en Internet.

El sistema Firewall opera en las capas superiores del modelo OSI y tienen información sobre las- funciones de la aplicación en la que basan sus decisiones.

Los Firewalls también operan en las capas de red y transporte en cuyo caso examinan los encabezados IP y TCP, (paquetes entrantes y salientes), y rechazan o pasan paquetes con base a reglas de filtración de paquetes programadas.

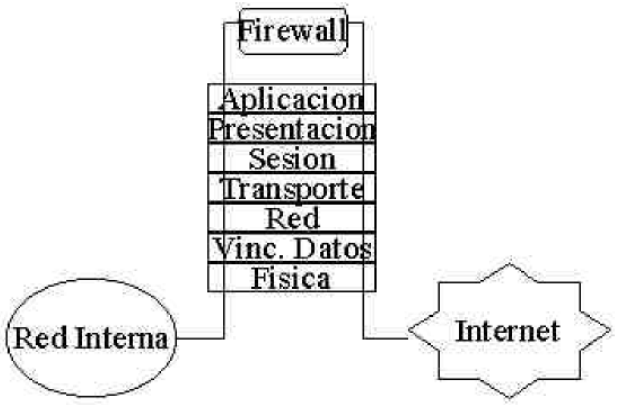


Figura 5.2.2

El firewall actúa como un punto de cierre que monitorea y rechaza el tráfico de red a nivel de aplicación, (Ver Figura. 2). Los Firewall son filtros que bloquean o permiten conexiones transmisiones de información por Internet, los filtros están diseñados para evitar que un usuario irreconocible ataque nuestro equipo por Internet y al mismo tiempo podrá ser inmune a la penetración.

FASE II
Los componentes del sistema Firewall

Un Firewall típico se compone de uno, o una combinación, de:

- Ruteador Filtra-paquetes.
- Gateway a nivel-aplicación.
- Gateway a nivel-circuito.

El ruteador toma las decisiones de rehusar y permitir el paso de cada uno de los paquetes que son recibidos. Este sistema se basa en el examen de cada datagrama enviado y cuenta con una regla de revisión de información de los encabezados IP, si estos no corresponden a las reglas, se descarta o desplaza el paquete. (Ver figura. 5.2.3).

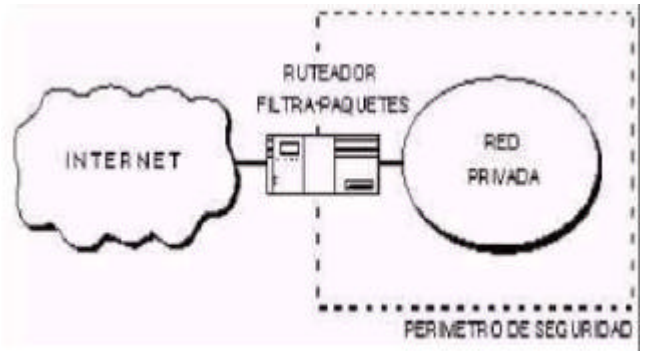


Figura 5.2.3

Gateways a nivel-aplicación

Los gateways nivel-aplicación permiten al administrador de red la implementación de una política de seguridad estricta que la que permite un ruteador filtra-paquetes.

Mucho mejor que depender de una herramienta genérica de filtra-paquetes para administrar la circulación de los servicios de Internet a través del firewall, se instala en el gateway un código de propósito-especial (un servicio Proxy) para cada aplicación deseada.

Gateway a nivel-circuito

Un Gateway a nivel-circuito es en si una función que puede ser perfeccionada en un Gateway a nivel-aplicación. A nivel-circuito simplemente trasmite las conexiones TCP sin cumplir cualquier proceso adicional en filtrado de paquetes.

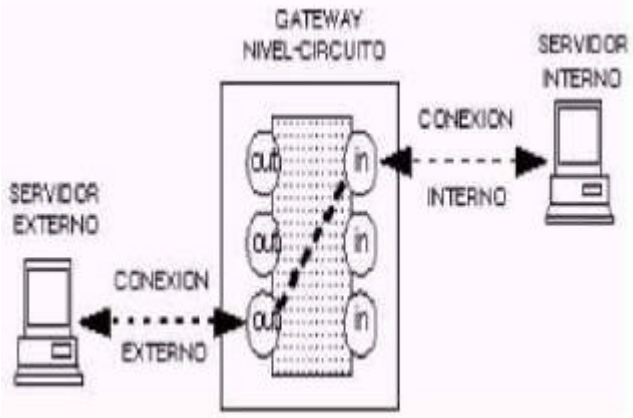


Figura 5.2.4

La figura 4.- Muestra la operación de una conexión típica Telnet a través de un gateway a nivel-circuito.

Tal como se menciono anteriormente, este gateway simplemente trasmite la conexión a través del firewall sin examinarlo adicionalmente, filtrarlo, o dirigiendo el protocolo de Telnet.

El gateway a nivel-circuito acciona como un cable copiando los bytes antes y después entre la conexión interna y la conexión externa.

FASE III

Características y ventajas del Firewall

Protección de la Red: mantiene alejados a los piratas informáticos (crakers) de su red al mismo tiempo que permite acceder a todo el personal de la oficina.

Control de acceso a los recursos de la red: al encargarse de filtrar, en primer nivel antes que lleguen los paquetes al resto de las computadoras de la red, el firewall es idóneo para implementar en el los controles de acceso.

Control de uso de Internet: permite bloquear el material no- adecuado, determinar que sitios que puede visitar el usuario de la red interna y llevar un registro.

Concentra la seguridad: el firewall facilita la labor a los responsables de seguridad, dado que su máxima preocupación de encarar los ataques externos y vigilar, mantener un monitoreo.

Control y estadísticas: permite controlar el uso de Internet en el ámbito interno y conocer los intentos de conexiones desde el exterior y detectar actividades sospechosas.
Choke-Point: permite al administrador de la red definir un (embudo) manteniendo al margen los usuarios no-autorizados fuera de la red, prohibiendo potencialmente la entrada o salida al vulnerar los servicios de la red, y proporcionar la protección para varios tipos de ataques.

Genera Alarmas de Seguridad: el administrador del firewall puede tomar el tiempo para responder una alarma y examina regularmente los registros de base.

Audita y registra Internet: permite al administrador de red justificar el gasto que implica la conexión a Internet, localizando con precisión los cuellos de botella potenciales del ancho de banda.

FASE IV

Diseño de decisión de un Firewall de Internet

Cuando se diseña un firewall de Internet, se toma algunas decisiones que pueden ser asignadas por el administrador de red:

Las políticas que propone el Firewall:
Posturas de la políticas “No todo lo específicamente permitido esta prohibido” y “Ni todo lo específicamente prohibido esta permitido”.

La primera postura asume que un firewall puede obstruir todo el tráfico y cada uno de los servicios o aplicaciones deseadas necesariamente y ser aplicadas caso por caso.

La segunda propuesta asume que el firewall puede desplazar todo el tráfico y que cada servicio potencialmente peligroso necesitara ser aislado básicamente caso por caso.

La Política interna propia de la organización para la seguridad total: la política de seguridad se basara en una conducción cuidadosa analizando la seguridad, la asesoria en caso de riesgo.

El costo Financiero del proyecto Firewall
Es el precio que puede ofrecer una organización por su seguridad, un simple paquete filtrado firewall puede tener un costo mínimo ya que la organización necesita un ruteador- conectado al Internet, y dicho paquete ya esta incluido como estándar del equipo.

FASE V

Limitaciones de un Firewall

Un firewall no puede protegerse contra aquellos ataques que se efectúen fuera de su punto de operación, en este caso:

- Conexión dial-out sin restricciones que permita entrar a nuestra red protegida, el usuario puede hacer una conexión SLIP o PPP a Internet. Este tipo de conexiones derivan de la seguridad provista por firewall construido cuidadosamente, creando una puerta de ataque.

- El firewall no puede protegerse de las amenazas a que esta sometido por traidores o usuarios inconscientes.

- El firewall no puede protegerse contra los ataques de la ingeniería social, en este caso, un craker que quiera ser un supervisor o aquel que persuade a los usuarios menos sofisticados.

- El firewall no puede protegerse de los ataques posibles a la red interna por virus informativos a través de archivos y software.

- Tampoco puede protegerse contra los ataques en la transferencia de datos, estos ocurren cuando aparentemente datos inocuos son enviados o copiados a un servidor interno y son ejecutados despachando el ataque.

5.3 Instalación y Configuración

Arquitectura

Las tecnologías de filtrado de paquetes que se emplean en los firewalls constituyen una manera eficaz y general para controlar el tráfico en la red. Tales tecnologías tienen la ventaja de no realizar ningún cambio en las aplicaciones del cliente y el servidor, pues operan en las capas IP y TCP, las cuales son independientes de los niveles de aplicación según se establece en el modelo OSI. Por otro lado, los enfoques de la filtración de paquetes no han declarado muchos requerimientos de seguridad, por la información incompleta con la que trabajan. Sólo la información de las capas de transporte y red, como las direcciones IP, los números de puerto y las banderas TCP están disponibles para las decisiones de filtración. En muchas implementaciones de los filtros de paquete, el número de reglas puede ser limitado; además, mientras mayor sea este número, habrá una

alta penalización en el desempeño, a causa del proceso adicional necesario para las reglas complementarias.
En vista de la falta de información de contexto, ciertos protocolos como el UDP y RPC no pueden filtrarse con efectividad. Además, en muchas implementaciones, faltan los mecanismos de intervención y alerta. Muchas de estas implementaciones de filtros pueden requerir un alto nivel de comprensión de los protocolos de comunicación y su comportamiento, cuando se utilizan por diferentes aplicaciones.

Los dispositivos de filtración de paquetes, casi siempre se mejoran mediante otros tipos dispositivos llamados barreras de protección. Las barreras de protección se llaman así porque operan en las capas superiores del modelo OSI y tienen información completa sobre las funciones de la aplicación en la cual basan sus decisiones. Estos constituyen la mayoría de los firewalls tal cual hoy los conocemos.

Existen varios métodos para construir una barrera de protección. Las organizaciones con talento en la programación y recursos financieros suficientes, en general prefieren usar un método personalizado de barreras de protección para proteger la red de la organización. Si se ejecuta de manera adecuada, tal vez éste sea el método más eficaz y por supuesto el más costoso.

Otras organizaciones prefieren usar los productos comerciales existentes, así como personalizarlos y configurarlos para cumplir la política de seguridad de red de esas organizaciones.

De aquí en adelante iremos describiendo las distintas arquitecturas con las cuales se puede implementar una barrera de protección para nuestra red,

De dos bases

Un firewall de dos bases no es nada más y nada menos que un firewall con dos interfaces de red, que permite asilar una red interna de una red externa no confiable. Como este anfitrión no envía ningún tráfico TCP/IP, bloquea por completo cualquier tráfico IP entre las redes no confiables interna y externa.

Muchos servicios Internet son en esencia de almacenaje y envío. Si estos servicios se ejecutan en el anfitrión, pueden configurarse para transmitir servicios de aplicación desde una red hacia la otra. Si los datos de aplicación deben cruzar la barrera, es factible configurar los agentes emisores de aplicación para hacer la ejecución en el anfitrión. Estos agentes son programas especiales, utilizados para enviar solicitudes de aplicación entre dos redes conectadas. Otro método es permitir que los usuarios se conecten al anfitrión de dos bases y después tengan accesos a los servicios externos desde la interfaz de red externa del anfitrión.

Si se usan los emisores de aplicación, el tráfico de la aplicación no puede cruzar la barrera, a menos que el emisor de aplicación se ejecute y se configure en el servidor de barrera de protección. Esta acción es la implementación de la política "si no está permitido de manera expresa, está prohibido". Si se autoriza a los usuarios conectarse en forma directa a la barrera de protección, puede comprometerse la seguridad de ésta porque la barrera es el punto central de la conexión entre la red externa y la interna. Por definición, la barrera de este tipo está en zona de riesgo. Si el usuario selecciona una contraseña débil o compromete su cuenta de usuario (al proporcionar la contraseña), la zona de riesgo quizá se extienda a la red interna y por lo tanto eliminará el objetivo de la barrera.

Si se mantienen registros adecuados de las conexiones de usuarios, es posible rastrear las conexiones no autorizadas a la barrera, en el momento que se descubra una brecha de seguridad. En cambio si se impide que los usuarios se conecten en forma directa a la barrera, cualquier intento de conexión directa se registrará como algo notorio y como una brecha potencial de seguridad.

Este tipo de firewall, con una interfaz mirando a cada red, es la configuración básica usada en las barreras de protección. Los aspectos delicados son que el enrutamiento se encuentra inhabilitado y que la única ruta entre los segmentos de red es a través de una función de capa de aplicación. Si el enrutamiento se ha configurado de manera errónea por accidente (o por diseño) para permanecer activo, se ignorarán las funciones de la capa de aplicación de las barreras de protección.

La mayoría de estas configuraciones están montadas sobre máquinas UNIX. En algunas implementaciones de este sistema operativo, las funciones de enrutamiento se activan de manera predeterminada, por lo cual es importante verificar que dichas funciones están inhabilitadas.

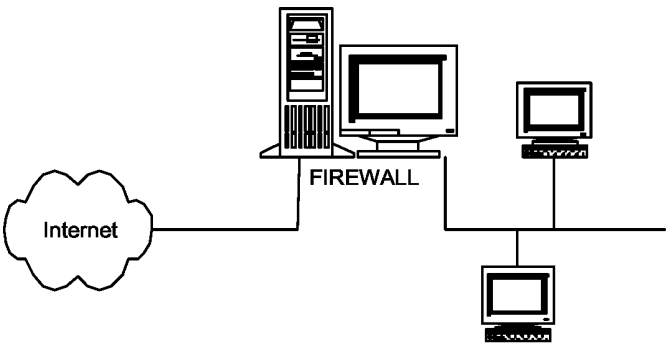


Figura 1. Como se compromete la seguridad en esta configuración

La mayor amenaza ocurre cuando el intruso obtiene el acceso directo de conexión al firewall. La conexión siempre se da mediante una aplicación apoderada del

servidor. Las conexiones desde redes externas requieren una autenticación más rigurosa.
Si el usuario obtiene acceso al servidor, la red interna puede ser inválida. Estas invasiones pueden tener cualquiera de las siguientes fuentes:

- Autorizaciones débiles en el sistema de archivos.
- Volúmenes montados en NFS en la red interna.
- Programas de respaldo de red que puedan restituir autorizaciones excesivas.
- El uso de scripts administrativos que no se hayan asegurado de manera adecuada.
- Comprensión del sistema a partir de antiguos niveles de revisión del software y notas que no se hayan asegurado de manera adecuada.
- La instalación de antiguos kernels de sistema operativo que activen el envío IP o la instalación de versiones de antiguos kernels de sistema operativo con problemas de seguridad conocidos.
- El uso de facilidades de Berkeley tal como el .rhosts o el hosts.equiv que definen equivalencia entre servidores o usuarios de distintos servidores, por lo cual se inhabilita el sistema de autenticación.
- Si el servidor firewall falla, la red interna no tendrá defensa ante futuros intrusos, a menos que el problema se detecte y corrija con rapidez.
- Como se mencionó antes, la variable ipforwarding del kernel de UNIX controla el desempeño del enrutamiento IP. Si el intruso obtiene suficientes privilegios del sistema, podrá cambiar el valor de esta variable y habilitar el envío IP, con lo cual se ignorará el mecanismo de la barrera.

Servicios

Además de inhabilitar el envío de IP, se debe eliminar todos los programas y servicios que puedan ser peligrosos en las manos de un intruso. Algunos de las precauciones a tomar son las siguientes:

- Eliminar las herramientas de programación: compiladores, enlazadores, etc.

- Eliminar los programas con autorizaciones de SUID y SGID que no se necesiten o no se comprendan. Si los programas no funcionan, siempre es factible colocar de nuevo aquellos que son esenciales.

- Utilizar particiones en el disco para que, en caso de haber una invasión para llenar todo el espacio del disco en la partición, la invasión queda confinada en dicho espacio.

- Eliminar todas las cuentas especiales y del sistema que no necesite.

- Eliminar los servicios de red que no se requieran. Utilizar el comando netstat –a para verificar que solo tiene los servicios precisos. Editar los archivos de servicio /etc/inetd.conf para eliminar las definiciones de servicio innecesarias.

Firewall como servidor bastión

Un firewall es un servidor de barrera de protección que es determinante para la seguridad en la red. Es el servidor central para la seguridad en la red de una organización y, por su función, debe estar en una buena fortaleza. Esto significa que el firewall lo monitorean con detenimiento los administradores de la red. La seguridad del sistema y del software del servidor debe revisarse con regularidad. Asimismo, es preciso observar los registros de acceso en busca de cualquier brecha potencial de seguridad y de un intento de asalto al servidor.

La configuración antes comentada es un caso especial del firewall. Como los firewalls actúan como un punto de interfaz para una red externa no confiable, casi siempre están sujetos a invasiones. La distribución más simple es aquella en la que el servidor constituye el primer y único punto de entrada para el tráfico de una red externa.

En vista de que el firewall es determinante para la seguridad de la red interna, por lo regular se coloca otra primera línea de defensa entre la red externa no confiable y la red interna. Esta línea casi siempre la proporciona un router de selección. En este esquema el firewall tiene una sola interfaz de red conectada a la red interna y el enrutador de selección tiene dos, una a Internet y la otra a la red interna enrutando todo el tráfico hacia el bastión.

Se debe configurar el router para que envíe primero hacia el firewall todo el tráfico recibido de las redes externas para la red interna. Antes de enviar el tráfico hacia este servidor, el router aplicará sus reglas de filtro en el tráfico del paquete. Sólo el tráfico de red que pase tales reglas será dirigido hacia el firewall; el resto del tráfico será rechazado. Esta arquitectura da un mayor nivel de confianza en la seguridad de la red. Un intruso necesita penetrar primero en el router de selección y, si lo logra, debe enfrentarse con el firewall.

El firewall utiliza funciones a nivel de aplicación para determinar si las solicitudes hacia y desde la red externa se aceptarán o negarán. Si la solicitud pasa el escrutinio del firewall, se enviará a la red interna para el tráfico de entrada. Para el tráfico de salida (tráfico hacia la red externa), las solicitudes se enviarán al router de selección.
Algunas organizaciones prefieren que su proveedor de acceso a Internet, IAP, proporcione las reglas de los filtros de paquetes para el tráfico en red enviado a la red de dicha organización. El filtro de paquetes aún actúa como la primera línea de defensa, pero se debe confiar al IAP el mantenimiento adecuado de las reglas del filtro de paquetes.

Otro punto a tener en cuenta es la seguridad del router de selección. Sus tablas de enrutamiento deben configurarse para enviar el tráfico externo al firewall. Dichas tablas necesitan estar protegidas contra las invasiones y los cambios no autorizados. Si la entrada a las tablas se cambia para que el tráfico no se envíe al firewall sino en forma directa a la red conectada localmente, el firewall se ignorará.

Además si el router responde a los mensajes ICMP (protocolo Internet de mensajes de control) de redirección, será vulnerable a los falsos mensajes ICMP que envíe el intruso. Por lo tanto, debe inhabilitarse la respuesta a los mensajes ICMP de redirección. Se deben eliminar los servicios de red innecesarios y utilizar el enrutamiento estático. En especial, asegurarse que los demonios “routed” y “gated” no se encuentren en ejecución; de lo contrario, las rutas serán anunciadas al mundo exterior. Por otro lado, realizar entradas permanentes en la tabla de caché ARP para señalar al firewall.

Entre los servicios a inhabilitar se encuentran: ARP, redirecciones ICMP, ARP apoderado, MOP y mensajes ICMP no alcanzables, TELNET. En una operación ARP normal, las tablas ARP de entrada se construyen de manera dinámica y expiran después de un tiempo determinado. Inicializar en forma manual la tabla cache ARP para el router y el firewall. Las entradas ARP realizadas en forma manual nunca expiran y actúan como entradas “estáticas”. Con el procesamiento ARP inhabilitado en el router, éste no proporcionará su dirección de hardware.

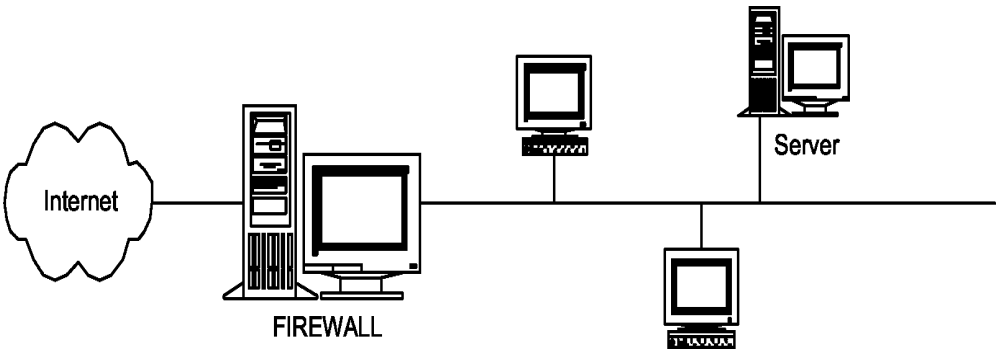


Figura 2. Firewall como servidor de bastión con dos interfaces de red

Bajo esta configuración una interfaz está conectada a la red "exterior" y la otra interfaz lo está a la red "interior". Uno de los puertos del router (el de la primera línea de defensa) está conectado a la red "interior" y el otro lo está a Internet. Nótese que hablamos de red "interior", "exterior" e Internet; aquí surge el nuevo concepto de red exterior que es la que se ubica entre el firewall y el router.

De nuevo el router debe configurarse para enviar todo el tráfico recibido de las redes externas para la red interna hacia la interfaz de red "interior" del bastión. Antes de enviar el tráfico, el router aplicará sus reglas de filtro de paquetes. Sólo el tráfico de red que pase estas reglas se dirigirá hacia el firewall; el resto habrá de rechazarse. Un intruso debe penetrar primero en el router y, si lo logra, se enfrentará al firewall.

No existen servidores en la red exterior más que el router y una de las interfaces de red del firewall. La red exterior forma una zona desmilitarizada DMZ. Ya que la DMZ sólo tiene dos conexiones de red, puede reemplazarla un enlace dedicado punto a punto. Este hecho dificultará más conectarse con este enlace mediante analizadores de protocolos. Si se utiliza una red Ethernet o token ring para la DMZ, una estación de trabajo que coloque su interfaz de red en el modo promiscuo puede capturar el tráfico de la red y tener acceso a los datos delicados.

