LICENCIATURA EN INFORMÁTICA

Trabajo final de grado

Seguridad en Internet

Héctor Javier Elsener

Director: **Luis Marrone**

Facultad de Informática
Universidad Nacional de La Plata
Año 2003

Introducción

Con el surgimiento y posterior difusión de Internet, aparece el comercio electrónico como una opción rentable dentro de las principales empresas, ya sea dedicadas al comercio interempresa (B2B, *bussiness to bussiness*) o directamente al consumidor final (B2C, *bussiness to consumer*). Gracias a este esquema de comercio electrónico, las empresas ven ampliado su mercado a uno de alcance mundial.

Dentro de esta área de la informática, una de las áreas de mayor relevancia es la seguridad de las transacciones comerciales, como un medio de evitar pérdidas, obtener clientes y fidelizar los ya existentes, reforzar la imagen de la empresa y permitir aprovechar las oportunidades del mercado.

Como primera parte de este trabajo, se tratará de realizar un estudio de la problemática de seguridad informática, teniendo como objetivo el obtener conocimientos sobre los distintos aspectos involucrados, investigando las diversas técnicas para alcanzar los objetivos de seguridad relacionados al comercio electrónico, atravesando distintos niveles de seguridad para llegar a una de las infraestructuras mas difundidas, probadas y aceptadas, denominada PKI o *infraestructura de clave pública,* basada en el uso de credenciales digitales para alcanzar un alto nivel de seguridad.

Luego de lograr familiaridad con esta arquitectura, se profundizará sobre los certificados digitales, estableciendo métodos para su identificación y procesamiento, que serán utilizados en la siguiente fase.

Como parte final, se configurará una autoridad de certificación que permita emitir certificados digitales y un sitio web de prueba que será accedido con estos certificados. Estos certificados serán administrados por una aplicación desde la cual se establecerán varias restricciones de seguridad, independientemente de las listas de revocación correspondientes, y que serán verificadas en tiempo real por un agente de seguridad, permitiendo o rechazando accesos al sitio mencionado. La aplicación de administración de los certificados digitales y el agente de seguridad encargado de realizar validaciones forman parte de este trabajo y son implementados en esta fase final.

Las restricciones impuestas constituyen un agregado al nivel de seguridad que se logra con una arquitectura de PKI, ya que pueden determinar la validez de un acceso en base a la dirección IP de origen, el horario o el destino del mismo. De esta manera, concluye el trabajo estableciendo una base para trabajos futuros que requieran conocimientos o implementaciones de seguridad en Internet, ya sean relacionados con comercio electrónico u otro tipo de aplicaciones.

Seguridad informática

Introducción

Internet

En los últimos años, organizaciones de todo tipo han adoptado Intranets y Extranets rápidamente. Esto no es sorprendente, considerando que estas relativamente nuevas tecnologías ofrecen un claro ahorro de costos y una gran facilidad de instalación, comparadas con otras anteriores como redes en línea o WANs basadas en tecnología propietaria. Además, permiten una alta productividad y nuevas formas de trabajo con costos más efectivos: pueden ser utilizadas para construir un amplio rango de aplicaciones de autoservicio que reducen los costos administrativos, mejorar la colaboración entre empleados de la misma organización o con socios de negocios, distribuir información mas rápidamente, etc.

Internet, como una tecnología, ha cambiado la forma en la cual las organizaciones conducen sus negocios, así también como el enfoque hacia la seguridad de la información. Los requerimientos de negocios del día de hoy demandan que la organización de seguridad de la información de una empresa comprenda como proteger el ambiente de web de ésta última, incluyendo sitios web, aplicaciones basadas en Internet y sus servidores de web. La organización de seguridad debe aportar el conocimiento necesario para soportar nuevas iniciativas de comercio electrónico, y al mismo tiempo, debe proveer el soporte y los conocimientos en una manera que no interfiera con el proceso de alcanzar los objetivos y metas de negocio.

Es usual que los objetivos de negocios cambien rápidamente, ya que surgen nuevas oportunidades que requieren una acción inmediata para ser aprovechadas a favor de la empresa. Gradualmente, la tecnología se ha ido transformado en un factor significante (e incluso determinante en algunos casos) para alcanzar los objetivos de negocio. Las soluciones automatizadas se vuelven cada vez más complejas, mientras que los tiempos de despliegue de las mismas se ven acortados. La tecnología para la protección de la información debe apoyar estos cambios, aún cuando las amenazas, exposiciones al mundo exterior y vulnerabilidades continúan incrementándose.