Por lo general, las interfaces de red sólo leen el paquete que se le dirija en forma directa. En el modo promiscuo, sin embargo, dichas interfaces leen todos los paquetes que ve la interface de red. Todos los servidores de la organización (excepto el firewall) están conectados a la red interior.

Esta configuración tiene otra ventaja sobre esa configuración de red en la cual sólo se empleaba una interfase de red del firewall. Esta ventaja es que el firewall no se puede ignorar al atacar las tablas de enrutamiento de los routers. El tráfico de la red debe pasar por este firewall para llegar a la red interior.

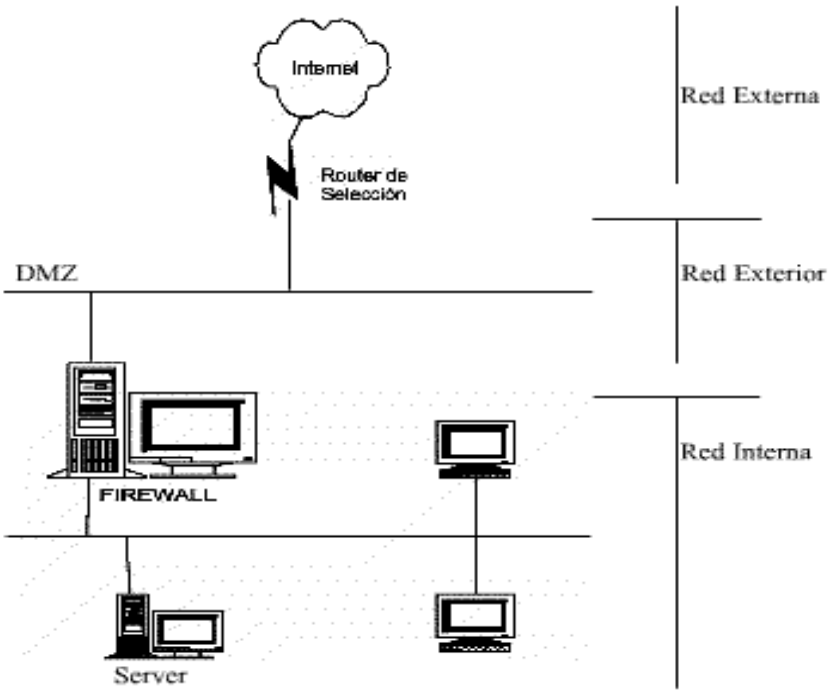


Figura 3. Dos Firewalls y dos DMZ

En este caso ambas interfaces de los firewalls se encuentran configuradas. Tres zonas de red están formadas en la red interna: la red exterior, la red privada y la red interior.

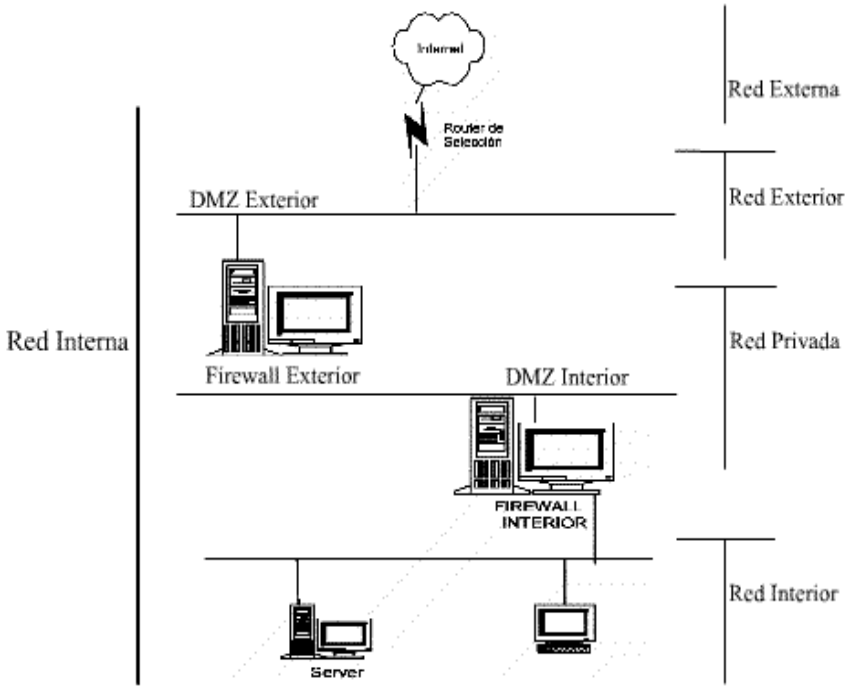
Existe una red privada entre los firewalls interior y exterior. Una organización puede colocar algunos servidores en la red privada y mantener los más delicados detrás del firewall interior. Por otra parte, una organización quizá desee una seguridad máxima y usar la red privada como una segunda zona de buffer o DMZ interior, además de mantener todos los servidores en la red interior.

Si una empresa quiere proporcionar acceso completo a una gran variedad de servicios, como FTP anónimo (protocolo de transferencia de archivos), Gopher y WWW (World Wide Web), puede proveer ciertos servidores de sacrificio en la DMZ exterior. Los firewalls no deben confiar en ningún tráfico generado desde estos servidores de sacrificio.

El router de selección debe estar configurado para enviar todo el tráfico recibido desde las redes externas para la red interna hacia el firewall interior. Antes de mandar el tráfico, el router aplicará las reglas de filtro de paquetes. Sólo el tráfico de la red que pasa esas reglas se dirigirá al firewall exterior; el resto será rechazado. Un intruso debe penetrar primero el router y, si lo hace, se enfrentará al firewall exterior.

Sin que le importe violar las defensas de la red exterior, el intruso penetra en el firewall interior. Por ello, si los recursos lo permiten, tal vez se desee dar a cada firewall la responsabilidad de un grupo administrativo diferente. Esto asegura que los errores de un grupo de administradores no los repitan los demás administradores. También se debe garantizar que los dos grupos compartan información acerca de debilidades descubiertas en los firewalls. Otro tipo de configuración de red se obtiene al utilizar dos firewalls, pero sólo una interfaz de red de cada anfitrión. Un segundo router llamado el ”ahogador” se agrega entre la DMZ y las redes interiores. Se debe asegurar que los firewalls no se ignoren y que los routers usen rutas estáticas.

A partir de estos elementos, firewalls con una o dos interfaces de red y routers, se pueden lograr diferentes configuraciones que surgen de la combinación de ellos. Cuando sólo esté en uso una interfaz de red del firewall, se deben utilizar rutas estáticas en los routers y configurar bien las entradas de las tablas de enrutamiento para asegurar que los firewalls no se ignoren.



Subredes Seleccionadas

En algunas configuraciones de barreras de protección, tanto la red externa no confiable como la red interna pueden tener acceso a una red aislada; sin embargo, ningún tráfico de red puede fluir entre ambas redes a través de la red aislada. El aislamiento de la red se lleva a cabo mediante una combinación de routers de selección configurados de manera adecuada. Dicha red se conoce como red seleccionada.

Algunas redes seleccionadas pueden tener compuertas a nivel de aplicación, que actúan como firewalls y permiten el acceso interactivo a los servicios exteriores. Los routers se utilizan para conectar a Internet con la red interna. El firewall es una compuerta de aplicación y rechaza todo el tráfico que no se acepte de manera expresa.

Como la única manera de tener acceso a la subred seleccionada es mediante el firewall, es bastante difícil que el intruso viole esta subred. Si la invasión viene por Internet, el intruso debe volver a configurar el enrutamiento en Internet, la subred seleccionada y la red interna para tener libre acceso (lo cual se logra con dificultad si los routers permiten el acceso sólo a los servidores específicos). Si alguien violara al firewall, el intruso forzaría su entrada hacia uno de los anfitriones en la red interna y después el router, para tener acceso a la subred seleccionada. Este tipo de invasión de tipo aislamiento es difícil de lograr sin desconectarse o sin activar alguna alarma.

Como las redes seleccionadas no permiten que el tráfico de red fluya entre Internet y la red interna, las direcciones IP de los anfitriones en dichas redes se ocultan unas de otras. Esto permite que una organización a la que el NIC (Centro de Información de red) aún no le haya asignado oficialmente números de red, tenga acceso a Internet mediante servicios de compuertas de aplicación, proporcionados por el firewall en la subred seleccionada. Si estos servicios usados con la compuerta de aplicación están restringidos, estas limitantes pueden estimular que la red interna asigne números de red de manera oficial.

Compuertas a nivel de aplicación

Las compuertas a nivel de aplicación pueden manejar tráfico de almacenaje y envío, así como tráfico interactivo. Dichas compuertas están programadas para comprender tráfico al nivel de aplicación del usuario (capa 7 del modelo OSI); por lo tanto, pueden proporcionar controles de acceso a niveles de usuario y de protocolos de aplicación. Además, es factible utilizarlas para mantener un registro inteligente de todos los usos de las aplicaciones. La habilidad para registrar y controlar todo el tráfico de entrada y de salida es una de las características principales de las compuertas a nivel de aplicación. Las compuertas por sí mismas pueden integrar, si es necesario, seguridad adicional.

Para cada aplicación que se transmita, las compuertas a nivel de aplicación utilizan un código de propósito especial. Gracias a este código, las compuertas de aplicación proporcionan un alto nivel de seguridad. Para cada nuevo tipo de aplicación que se agregue a la red y que requiera protección, tiene que escribirse un nuevo código de propósito especial; por lo tanto, la mayoría de este tipo de compuertas provee un subgrupo limitado de aplicaciones y servicios básicos.
Para utilizar las compuertas a nivel de aplicación, los usuarios deben conectarse a la máquina de la compuerta de aplicación o implementar un servicio específico

para la aplicación de cliente para cada servidor que empleará el servicio. Cada módulo de compuerta específica para la aplicación puede tener su propio grupo de herramientas de administración y lenguaje de comandos.

Una desventaja de muchas compuertas es que debe escribirse un programa personalizado para cada aplicación. Sin embargo, este hecho también es una ventaja en seguridad, pues no se pueden pasar las barreras de protección a menos que haya estipulado una compuerta explícita a nivel de aplicación.

El programa personalizado de aplicación actúa como un “apoderado” que acepta las llamadas entrantes y las verifica con la lista de acceso de los tipos de solicitudes permitidas. En este caso, se trata de un servidor apoderado de aplicación. Al recibir la llamada y verificar que sea permitida, el apoderado envía la solicitud al servidor correspondiente; por lo tanto, actúa como un servidor y como cliente. Trabaja como un servidor para recibir la solicitud entrante y como un cliente al enviarla. Después de establecer la sesión, el apoderado de aplicación funciona como un relevo y copia los datos existentes entre el cliente que inició la aplicación y el servidor. Dado que todos los datos ente el cliente y el servidor los intercepta el apoderado de la aplicación, éste tiene control total sobre la sesión y puede hacer un registro tan detallado como usted lo requiera.

Estado actual en el mercado

Hoy por hoy en el mercado existen infinidad de proveedores de aplicaciones de firewalls, que están complementadas con los esfuerzos por elevar el nivel de seguridad de los distintos sistemas operativos. Estas aplicaciones de firewalls están disponibles para los distintos sistemas operativos y si bien en líneas generales todos cumplen básicamente las mismas funciones, cada proveedor tiene sus características particulares que se van igualando de uno a otro con el correr del tiempo y el desarrollo de los distintos releases.

Entre los productos más utilizados se encuentran Firewall-1, Pix y Raptor, de las empresas Chekpoint, Cisco y HP respectivamente.

Indagando un poco en el mercado local se puede comprobar que el producto de Cisco esta siendo muy utilizado, debido a su integración “nativo” en todas aquellas organizaciones que están utilizando routers Cisco.

Además de verificar las características particulares de cada uno se pueden bajar de Internet versiones de evaluación para corroborar las distintas facilidades como así también observar como se adapta a la organización donde se implementará.

Ejercicio 1

Firewalls del XX de XX

Instrucciones

1.- Realiza la búsqueda en Internet de 2 tipos de software de firewall para computadora personal, instala 1 de ellos en una computadora y observa que características de protección tiene, así como los resultados obtenidos. Día 1 y 2

2.- Participa en el grupo de discusión con una aportación que contenga las características y url de descarga del Firewall que seleccionaste, así como los resultados que obtuviste, menciona las ventajas que presenta. Día 3 y 4

3.- Consulta la aportación de otro de tus compañeros y visita el sitio que sugiere, compáralo con el que utilizaste y contesta su propuesta.

Páginas de Referencia Electrónicas

• Elorreaga, Daniel. ***Firewalls y Seguridad en Internet***. UNAM.
Disponible en http://www.monografías.com
Consultado 30 de septiembre de 2004.

• Wanadoo. ***Arquitectura de Firewalls***. Disponible en
http://html.rincondelvago.com/arquitectura-de-firewalls.html
Consultado 30 de septiembre de 2004.

• Wanadoo**. Firewalls**. Disponible en
http://html.rincondelvago.com/firewalls.html
Consultado 30 de septiembre de 2004.

6. Seguridad en Internet

6.1 Footprinting

Uno de los primeros pasos que realiza un atacante es el footprinting, es decir, el hecho de la recopilación de información sobre la máquina o red que será la víctima. Por ejemplo, cuando los ladrones deciden robar un banco, no andan caminan adentro y no comienzan a exigir el dinero (al menos no los elegantes). En cambio, ellos toman grandes cantidades de información recopilada acerca del banco, las rutas y fechas de expedición de los vehículos blindados del banco, las cámaras de video, y los números de las cajas, las salidas para escapar, y cualquier cosa que ayudara en una desventura acertada.

Los mismos requisitos se aplican a los atacantes acertados. Ellos deben reunir información en abundancia para ejecutar un ataque enfocado y quirúrgico (uno que no será detectado fácilmente). Como resultado, los atacantes recopilarán tanta información como les sea posible sobre todos los aspectos de la postura de la seguridad de una organización. Los atacantes terminan con un único footprint o un perfil único de su, acceso remoto de Internet, y la presencia de Intranet/Extranet. Siguiendo una metodología estructurada los atacantes pueden dividir sistemáticamente la información de la multitud de fuentes para compilar el crítico footprint sobre cualquier organización.

¿QUÉ ES FOOTPRINTING?

El sistemático footprinting de una organización habilita a los atacantes a crear un perfil completo de la postura de seguridad de la organización. Pero usando una combinación de herramientas y técnicas, los atacantes pueden tomar una cantidad desconocida (Widget Company's Internet connection) y reducirlo a un rango especifico de nombres de dominios, bloques de redes, y direcciones IP individuales del sistema directamente conectadas a Internet. Mientras hay muchos tipos de técnicas de footprinting, ellos principalmente se dirigen al descubrimiento de la información relacionada con los ambientes siguientes: Internet, Intranet, acceso remoto, y extranet. Tabla 1-1 representa estos ambientes y la información crítica que un atacante tratará de identificar.

¿POR QUÉ ES FOOTPRINTING NECESARIO?

Footprinting es necesario para asegurar sistemáticamente y metódicamente que todas las piezas de información relacionadas con las tecnologías ya mencionadas sean identificadas. Sin una metodología buena para realizar este tipo de reconocimiento, probablemente se omiten piezas claves de información relacionada con una tecnología específica de la organización. Footprinting es a menudo la tarea más ardua de intentar determinar la postura de la seguridad de

una entidad; sin embargo, es uno de los más importantes. Footprinting se debe realizar correctamente y en una manera controlada.

Tecnología	**Identificador**
Internet	Nombre del Dominio
	Bloques de Red
	Direcciones IP especificas del sistema accesibles vía Internet
	Funcionamiento de los servicios TCP y UDP en cada sistema identificado
	Arquitectura del sistema (por ejemplo, SPARC vs. X86)
	Mecanismos del control de acceso y listas relacionadas del control de acceso (ACLs)
	Sistemas de detección de intrusos (IDSes)
	Enumeración del sistema (usuarios y nombres de grupos, banderas del sistema, tablas de enrutamiento, información del SNMP)
Intranet	Protocolos de red en uso (por ejemplo, IP, IPX, DecNET, y etc.)
	Nombres Internos del Dominio
	Bloques de Red
	Direcciones IP especificas del sistema accesibles vía Internet
	Funcionamiento de los servicios TCP y UDP en cada sistema identificado
	Arquitectura del sistema (por ejemplo, SPARC vs. X86)
	Mecanismos del control de acceso y listas relacionadas del control de acceso (ACLs)
	Sistemas de detección de intrusos (IDSes)
	Enumeración del sistema (usuarios y nombres de grupos, banderas del sistema, tablas de enrutamiento, información del SNMP)
Acceso Remoto	Números telefónicos de análogo/digital
	Tipo del sistema remoto
	Mecanismos de autentificación
	VPNs y protocolos relacionados (IPSEC,PPTP)
Extranet	Origen y destino de la conexión
	Tipo de conexión
	Mecanismos de control de acceso

Tabla 1-1 Ambientes e información crítica que los atacantes pueden identificar.

Es difícil proporcionar una guía paso a paso de footprinting porque es una actividad que puede conducirle a tomar varios caminos. Uno de los principales pasos del footprinting es hacer el llamado fingerprinting o detección del sistema operativo que se describe a continuación.

Un stack fingerpinting es una técnica extremadamente poderosa que permite comprobar o detectar el sistema operativo de un host con una probabilidad muy alta de acertar. Esto se logra mediante el stack o pila de protocolos TCP/IP. Cada desarrollador de software que escribe su propia pila del protocolo TCP/IP le agrega cosas particulares, de esta manera probando esas particularidades o diferencias es posible intentar adivinar el sistema operativo que se está manejando. Para una mayor confiabilidad, el snack fingerprinting, requiere de un puerto abierto, aunque existe el programa NMAP que no requiere de un puerto abierto para adivinar el sistema operativo (una descripción completa de como funciona se puede encontrar en http://www.insecure.org/nmap/nmapfingerprinting-article.html).

Los tipos de pruebas que pueden ser lanzados para ayudar a distinguir un sistema operativo de otro son:

- FIN probe. Un paquete FIN es enviado a u puerto abierto. El RFC 793 indica que lo correcto es no responder, aun así implementaciones (como la de Windows NT) responden con una respuesta FIN/ACK.

- Prueba de la bandera falsa. Una bandera indefinida TCP es fijada en el encabezado TCP de un paquete SYC. Algunos sistemas operativos, como Linux, responden con la bandera en el paquete de respuesta.

- Muestreo de número de secuencia inicial (ISN). La idea es poner una "bandera" TCP (64 o 128) en el encabezado TCP de un paquete SYN. El sistema operativo Linux antes de 2.0.35 mantiene la bandera puesta en su respuesta. Por el momento no hay otro sistema operativo que tenga este defecto. Sin embargo, algunos sistemas operativos parecen cancelar la conexión cuando reciben un paquete SYN+MANIACA. Este comportamiento podría ser útil para identificarlos.

- Probar TCP ISN -- La idea aquí es encontrar patrones en los números de secuencia inicial seleccionados por las implantaciones de TCP al responder a solicitudes de conexión. Esto puede ser categorizado en muchos grupos como el tradicional 64K (muchos sistemas UNIX viejas), incrementos aleatorios (nuevas versiones de Solaris, IRIX, FreeBSD, Digital UNIX, Cray, y muchas otras), "Aleatoriedad" verdadera (Linux 2.0.*, OpenVMS, AIX más recientes, etc). Sistemas Windows (y unas cuantas más) usan un modelo "dependiente del tiempo" donde el ISN se incrementa por una pequeña cantidad estática cada periodo de tiempo. Sin necesidad de decirlo, esto se vence tan fácilmente como el viejo comportamiento de 64K. Las maquinas SIEMPRE usan exactamente el mismo ISN. :). En algunos hubs 3com (usan 0x803) y en impresoras Apple LaserWriter (usan 0xC7001). También se puede subdividir en grupos tales como incremento aleatorio por varianzas computacionales, máximos comunes divisores, y otras funciones en el conjunto números de secuencia y las diferencias entre los números. Nmap es un programa que usa esto para identificación de SO.

- Bit de No-fragmentación -- Muchos sistemas operativos están empezando a poner el bit "no-fragmentación" de IP en algunos paquetes que manda. Esto da varios beneficios de desempeño (aunque también puede ser engañoso -- por esto las exploraciones de fragmentación de nmap no funcionan desde cajas Solaris). En cualquier caso, no todos los SO's hacen esto y algunos lo hacen en diferentes casos, así que poniendo atención a este bit es posible obtener aún más información sobre el SO objetivo.

- Ventana Inicial TCP -- Esto solo requiere revisar el tamaño de la ventana de los paquetes regresados. Escáneres más viejos simplemente utilizan en una ventana no-zero un paquete RST para querer decir "derivado de BSD 4.4". Escáner más nuevos como queso y nmap dan seguimiento a la ventana exacta, ya que es en realidad bastante constante según el tipo de SO. De hecho esta prueba nos da mucha información, ya que algunos sistemas operativos pueden ser identificados de manera única por la ventana en sí (por ejemplo, AIX es el único SO que he visto usa 0x3F25). En su pila TCP "completamente re-escrita" para NT5, Microsoft usa 0x402E. Interesantemente, ese es exactamente el número usado por OpenBSD y FreeBSD.

- Valor ACK -- A pesar de que podría pensarse que esto seria completamente estándar, en algunos casos las implantaciones difieren en el valor que usan para el campo ACK. Por ejemplo, digamos que mandas un FIN|PSH|URG a un puerto TCP cerrado. La mayor parte de las implementaciones pondrán el ACK igual que tu número de secuencia inicial, pero Windows y algunas impresoras mandan la secuencia + 1. Si se manda un SYN|FIN|URG|PSH a un puerto abierto, Windows es muy inconsistente. Algunas veces regresa tu secuencia, otras veces manda s++, aun otras veces regresa un valor aparentemente aleatorio. Uno tiene que preguntarse que tipo de código este escribiendo MS que cambia su parecer de esta manera.

- Control de Error de Mensaje ICMP -- Algunos sistemas operativos siguen la sugerencia RFC 1812 de limitar la cantidad en que diferentes mensajes de error son mandados. Por Ejemplo, el kernel de Linux (en net/ipv4/icmp.h) limita la generación de mensajes de destino inalcanzable a 80 en 4 segundos, con 1/4 de segundo de castigo si se sobrepasa ese limite. Una manera de probar esto es mandar una bola de paquetes a algún puerto UDP alto seleccionado al azar y contar el numero de inalcanzables recibidos.

- Referenciación de Mensaje ICMP -- Los RFCs especifican que los mensajes de error ICMP hacen referencia a una pequeña cantidad de un mensaje ICMP que provoca errores variados. Para un mensaje de puerto inalcanzable, casi todas las implementaciones mandan solo el encabezado IP requerido + 8 bytes de regreso. Sin embargo, Solaris regresa un poco más y Linux regresa aun más que eso. Lo

interesante de esto es que permite a nmap identificar a maquinas Linux y Solaris aun cuando no tengan ningún puerto escuchando.

- Eco de Integridad de Mensajes de Error ICMP. Las maquinas tienen que regresar parte del mensaje original junto con un error de puerto inalcanzable. Y aun otras maquinas tienden a usar los encabezados enviados como 'espacio para escribir' durante su procesamiento inicial, así que están un tanto alterados cuando son regresados. Por ejemplo, AIX y BSDI regresan un campo 'longitud total' de IP que es 20 bytes demasiado grande. Algunos BSDI, FreeBSD, OpenBSD, ULTRIX, y VAX modifican el ID del IP que se les manda. Mientras la suma de control va a cambiar debido que se cambia el TTL de todas formas, hay algunas maquinas (AIX, FreeBSD, etc.) que regresan una suma de control inconsistente o de 0. Lo mismo sucede con la suma de control UDP. Al final, nmap hace nueve diferentes pruebas de errores ICMP para detectar diferencias sutiles como estas.

- Tipo de Servicio -- Para los mensajes de inalcanzable del puerto ICMP se analiza el valor del tipo de servicio (TOS) en el paquete que regresa. Casi todas las implantaciones usan 0 para este error ICMP, aunque Linux usa 0xC0. Esto no indica uno de los valores TOS estándar, pero en vez de eso, es parte del campo de precedencia (AFAIK) que no es usado.

- Manejo de fragmentación -- Esta es una técnica favorita de Thomas H. Ptacek de Secure Networks, Inc. Esto se aprovecha del hecho de que diferentes implantaciones seguido manejan fragmentos IP contranslape de manera diferente. Algunos sobrescribirán las viejas porciones con las nuevas, y en otros casos las cosas viejas tienen precedencia. Hay muchas pruebas diferentes que se pueden usar para determinar como el paquete fue reensamblado. Para mas información sobre fragmentos traslapados, pueden leer este articulo IDS (www.secnet.com).

6.2 Negación de Servicio (DOS)

Negación de Servicio (Denial of Service)

Las negaciones de servicio (conocidas como DoS, *Denial of Service*) son ataques dirigidos contra un recurso informático (generalmente una máquina o una red, pero también podría tratarse de una simple impresora o una terminal) con el objetivo de degradar total o parcialmente los servicios prestados por ese recurso a sus usuarios legítimos. Los ataques de negación de servicio son en Internet comunes por la facilidad con que pueden hacerse. Lo único difícil o complicado es conseguir un servidor desde donde lanzar el ataque.

Un ataque de negación de servicio es una acción que priva al sistema de los recursos requeridos. También una negación de servicio es considerado una rutina de troyano que interrumpe o inhibe el flujo normal de datos hacia adentro o afuera

de un sistema. Algunos ataques DoS intentan aprovecharse de bugs en la red para quebrar el sistema con un solo paquete. Las negaciones de servicio más habituales suelen consistir en la inhabilitación total de un determinado servicio o de un sistema completo, bien porque ha sido realmente bloqueado por el atacante o bien porque está tan degradado que es incapaz de ofrecer un servicio a sus usuarios. Otra forma de ataque DoS ocurre cuando se tiene acceso a un servicio Web de manera masiva y repetida desde diferentes lugares, impidiendo a otros sistemas tener acceso al servicio y obtener información de él. Los ataques con paquetes enmascarados, por ejemplo, son casi imposibles de detener, el cual puede desconectar el sistema de Internet. Pueden no dañar el sistema, pero saturan la conexión a Internet.

Haciendo uso de este tipo de ataque (DoS) la persona que lo realiza, puede obtener acceso a un sistema, ya que el ataque dispara otro tipos de 'agujeros' en un sistema. Algunos ejemplos pueden ser las versiones antiguas de X-lock que permitían el acceso al sistema una vez este ha sido colapsado.

En la mayor parte de los casos la negación de servicio tiene éxito porque el objetivo utiliza versiones no actualizadas de demonios (si se detiene un servicio concreto) o del propio núcleo del sistema operativo, si se detiene o degrada por completo la máquina. Para evitar esto, la norma a seguir es evidente: mantener siempre actualizados nuestros sistemas, tanto en lo referente al nivel de parches de seguridad o versiones del núcleo, como en lo referente a programas críticos encargados de ofrecer un determinado servicio.

Negación de Servicio Distribuido

La negación de servicio distribuido (*Distributed Denial of Service*, DDoS): es un ataque en el que un pirata compromete en primer lugar un determinado número de máquinas y, en un determinado momento, hace que todas ellas ataquen masiva y simultáneamente al objetivo u objetivos reales enviándoles diferentes tipos de paquetes; por muy grandes que sean los recursos de la víctima, el gran número de tramas que reciben hará que tarde o temprano dichos recursos sean incapaces de ofrecer un servicio, con lo que el ataque habrá sido exitoso. Si en lugar de cientos o miles de equipos atacando a la vez lo hiciera uno sólo las posibilidades de éxito serían casi inexistentes, pero es justamente el elevado número de pequeños atacantes lo que hace muy difícil evitar este tipo de negaciones de servicio.

Según el CERT Cordination Center (Centro mayor de reporte de problemas de seguridad en Internet), los ataques de negación de servicio distribuidos más habituales consisten en el envío de un gran número de paquetes a un determinado objetivo por parte de múltiples *hosts*, lo que se conoce como *packet flooding* (en función del tipo de paquetes utilizados se habla de *ping flood*, de SYN *flood, etc*.). Defenderse de este tipo de ataques es difícil: en primer lugar, uno piensa en bloquear de alguna forma (firewall o router) todo el tráfico proveniente de los

atacantes; pero en un ataque distribuido no es posible ya que tenemos miles de computadoras atacando desde un gran número de redes diferentes. Como podemos notar, la defensa ante una negación de servicio distribuida no es inmediata; en cualquier caso, podemos tomar ciertas medidas preventivas que nos ayudarán a limitar el alcance de uno de estos ataques. De entrada, un correcto filtrado del tráfico dirigido a nuestras máquinas es vital para garantizar nuestra seguridad: no hay que responder a *pings* externos a nuestra red, es necesario activar el *antispoofing* en nuestros cortafuegos y en los dispositivos de red que lo permitan, etc. Establecer correctamente límites a la utilización de nuestros recursos, como ya hemos visto, es también una importante medida preventiva; es posible limitar el ancho de banda dedicado a una determinada aplicación o a un protocolo, de forma que las utilizaciones por encima del margen son negadas.

A pesar de las dificultades con las que nos podemos encontrar a la hora de prevenir ataques de negación de servicio, una serie de medidas sencillas pueden ayudarnos de forma relativa en esa tarea; las negaciones de servicio son por desgracia cada día más frecuentes, y ninguna organización está a salvo de las mismas. Especialmente en los ataques distribuidos, la seguridad de cualquier usuario conectado a Internet (aunque sea con una sencilla PC y un módem) es una parte importante en la seguridad global de la red, ya que esos usuarios se convierten muchas veces sin saberlo en cómplices que colaboran en un ataque masivo contra organizaciones de cualquier tipo.

Objetivos Básicos de un Ataque de Negación de Servicio

Espacio swap

La mayoría de los sistemas tienen cientos de Megabytes reservados al espacio swap para resolver las peticiones del cliente. El espacio swap es típicamente usado para almacenar procesos menores que tienen un periodo de vida corto. Así pues el espacio swap bajo ninguna causa deberá ser usado rigurosamente. Una negación de servicio podría estar basada en un método que trate de llenar el espacio swap.

Tablas del Kernel

Es trivial sobrecargar las tablas del kernel para ocasionar serios problemas al sistema. Los sistemas que escriben a través de caches y buffers pequeños son especialmente sensitivos.

Otro objetivo susceptible es el kernel memory allocation. Todo Kernel tiene un límite para la memoria en su mapa de memoria, si el sistema alcanza este limite el kernel no puede colocar más memoria y el sistema debe ser reiniciado. La memoria del kernel no solo es usada para procesos del CPU y monitores sino

también para procesos ordinarios lo que significa que cualquier sistema puede ser colapsado con un buen algoritmo lo bastante rápido.

RAM

Una negación del servicio que ocupe grandes cantidades de memoria RAM puede ocasionar bastantes problemas. NFS y servidores de correo pueden ser usados para realizar un ataque porque casi nunca ocupan mucha memoria RAM por lo tanto tienen poca memoria RAM asignada.

Discos

Un ataque común es llenar el disco duro.

Tipos de Ataques de Negación de Servicio

Existe una clasificación de los diferentes ataques de negación de servicio, tomando en cuenta el recurso que atacan y los posibles daños que pueden causar, entre los más comunes podemos encontrar los siguientes:

JAMMING o FLOODING

Son ataques que saturan los recursos del sistema de la víctima dejándola sin memoria, sin espacio libre en su disco duro o saturando sus recursos de red.
Por ejemplo, el atacante satura el sistema con peticiones de conexión. Sin embargo, en vez de enviar la IP real del emisor, envía una falsa. Al no encontrar el sistema una respuesta desde esa IP falsa, mantiene ese buffer abierto esperando información pero bloqueando la comunicación con la IP verdadera. Los ataques más frecuentes a proveedores son usando el ping de la muerte (que bloquea el equipo) o enviando miles de correos electrónicos los usuarios de ese servidor, de forma continuada, saturando los sistemas. Relacionados con el tema están los rabbits (conejos), que son programas que provocan procesos inútiles y se reproducen como conejos hasta que satura la capacidad del sistema, provocando su colapso.

SYN FLOOD (Inundación con Paquetes SYN)

En el escaneo TCP SYN el protocolo TCP se inicia con una conexión en tres pasos (Se envía un paquete SYN y se espera por la respuesta. Al recibir un SYN/ACK y se inicia la conexión). Si el paso final no llega a establecerse, la conexión permanece en un estado denominado "semiabierto". El Syn Flood es el

más famoso de los ataques tipo Denial of Service (DoS). Se basa en un "saludo" incompleto entre los dos sistemas.

El Cliente envía un paquete SYN pero no responde al paquete ACK del 2º paso del saludo ocasionando que el servidor permanezca a la escucha un determinado tiempo hasta cancelar la llamada. Si se envían muchos saludos incompletos, se consigue que el servidor se detenga.

Para operar con este sistema hay que mantener el Sys Flood activo, ya que la mayoría de los sistemas tienen un límite de espera muy corto para conexiones semiabiertas. Cuando se ha esperado mucho tiempo, se libera un hueco para aceptar otras posibles peticiones, lo cual puede ser aprovechado para “colar” un paquete SYN destructivo o malicioso. Así que, este ataque se puede utilizar tanto para consumir los recursos de un sistema como para abrir el camino a otro tipo de ataque.

Este ataque suele combinarse también con el IP Spoofing (suplantación de una IP), para ocultar el origen del ataque.

CONNECTION FLOOD (Inundación de la Conexión)

Se basa en que la mayoría de los proveedores de Internet (ISP) tienen un tope máximo de conexiones simultaneas, que tras ser alcanzado no acepta más conexiones. Si por ejemplo un servidor Web tiene un tope de 1000 conexiones, y el atacante establece mil conexiones y no realiza ninguna petición sobre ellas, monopolizará la capacidad del servidor. Las conexiones van caducando por inactividad poco a poco, pero el atacante sólo necesita establecer nuevas conexiones para mantener fuera de servicio el servidor.

NET FLOOD (Inundación de la Red)

En este ataque se envían tantas solicitudes de conexión que las conexiones de los demás usuarios no pueden llevarse a cabo.

Es un ataque muy dañino y con poca defensa por parte de la red atacada. Para solucionarlo el Proveedor tiene que detectar el origen del ataque, bloquear la comunicación desde esa dirección y avisar al administrador de la misma para que actúe, ya que lo normal es que el administrador de esa dirección no sepa nada y que esté siendo utilizada su red para llevar a cabo el ataque por medio de algún spoofing (sustitución). De cualquier forma, saber el origen real del ataque es prácticamente imposible.

LAND ATTACK

Este ataque consiste en un Bug (error) en la implementación de la pila TCP/IP en sistemas Windows.

El ataque envía a algún puerto abierto de un servidor (generalmente al 113 o al 139) un paquete, maliciosamente construido con la IP y puerto origen igual que la IP y puerto destino. Al final la máquina termina por colapsarse.

Existen ciertas variantes a este método, por ejemplo, en enviar el mensaje a una dirección específica sin especificar el puerto Smurf o Broadcast Storm. Este ataque es bastante simple y a su vez devastador. Consiste en recolectar una serie de direcciones para a continuación mandar una petición ICMP (simulando un Ping) a cada una de ellas en serie, varias veces, falsificando la dirección IP de origen. Este paquete maliciosamente manipulado, será repetido en Broadcast, y cientos ó miles de hosts (según la lista de direcciones de Broadcast disponible) mandarán una respuesta a la víctima cuya dirección IP figura en el paquete ICMP.

MAIL BOMBING-MAIL SPAMMING-JUNK MAIL

El Mail Bombing consiste en un envío indiscriminado y masivo de un correo idéntico a una misma dirección, saturando así buzón de correo (mailbox) del destinatario.

El Mail Spamming en cambio es un bombardeo publicitario que consiste en enviar un correo a miles de usuarios, hayan estos solicitado el mensaje o no. Es muy utilizado por las empresas para publicitar sus productos. El Spamming esta siendo actualmente tratado por las leyes como una violación de los derechos de privacidad del usuario.

El junk mail o correo basura es una propaganda indiscriminada y masiva a través de correos. El mail spamming y el junk mail no pueden ser considerados como ataques DoS auténticos, porque por mucho que molesten no están encaminados a saturar ningún sistema.

6.3 Spoofing

El Spoofing es básicamente el aprovechamiento de una relación de confianza entre dos "hosts de confianza". El Spoofing se produce cuando los atacantes autentifican una máquina con otra mediante falsificación de paquetes de un host en el que se confía. Existen varias técnicas de spoofing, entre las cuales podemos encontrar las siguientes:

- Spoofing de IP
- Spoofing de ARP
- Spoofing de DNS
- Spoofing de Web

Spoofing de IP

El protocolo de Internet, IP, es uno de los más usados para la interconexión de redes tanto en ambientes académicos como corporativos, y naturalmente lo es también en la Internet. Su flexibilidad y sus poderosas capacidades lo han impuesto como una herramienta de interconectividad muy importante. La fuerza de IP radica en su facilidad y su flexibilidad para el envío de grandes volúmenes de información en pequeños datagramas a través de los diversos esquemas de enrutamiento. Sin embargo, IP presenta ciertas debilidades. La forma en que el protocolo enruta los paquetes hace que las grandes redes IP sean vulnerables a ciertos riesgos de seguridad. Uno de ellos es el Spoofing de IP, en el cual un intruso ataca cambiando la dirección del paquete IP, haciéndolo aparecer como si éste se hubiese originado en otro lugar.

El protocolo de Internet IP mantiene dos funciones. Define un paquete que puede ser enviado a través de Internet, y proporciona unos medios para fragmentar datagramas en paquetes y reensamblar paquetes en el datagrama original.

En la figura 1 podemos observar en la 4 línea del datagrama IP que contiene la dirección de origen del datagrama. En la forma más simple de una falsificación de dirección IP, el atacante sólo necesita crear un paquete que contiene una dirección de origen falsa y enviarlo a Internet escribiéndolo en el dispositivo utilizado para la comunicación con Internet. La infraestructura de Internet consiste principalmente en un conjunto de computadoras de entrada (*gateways*) y ruteadores de paquetes (*routers*). Estos sistemas tienen múltiples interfaces hardware. Mantienen tablas de enrutamiento para decidir por qué interfaz de salida enviar un paquete basándose en la interfaz de entrada por el que llegó y la dirección IP destino que en él se especifica. Cuando un paquete falsificado llega a un elemento de la infraestructura, éste dirigirá el paquete tal cual hacia la dirección destino, exactamente como lo habría hecho con un paquete legítimo.

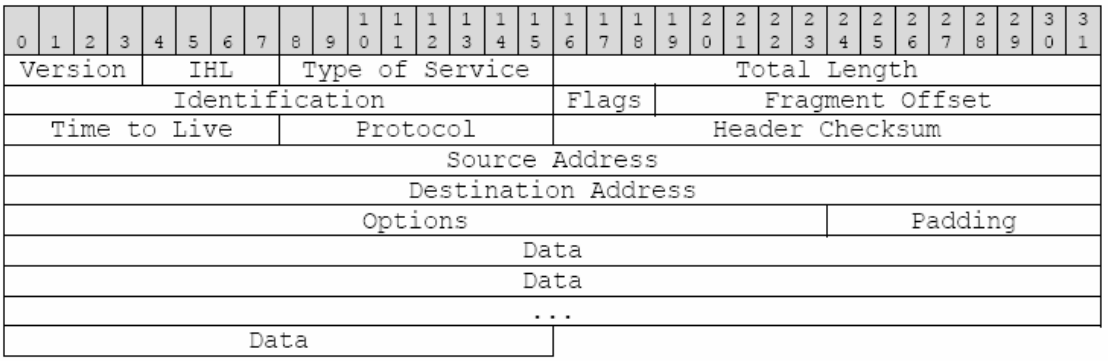


Figura 1 - Datagrama IP

Servicios vulnerables al Spoofing de IP

El Spoofing de IP afecta solamente a ciertas máquinas que ejecutan servicios específicos. Entre las configuraciones y los servicios que se conoces que son vulnerables se incluyen:

- RPC (Servicio de llamada a procedimientos remotos).
- Todos los servicios que utilizan la autenticación de direcciones IP.
- El sistema X Windows.
- Los servicios R.

El efecto que causa el ataque a dichos servicios afectan la información en los sistemas, tales como:

- **Corrupción de información**: Las direcciones de IP se usan a menudo en Internet como base para la toma de decisiones. Por ejemplo, para las actualizaciones de DNS se designa a menudo que sólo pueden venir de unos servidores específicos. Con direcciones IP falsificadas el sistema DNS entero podría adulterarse, causando que todos los paquetes serían redirigidos a través de los servidores ilegales.

- **Negación de servicios**: Internet es básicamente una red frágil que depende de la conducta apropiada y buena fe de los participantes para su apropiado funcionamiento. Sin grandes cambios en la forma en que Internet trabaja, la denegación de servicios es casi imposible prevenir. Por ejemplo, el mismo ataque de DNS podría usarse para causar el rechazo extendido de servicios, o quizá para modificar los mecanismos de entrega de paquetes de forma que la información viaje en círculos a través de la columna vertebral de Internet.

Los controles de acceso a la red que utilizan host son las partes importantes de la seguridad en Internet, aunque son distintos en cada una de las aplicaciones.

Algunos se han diseñado para proteger a un solo servidor, mientras que otros, como los empaquetadores de TCP, protegen varios servicios simultáneamente. Un pequeño número de estas herramientas, como los firewalls, tienen un alcance mayor y protegen redes enteras.

Escenario de Spoofing de IP

Supongamos que nosotros trabajamos diariamente en un servidor llamado FEI y también lo hacemos diariamente en otro servidor llamado ALUMNOS, sería muy molesto tener que estar escribiendo en FEI para ingresar el usuario y el password, y luego desconectarnos para abrir una sesión a **ALUMNOS** y volver a poner el usuario y el password, y si tenemos que hacer distintas tareas secuencialmente en cada uno de los servidores es mucho más tedioso estar realizando dichas tareas. Para evitar esto podemos hacer uso de host de confianza, es decir que si somos un usuario del servidor **FEI** y también lo soy del servidor **ALUMNOS** y no quiero estar escribiendo el usuario y el password cada vez en cada uno de estos servidores, se debe establecer una relación de confianza entre ambos hosts, y de esta manera en **FEI** puedo pasar directamente a **ALUMNOS** usando un comando y sin necesidad de volver a identificarse. En sistemas Unix el comando que se utiliza para pasar de un servidor a otro es rlogin.

Con esta relación de confianza, se podría sacar provecho de la situación, por ejemplo, si queremos entrar al servidor **ALUMNOS** aprovechando la relación de confianza que tiene con **FEI**, lo que hacemos es ir al servidor **ALUMNOS** y decir que somos un usuario de **FEI** y como existe una relación de confianza entre esos servidores debe dejarnos pasar, y como el servidor **ALUMNOS** verifica que venimos del servidor **FEI** nos deja tener acceso.

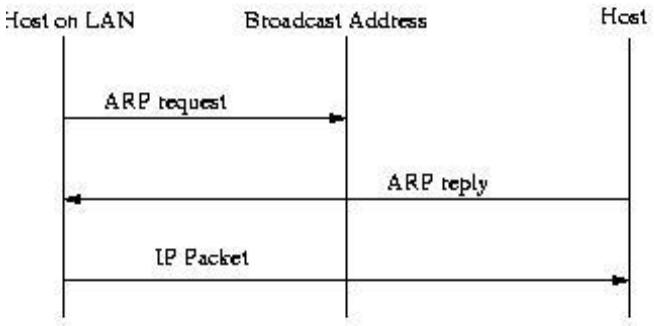
Spoofing de ARP

La segmentación de redes mediante el uso de Switches parece que es la solución perfecta para evitar ser espiados en la red por los programas llamados sniffers. Pero no es así y es posible aprovechar una inseguridad en el protocolo ARP para espiar en la red.