Bajo estas circunstancias, es fácil ver por qué el mantenimiento de un ambiente corporativo seguro en web representa un desafío creciente continuamente.

Consideraciones de seguridad

Internet se basa en una arquitectura de cliente-servidor, comunicados mediante ciertos protocolos y que, mediante un lenguaje de hipertexto como es HTML, provee al usuario final herramientas gráficas y fáciles de usar,

Mediante el uso de TCP/IP, la familia de protocolos estándar de Internet, diferentes sistemas de computación pueden interconectarse con otros, dentro y fuera de la empresa.

Estos protocolos han sido diseñados para ser inherentemente abiertos: los datos son divididos en paquetes que son transportados libremente a través de la red, buscando la mejor ruta posible para alcanzar su destino final. Entonces, si no se toman las debidas precauciones, estos datos pueden ser interceptados y/o modificados, a menudo sin que la parte que los envía o los recibe sea conciente de esto. Ya que el enlace entre ambas partes no se establece de antemano, es fácil para una tercera parte intervenir esta comunicación.

En su estado inicial, Internet no es un sitio seguro para colocar información privada o confidencial. Esto debe ser modificado si una empresa desea utilizar sus ventajas para obtener beneficios o aprovechar nuevas oportunidades de negocios.

En Internet se deben tener en cuenta ciertos aspectos que usualmente no son tenidos en cuenta, así como también sus implicancias:

- Internet es un medio de publicación de información bidireccional. A diferencia de otros medios tradicionales de publicación como teletexto, respuesta por voz, o sistemas de fax, los servidores de Internet están sujetos a ataques a través de Internet.
- Internet es usada cada vez mas como medio de muestra de productos, lo que permite la realización de una plataforma de transacciones de negocios. La reputación de una empresa puede verse dañada si sus servidores son vulnerables a ataques, y además puede sufrir pérdidas económicas.
- A pesar que los navegadores de Internet son muy fáciles de usar, los servidores de web son fáciles de configurar y administrar, y el contenido de Internet es cada vez más fácil de desarrollar, el software de base es de una extraordinaria complejidad. Al ser tan complejo, es probable que posea ocultas varias fallas de seguridad. Existen varios ejemplos en los que software nuevo (o actualizaciones) correctamente instalado es vulnerable a una variedad de ataques.
- Un servidor de Internet puede ser utilizado con otros fines adicionales dentro del entorno de la empresa. Si es vulnerable a ataques, puede que el atacante consiga acceso a otros recursos que no son parte del entorno de Internet pero que están conectados localmente a dicho servidor.
- Usuarios finales sin conocimientos de temas de seguridad son clientes comunes de servicios de Internet, y no poseen el conocimiento necesario para tomar medidas necesarias, ya sean de precaución o corrección. Un usuario final puede entregar una palabra clave a alguien ajeno a la empresa, etc.

La tarea de proveer un ambiente de web seguro a una empresa requiere una atención constante e involucra significantes decisiones técnicas y de negocios.

Objetivos de seguridad

Existe una amplia variedad de técnicas para alcanzar distintos niveles de seguridad en un ambiente de Internet, pero antes de proceder a ellas, debe tenerse un conocimiento de cuales son los objetivos a cumplir.

Hay aspectos principales a tener en cuenta, que deben ser evaluados para formular una estrategia de seguridad adecuada, dependiendo de las necesidades de negocios de la empresa.

Confidencialidad o privacidad

La información no debe estar disponible para quién no está autorizado a acceder a ella. Deben implementarse controles muy estrictos para asegurarse que sólo aquellas personas que necesitan acceso a cierta información tienen acceso a la misma; en algunas situaciones que involucran información confidencial y/o secreta, las personas sólo deben tener acceso permitido a los datos necesarios para realizar su trabajo. El concepto de permitir acceso a información (o recursos en general) solamente a aquellos que verdaderamente lo necesitan, se denomina *control de acceso*.

Un aspecto referente al control de acceso es la limitación de recursos disponibles para un empleado una vez que él ha ingresado en la red corporativa. Algunos de ellos sólo tendrán acceso a realizar consultas dentro del sistema, mientras que otros podrán modificar o crear nuevos datos.

La confidencialidad también implica que la información requerida por un usuario correctamente habilitado no sea interceptada en su camino por personas no autorizadas que puedan hacer uso de la misma. Como es casi imposible impedir que la información sea interceptada (para alcanzar costos baratos, la información debe ser transportada por medios comunes, como líneas telefónicas o microondas), en estos casos debe lograrse que los datos obtenidos sin autorización sean inútiles para quién los obtuvo.