ARP significa Protocolo de Resolución de Direcciones. ARP resuelve la IP a una dirección física (dirección MAC). Una dirección MAC es un número compuesto por 12 dígitos hexadecimales que identifican de forma única a cada dispositivo Ethernet. La dirección MAC se compone de 48 bits. Los 24 primeros bits identifican al fabricante del hardware, y los 24 bits restantes corresponden al número de serie asignado por el fabricante, lo que garantiza que dos tarjetas no

puedan tener la misma dirección MAC ya que si esto sucede causaría problemas en la red. La Ethernet fue concebida en torno a una idea principal: todas las máquinas de una misma red local comparten el mismo medio (el cable o canal en el caso de redes inalámbricas). Todas las máquinas son capaces de ver todo el tráfico originado en la red. Debido a esto, las tarjetas Ethernet incorporan un filtro que ignora todo el tráfico que no está destinado a él. Esto se consigue ignorando aquellos paquetes cuya dirección MAC (Media Access Control) no coincide con la suya. Un sniffer elimina este filtro de la tarjeta de red y la coloca en modo promiscuo, es decir, escucha y toma todo lo que pasa por el medio. Solo es cuestión de colocar los filtros adecuados y comenzar a capturar los paquetes que más nos interesen (login/passwd de conexiones de telnet, POP3, http, etc.).El empleo de switches soluciona este problema. Mediante la segmentación de la red el único tráfico que seremos capaces de ver será el nuestro, ya que el Switch se encarga de enrutar hacia nuestro segmento solo aquellos paquetes destinado a nuestra dirección MAC.
Cuando un host quiere una sesión, manda un broadcast ARP con la IP del host al que se quiere conectar. Nuestro sistema crea una tabla de direcciones ARP en forma de caché para que se pueda conectar a hosts conocidos sin tener que volver a realizar otro broadcast. El protocolo arp es el encargado de traducir las direcciones IP de 32 bits a las correspondientes direcciones de hardware. En Ethernet y Token Ring estas direcciones tienen 48 bits. La traducción inversa la hace el protocolo RARP o Reverse ARP. Con este caché los atacantes pueden utilizarlo para realizar el ataque ya que lo que guarda es la hardware dirección física o MAC.

Esquemáticamente el proceso es:



El Spoofing de ARP es una variante de IP Spoofing porque explota una debilidad del protocolo TCP/IP. La autentificación ARP está basada, al igual que el Spoofing de IP en la dirección fuente, es decir, la dirección de donde provienen los paquetes, lo que lo diferencia a ARP es que se fija en una dirección MAC.

Si queremos ver las direcciones de otros equipos de la red, haremos uso de la caché arp de nuestro equipo, mediante el comando arp -a. Este comando nos

mostrará la relación IP/MAC de los equipos que la caché tiene almacenados en ese momento (Si queremos obtener la dirección Ethernet, de una máquina, primero le haremos un ping. De esta forma almacenaremos la dirección MAC en nuestra caché y mediante arp -a podremos obtener su dirección MAC). Cada vez que deseamos comunicarnos con un equipo de la red, debemos de conocer su dirección MAC.

El proceso de Spoofing de ARP consiste en engañar la caché arp de las dos máquinas a las que se desea espiar. Una vez que las caches han sido engañadas, los dos hosts comenzarán la comunicación, pero los paquetes serán para nosotros, los escuchamos y los enrutaremos de nuevo al host apropiado. De esta forma la comunicación es transparente para los dos hosts. La única forma de descubrir que existe alguien escuchando en nuestra conexión, es ver la caché arp de nuestra máquina y comprobar si existen dos máquinas con la misma dirección MAC. Un ejemplo simple podemos tener el siguiente escenario:

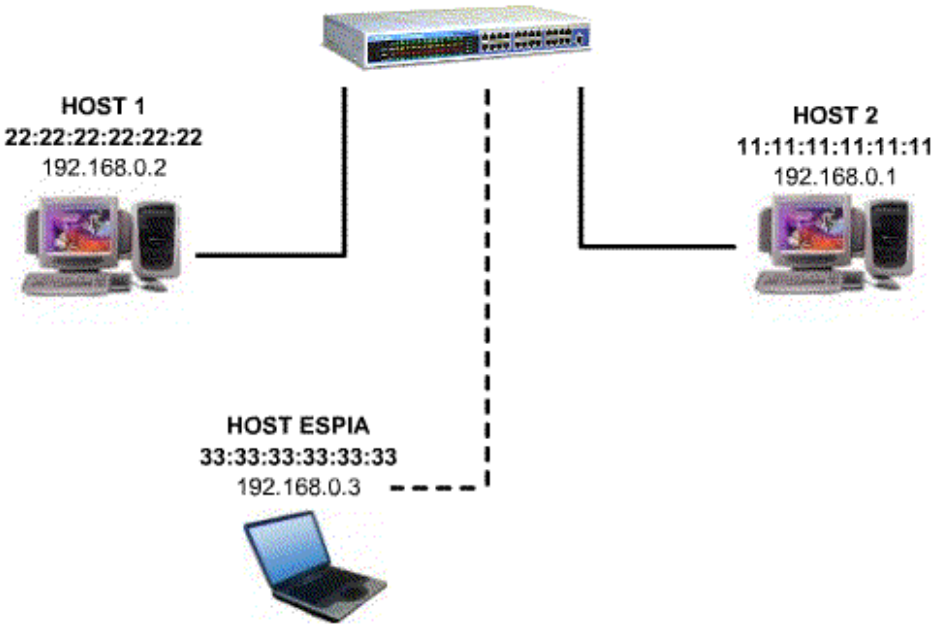
Enviamos constantemente datagramas arp_reply (para evitar que la cache arp de las máquinas se actualice con la información correcta) al host 1 y host 2 con los siguientes datos:

HOST 1: arp_reply diciendo que 192.168.0.2 tiene dirección MAC 33:33:33:33:33:33
HOST 2: arp_reply diciendo que 192.168.0.1 tiene dirección MAC 33:33:33:33:33:33

De esta forma estamos engañando las caché arp. A partir de ahora lo paquetes que se envíen entre ambas nos llegarán a nosotros, pero para que ambos hosts no noten nada extraño, deberemos de hacer llegar los paquetes a su destino final. Para ello deberemos de tratar los paquetes que recibamos en función del host de origen (figura 2):

Datagramas del HOST 1 ----------> reenviar a 22:22:22:22:22:22
Datagramas del HOST 2 ----------> reenviar a 11:11:11:11:11:11

De esta forma la comunicación entre ambos no se ve interrumpida, y podemos espiar todo el tráfico entre ellos. Solo tendremos que utilizar un programa sniffer para poder capturar y filtrar el tráfico entre ambos. Existen varios programas para poner en práctica el Spoofing de ARP tales son: Ettercap, Arptool, Arp_ Fun,etc.



Las vulnerabilidades del protocolo ARP son múltiples: ataques DoS (Negación de servicio), si engañamos la caché arp de una máquina podemos hacernos pasar por otra. Algunos switches pueden ser manipulados mediante paquetes ARP para que en vez de actuar en modo segmentado lo hagan en modo de repetición. Es decir, que en vez de enviar los paquetes por el puerto correspondiente del switch, los enviará a todas las máquinas. Esto se consigue inundando la tabla de direcciones con gran cantidad de direcciones MAC falsas. El switch al recibir un paquete cuya dirección MAC de destino no tenga en su caché, lo enviará a todos los equipos, esperando la respuesta del equipo para poder almacenar su MAC en la caché. Pero como estamos saturándola con direcciones MAC falsas, esto no ocurrirá.

Spoofing de Web

El atacante crea un sitio web (falso) 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: datos, contraseñas, números de tarjeta de créditos, etc. El atacante también es capaz de modificar cualquier dato que se esté transmitiendo entre el servidor original y la víctima o viceversa.
Es un ataque peligroso, y difícilmente detectable, que hoy por hoy se puede llevar a cabo en Internet. Afortunadamente hay algunas medidas preventivas que se pueden llevar a cabo:

1. Desactivar la opción de JavaScript en el navegador.
2. Asegurarse en todo momento que la barra de navegación esta activa.

3. Poner atención a las URL que se enseñan en la barra de estado, asegurándote que siempre apuntan al sitio que quieres conectar.

Hoy en día tanto JavaScript como Active-X, como Java tienden a facilitar las técnicas de spoofing, así que recomendamos que se desactiven del navegador, al menos en los momentos que vaya a transferir información crítica como login, password, números de cuenta bancaria, etc.

Medidas Preventivas

Descubrir los ataques: Si supervisa paquetes usando software como *netlog* y se encuentra paquetes en su interfaz externa que tienen IP de origen y destino dentro del dominio local, podemos estar siendo atacados. Otra manera de descubrir la falsificación de IP es comparar los logs de acceso al sistema en su red interna. Si el ataque ha tenido éxito, se podrá conseguir una entrada de acceso remoto en el log de la máquina víctima, mientras que en la máquina de origen (también en su dominio) no habrá ninguna entrada correspondiente al inicio de ese acceso remoto.

No confiar en nadie: Una sencilla solución para prevenir este ataque es no utilizar la autentificación basada en direcciones. Desactivar los comandos "r", borrar todos los archivos *.rhosts* y vaciar el archivo */etc/hosts.equiv*. Esto forzará a todos los usuarios a emplear otras medidas de acceso remoto (telnet, ssh, skey, etc.)

Filtrado de Paquetes: Si su host tiene una conexión directa a Internet, puede utilizar su router como ayuda. Primero asegúrese de que únicamente hosts de su propia LAN interna pueden participar en relaciones de trust (ningún host interno debe ser trusted host de otro sistema externo a la LAN, o viceversa). Luego utilice el filtrado de paquetes para evitar que paquetes con una dirección de origen que pertenezca a su LAN llegado desde Internet pueda llegar hasta el interior.

6.4 Sniffers

Introducción

¿Alguna vez se ha preguntado como su computadora se comunica con otras en una red? ¿Le gustaría escuchar o husmear, en la conversación? Los ingenieros de red, administradores de sistemas, profesionales de seguridad y, desafortunadamente, los crackers han usado por mucho tiempo una herramienta que les permite hacer exactamente eso. Esta utilidad husmeadora, conocida como **sniffer**, puede encontrarse en el arsenal de cualquier guru de redes y que la utiliza todos los días para una infinidad de tareas. Este articulo dará una breve introducción de **sniffers**, incluyendo, que hacen, como trabajan, porque los

usuarios deben conocerlos y que pueden hacer los usuarios para protegerse del uso inapropiado de los **sniffers**.

¿Qué es un sniffer?

Un **sniffer** es una porción de software que toma todo el tráfico que está fluyendo hacia y desde una computadora conectada a una red. Están disponibles para muchas plataformas en variedades comerciales y **open-source**. Algunos de los paquetes más simples son en realidad sencillos de implementar en C o Perl, utilice una interfaz de comandos de línea y coloque la información capturada a la pantalla. Algunos proyectos más complejos utilizan una GUI, grafican estadísticas de tráfico, siguen sesiones múltiples y ofrecen varios puntos de configuración. Los **sniffers** son también los motores para otros programas. Los Sistemas de Detección de Intrusos (IDS) utilizan **sniffers** para comparar los paquetes contra una regla para detectar cualquier cosa maliciosa o extraña. Programas para saber la utilización de la Red y monitoreo utilizan a menudo **sniffers** para recolectar la información necesaria para análisis y mediciones. Los juzgados que necesitan investigar los correos electrónicos, usualmente utilizan **sniffers** diseñados para capturar tráfico muy específico. Sabiendo que los **sniffers** solamente toman información de la red, nos da una idea de cómo trabajan.

¿Cómo trabaja un sniffer?

Antes de que podamos saber como opera un **sniffer**, puede ser útil examinar cuáles son las partes que le permiten a la herramienta trabajar. Durante operaciones normales como navegación en la red y envío de mensajes, las computadoras se están constantemente comunicando con otras máquinas. Obviamente, un usuario debería ser capaz de ver todo el tráfico hacia o desde su máquina. Si la red no esta conectada a un **switch** (un **switch** es un dispositivo que filtra y reenvía paquetes entre segmentos de una LAN), el tráfico destinado para cualquier máquina en un segmento de la LAN es enviado a todas las máquinas del segmento (**broadcast**). Esto significa que la computadora realmente ve la información viajando hacia y desde todos sus vecinos, pero los ignora, a menos que se le indique que son para él.

Ahora podemos comenzar a entender la magia detrás de un **sniffer**. El programa husmeador le dice a una computadora, específicamente a su Tarjeta de Interfaz de Red (NIC), que deje de ignorar el tráfico que no es enviado para él y que ponga atención en toda la información en la red. Hace esto colocando a la NIC en un estado conocido como modo *promiscuo*. Una vez que la NIC entra en este modo, un estado que requiere privilegios administrativos, una máquina puede ver toda la información transmitida en el segmento. El programa entonces lee constantemente toda la información entrando a la PC a través de la Tarjeta de Red. La información que viaja a través de la red viene en tramas, o paquetes, ráfagas de bits formateadas con un protocolo especifico. Debido a este formateo, un **sniffer**

puede quitar las capas de encapsulación y decodificar la información relevante almacenada dentro de la capsula: computadora fuente, puerto de entrada marcado, carga, etc., en síntesis, todos los pedazos de información que están intercambiando las máquinas en la red.

¿Cómo se ve la información que regresa un sniffer?

Es fácil tomar los conceptos discutidos arriba mirando un **sniffer** en acción. La información en el ejemplo siguiente fue derivada usando el tcpdump, un programa que ha estado circulando por mucho tiempo y está disponible para muchas plataformas. Este **sniffed** particular es un intercambio abreviado entre una máquina y el servidor Web de SecurityFocus.

```
21:06:30.786814     0:1:3:e5:46:6b     0:4:5a:d1:46:ad     0800     650:
192.168.1.3.32946 >
66.38.151.10.80:   P   [tcp   sum   ok]   1:585(584)   ack   336   win   64080
<nop,nop,timestamp 608776
899338> (DF) (ttl 64, id 7468, len 636)
0x0000   4500 027c 1d2c 4000 4006 8074 c0a8 0103      E..|.,@.@..t....
0x0010   4226 970a 80b2 0050 54ac b070 78ef d6c3      B&.....PT..px...
0x0020   8018 fa50 c663 0000 0101 080a 0009 4a08      ...P.c........J.
0x0030   000d b90a 4745 5420 2f63 6f72 706f 7261      ....GET./corpora
0x0040   7465 2f69 6d61 6765 732f 6275 696c 642f      te/images/build/
0x0050   626c 6c74 5f72 645f 312e 6769 6620 4854      bllt_rd_1.gif.HT
0x0060   5450 2f31 2e31 0d0a 486f 7374 3a20 7777      TP/1.1..Host:.ww
0x0070   772e 7365 6375 7269 7479 666f 6375 732e      w.securityfocus.
0x0080   636f 6d0d 0a55 7365 722d 4167 656e 743a      com..User-Agent:
0x0090   204d 6f7a 696c 6c61 2f35 2e30 2028 5831      .Mozilla/5.0.(X1
0x00a0   313b 2055 3b20 4c69 6e75 7820 6936 3836      1;.U;.Linux.i686

21:06:30.886814 0:4:5a:d1:46:ad 0:1:3:e5:46:6b 0800 402: 66.38.151.10.80
>
192.168.1.3.32949:   P   [tcp   sum   ok]   2363393025:2363393361(336)   ack
1437810754 win 8616
<nop,nop, timestamp 899338 608766> (ttl 61, id 10825, len 388)
0x0000   4500 0184 2a49 0000 3d06 b74f 4226 970a      E...*I..=..OB&..
0x0010   c0a8 0103 0050 80b5 8cde 8401 55b3 4042      .....P......U.@B
0x0020   8018 21a8 0543 0000 0101 080a 000d b90a      ..!..C..........
0x0030   0009 49fe 4854 5450 2f31 2e31 2032 3030      ..I.HTTP/1.1.200
0x0040   204f 4b0d 0a41 6765 3a20 320d 0a41 6363      .OK..Age:.2..Acc
0x0050   6570 742d 5261 6e67 6573 3a20 6279 7465      ept-Ranges:.byte
0x0060   730d 0a44 6174 653a 2054 7565 2c20 3132      s..Date:.Tue,.12
0x0070   2046 6562 2032 3030 3220 3033 3a30 343a      .Feb.2002.03:04:
0x0080   3538 2047 4d54 0d0a 436f 6e74 656e 742d      58.GMT..Content-
0x0090   4c65 6e67 7468 3a20 3433 0d0a 436f 6e74      Length:.43..Cont
0x00a0   656e 742d 5479 7065 3a20 696d 6167 652f      ent-Type:.image/
0x00b0   6769 660d 0a53 6572 7665 723a 2041 7061      gif..Server:.Apa
0x00c0   6368 652f 312e 332e 3232 2028 556e 6978      che/1.3.22.(Unix
0x00d0   2920 6d6f 645f 7065 726c 2f31 2e32 360d      ).mod_perl/1.26.
```

En la imagen se muestran dos paquetes: una petición HTTP del cliente y la respuesta del servidor. Note que las primeras líneas de cada paquete dan un

sumario de la transacción: tiempos, direcciones MAC fuente y destino, direcciones IP fuente y destino, y otros bit de información más. Las líneas numeradas (0X00##) muestran la información transmitida por cada paquete en formato hexadecimal. Adicionalmente, una decodificación ASCII de la carga esta localizada a la derecha – una característica conveniente para los **crackers** y vecinos ruidosos que lo quieren ver en la red.

¿Por qué los usuarios deben conocerlos?

En una LAN normal hay miles de paquetes intercambiados por múltiples máquinas cada minuto, fuente suficiente para cualquier enemigo. Cualquier cosa transmitida como texto plano en la red es vulnerable, como contraseñas, páginas Web, consultas a bases de datos y mensajes por nombrar algunos. Un **sniffer** puede ser fácilmente personalizado para capturar tráfico específico como sesiones de telnet o e-mail. Una vez que el tráfico es capturado, los **crackers** pueden extraer fácilmente la información que necesitan como, contraseñas, logins, y el texto de algunos mensajes. Además los usuarios no se darán cuenta que fueron comprometidos, debido a que los **sniffers** no causan problemas o deterioros a la red, por lo cuál no son detectados.

¿Cómo se pueden proteger los usuarios?

Herramientas Anti-Sniffing

Un aspecto aterrador de estas herramientas es quien puede, y debe, usarlas. Como se menciono anteriormente, los **sniffers** pueden ser usados para propósitos legítimos e ilegítimos. Por ejemplo, un administrador de red puede usarlos para monitorear el flujo de tráfico en la red para asegurarse que la red funciona de manera adecuada. Sin embargo, los **sniffers** también pueden ser usados por usuarios maliciosos para obtener información personal valiosa. Ya sean contraseñas o comunicaciones privadas, tanto los **crackers** como los empleados se pueden beneficiar de leer dicha información. El defenderse de los **sniffers** como de cualquier otro ataque, tiene que empezar en la cima y filtrar a los usuarios. Como en cualquier red, los administradores tienen que asegurar máquinas individuales y servidores. Un **sniffer** es una de las primeras cosas que un **cracker** cargará para ver que esta pasando dentro y alrededor de la máquina que desean atacar.

Otro método de protección incluye herramientas, que analizan la red para determinar si alguna NIC esta trabajando en modo *promiscuo*. Estas herramientas de detección deben trabajar regularmente, ya que actúan como una alarma de clases, disparadas por evidencia de existencia de un **sniffer.**

Redes Switcheadas

Una red switcheada es un buen impedimento. En el entorno no switcheado, los paquetes son visibles para todos los nodos en la red, en un entorno switcheado, los paquetes solo son enviados a la dirección destino. Aunque son mas caros que los hubs, el costo de los switches ha disminuido con el tiempo, llevándolos dentro de los alcances de la mayoría de los bolsillos. A diferencia de los hubs, los switches solo envían tramas al recipiente designado; por lo tanto una NIC en modo *promiscuo* en una red switcheada no capturará cada pieza del tráfico local. Pero programas como dsniff, le permiten a un enemigo analizar una red switcheada con una técnica conocida como arp-spoofing. A pesar de que usa diferentes métodos, arp-spoofing puede dar resultados similares al spoofing.

Encripción

La encripción es la mejor forma de protección contra cualquier forma de intercepción de tráfico. Es razonable asumir que en algún punto a lo largo del camino, la información pueda ser comprometida. Por lo tanto, su mejor defensa es asegurarse que el tráfico sea ilegible esencialmente para todos excepto para el receptor especificado. Esto no es difícil de realizar porque muchas organizaciones han desarrollado servicios que utilizan Capaz de Sockets Seguras (SSL), Seguridad de Capa de Transporte (TLS) y otros métodos para asegurar los mensajes, la navegación en red y más. Solo las cargas son revueltas, asegurándose de que los paquetes lleguen a sus destinos fijados. Por lo tanto un enemigo puede ver hacia donde fue enviado el tráfico y de donde vino, pero no que información contiene.

```
0x0000 4500 00a4 fea8 4000 ed06 43e2 8184 0799 E.....@...C.....
0x0010 c0a8 0103 01bb 80a5 be10 d77f 19a2 0520 ...............
0x0020 5018 2744 8303 0000 4d3a a587 805e e2bc P.'D....M:...^..
0x0030 9a2a 8ff3 fe95 46d4 930e b2bc 74f0 a484 .*....F.....t...
0x0040 fcae 33ad 6d1f 0198 6020 aee5 0c26 908e ..3.m...`....&..
0x0050 a1b5 17b4 84b7 44bc 1b0b 434e bbae a483 ......D...CN....
0x0060 1e23 38d3 520f 687e c5e3 b62e 5225 aa2f .#8.R.h~....R%./
0x0070 f747 1a71 669c 8fd1 55bd 511c 4988 b78a .G.qf...U.Q.I...
0x0080 a08d 554e a3fe bb7d 36ca e66b fb8b 0392 ..UN...}6..k....
0x0090 a3f3 4cef 7b04 af5a 7a94 cb4c a1e6 e7fa ..L.{..Zz..L....
0x00a0 9610 a5ee ....
```

Compare este ejemplo de **sniffer** de una sesión Web con el servidor Web OpenSSL con el ejemplo anterior. Note que la información de la cabecera permanece legible, pero el código ASCII de la carga contiene caracteres aleatorios, generados con la encripción. Los dos participantes en el intercambio, sin embargo, pueden desencriptar y procesar la información una vez que se recibe. Este tipo de salvavidas puede aplicarse a todos los procesos de red y debe emplearse cuando sea posible.

¿Puedo usar un Sniffer?

Un **sniffer** puede ser una herramienta invaluable para los administradores, profesionales de seguridad, programadores e incluso principiantes. Son utilidades excelentes para examinar cualquier problema en la red, debido a que son una ventana al tráfico local. Podemos usar **sniffers** para detectar máquinas que están enviando información periódicamente a **crackers** sobre datos de la organización, contraseñas, logins, etc. Para programación en red los **sniffers** son una necesidad para depurar en las etapas de desarrollo. Los **sniffers** son una excelente fuente para los principiantes curiosos, que desean entender las redes y la seguridad. Nada lo puede acercar más a lo que pasa realmente cuando las computadoras se comunican, que estas herramientas.

Debe notarse que el usuario casual debe ser muy cuidadoso sobre cuando, como y donde usar estos programas. Nunca utilice un **sniffer** en una red local sin examinar antes con el administrador. Es mejor probar estas técnicas en casa o en una red que usted maneje.

6.5 Intrusion Detection Systems (IDS)

Puesto que la tecnología y las redes globales se han expandido y han evolucionado rápidamente, los ataques a este tipo de sistemas de información se han disparado en los últimos años. En ocasiones, los equipos de seguridad contra ataques solían atacar el sistema como un procedimiento de prueba para informarle a la víctima que era un objetivo para un ataque determinado. La víctima puede ser cualquiera; si se trata de una red conectada a Internet, es susceptible a un ataque. Este rápido crecimiento por lo general significa sacrificios para establecer seguridad y privacidad en tales sistemas. Por ejemplo, existe el caso del gusano Internet que se extendió por la totalidad de Internet y causó millones de dólares de daños.

Los sistemas de detección de intrusiones (IDS) son un conjunto de programas y algoritmos utilizados con un conjunto de referencias; este conjunto puede ser un objeto estático o un conjunto de reglas predefinidas. Están diseñados para asegurar que el uso de un ambiente de cómputo sea acorde con la política de seguridad predefinida.

Las redes son vulnerables debido a que su identidad puede revelarse simplemente al realizar una consulta “whois” en el InterNIC. Un a vez que se ha completado la consulta “whois”, el siguiente paso para el atacante es hacer un mapa interno de la topología de la red. No es difícil para un atacante hacer esto; un cortafuego no bloquea esta consulta. La única para interrumpir esto es tener un servidor DNS instalado en una DMZ en el punto de conexión a Internet, y que las consultas

del DNS vayan a esa máquina. El atacante podría intentar una “ruta trazada” a la red para ver dónde se sueltan finalmente los paquetes y utilizar la dirección IP como un posible punto de partida.

Las consultas del DNS pueden revelar el mapa de la topología de una compañía grande. Los servidores DNS que se consideren “no confiables” (por ejemplo, aquellos en la DMZ) no deberían tener ningún mapa del DNS de la arquitectura interna. Sólo deberían saber qué se ha hecho a grandes rasgos para la comunidad de Internet. En otras palabras, un servidor DNS no confiable sólo deberá saber de los servidores Web, FTP y HTTP lo que las personas ajenas a su organización necesitan conocer.

NOTA SOBRE SEGURIDAD: si usted implementa un firewall que utilice NAT y permite consultas DNS a través de él, un atacante puede trazar la topología de su red. En un paquete de datos y no será modificada por el firewall.

Después de que un atacante ha consultado a la InterNIC y a distintos servidores DNS para conseguir información sobre el sitio, el siguiente paso es la proyección del DNS. Si tiene un servidor DNS que maneje tanto la información interna como la externa, un atacante puede trazar el mapa de la red. La razón es que el cortafuego no modifica el paquete de datos, el cual puede contener una dirección IP del anfitrión interno.

Muchas empresas tienen una dirección pública enrutable válida que obtuvieron anteriormente. Están conectadas a Internet y, por razones de seguridad, no quieren que esta dirección esté enrutada a Internet. Por lo tanto, consiguen un espacio en la dirección IP de su PSI e instalan una traducción de direcciones de red en su firewall. La única función del cortafuego es proyectar la dirección interna a la dirección externa. Ahora, para tener un funcionamiento al DNS y que la gente encuentre el servidor Web, el servidor FTP, etc., muchas compañías permiten la entrada de tráfico al DNS. Si este DNS también se emplea para proyecciones internas de DNS, un atacante puede consultar a este servidor y obtener un mapa de la red. La única manera de evitar esto es con la utilización de un tercero como el anfitrión DNS de la compañía, o instalar un servidor DNS por separado en una zona DMZ.

El siguiente paso en el proceso podría ser el uso de un estroboscopio para que realice exploraciones de los puertos en los distintos dispositivos que existen en la DMZ y, si es posible, a lo largo de cualquier proyección de DNS disponible. Entonces, el siguiente paso del atacante podría ser el intento de provocar que cualquier máquina revelara su archivo de contraseñas para listar los nombres de las cuentas de usuarios. Después, si es posible, el atacante puede utilizar distintos programas para adivinar contraseñas e intentar descubrir cualquier identidad de usuario y, finalmente, intentar forzar la cuenta del administrador.

NOTA SOBRE SEGURIDAD: la exploración de puertos no es la única vulnerabilidad que debe impedirse. Si permite protocolos no autorizados (por ejemplo, el protocolo GRE) para pasar un cortafuego, los ataques pueden montarse en ellos.

Los primeros sistemas de detección de intrusiones utilizaban varios guiones y programas caseros diseñados para penetrar la detección. Los sistemas de detección de intrusiones son una forma de sistemas expertos; están en la misma categoría que los sistemas de inteligencia artificial, los sistemas lógicos basados en reglas y las redes neuronales. COPS, es un sistema de detección de intrusiones diseñado para hacer revisiones en busca de puntos vulnerables en un sistema. COPS utiliza un conjunto de herramientas que automatizan el proceso de recopilación de información para análisis.

Necesidad de sistemas de detección de intrusiones

Muchas organizaciones asumen que el mejor tipo de dispositivo contra intrusiones y ataques es el cortafuego. Lamentablemente, esta lógica no es cierta. Diariamente se dan a conocer reportes de nuevos ataques, vulnerabilidades y equipos comprometidos. Si los cortafuegos fueran la respuesta, entonces estos ataques nunca habrían ocurrido. Los cortafuegos son parte del cuadro, una buena parte, pero no todo. Si se configuran apropiadamente, los cortafuegos bloquean internamente los ataques dirigidos, pero los ataques pueden ocurrir en dispositivos que son accesibles al público, por ejemplo, los servidores Web, y los ataques pueden ocurrir cuando se permite que el tráfico pase el cortafuego, por ejemplo, las consultas al DNS. Los sistemas de detección de intrusiones no reemplazan a los cortafuegos, sino que los complementan.

Se supone que los dispositivos de autenticación proporcionan un mecanismo de acceso a los usuarios, así que además de los cortafuegos, los usuarios tienen que ser autenticados para conseguir el acceso. Sin embargo, incluso los sistemas de autenticación fallan, permitiendo que los atacantes consigan el acceso. Los sistemas de detección de intrusiones pueden complementar a este tipo de sistemas. Las redes y los sistemas de detección de intrusiones pueden vigilar los servidores y las redes mismas, así que incluso si un atacante descifra una cuenta en un servidor de autenticación más allá del cortafuego, el IDS puede detectar este tráfico sospechoso.

Categorías de los sistemas de intrusión

Todos los sistemas de detección de intrusiones tienen una máquina y un conjunto de reglas (u objetos de datos) sobre los cuales actúa la máquina. Una máquina puede actuar con base en un conjunto de reglas para buscar intrusiones, o puede utilizar un conjunto de objetos básicos para actuar contra éstas. Las máquinas de IDS se dividen en dos categorías máquinas de uso erróneo, o patrón, y máquinas de anomalías.

Máquinas de uso erróneo (patrón)

Estas máquinas se organizan de acuerdo a los ataques comunes en puntos conocidos del sistema, por ejemplo, puntos vulnerables en un sistema operativo que llevan a un problema de desbordamiento de la memoria intermedia. Por lo general, se detectan al vigilar ciertos objetos y notar una desviación de la normalidad. Estos sistemas generalmente utilizan alguna forma de patrón de concordancia que buscan desviaciones.

Algunos de los beneficios de esta clase de sistemas son la eficiencia y la centralización. Se pueden ajustar para un servidor en específico y solo requieren un conjunto limitado de reglas para funcionar. Las limitaciones con esta clase son la carencia de crecimiento, la necesidad de establecer nuevos patrones de concordancia y un área establecida reducida. No es posible tomar este tipo de sistemas y esperar que concuerde con toda la infraestructura. Además para configurar una nueva clase de ataques en estos sistemas, se necesita programar todo un nuevo conjunto de patrones. De hecho, estos sistemas generalmente se implementan para un objetivo específico.

Máquinas de anomalías

Los sistemas de anomalías son aquellos que se establecen al observar desviaciones de la norma; por lo general se construyen perfiles y se señalan las desviaciones de estos perfiles. Por ejemplo, se revisa un archivo base, cuyo tamaño y resumen se conocen para determinar si ha ocurrido un cambio en el tamaño y el resumen del archivo original. Este tipo de sistemas generalmente utiliza alguna métrica para establecer las bases. Se puede pensar en los sistemas de detección de anomalías como en sistemas que utilizan algún tipo de inteligencia artificial. Este sistema puede evolucionar con su propio conocimiento.

Los beneficios de este tipo de sistemas incluyen la capacidad para establecer umbrales y para trabajar en tiempo real. Se pueden ver las cosas conforme aparecen; no es necesario esperar a que la alerta se active o hasta tener una oportunidad para vigilar los registros. También se ajustan para permitir configuraciones de umbral; por lo tanto, es posible detectar que ocurre un ataque antes de que cause cualquier daño.

Sin embargo, sufren el mismo problema de "base incorrecta" : ¿quién decide qué y cómo se define una base, de modo que el mecanismo pueda detectar exactamente las intrusiones? Incluso, un ataque que sabe que se emplea un IDS de anomalías posiblemente puede evitarlo. El atacante puede utilizar distintas cuentas y realizar varios ataques; en otras palabras, el atacante ataca con una cuenta en particular, y después ataca con otra cuenta y con otro tipo de ataque. Además, la sensibilidad en este tipo de ataques tiene que ser dirigida, puesto que depende del sistema decidir el momento en que un usuario tiene demasiados archivos abiertos y ha intentado demasiados registros. Cualquiera que utilice un dispositivo de autenticación en el servidor de la red donde el tiempo es un poco excesivo entre el cliente y el servidor sabe cómo un atacante fácilmente puede intentar varias veces el acceso al sistema.

Los sistemas de anomalías tienen menos tiempo para detectar las intrusiones que su contraparte, los sistemas de uso erróneo. No hay patrones fijos que puedan vigilarse y, si se intenta un ataque, no hay una base para que el mecanismo reaccione en contra. Por lo tanto, estos sistemas necesitan combinar un patrón de razonamiento humano con un programa tipo computadora. Algunos sistemas de intrusiones basan su análisis en un seguimiento de auditorias. Este patrón se construye con el tiempo y los datos se emplean como una fuente de referencia. Se miden cosas tales como los ciclos del CPU, el empleo de la memoria y el espacio en disco.

Aunque existen dos categorías de máquinas IDS, también existen diferentes tipos de sistemas que se emplean con esas máquinas: los que vigilan la red, los que vigilan el sistema y los que vigilan los archivos de registro.

Sistemas de detección de intrusiones en redes

Este tipo de sistemas vigila los paquetes de datos en la red. Estos sistemas buscan anomalías en el circuito, como los ataques de negación del servicio, los cuales pueden comenzar con una gran cantidad de solicitudes TCP SYN, o cuando alguien realiza una exploración de puertos en una de las máquinas. Los sistemas NIS supervisan una gran cantidad de máquinas.

Sistemas de detección de intrusiones en el sistema

Este tipo de sistemas vigila los archivos del sistema. Observa los cambios en un archivo que se desvía de una línea establecida. Tales cambios pueden ocurrir si un atacante construyó una puerta trasera a través de uno de estos archivos principales del sistema.

Sistemas de detección de intrusiones de registros

Este tipo de sistema vigila los archivos de registro generados por varios servicios en la máquina. Busca un patrón de tráfico que puede indicar que un atacante ha intentado conseguir el acceso al sistema a través de las vulnerabilidades conocidas.

Sistemas de detección de intrusiones de engaños

Estos sistemas en realidad son trampas; están diseñados para actuar como un servicio conocido que incita al atacante y lo atrapa.

NOTA: los sistemas de detección de intrusiones no pueden vigilar el tipo de tráfico en las redes privadas virtuales, el cual normalmente está cifrado. Por lo tanto, su ubicación debe estar después del dispositivo de cifrado / descifrado.

Los paquetes de datos que viajan por el flujo de las RPV normalmente están cifrados, por ejemplo, DES, IDEA, etc. Por lo tanto, puede ser que un IDS no realice el análisis necesario del flujo de datos. La situación se complica más con los modos de túneles, donde todo un paquete se cifra detrás de una dirección IP nueva. Para que los sistemas de detección de intrusiones funcionen apropiadamente, deben trabajar después del descifrado. En el caso de un flujo de cifrado PPTP punto a punto, esto puede significar que los sistemas de detección de intrusiones deben residir en el propio anfitrión.

Características de un buen sistema de detección de intrusiones

Se ha escrito mucho sobre qué es lo que integran un buen sistema de detección de intrusiones. Algunos de estos aspectos se mencionan a continuación:

***Seguridad**:* el sistema que se emplee tiene que ser seguro por sí mismo, o no será lo suficientemente seguro como para vigilar otros sistemas. Si ocurre un ataque o una intrusión, ningún dato que produzca podrá considerarse confiable.

***Desviaciones de la norma**: l*os sistemas de intrusión deben señalar cuando encuentren desviaciones de la norma.

***Intervención humana no necesaria**:* los sistemas de intrusión necesitan ejecutarse sin obstáculos en el sistema principal. No deberán interferir con el sistema ni con la red que vigilan. Sin embargo, deberían ser accesibles desde el exterior para que sus resultados puedan supervisarse.

***Imposición mínima de sobrecarga**:* no se puede instalar un sistema de detección de intrusiones si se crea una situación donde dicho sistema consume más recursos que para lo que se diseñó.

Alerta: los sistemas de detección de intrusiones deben estar configurados para emitir cualquier conjunto de alertas o páginas al personal apropiado. Éstas pueden incluir trampa SNMP, eventos NT, edición en el syslog, correo electrónico, páginas y lanzamientos de otros programas que inician un procedimiento para reducir el impacto del ataque.

Flexibilidad*:* así como la política de seguridad debe ser flexible, también debe serlo el sistema de detección de intrusiones. Debe existir una manera de agregar esto a las bases, a través de medidas o lo que sea que emplee como mecanismo para activar las intrusiones.

Cliente/servidor*: d*ebe elegir sistemas y redes de detección de intrusiones que sean heterogéneos y pueden trabajar a través de distintos entornos, por ejemplo, UNIX y NT.

Informe: no tiene sentido instalar un IDS sin tener capacidad para crear informes o tener informes que sean complicados de leer. Se requieren informes claros que puedan dividirse sólo en aquellas piezas de información necesaria.

Detección de intrusiones/huellas

Para que los sistemas de detección de intrusiones funcionen, deben colocarse apropiadamente en la red. Una vez que la intrusión ha ocurrido, ya sea que la seleccione o no el IDS, por lo general queda una huella.

En cualquier organización la red se divide en subredes, separadas por conmutadores, enrutadores y cualquier otro dispositivo de red que pueda enlutar el tráfico, por ejemplo, un cortafuego. El tráfico es dirigido por la dirección de destino y por la política de enrutamiento aplicada en los enrutadores en toda la organización; en otras palabras, la rita predeterminada. Los sistemas de detección de intrusiones en redes examinan los paquetes de datos que pasan por el cable.

NOTA GENERAL: los sistemas de detección de intrusiones no modifican los paquetes de datos; no son software de almacenamiento y reenvío. De hecho, para que sean efectivos, tienen que estar donde puedan vigilar el tráfico que la compañía considera crítico.

NOTA SOBRE SEGURIDAD: los sistemas de detección de intrusiones no trabajan en redes conmutadas; por lo tanto, la ubicación es crítica para el éxito de la señalización de intrusiones.

La ubicación es un paso crítico cuando se implementa un sistema IDS. Los sistemas de detección de intrusiones en redes por lo general tienen sus

adaptadores de red en modo de escucha, lo que significa que escuchan todos los paquetes de datos que pasan por el cable.

Las redes conmutadas no son redes de transmisión general; por lo tanto, segmentan el tráfico. El sistema de detección de intrusiones se coloca detrás del cortafuego, el enrutador interno y del conmutador, y se pone en el anillo FDDI. El IDS sólo se vigilará el tráfico que se encuentre en el anillo FDDI; cualquier otro dispositivo será vulnerable.

Ahora vea dos ubicaciones, en la primera ubicación se localiza en un concentrador que esté conectado entre el cortafuego y el enrutador interno. En la segunda ubicación, actúa como una compuerta que reenviará el tráfico en ambas direcciones.

El empleo de cualquiera de las configuraciones anteriores le proporciona al IDS acceso a todo el tráfico de datos que pasa por el cable y por lo tanto le permite vigilar todos los datos, y al mismo tiempo, incrementar la utilización de sus recursos. El IDS está escuchando y capturando los paquetes. En la parte inferior se convierte en un enrutador en sí. Aunque su carga se incrementa en la configuración inferior, esta configuración puede proporcionar más flexibilidad al sistema. Además de firewall, el IDS puede actuar como otro filtro, dispositivo de autenticación o máquina de filtrado basada en el Web. Sin embargo, en el mundo real el IDS es tan sólo otra pieza de software y muchas compañías tienen un servidor que realiza estas funciones. Sólo es necesario decidir si esta configuración es la apropiada para una compañía.

En lugar de que instale un IDS en una ubicación específica en la red donde parezca que resulta mejor, los agentes IDS están dispersos por toda la red. En algún lugar de la red, un IDS maestro recolectará toda la información de los distintos anfitriones, enrutadores, conmutadores y cortafuegos de la red. Después llevará un proceso de correlación del tráfico para buscar las intrusiones. Con este tipo de configuración puede tener varios IDS maestros que compartan datos y recolecten información de las distintas máquinas para después procesarlos. El IDS maestro podría utilizar estos recursos para analizar los datos, y las otras máquinas podrían informar al maestro.

6.6 Comercio Electrónico

Comercio electrónico

El comercio electrónico, o *e-commerce*, es el proceso que permite establecer acuerdos y llevar a cabo transacciones en línea con clientes y socios. En la actualidad, más que cambiar la forma en la que se llevan a cabo los intercambios mercantiles, está cambiando la forma en que se establece la relación entre compradores y vendedores. Anteriormente sólo involucraba la venta al menudeo de bienes y servicios sobre redes; ahora, el comercio empresa-empresa se encuentra en plena expansión y se espera que constituya la mayor parte del crecimiento potencial en los próximos años.

Novedades

- **Soluciones de comercio electrónico escalables**

A medida que aumenta la demanda de comercio electrónico, las empresas necesitan soluciones de Internet escalables y de alto rendimiento que puedan controlar elevados volúmenes de transacciones y un gran número de usuarios.

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/articulos/200103/art04/default.asp

- **Business Internet Analytics (BIA) de Microsoft**

Aprovechamiento de la tecnología integrada para ayudar a las empresas a analizar su comercio electrónico.

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/articulos/200103/art02/default.asp

- **Trabajar con un proveedor de hospedaje**

Preparación técnica para el comercio electrónico

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/articulos/200107/art05/default.asp

- **Captura y análisis de registros**

Cualquiera que tenga que crear informes sobre sitios Web encontrará de utilidad la lectura de este documento. También será de ayuda para aquellas personas que leen los informes. Proporcionará a todos los lectores una mejor comprensión sobre el proceso de captura y análisis de registros.

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/articulos/200107/art06/default.asp

- **Administración de versiones para un entorno punto-com**

Administración de servicios tal y como se aplica a una solución de comercio electrónico entre empresas y clientes.

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/articulos/200012/art01

- **Lista de comprobación de Servicios de Internet Information Server 5 seguro**

En este documento se enumeran algunas recomendaciones y procedimientos para proteger un servidor del Web que ejecuta Microsoft Windows 2000 y Servicios de Internet Information Server (IIS) 5.

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/articulos/200012/art03

- **Distribuir Windows 2000 con IIS 5.0 para entornos punto-com: Procedimientos recomendados**.

Implementación de Windows 2000 con IIS 5.0 para entornos punto-com (Dot Coms).

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/articulos/200011/art01/default.asp

Distribución administrada

La estrategia de implementación se refiere al proceso de mover un conjunto de códigos desde un entorno distribuido a otro. La estrategia propuesta es implementar la arquitectura WinDNA como una entidad única en las plataformas de Windows 2000 y Windows NT 4.0 Server.

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/articulos/200011/art02/default.asp

- **Ejecutar Site Server 3.0 Commerce Edition en Windows 2000**

La migración a Windows 2000 Server proporciona una mejora de rendimiento para sitios SSCE y también permite aprovechar otras ventajas de Windows 2000 Server.

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/articulos/200010/art06/default.asp

Comparación de rendimiento entre IIS de Windows NT 4.0 y Windows 2000 para CompanyX.tld

CompanyX.tld junto con Microsoft Consulting Services determinaron el rendimiento de su actual sitio Web en Windows NT 4.0, para luego ejecutar las pruebas de la misma base en el sitio después de actualizar a Windows 2000.

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/articulos/200006/art07/

Introducción a la infraestructura de claves públicas de Windows 2000
En este documento se describen los fundamentos de los sistemas de seguridad de claves públicas, incluidas las ventajas que ofrecen y los componentes necesarios para implementarlos.

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/articulos/windows2k/pkiintro/

Distribución rápida de un sitio de comercio electrónico altamente escalable
Este documento describe cómo un sitio de Microsoft Site Server 3.0 Commerce Edition fue implementado en un sitio de hardware de Compaq en un mínimo de tiempo.

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/articulos/200004/art01/

Diez sugerencias para optimizar Personalization y Membership de Site Server 3.0 Commerce Edition
En este documento se ofrecen diez sugerencias para optimizar el funcionamiento de los servicios Personalization y Membership de Site Server 3.0 Commerce Edition.

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/articulos/200002/art03/

Caso práctico de MS® Market
MS® Market es una aplicación para la obtención de productos basada en Intranet que permite a los empleados de Microsoft realizar fácilmente y en pocos minutos pedidos de productos o servicios, y además recibirlos en 24 horas.