Para lograr confidencialidad no se requiere autenticación o autorización previas.

Integridad

Es necesario que la información no pueda ser modificada de manera inesperada, ya sea por error humano, por un tratamiento erróneo de la información, o hasta fallas catastróficas. Las consecuencias del uso de información inexacta pueden ser desastrosas, ya que si es modificada inadecuadamente los datos se vuelven inútiles o, peor aún, peligrosos. Debe asegurarse la exactitud y coherencia de los datos en todo momento.

En aquellos casos en que la validez de la información es crítica, es muy imprescindible diseñar controles y chequeos que aseguren exactitud completa.

Un esquema de seguridad bien balanceado tendrá complementariamente componentes proactivos y reactivos. Los primeros involucran la utilización de fuertes controles de seguridad, mientras que los segundos incluyen herramientas de auditoria y monitoreo de estos controles. El administrador de red combina estos componentes para visualizar los accesos realizados al sistema (que son registrados en forma de *logs*) para encontrar actividades sospechosas e investigar desviaciones del uso normal del sistema.

Disponibilidad

Los recursos y la información no deben tornarse inaccesibles, el usuario correspondiente debe poder acceder a ellos en el momento necesario. La imposibilidad de acceder a aquellos recursos requeridos es denominada *negativa de servicio*. Los ataques intencionales contra el sistema a menudo requieren que accesos a datos sean deshabilitados para evitar que los mismos sean sustraídos o modificados por el ataque en sí mismo.

Asegurar la seguridad física de un sistema o una red es una manera de mantener disponibilidad de recursos. Mediante la limitación de accesos a máquinas críticas o fuentes de datos, la probabilidad de inaccesibilidad se ve reducida; si el contacto con estos recursos es restringido, se reducen los accidentes y errores internos. Similarmente, proteger electrónicamente la red es importante si existen varios puntos de entrada posibles, especialmente desde un dominio público como lo es Internet.

Brindando redundancia dentro de un sistema, en forma de datos de respaldo (*backup*), máquinas, e inclusive fuentes de energía a menudo se asegura la disponibilidad del sistema en cuestión. El almacenamiento de datos en un lugar diferente al sitio donde el sistema funciona normalmente puede ser útil si la seguridad de éste último es quebrada. Adicionalmente, servidores de respaldo pueden permitir que el flujo de trabajo normal continúe en estos casos. Mientras que estas características aumentan la disponibilidad, es importante protegerlas de intrusos y mantener la confidencialidad de sus datos

Autenticación o identificación

Autenticar consiste en asegurar que la entidad que envía mensajes, los recibe, o accede al sistema es quién dice ser, y además posee los permisos para emprender tales acciones. En todo sistema basado en un entorno de Internet se debe asegurar un esquema de autenticación correcto, para evitar que una persona tome el lugar de otra y de esta forma obtenga beneficios a costa de los demás: si este tipo de problemas se suscitan, la empresa poseedora del sistema pierde credibilidad, genera problemas entre los usuarios y finalmente no le es posible seguir adelante con el sistema.

Lograr la autenticación de los usuarios que ingresan al sistema es fundamental para determinar sus permisos, datos de transacciones a realizar, realizar auditorias, etc.

En algunos emprendimientos basados en Internet, la autenticación de usuarios del sistema puede tornarse un problema con implicancias legales.

No repudiación

Es un método para probar que un usuario ha realizado una acción en particular, o que ha enviado o recibido cierta información en un instante en particular. Esto impide que una persona niegue haber realizado una transacción, que realmente ha realizado. En las áreas de comercio electrónico, este tipo de problemas son los más difíciles de solucionar y a la vez son los más comunes y causan importantes pérdidas económicas.

Un plan que contemple este objetivo usualmente requiere autenticación, autorización, integridad de datos y mecanismos de auditoria. Adicionalmente, se requiere un mensaje advirtiendo al usuario que la acción que está a punto de tomar tiene implicancias legales. Esto hace mas seguras las transacciones mas seguras.

Debe ser uno de los aspectos mas cuidadosamente contemplados, ya que un error implicaría problemas de diversa índole como económica o legal. Además, se tienen que contemplar las variaciones que suele haber entre diferentes países (principalmente en cuestiones legales).