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/articulos/200002/art20/

Ya están disponibles Microsoft Site Server 3.0 y Site Server Commerce Edition
Durante el primer semestre de este año, Microsoft Corp. anunció el lanzamiento de Microsoft® Site Server 3.0 y Microsoft Site Server 3.0 Commerce Edition, la herramienta para la creación de intranets y para soluciones de comercio electrónico a través de Internet

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/ecommerce/art03/art031.asp

Nuevas soluciones para comercio electrónico
Microsoft® Site Server Commerce Edition 3.0 es una herramienta que permite administrar todas las fases que intervienen en el comercio electrónico: involucrar al cliente, tramitar una orden y analizar los resultados. Asimismo, puede ser usado en tres distintos escenarios de comercio: la venta al menudeo en línea, la compra corporativa y la cadena extendida de valor.

Para mayor información consultar la siguiente página electrónica:
http://www.microsoft.com/latam/technet/ecommerce/art01/art011.asp

Ejercicio 1

Investigar y describir los siguientes ataques a través de spoofing:

- Land attack
- Ping of death
- TCP SYN flood
- UDP port loopback
- Smurf
- IP fragment overlap
- ICMP unreachable storm
- Boink DoS

En esta actividad debemos dejar claro para cada uno de los ataques antes mencionados, la descripción del ataque, la forma en la que lo realiza y la posible solución o la manera en la que nos podemos proteger del ataque. Esta tarea la deberá reenviar resuelta.

Fuentes de Referencias Bibliográficas

Molva, Refik, Internet security architecture, *Computer Networks*, Vol. 31, No. 8, april 1999, Elsevier.

Stanger, James, Lane, Patrick T. Hack Proofing Linux. Syngress. 2001.

Páginas de Referencia Electrónicas

http://www.iss.net/security_center/advice/Underground/Hacking/Methods/Technical/Spoofing/default.htm

http://www.cert.org/advisories/CA-1995-01.html

http://www.cs.dartmouth.edu/~pkilab/demos/spoofing/

7. Redes Privadas Virtuales (VPN)

7.1 Introducción

En las últimas décadas el mundo ha sido testigo del gran crecimiento de Internet, las empresas están aprovechando las grandes ventajas que ofrece las tecnologías de la red para nuevas oportunidades de negocios, lo que favorece en gran medida al crecimiento de la "Red de Redes". Este crecimiento no se refieres tan solo al número de host que se agregan a la red, si no también en la manera en que Internet se utiliza en varias aspectos de nuestra vida, como los son prioridades para negocios, requerimientos de clientes y actitudes generales de comercio.

Al inicio de este crecimiento, las empresas, utilizaron Internet para promover sus productos y servicios a través de sus sitios corporativos. En la actualidad el enfoque ha cambiado al e-commerce y e-bussiness, lo que representa nuevos retos de seguridad, costos y acceso global a las aplicaciones de negocios y datos almacenados en los sistemas de las empresas. El problema de realizar transacciones de datos sobre "redes inseguras" (como lo es Internet) ha dado lugar a las Redes Privadas Virtuales (Virtual Private Networks, VPNs).

Una VPN se define como un proceso de comunicación cifrada o encapsulada que transfiere datos desde un punto hacia otro de manera segura; la seguridad de los datos se logra gracias a una tecnología robusta de cifrado, y los datos que se transfieren pasan a través de una red abierta, insegura y enrutada. Otra manera de ver una VPN es el envío de datos de manera segura a través de medios inseguros. Las premisas de trabajo de una VPN son la seguridad, el rendimiento y la confiabilidad en la transmisión de la información.

Es posible conceptuar una VPN como una extensión de una intranet privada a través de una red pública, lo que permite una conexión segura, a bajo costo y eficiente. Para que la intranet pueda extenderse hace uso de "túneles lógicos" (tunnels). Cada túnel habilita los dos puntos de intercambio de datos de manera que cada extremo del túnel reensambla la información de la comunicación punto a punto. Esto puedo verse en la figura 1.

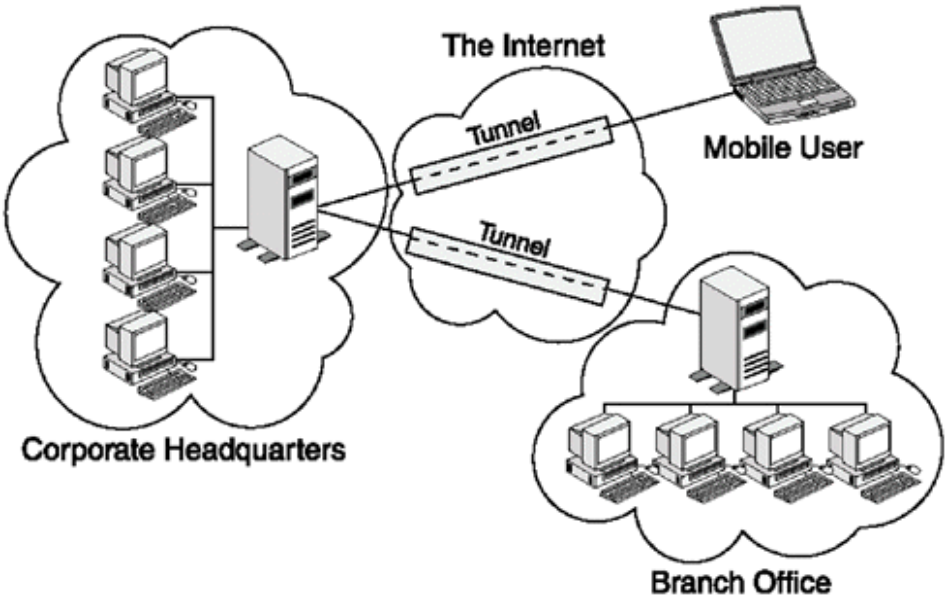


Figura 1. Ejemplo de un túnel en una VPN

Los mecanismos de seguridad utilizados por una VPN son:

- **Cifrado:** es el proceso de cambiar los datos de manera que solo pueda ser leída por el o los receptores autorizados. Las VPNs pueden utilizar cifrado simétrico o de llave privada o cifrado asimétrico (si se requiere mayor información sobre cifrado consultar el capítulo 3).

- **Autenticación:** es el proceso de asegurar que los datos son enviados y recibidos por quién ser enviados y por quienes deben ser recibidos. En una

manera simple la autenticación requiere de un nombre de usuario y contraseña para acceder a un recurso.

- **Autorización:** es el proceso de otorgar o negar acceso a los recursos localizados en una red después de que el usuario se ha identificado y autenticado.

Los principales objetivos de una VPN son:

- Acceso remoto, de los empleados, en cualquier hora y en cualquier lugar, a través de móviles y comunicaciones, hacía los recursos de la red corporativa.
- Interconectividad entre diferentes oficinas remotas.
- Acceso controlado de recursos de red a clientes, proveedores y otras entidades externas que son importantes para el negocio.

Evolución de la VPNs

La tecnología VPN no es nueva, tiene alrededor de 15 años y ha pasado por varias generaciones hasta llegar a lo que es hoy en día. La primera VPN conocida fue ofrecida por la AT&T a finales de los 80's y fue conocida como “Software Defined Networks” (SDNs). Estas SDNs eran WANs de larga distancia que hacían uso tanto de conectividad dedicada como switcheada y se basaban en bases de datos que clasificaban todos los intentos de acceso tanto locales como remotos. Basados en esta información, los paquetes de datos eran encaminados a su destino a través de redes públicas utilizando una infraestructura swicheada.

La segunda generación de VPNs surgió con la aparición de X.25 e ISDN (Integrated Services Digital Networks) en los años 90s. Dichas tecnologías permiten transmisiones de paquetes a través de redes públicas a un bajo costo. El estándar de las VPNs fue X.25 aunque no duró mucho tiempo dado los problemas de transmisión que presentaba.

Después de la segunda generación de VPNs hubo un receso hasta que surgieron las tecnologías basadas en celdas, como Frame Relay y Asynchronous Transfer Mode (ATM). La tercera generación de VPNs tomó de base dichas tecnologías. Estas tecnologías crean un circuito virtual desde el nodo fuente y hasta el nodo destino.

A mediados de los 90s, con la llegada del e-commerce, los requerimientos de los usuarios y empresas cambiaron: se requería de soluciones que dieran fácil de implementar, escalables, administrables, globalmente accesible y capaces de proporcionar altos niveles de seguridad en todos los niveles de la comunicación. De esta forma nació la generación actual de VPNs, las VPNs basadas en IP (Internet Protocolo). Con esto las grandes y pequeñas empresas que no pueden

pagar soluciones caras de renta de líneas dedicadas pueden tener presencia a nivel mundial.

Tunneling

El "Tunneling" es la técnica de encapsular un paquete de datos en un protocolo de tunelado, como por ejemplo IPSec (IP Security), PPTP (Point-to-Point Tunneling Protocol) o L2TP (Layer 2 Tunneling Protocol, y finalmente empaquetar el paquete tunleado en un nuevo paquete de tipo IP (Internet Protocol). Este paquete resultante es encaminado a su red destino utilizando la información de IP. Dado que el paquete de datos original puede ser de cualquier tipo, el tunelado soporta tráfico multi-protocolo, incluyendo IP, ISDN, Frame Relay y ATM.

Protocolos de tunneling

Los principales protocolos para hacer tunneling son:

- **IP Security (IPSec)**: desarrollado por la Internet Engineering Task Force (IETF), IPSec es un estándar abierto que garantiza la seguridad de la transmisión y autenticación de usuarios sobre redes públicas. A diferencia de otras técnicas de cifrado, IPSec trabaja en la capa de red del modelo OSI (Open System Interconnect), por lo que puede ser implementado independientemente de las aplicaciones que corran en la red, lo que garantiza una red segura, sin importar las aplicaciones.

- **Protocolo de Tunneling Punto-a-Punto (PPTP):** desarrollado por Microsoft, 3COM y Ascend Communications, se propuso como una alternativa a IPSec. Trabaja en la capa 2 (enlace de datos) del modelo OSI. Se utiliza para transmisiones seguras de tráfico basado en Windows.

- **Layer 2 Tunneling Protocol (L2TP):** desarrollado por Cysco Systems, L2TP se creó con la intención de reemplazar a IPSec como el estándar de facto. L2TP es una combinación de Reenvío de capa 2 (Layer 2 Forwarding, L2F) y PPTP y es utilizado para encapsular tramas de tipo PPP (Point-to-Point Protocol) a través de redes X.25, Frame Relay y ATM.

Ventajas y desventajas del uso de VPNs

Las principales ventajas de utilizar VPNs son:

Reducir costos de implementación: los costos de implementación son mucho menores a una solución tradicional basada en líneas privadas rentadas, Frame Relay o ISDN, ya que la VPN elimina la necesidad de conexiones de larga distancia, utilizando conexiones locales, a un proveedor de servicios de Internet (Internet Server Provider, ISP) o un punto de presencia (Point of Presence, POP) de un ISP.

Incrementa la conectividad: las VPNs utilizan Internet para comunicarse con la Intranet por lo que el acceso puede llevarse a cabo prácticamente desde cualquier lugar que tenga acceso a Internet.

Seguridad en las transacciones: las VPNs utilizan tecnología de tunelado para transmitir información a través de redes inseguras, lo que asegura la confiabilidad de la información. Pero además del tunelado utiliza otras medidas de seguridad como el cifrado, autenticación y autorización.

Mejor uso del ancho de banda: el crear un túnel para la transmisión de información cuando es requerida hace que el ancho de banda sea utilizado solo cuando hay una conexión activa hacia Internet, lo que permite su mejor aprovechamiento.

Incrementa la escalabilidad: dado que las VPNs están basadas en Internet, permite a intranets corporativas un rápido crecimiento y adaptación cuando los negocios necesitan de cambiar con mínimo de equipamiento extra. Esto hace a la intranet basada en VPN escalable y adaptable a futuros crecimientos sin preocupar demasiado en la organización de la red.

Desventajas del uso de una VPN

Dependencia de Internet: El rendimiento de la red basada en VPN es dependiente del rendimiento de Internet. Líneas rentadas garantizan un ancho de banda, pero no es posible garantizar el rendimiento de Internet. Una sobrecarga de Internet puede afectar negativamente el rendimiento de toda la VPN.

Carencia de soporte de estándares de protocolos para la VPN: Actualmente las VPN están basadas en tecnología IP y aun muchas compañías descontinúan utilizando mainframes y dispositivos que utilizan sus protocolos nativos que son incompatibles con IP

Consideraciones al crear una VPN

Algunos de los puntos a considerar al implementar una VPN son:

***Seguridad*:** dado la naturaleza de los datos que viajarán por la red insegura, es necesario contar con mecanismos que garanticen la seguridad de la información.

Deben tomarse medidas para asegurar que los datos no puedan ser interceptados, escuchados, o dañados mientras se están enviando. Se deben contar con mecanismos para cifrar los datos. Otro aspecto importante es que la solución elegida como VPN debe ser compatible con la infraestructura de red, como lo es firewall, antivirus, proxy, actual de la empresa.

***Interoperatibilidad de dispositivos desde varios proveedores*:** se recomienda que todo el equipo para la VPN sea del mismo proveedor para evitar incompatibilidades y asegurar un mejor rendimiento.

***Administración centralizada de la VPN*:** esto permite configurar, administrar y resolver problemas relacionados con la VPN. Es importante que la administración genere siempre una bitácora, ya que ayuda a encontrar problemas.

***Facilidad de implementación*:** la solución basada en VPN debe se fácil de implementar y configurar, si se está implementando una solución a gran escala, es necesario asegurar que el software de administración sea capaz de grabar y mantener la pista de un gran número de túneles que el sistema implementa.

***Facilidad de Uso*:** el software de la VPN debe ser simple y sin complicaciones al momento de instalar y configurar, sobre todo en los clientes, aún para usuarios finales con poca o nula experiencia.

***Escalabilidad*:** la VPN debe ser capaz de adaptarse a futuras demandas con mínimos cambios a la infraestructura actual.

***Rendimiento*:** el cifrado es muy importante en la VPN, pero también es una operación que utiliza mucho tiempo de procesador, es por ello que deben seleccionarse dispositivos que puedan llevar a cabo tareas, como el cifrado, de manera rápida y eficiente.

***Administración del ancho de banda*:** para asegurar un alto rendimiento, alta disponibilidad y garantizar una calidad de servicio (QoS) es esencial manejar el ancho de banda de manera eficiente. Es importante administrar el ancho de banda de usuarios, grupo y aplicaciones de manera que sea posible asignar prioridades a los usuarios, grupos y aplicaciones de acuerdo a las políticas de la compañía.

Elección de un ISP: el proveedor de servicios de Internet debe ser confiable y debe ser capaz de proveer soporte para usuarios y administradores de VPN. Es muy importante conocer los tipos de servicios y localidades donde se ofrecen.

***Proteger la red de datos no solicitados*:** el funcionamiento de la VPN puede ser entorpecida por los datos de la intranet que no necesitan pasar por la VPN, por lo

que los túnels deben de ser capaces de proveer mecanismos para filtrar el tráfico no-VPN.

Tipos de VPNs

Existen tres tipos de VPNs: VPNs de acceso remoto, VPNs de Intranet y VPN de Extranet.

VPN de acceso remoto:

Permiten acceso remoto a través de móviles y equipo de teleconmutación de empleados de una organización a los recursos de la red corporativa. Por lo general este tipo de acceso es requerido rpo usuarios que constantemente se están moviendo de un lugar a otro. La figura 2 muestra una configuración sin configuración de acceso remoto.

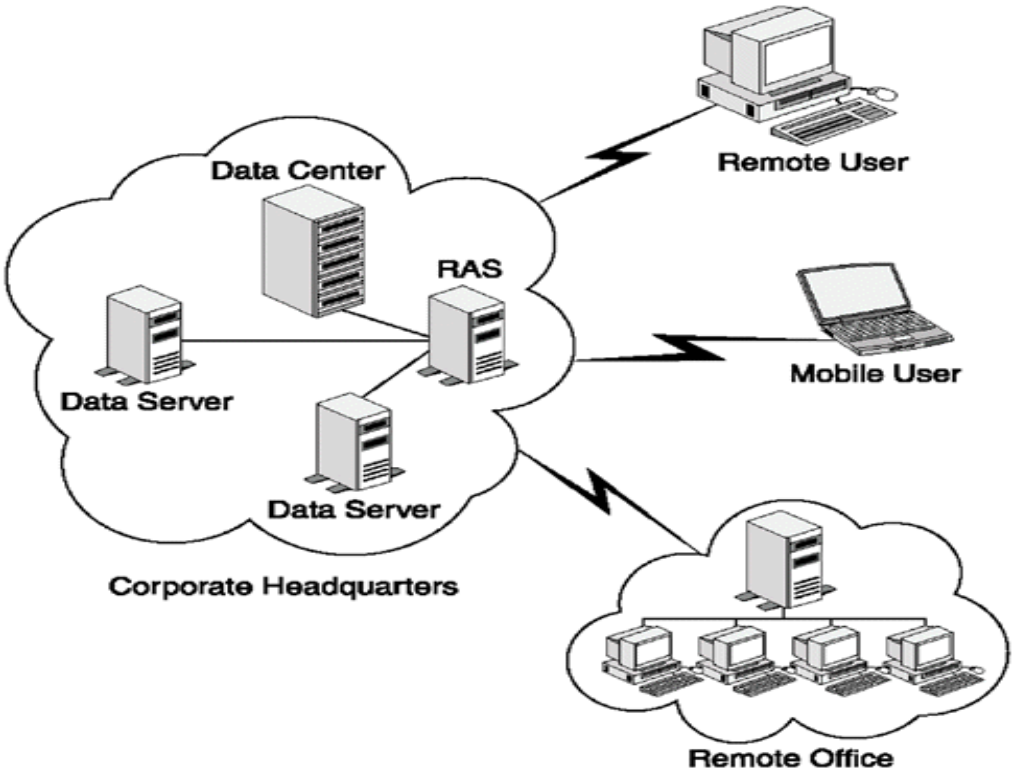


Figura 2. VPN sin configuración de acceso remoto

El Servidor de Acceso Remoto (RAS) está localizado en el site central para autenticar y autorizar usuarios remotos.

La conexión Dial-up permite la conexión al site central.

Se necesita personal de soporte para configurar, mantener y administrar el RAS para acceso remoto.

Si se utiliza una VPN con soporte para usuarios remoto y oficinas, solo es necesario tener una conexión al ISQP o hacías los ISPs y conectar a la red corporativa a través de Internet como muestra la figura 3:

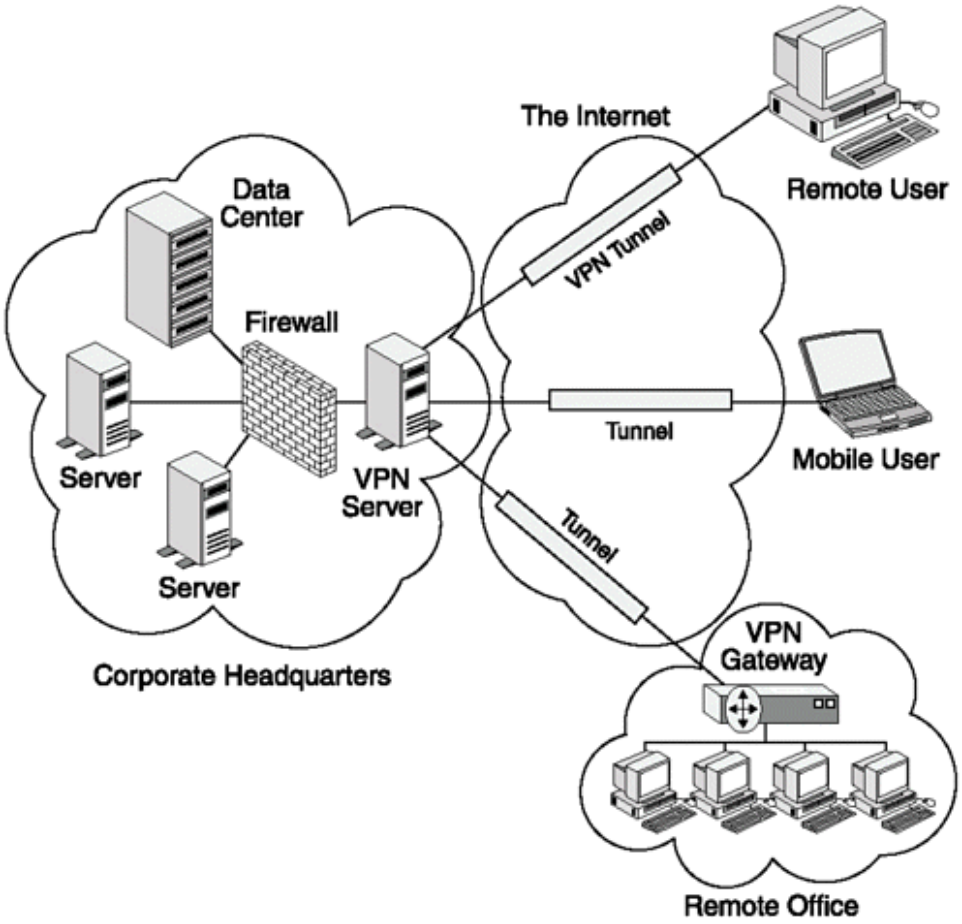


Figura 3. Configuración vía acceso remoto

Las principales ventajas del acceso remoto son:

- No es necesario tener un RAS un conjunto de modems.
- La necesidad de personal de soporte ya no es necesaria y las conexiones de larga distancia con reemplazadas por conexiones dial-up locales.

- Ahorro significativo en servicio de larga distancia.
- Dado que el acceso es local, las velocidades de transferencia son más rápidas
- Las VPNs proveen mejor accesibilidad a los sitios corporativos porque soportan un nivel mínimo de acceso a servicios, a pesar de un incremento en el número de conexiones simultáneas a la red.

Desventajas del acceso remoto en la VPN son:

- El acceso remoto a la VPN no garantiza la calidad de servicio.
- La posibilidad de perder los datos es muy alta.
- La sobrecarga de los protocolos se incrementa considerablemente el trabajo de la red sobre todo por los algoritmos de cifrado.
- Las transmisiones multimedia pueden ser muy lentas.

VPN de Intranet

Se utilizan para interconectar oficinas remotas de una organización a la red corporativa. En la configuración de una intranet sin utilizar tecnología VPN cada sitio remoto debe ser conectado a la intranet corporativa utilizando un router del campus, como se muestra en la figura 4

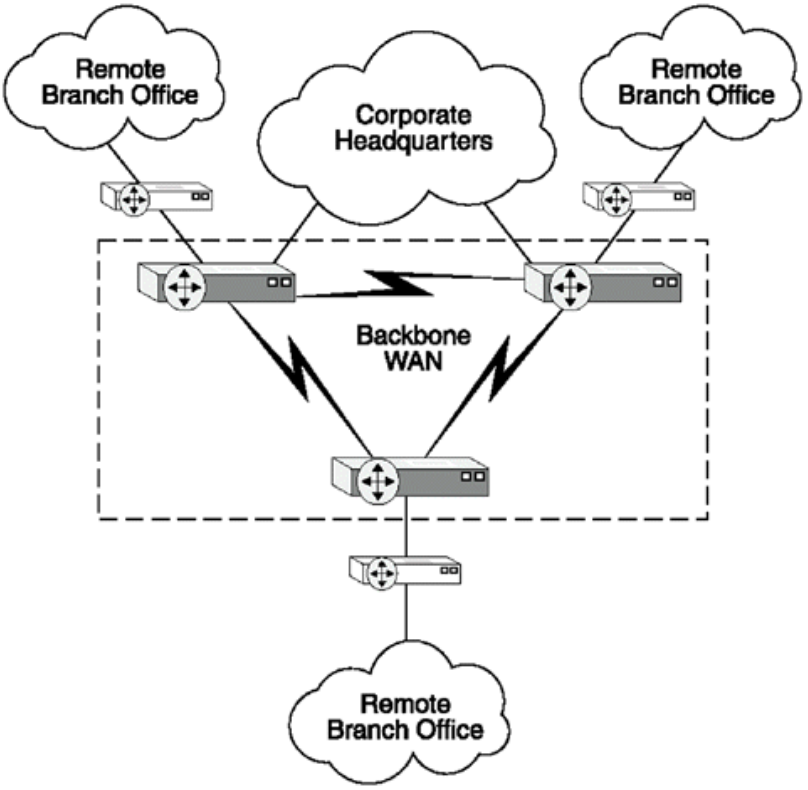


Figura 4. Acceso a la Intranet sin VPN

La configuración de la figura 4 es muy cara ya que menos dos routers son requeridos para conectar un campus u oficina remota a la intranet de la organización. Por lo que la implementación, mantenimiento y administración del backbone de la intranet puede ser muy caros, dependiendo del volumen de tráfico, por lo que si crece el número de clientes mayores será el tráfico y mayor será el costo.

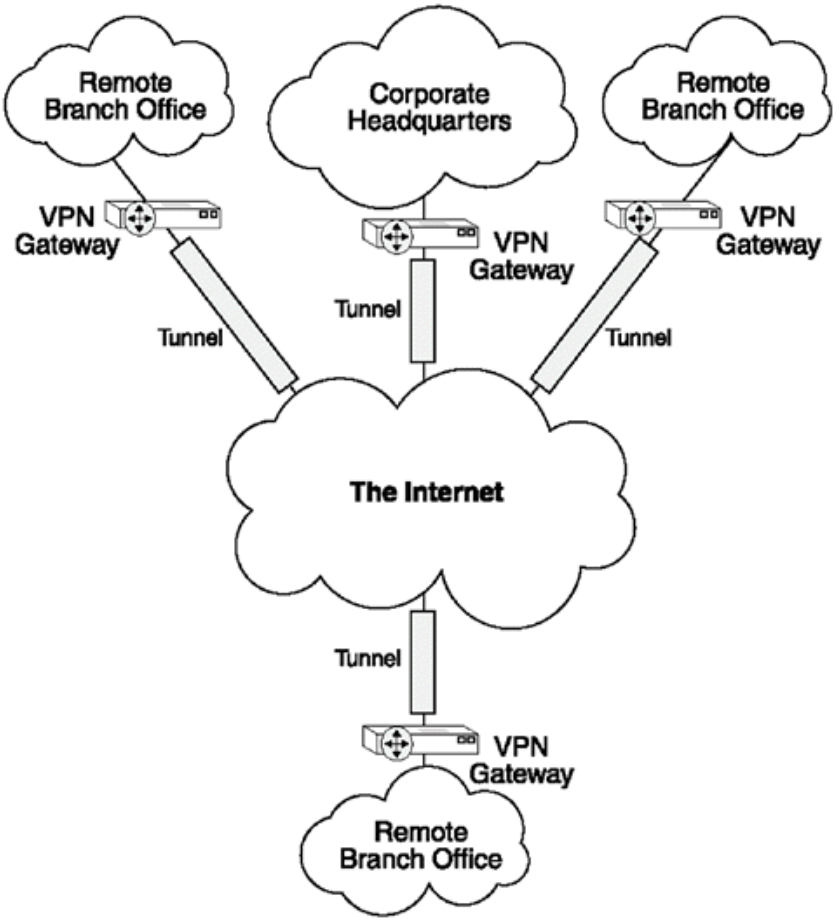


Figura 5. Configuración con VPN

Las principales ventajas de una VPN para la Intranet son:

- Un menor costo dado que se elimina el uso de routers.
- Se reduce el número de personal de soporte requerido.
- Dado que Internet actúa como el medio de conexión, es más sencillo agregar nuevas conexiones punto a punto.
- Mayor facilidad para realizar respaldos utilizando los tunnels de la VPN.
- La conectividad vía dial-up es más rápida y se eliminan las llamadas de larga distancia.

Algunas de las desventajas de una VPN de Intranet son:

- Es muy sencillo sufrir un ataque de negación de servicio, dado que los túnels están sobre Internet.
- La posibilidad de pérdida de paquetes en tránsito es alta.
- La latencia al transmitir contenidos multimedia puede ser muy alta dado el rendimiento de Internet.
- La calidad de servicio no puede se garantizada.

VPN de Extranet

A diferencia de una solución basada en una VPN remota, las VPNs de extranet no se segregan completamente desde el "mundo exterior", es decir, la VPN de extranet controla el acceso a recursos necesarios de la red de las entidades de negocios externas, como socios comerciales, clientes y proveedores y que juegan un papel en la organización del negocio. Esta solución se presenta en la figura 6.

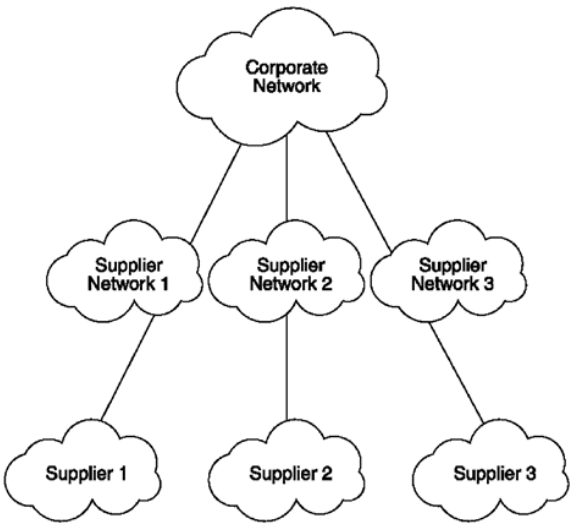


Figura 6. Escenario de una VPN de Extranet

Puede verse en la figura 6 que la instalación tradicional es muy cara, porque cada red separada en la intranet debe ser adaptada de acuerdo a las redes externas, lo que da como resultado redes complejas de implementar y administrar. También es necesario contar con personal calificado para mantener y administrar la red. Es muy complicado extender este tipo de redes.

La solución para la VPN de extranet se muestra en la figura 7.

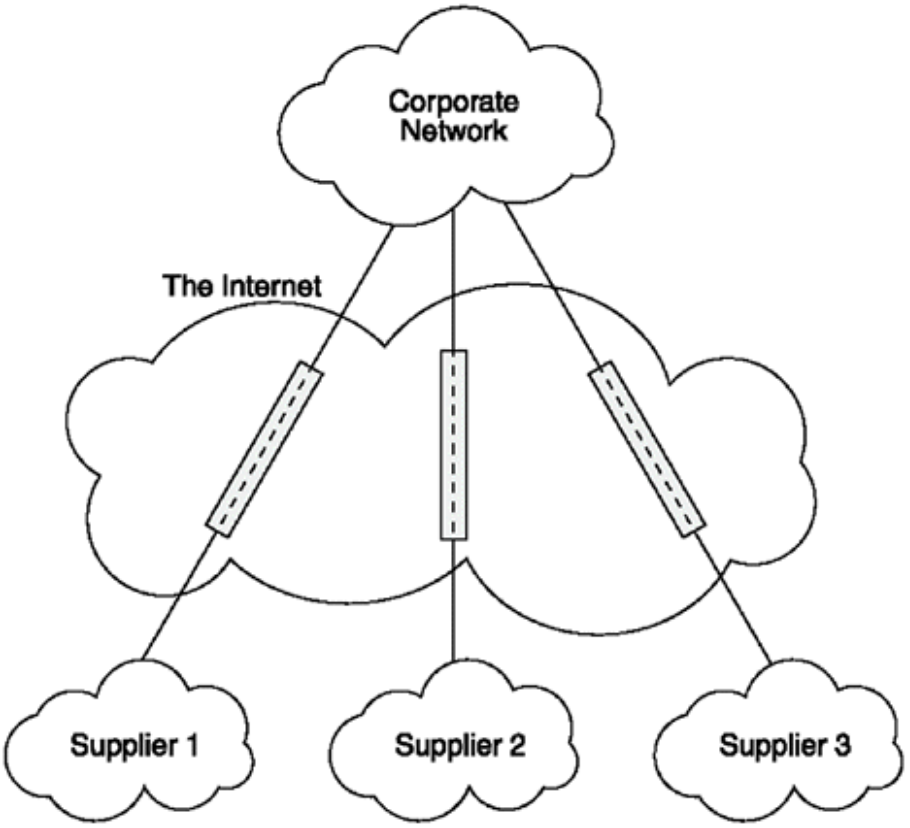


Figura 7. Implementación de la VPN para una Extranet

Las principales ventajas de la solución de la figura 7 son:

- Costos divididos a diferencia de una instalación tradicional.
- Facilidad de implementación, mantenimiento y modificaciones.
- Mayores opciones al elegir un proveedor dado que la base es Internet.
- Menores costos de instalación y operación.
- Algunas de las desventajas de este modelo son:
- Amenazas de seguridad como negación de servicio.
- Se incrementa el riesgo de penetración hacía la intranet de la organización.

- Dado que la base es Internet se genera una alta latencia al transferir contenidos multimedios.

- La calidad de servicio no puede ser garantizada.

Requerimientos para una VPN

Es posible ver a una VPN como una red privada de datos que permite conectarse con otras redes públicas o privadas de una manera segura y económica. Es por ello que muchos de los requerimientos de una VPN son similares a los de una red privada tradicional. Los puntos más importantes de la VPN, a considerar son:

- Seguridad.
- Disponibilidad.
- Calidad de Servicio (QoS).
- Confiabilidad.
- Compatibilidad.
- Flexibilidad.

Seguridad

Una VPN debe ofrecer una medio ambiente con una seguridad muy alta, dado que los recursos de la red no son accesibles al público en general, por lo que la posibilidad de acceso no autorizado a los recursos de la intranet deben se extremadamente bajos. Conseguir lo anterior es muy complicado ya que la VPN utiliza redes públicas, como lo es Internet, para la comunicación, por lo que la VPN debe contar con medidas avanzadas de seguridad.

Algunas formas para proteger la red son:

- Implementación de mecanismos periféricos de defensa que permitan acceso solo a tráfico autorizado proveniente de redes confiables y bloquen el demás tráfico. Firewalls y Network Address Tranlation (NAT) son ejemplos de mecanismos que pueden implementarse en el punto donde la red privada o intranet se conecta a la red pública. Los firewalls revisan tanto los paquetes entrantes como los salientes. Nat, por otra parte, no revela las direcciones reales de los recursos de la red interna.
- Implementar autenticación de usuarios y paquetes para establecer la identifiación de los usuarios y determinar si se permite el acceso o no. El modelo de Autenticación y Autorización de Cuentas (AAA, Autentication Authorization Accounting) es un ejemplo de un sistema compresivo para la autenticación de usuarios. Después de que el usuario se ha autenticado correctamente, podrá acceder solo a los recursos asignados para el. Debe mantenerse una bitácora detallada de todas las actividades que llevan a

cabo los usuarios en la red, lo que permite monitorear las actividades no autorizadas.

- Implementar mecanismo de cifrado de datos para asegurar la autenticidad, integridad y confidencialidad de los datos mientras esto son transmitidos a través de redes inseguras. IPSec es uno de los protocolos más poderosos que existen, ya que no tan solo encripta datos, si no que también habilita autenticación de cada usuario y cada paquete de manera individual.

- Los métodos de seguridad deben elegirse con cuidado, ya que no tan solo deben de ser fáciles de implementar y administrar, si no que también registrar cualquier intento de violación de usuarios internos. El proceso de ingreso del usuario debe ser rápido y fácil para ellos y no presentar mayores problemas a los usuarios.

Disponibilidad y confiabilidad

La disponibilidad se refiere al tiempo total que la red esta en funcionamiento de una manera adecuada. En una red privada y una intranet, este tiempo debe ser alto ya que la infraestructura completa y su control pertenece a la misma organización. En la VPN la instalación debe considerar que se utilizan redes intermedias para acceder, por lo que hay una gran dependencia de las interredes, por lo que la disponibilidad depende en gran medida del Proveedor de Servicios de Internet (Internet Server Provider, ISP).

Por lo general un ISPasegura la disponibilidad mediante un acuerdo de nivel de servicio (Service Level Agreement, SLA), este es un documento escrito que garantiza el acceso a la red. Por un precio alto un ISP puede asegurar un porcentaje de hasta un 99 por ciento.

Si se busca un servicio de alta disponibilidad es necesario buscar un proveedor que ofrezca una infraestructura de backbone con switcheo altamente resistente, lo cual debe incluir:

- Capacidad de ruteo que permita reencaminar el tráfico a través de un camino alternativo en caso de que la ruta principal falle o esté congestionada.

- Redundancia de líneas de acceso, las cuales pueden ser utilizadas para incrementar el ancho de banda.

- Infraestructura redundante que no tan solo soporte dispositivos de tipo hot-swap si no también sistemas redundantes de poder y enfriamiento.

La confiabilidad es otro de los requerimientos más importantes de la VPN. La confiabilidad de las transacciones en la VPN asegura el envío de datos extremo a extremo en todas las transmisiones. Si una ruta falla es necesario que los paquetes sean enviados por diferentes rutas si alguna ruta falla. Dicho proceso debe ser transparente para el usuario final y puede ser implementado por medio de ligas redundantes de hardware.

Calidad de Servicio

La calidad de servicio (Quality of Service, QoS) es la capacidad de una red de responder a situaciones críticas asignando un mayor porcentaje del ancho de banda de una red y recursos a una aplicación de misión crítica y aplicaciones sensibles a retraso de datos. Una aplicación como una transacción financiera y proceso de una orden de venta son más importantes, desde el punto de vista del negocio, que navegar en Internet. De igual forma aplicaciones como videoconferencias requieren de un ancho de banda mayor que leer un correo electrónico. Es responsabilidad del QoS asignar el ancho de banda suficiente a esas aplicaciones.

La QoS abarca la latencia y el rendimiento de procesamiento. La latencia es el retardo de la comunicación y es extremadamente importante en aplicaciones de audio y vídeo. El rendimiento de procesamiento se refiere a la disponibilidad del apropiado ancho de banda de todas las aplicaciones, especialmente aplicaciones de misión crítica y aplicaciones de ancho de banda intensiva.

Al igual que la disponibilidad, la QoS es dependiente de la SLA. Con la ayuda de un SLA un ISP garantiza cierto nivel de latencia y rendimiento de procesamiento al cliente. Dependiendo del nivel de latencia y rendimiento de procesamiento la QoS puede dividirse en tres categorías:

- QoS de mejor esfuerzo: en este servicio se indica la ausencia de QoS ya que el ISP no la puede garantizar. Este es el servicio menos caro y no debe ser utilizado para aplicaciones que necesitan ancho de banda intensiva.

- QoS relativa: Permite establecer prioridades al tráfico de datos, con lo que el rendimiento de procesamiento está asegurado, pero depende de la carga de la red y el porcentaje de tráfico que necesite ser prioritizado.

- QoS Absoluto: Garantiza latencia y rendimiento de procesamiento, es el servicio más caro.

Flexibilidad

El control completo de los recursos de la red y sus operaciones, así como su correcta administración es uno de los aspectos más importantes de toda

organización. La mayoría de las compañías utilizan ISP como intermediarios para conectarse a Internet y la compañía misma administra su propia intranet lo que da como resultado, en la mayoría de los casos, en incompatibilidades en sus configuraciones.

Con la disponibilidad de los dispositivos actuales de la VPN y con acuerdos entre el ISP y la organización es posible eliminar las barreras de la administración de recursos y así administrar la red completa (la intranet y el acceso a Internet) como parte de la VPN

7.2 SSH, SSH2

SSH (Secure Shell) es un programa que permite conectarse a otra computadora, a través de una red, para ejecutar comandos en la máquina remota y para copiar archivos entre ambos equipos. SSH provee "autenticación fuerte" y segura sobre canales inseguros. SSH es el sustituto de telnet, rlogin, rsh y rcp ya que estos programas llevan a cabo una conexión en texto plano, es decir, fácilmente con un sniffer es posible ver la información que intercambian ambos host.

SSH está basado en criptografía de llave pública. Las versiones anteriores a la 2.0 utilizan autenticación basada en RSA, mientras que versiones posteriores a 2.0 utilizan DSA.

Por lo general SSH viene instalado en la mayoría de las distribuciones GNU/Linux actuales, tanto el programa para el cliente como el programa para realizar el servidor de SSH.

De una manera rápida es posible conectarse a un host remoto (que debe estar corriendo un servidor de SSH) solo dando el comando, ssh usuario@máquina_remota, por ejemplo:

gerardo@puntog:~$ssh gerardo@www.miservidor.com

La primera vez que se conecta al servidor remoto se presenta el "fingerprint" o "huella digital" del host remoto, esta información nos sirve para saber que realmente se está conectando al host que le indicamos (por lo general debemos conocer el fingerprint del host remoto). Es necesario indicar "Yes" para que el host remoto pida la contraseña y con ello ingresar al sistema. Si se conoce la huella y no coincide con la que se mostró al conectarse, hay que contestar "no" para no llevar a cabo la conexión.

Para conectarse es necesario que SSH se encuentre instalado en ambos equipos. Para mostrar los pasos necesarios para conectarse se mostrará un sencillo ejemplo, la máquina local se llama "puntog" y el usuario en dicha máquina es "gato"; la máquina remota es "www.servidor.com" y el usuario remoto es "gerardoc".

El primer paso es crear el par de llaves (pública y privada) para el sistema local, esto se hace con el comando ssh-keygen –d

Ejemplo:

gato@puntog:~$ ssh-keygen –d
Generating public/private dsa key pair.
Enter file in which to save the key (/home/gato/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gato/.ssh/id_dsa.
Your public key has been saved in /home/gato/.ssh/id_dsa.pub.
The key fingerprint is: 17:6f:91:5c:ae:ee:6b:8d:7c:18:2b:06:15:00:ab:75
gato@puntog gato@puntog:~$

Lo primero que se pide es el nombre donde se almacenará la llave privada a crear, por default se crea en el directorio hogar/.ssh/ con el nombre id _dsa. A continuación se pide una frase (passphrase), esta frase es utilizada por el algoritmo 3DES para encriptar la parte privada de la red. Para el ejemplo no se escribe ninguna frase. Las llaves han sido creadas, en el directorio hogar/.ssh hay dos archivos: id_dsa e id_dsa.pub, el primero almacena la llave privada y el segundo la llave pública, si se desea puede verse el contenido de los archivos ya que se encuentran en texto plano.

Ejemplo:

```
gato@puntog:~$ cd .ssh/
gato@puntog:~/.ssh$ ls id_dsa id_dsa.pub
gato@puntog:~/.ssh$ more id_dsa
-----BEGIN DSA PRIVATE KEY-----
MIIBugIBAAKBgQCnBx4cWt8F9gRC3FMkUcWB/+MwjYZ6ihWpJ3S4vCVjHrDSJlGj
6UvM2W2IvrrIvRqYBxFgEdqllyEwxWm3Mx5u+e276xb5h0v346osolWrVL5ErSxc
TKIkKCL/rF/Ht123Foqes4mMQLNftnII9tisEP4DYR9Z4fyQy2Zw94e1dwIVAK8s
TF20Bbmhc/xfT8KAtJKx2qKrAoGAH9KRNXpWNT5WwpBWBkM7Ok5tMHmucBH6UcUQ
OmzqINKoDCSXrXgeAX98oH2sdz/WhEx4+bY92sZjU6h2953N7iJwWvVWiJf1FIot
XuLT1TUkouArfTvCZS4ukWzcUNGfHaYa62OFXxVItVxCPTXAxl/GiZ9T5g8x/av5
fFlag8ICgYBgmHbKIzBW6Mb+8nX7hi+98b580KVj+MoMnh1o3l/pzF6H67OnpULk
o85OlZmq7uT2DU+7adWYyfHNg63qd4MYExlPpPTtJyur+rd4MM5lL/6tUi4UDkL6
rGXBuiPP2fCbHuuAnmKoU4lB2un40AiQou21JOuQhZCKHQBY+q3VsAIURl8KWiZ5
Q7HXRgOJKoEK2QgMbpc=
-----END DSA PRIVATE KEY-----
gato@puntog:~/.ssh$ more id_dsa.pub ssh-dss
AAAAB3NzaC1kc3MAAACBAKcHHhxa3wX2BELcUyRRxYH/4zCNhnqKFakndLi8JWMesNIkgaPpS8zZb
Yi+usi9GpgHEWAR2qWXITDFabczHm757bvrFvmHS/fjqiyiVatUvkStLFxMoiQoIv+sX8e3XbcWip
6ziYxAs1+2cgj22KwQ/gNhH1nh/JDLZnD3h7V3AAAAFQCvLExdtAW5oXP8X0/CgLSSsdqiqwAAAIA
f0pE1elY1PlbCkFYGQzs6Tm0wea5wEfpRxRA6bOog0qgMJJeteB4Bf3ygfax3P9aETHj5tj3axmNT
qHb3nc3uInBa9VaII/UUii1e4tPVNSSi4Ct9O8JlLi6RbNxQ0Z8dphrrY4VfFUi1XEI9NcDGX8aJn
1PmDzH9q/l8WVqDwgAAAIBgmHbKIzBW6Mb+8nX7hi+98b580KVj+MoMnh1o3l/pzF6H67OnpULko8
5OlZmq7uT2DU+7adWYyfHNg63qd4MYExlPpPTtJyur+rd4MM5lL/6tUi4UDkL6rGXBuiPP2fCbHuu
AnmKoU4lB2un40AiQou21JOuQhZCKHQBY+q3VsA== gato@puntog
gato@puntog:~/.ssh$
```

Ahora en el host remoto es necesario crear el directorio .ssh dentro del directorio hogar y copiar el archivo id_dsa.pub con el nombre authorized_keys2 (si se utiliza una versión anterior a la 2.0 el archivo debe llamarse authorized_keys) y listo, ahora desde la máquina cliente tan solo con dar ssh gerardoc@servidor.com se establece una conexión segura con el host remoto y sin la necesidad de proporcionar password.