Fácil uso

Los sistemas de seguridad no deben restringir la habilidad de personas o empresas para sus emprendimientos de negocios, ni interferir con sus tareas diarias. El sistema de seguridad en su conjunto debe ser una herramienta con la cual el usuario se sienta seguro en sus tareas, de tal manera que sea agradable su uso, sin complicaciones.

Auditoria

Se refiere a mantener una lista segura de eventos en el sistema, tales como cual usuario ingresó en un determinado instante de tiempo, qué archivos accedió, etc. Es un elemento importante para cumplir el objetivo de no repudiación.

Amenazas, vulnerabilidades, ataques

Cuando se considera la seguridad de un sistema, es necesario determinar todas las amenazas posibles, vulnerabilidades y ataques involucrados.

- Una *amenaza* es una posibilidad de que la seguridad del sistema sea quebrantada, usualmente involucrando información confidencial
- Una *vulnerabilidad* es una debilidad en el sistema que puede convertir una amenaza en una realidad. Pueden ser errores o defectos de software o hardware, o el resultado de una falla de administración.
- Un *ataque* toma ventaja de una vulnerabilidad existente para obtener ventajas en el sistema. Hay distintos tipos de ataques, que pueden ser categorizados en:
    a) Acceso a datos confidenciales no autorizado o inapropiado: es probablemente el tipo de ataque mas común.

b) Corrupción de datos: es particularmente peligroso, sobre todo si la existencia de datos de *back-up* es pobre. Este ataque es principalmente llevado a cabo por virus informáticos, ya que pocos atacantes desean destruir información, la mayoría consiste en obtener ventajas económicas (alterando transacciones) o simplemente un desafío intelectual o mero entretenimiento.

c) Negativa de servicio: este tipo ataque se ha convertido en el mas común en ambientes de Internet, ya que puede ser llevado a cabo remotamente. Existen dos tipos: uno ocurre cuando el sistema se ve sobrecargado y no puede obtener datos para nuevos usuarios, que ven negada su petición; y otro cuando el sistema es obligado a caerse, por lo que tampoco puede servir nuevas peticiones. Un ejemplo del primer caso es un programa que obtiene acceso a todas las impresoras, con trabajos de alta prioridad, lo que genera que los usuarios no puedan acceder a ellas. En el segundo caso, se puede considerar un programa que envía un bloque de datos muy grande sabiendo que producirá el colapso de quién intente procesarlo (esto se conoce como ataque "*buffer overflow*".

Niveles de seguridad

Existen distintos niveles de seguridad, desde el mas bajo que no contempla ningún tipo de protección de la información, brindando libre acceso, hasta el mas alto, donde un usuario debe ser identificado o autenticado correctamente antes de tener acceso a determinados datos, atravesando una serie de controles.

Para cada problemática se debe seleccionar cuidadosamente el nivel de seguridad deseado. Por ejemplo, si los datos son irrelevantes, y su conocimiento debe ser público y tratar de lograr la mayor difusión posible de los mismos, no será deseable restringir el acceso a ellos, ni establecer controles adicionales que perturben el accionar de los usuarios (caso típico de la mayoría de los sitios de Internet, donde se exhibe información libremente). Si, en cambio, la información que será expuesta es confidencial, sólo deberá ser accesible a determinados usuarios quienes deberán sortear una serie de controles para permitir su debida autenticación (deben ser conscientes del valor que posee esta información para que estos controles no sean tomados como impedimentos, sino como protecciones que los ayudan en su tarea).

La decisión que establece el nivel de seguridad adecuado para un sistema informático debe estar sustentada en un análisis de riesgo y de costos: la inversión en asegurar el sistema debe justificarse por los riesgos que se corren al distribuir la información: debe tenerse en cuenta qué datos hay que proteger, de quién hay que protegerse, cuáles son los recursos involucrados, (un error muy común es omitir entre éstos a las personas que manipulan los datos), etc. Luego es necesario identificar las amenazas posibles a los recursos.