Ejemplo:

```
gato@puntog:~/.ssh$ ssh gerardoc@servidor.com
Linux sunserv 2.2.19 #1
Sat Jun 9 13:04:06 EST 2001 i686 unknown
Most of the programs included with the Debian GNU/Linux system
are freely redistributable; the exact distribution terms for
each program are described in the individual files in
/usr/share/doc/*/copyrigh Debian GNU/Linux comes with ABSOLUTELY NO
WARRANTY, to the extent permitted by applicable law.
No mail.
Last login: Mon Aug 9 10:22:07 2004
from dup-148-221-112-3.prodigy.net.mx
gerardoc@sunserv:~$
```

Copia de archivos SSH también pe rmite copiar archivos, para ello utilizamos el comando scp (Secure Copy), si se desea copiar un archivo del host local al remote la sintáxis es:

scp nombre_archivo _local usuario@nombre_servidor:directorio_remoto

Por ejemplo, se quiere copiar el archivo local index.html a la máquina www.servidor.com, al directorio /home/gerardoc/public_html, como el usuario gerardoc (remoto):

```
gato@puntog:~$scp index.html
gerardoc@www.servidor.com:/home/gerardoc/public_html
```

Para copiar del host remoto al local la sintáxis es:

scp usuario@nombre_equipo_remoto:directorio/archivo directo rio_local

Ejemplo, copiar todo el contenido del directorio remoto /home/gerardoc/tareas/ al directorio local /home/gato/:

```
gato@puntog:~$scp -r
gerardoc@www.miservidor.com:/home/gerardoc/tareas \ /home/gato
```

Para ejecutar un comando de manera remota solo es necesario incluir el comando al final de la línea y este se ejecuta en el host remoto, por ejemplo, se quiere ver el contenido del directorio hogar del host remoto www.servidor.com:

gato@puntog:~$ssh gerardoc@www.servidor.com ls

SSH también permite hacer seguro un protocolo inseguro (encapsula un protcolo en otro, también conocido como tunnel), esto se realiza con:

ssh -L puerto_local:máquina_remota:puerto_remoto cuenta@máquina_remota

Por ejemplo, quiero leer mi correo electrónico de la máquina www.servidor.com, utilizando POP3 (POP3 no es protocolo seguro) lo que se hace es crear un tunnel de mi máquina local (yo asigno el puerto, en este caso utilizo el puerto 1100, ya que si se desea utilizar puerto menores al 1024 es necesario ser el usuario root) a la máquina remota (el puerto del pop3 es el 110):

```
gato@puntog:~$ ssh -L 1100:www.servidor.com:110 \
gerardoc@www.servidor.com
```

Ahora lo único que se debe hacer es configurar mi cliente de correo favorito e indicarle que se conecte al puerto local 1100, también se puede probar haciendo un telnet a 127.0.0.1 al puerto 1100 (y dar comando de POP3).

Existe un cliente de ssh para windows que también permite hacer tunnel el que utilizo es el putty (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)

7.3 PPTP

El PPTP es un protocolo de red que permite el tráfico seguro de datos desde un cliente remoto a un servidor corporativo privado, estableciéndose así una Red Privada Virtual (VTN) basada en TCP/IP. PPTP soporta múltiples protocolos de red (IP, IPX y NetBEUI) y puede ser utilizado para establecer dichas redes virtuales a través de otras redes públicas o privadas como líneas telefónicas, redes de área local o extensa (LAN's y WAN's) e Internet u otras redes públicas basadas en TCP/IP.

Una red privada virtual consiste en dos máquinas (una en cada "extremo" de la conexión) y una ruta o "túnel" que se crea dinámicamente en una red pública o privada. Para asegurar la privacidad de esta conexión los datos transmitidos entre ambos ordenadores son encriptados por el Point-to-Point protocol (PPP), un protocolo de acceso remoto, y posteriormente enrutados o encaminados sobre una conexión previa (también remota, LAN o WAN) por un dispositivo PPTP. En

el entorno Windows 95 o NT 4.0, este dispositivo se conoce como adaptador de red privada virtual.

El escenario más común es el siguiente: Una máquina cliente (Windows 95 o NT 4.0) se conecta a un Proveedor de Servicios de Internet (ISP) utilizando una conexión de acceso telefónico a redes. En otro punto de Internet existe una máquina (NT 4.0 Server) con el servicio de servidor de acceso remoto (RAS) conectado a ella, bien directa y permanentemente, mediante un adaptador de Red o también mediante una conexión de acceso telefónico a redes, a otro ISP que no tiene porqué coincidir con el primero. El cliente puede en ese momento lanzar otra conexión de acceso remoto, usando su adaptador de red privada virtual, a dicho servidor, creándose un túnel privado sobre Internet, que conecta ambas máquinas como si estuvieran en la misma red local, pudiendo así tener acceso a recursos compartidos tales como carpetas o impresoras.

Este escenario puede variar según el tipo de conexión que tengan tanto el cliente como el servidor. Organizaciones con acceso permanente a Internet, pueden configurar servidores de acceso remoto para que soporten PPTP. Esto permite que colaboradores en cualquier parte del mundo puedan conectarse a ellos, usando sus accesos a Internet habituales. Así, será posible participar de los recursos de la red corporativa, con seguridad garantizada, y sin los costes habituales de las llamadas de larga distancia a estos servidores de acceso remoto.

La máquina configurada como PPTP Server debe tener la configuración mínima requerida para correr Windows NT 4.0 Server. Además, debe tener dos adaptadores de red (NIC, módem, RDSI, X25), uno conectado a la red local LAN, y otro a Internet.

Una de las ventajas más palpables del PPTP es que reduce o elimina la necesidad del uso de sofisticados y caros equipos de telecomunicaciones para permitir las conexiones de equipos portátiles y remotos. PPTP puede usar redes telefónicas normales de forma totalmente segura.

En el cliente

El cliente puede ser una máquina con Windows NT 4.0, tanto Server como Workstation, o Windows 95. En ambos casos, se requiere de un módem o tarjeta RDSI, además de un equipo de conexión a una red telefónica (plaqueta en pared, teléfono móvil que soporte este tipo de conexiones etc.). Por otro lado, si el

cliente está accediendo al servidor PPTP a través de una red de área local (LAN), se precisa de un adaptador de red (NIC) que lo conecte físicamente a ella.

PPTP permite el uso de redes privadas virtuales sobre redes TCP/IP ya existentes, como Internet, pero preservando el uso de los protocolos de red, direcciones de nodo y nombres de máquinas ya existentes en la red privada. Por tanto, no se requiere el uso de nuevos protocolos ni cambios en las aplicaciones de red. Mediante el "túnel privado" que se establece a través de la red pública TCP/IP, pueden usarse, por ejemplo, los protocolos IPX y NetBEUI usados en la red privada para la ejecución de las aplicaciones de red.

Por otro lado, los métodos de resolución de nombres de la red privada (WINS, DNS, SAP) no necesitan modificarse. Además, en el caso de las direcciones IP, pueden usarse para la red privada direcciones no válidas en Internet. Por ejemplo:

Una organización puede usar internamente un conjunto de direcciones IP de clase B, sin preocuparse de que dichas direcciones ya estén siendo utilizadas por otra en Internet. Sin embargo, tanto el servidor PPTP como el cliente tienen que tener asignadas direcciones IP válidas en Internet antes de poder establecer el "túnel". En ese momento, cada uno tendrá dos direcciones IP, una valida en Internet y otra válida en la red privada. PPTP se encargará de "encapsular" los paquetes IP con destino a la red privada dentro de otro paquete que viajará por la red pública (Internet) hasta el servidor PPTP. Éste extraerá el paquete IP con destino a la red privada y se lo transmitirá.

La autentificación de usuarios se realiza a través de los protocolos existentes en el Servicio de Acceso Remoto de NT 4.0 Server (PAP y CHAP). Además, puede existir una seguridad adicional establecida por el Proveedor de Servicios de Internet o de la red pública TCP/IP. PPTP hace uso de la seguridad que da el protocolo PPP. La autentificación MS-CHAP se utiliza para validar las credenciales del usuario remoto contra dominios NT. Sólo los usuarios con permiso para ello pueden realizar la conexión. Una vez que se comprueba que el usuario tiene permiso para iniciar una sesión, se genera una "llave" de 40 bits a partir de la clave del usuario (que en el entorno Microsoft SIEMPRE viaja ya encriptada por la red) que es utilizada para encriptar a su vez los datos del usuario (encriptación RC-4).

Por otro lado, la implementación de Microsoft del protocolo CCP (Compression Control Protocol) tiene un bit que se usa para negociar la encriptación. Los clientes de RAS pueden pedir ser validados solamente con la encriptación habilitada. Por su parte, los servidores de RAS pueden configurarse a su vez para exigir que los clientes usen encriptación.

Además, puede defenderse a la red privada de un uso malintencionado

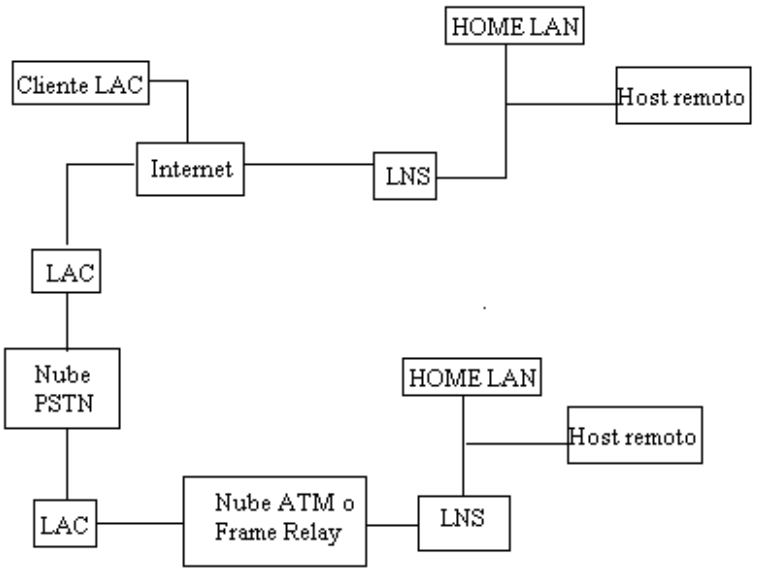
habilitando el filtrado PPTP del servidor. En este caso, el servidor solamente acepta y enruta paquetes enviados por usuarios validados. Esto evita que cualquier otro tipo de paquete ajeno pueda ser introducido en la red privada.

PPTP complementa el uso de firewalls y cubre la necesidad de otro tipo diferente de seguridad. Los firewalls aseguran la red corporativa privada regulando estrictamente los datos que llegan desde Internet. PPTP asegura la privacidad de los datos intercambiados entre cliente y servidor a través de Internet.

El trafico PPTP usa el puerto TCP 1723, y el identificador IP 47. Por tanto, un firewall puede configurarse para habilitar el tráfico a través de ellos. Un servidor PPTP situado detrás de él aceptará los paquetes PPTP que éste le envíe, extraerá el paquete PPP del datagrama IP, descifrará el paquete, y lo enviará a la máquina destino de la red privada. Por tanto, se habrán combinado la seguridad del firewall en cuanto a la defensa de la red privada de paquetes ajenos a ella, y la del PPTP, en lo que respecta a la seguridad con la que los paquetes de datos relativos a la red privada que han sido enviados a través de la red pública TCP/IP (Internet).

7.4 L2TP

Layer-2 Tunneling Protocol (L2TP) facilita el entunelamiento de paquetes PPP a través de una red de manera tal que sea lo más transparente posible a los usuarios de ambos extremos del túnel y para las aplicaciones que éstos corran.
El escenario típico L2TP, cuyo objetivo es la creación de entunelar marcos PPP entre el sistema remoto o cliente LAC y un LNS ubicado en una LAN local, es el que se muestra en la siguiente figura:



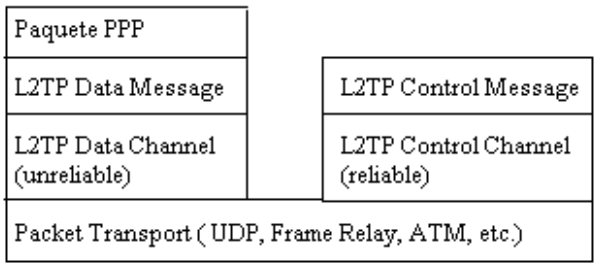
Un L2TP Access Concentrator (LAC) es un nodo que actúa como un extremo de un túnel L2TP y es el par de un LNS. Un LAC se sitúa entre un LNS y un sistema remoto y manda paquetes entre ambos. Los paquetes entre el LAC y el LNS son enviados a través del túnel L2TP y los paquetes entre el LAC y el sistema remoto es local o es una conexión PPP.

Un L2TP Network Server (LNS) actúa como el otro extremo de la conexión L2TP y es el otro par del LAC. El LNS es la terminación lógica de una sesión PPP que está siendo puesta en un túnel desde el sistema remoto por el LAC.
Un cliente LAC, una máquina que corre nativamente L2TP, puede participar también en el túnel, sin usar un LAC separado. En este caso, estará conectado directamente a Internet.

El direccionamiento, la autenticación, la autorización y el servicio de cuentas son proveídos por el Home LAN's Management Domain.

L2TP utiliza dos tipos de mensajes: de control y de datos. Los mensajes de control son usados para el establecimiento, el mantenimiento y el borrado de los túneles y las llamadas. Utilizan un canal de control confiable dentro de L2TP para garantizar el envío. Los mensajes de datos encapsulan los marcos PPP y son enviados a través del túnel.

La siguiente figura muestra la relación entre los marcos PPP y los mensajes de control a través de los canales de control y datos de L2TP.



Los marcos PPP son enviados a través de un canal de datos no confiable, encapsulado primero por un encabezado L2TP y luego por un transporte de paquetes como UDP, Frame Relay o ATM. Los mensajes de control son enviados a través de un canal de control L2TP confiable que transmite los paquetes sobre el mismo transporte de paquete.

Se requiere que haya números de secuencia en los paquetes de control, que son usados para proveer el envío confiable en el canal de control. Los mensajes de datos pueden usar los números de secuencia para reordenar paquetes y detectar paquetes perdidos.

Al correr sobre UDP/IP, L2TP utiliza el puerto 1701. El paquete entero de L2TP, incluyendo la parte de datos y el encabezado, viaja en un datagrama UDP. El que inicia un túnel L2TP toma un puerto UDP de origen que esté disponible, pudiendo ser o no el 1701 y envía a la dirección de destino sobre el puerto 1701. Este extremo toma un puerto libre, que puede ser o no el 1701, y envía la respuesta a la dirección de origen, sobre el mismo puerto iniciador. Luego de establecida la conexión, los puertos quedan estáticos por el resto de la vida del túnel.

En la autenticación de L2TP, tanto el LAC como el LNS comparten un secreto único. Cada extremo usa este mismo secreto al actuar tanto como autenticado como autenticador.

Sobre la seguridad del paquete L2TP, se requiere que el protocolo de transporte de L2TP tenga la posibilidad de brindar servicios de encriptación, autenticación e integridad para el paquete L2TP en su totalidad. Como tal, L2TP sólo se preocupa por la confidencialidad, autenticidad e integridad de los paquetes L2TP entre los puntos extremos del túnel, no entre los extremos físicos de la conexión.

7.5 IPSEC, FreeSWAN

IPSEC

IPSec trata de remediar algunas falencias de IP, tales como protección de los datos transferidos y garantía de que el emisor del paquete sea el que dice el paquete IP. Si bien estos servicios son distintos, IPSec da soporte a ambos de una manera uniforme.

IPSec provee confidencialidad, integridad, autenticidad y protección a repeticiones mediante dos protocolos, que son Authentication Protocol (AH) y Encapsulated Security Payload (ESP).

Por confidencialidad se entiende que los datos transferidos sean sólo entendidos por los participantes de la sesión.

Por integridad se entiende que los datos no sean modificados en el trayecto de la comunicación.

Por autenticidad se entiende por la validación de remitente de los datos.

Por protección a repeticiones se entiende que una sesión no pueda ser grabada y repetida salvo que se tenga autorización para hacerlo.

AH provee autenticación, integridad y protección a repeticiones pero no así confidencialidad. La diferencia más importante con ESP es que AH protege partes del header IP, como las direcciones de origen y destino.

ESP provee autenticación, integridad, protección a repeticiones y confidencialidad de los datos, protegiendo el paquete entero que sigue al header.

AH sigue al header IP y contiene diseminaciones criptográficas tanto en los datos como en la información de identificación. Las diseminaciones pueden también cubrir las partes invariantes del header IP.

El header de ESP permite rescribir la carga en una forma encriptada. Como no considera los campos del header IP, no garantiza nada sobre el mismo, sólo la carga.

Una división de la funcionalidad de IPSec es aplicada dependiendo de dónde se realiza la encapsulación de los datos, si es la fuente original o un gateway:

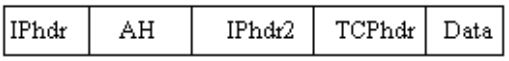
- El modo de transporte es utilizado por el host que genera los paquetes. En este modo, los headers de seguridad son antepuestos a los de la capa de transporte, antes de que el header IP sea incorporado al paquete. En otras

palabras, AH cubre el header TCP y algunos campos IP, mientras que ESP cubre la encriptación del header TCP y los datos, pero no incluye ningún campo del header IP.

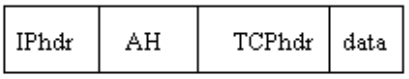
- El modo de túnel es usado cuando el header IP entre extremos está ya incluido en el paquete, y uno de los extremos de la conexión segura es un gateway. En este modo, tanto AH como ESP cubren el paquete entero, incluyendo el header IP entre los extremos, agregando al paquete un header IP que cubre solamente el salto al otro extremo de la conexión segura, que, por supuesto, puede estar a varios saltos del gateway.

Los enlaces seguros de IPSec son definidos en función de Security Associations (SA). Cada SA está definido para un flujo unidireccional de datos y generalmente de un punto único a otro, cubriendo tráfico distinguible por un selector único. Todo el tráfico que fluye a través de un SA es tratado de la misma manera. Partes del tráfico puede estar sujeto a varios SA, cada uno de los cuales aplica cierta transformación. Grupos de SA son denominados SA Bundles. Paquetes entrantes pueden ser asignados a un SA específico por los tres campos definitorios: la dirección IP de destino, el índice del parámetro de seguridad y el protocolo de seguridad. El SPI puede ser considerado una cookie que es repartido por el receptor del SA cuando los parámetros de la conexión son negociados. El protocolo de seguridad debe ser AH o ESP. Como la dirección IP de destino es parte de la tripleta antes mencionada, se garantiza que este valor sea único.

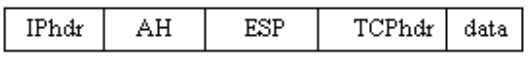
Un ejemplo de paquete AH en modo túnel es:



Un ejemplo de paquete AH en modo transporte es:

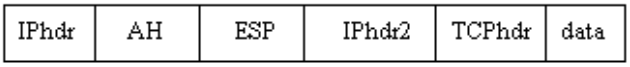


Como ESP no puede autentificar el header IP más exterior, es muy útil combinar un header AH y ESP para obtener lo siguiente:



Este tipo de paquete se denomina Transport Adjacency.

La versión de entunelamiento sería:



Sin embargo, no es mencionado en las RFC que definen estos protocolos. Como en Transport Adjacency, esto autenticaría el paquete completo salvo algunos pocos campos del header IP y también encriptaría la carga. Cuando un header AH y ESP son directamente aplicados como en esta manera, el orden de los header debe ser el indicado. Es posible, en el modo de túnel, hacer una encapsulación arbitrariamente recursiva para que el orden no sea el especificado.

FreeSWAN

Es una implementación IP SEC sobre Linux, provee Encriptación y Autenticación en IP.

La Encriptación Oportuna, permite la habilidad para configurar FreeS/Wan gateways de tal manera que entre ellos negocie el establecimiento de encriptación, de forma dinámica sin coordinación. FreeSwan es software libre.

Ejercicio 1

Creación de un tunnel para leer correo electrónico

Objetivo: El alumno demuestra la importancia de utilizar una Red Privada Virtual en un medio inseguro como lo es Internet, en una actividad de uso común con lo es la lectura de correo electrónico.

Instrucciones

1. Correr un analizador de protocolos o sniffer (puedes utilizar Ethereal http://www.ethereal.com), configurarlo para capturar el trafico local.

2. Conectarse a un sitio de correo (que no maneje auntenticación segura, SSL o https, ni shttp) y leerlo.
3. Grabar la captura en un archivo.

4. Correr de nueva cuenta el sniffer y configurarlo para capturar el tráfico local.

5. Crear un tunnel con algún cliente de ssh (puedes utilizar putty para windows o el openssh para linux o *BSD).

6. Leer el correo.

7. Grabar la captura en un nuevo archivo.

8. Comparar los archivos.

9. Envíar un reporte por correo electrónico de las diferencias encontradas al momento de autenticarse dentro del sistema de correo y el porqué de esas diferencias.

Productos esperados:

- Reporte de captura de paquetes sin utilizar el tunnel y utilizando el tunnel.

- Reporte de diferencias y explicación de lo que sucede al crear el tunnel.

Fuente de Referencia Bibliográfica

Hack Proofing Linux, James Stanger, Ph D.; Patrick T. Lane; Edit. Syngress

Páginas de Referencia Electrónicas

http://www.rz.uni-karlsruhe.de/~ig25/ssh-faq/

http://www.suso.org/linux/tutorials/ssh.phtml

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

man ssh
man scp