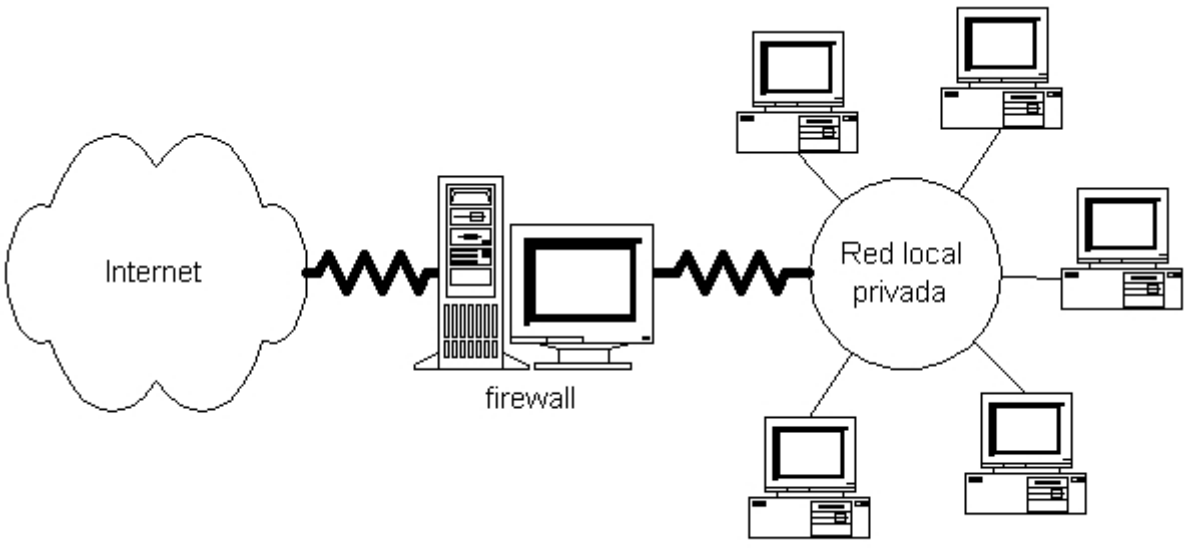
Seguridad perimetral

La mayoría de las empresas y organizaciones tienen conectadas sus redes internas, redes externas e Internet en una red global, obteniendo así beneficios en términos de comunicaciones y accesos a información. Pero esta interconexión puede exponer la red completa de la empresa y/o información crítica a accesos no autorizados.

El concepto de seguridad perimetral es el siguiente: antes que una persona sea admitida en la red interna de la organización, y obtenga acceso a las computadoras que la integran, debe identificarse a sí misma ante un dispositivo que será el encargado de permitir o negar la solicitud. Pueden existir varios puntos de entrada, o uno solo, según la arquitectura de la red corporativa. Si se asegura que una persona puede ingresar a la red sólo a través de estos puntos controlados, entonces se tiene un perímetro de seguridad para la red.

Firewalls

Para controlar el acceso a una red se utilizan los denominados *firewalls*. Este consiste en uno o más sistemas que fuerzan el establecimiento de una política de control de acceso para la mencionada red, de tal manera que todas las comunicaciones entre la misma e Internet se ajusten a la política de seguridad de la organización. Un *firewall* se interpone entre la empresa e Internet, vigilando los datos que entran y salen de la conexión a la red, controlando el acceso de los usuarios internos (desde el interior de la empresa) a Internet y viceversa: los usuarios externos que acceden a la empresa desde Internet. Es decir, realiza un filtro de los paquetes de datos que circulan desde y hacia Internet.



Actualmente los *firewalls* brindan muchas características que están mas allá de éste tradicional filtro de paquetes de información. Otorgan varios niveles de protección, desde comprobar las direcciones IP hasta examinar el contenido de las transmisiones. Poseen mecanismos sofisticados para identificar potenciales intrusos, realizar un *log* de las actividades de los usuarios, notificaciones a personal correspondiente, etc. Algunos de ellos implementan mejoras en el sistema como *caché* de

contenido, para mejorar el rendimiento, soporte para redes virtuales privadas, (*VPN, Virtual Private Network*), administración del sistema mediante web, mecanismos de autenticación y demás.

Tipos de firewall

- Filtro de paquetes: esta es la primera de tres aproximaciones que un *firewall* utiliza para proteger una red. En este caso, un *firewall* es un *router* especializado: examina cada paquete de información que llega al sitio para asegurar que la dirección sea la apropiada, mediante el examen del encabezado del paquete, sin realizar cambio alguno en él, simplemente lo acepta o lo rechaza. Al ser esta operación muy simple, este mecanismo es muy rápido y eficiente, y aún es una parte esencial de un *firewall*. Además de verificar la dirección de IP de origen del paquete, opcionalmente comprueba el *port*. Un beneficio adicional del filtrado de paquetes es que no requiere conocimiento adicional por parte del usuario final, y tampoco se necesita su cooperación.
- Circuito *proxy*: la segunda de las aproximaciones es a través del llamado circuito *proxy*. Todos los comunicadores (clientes y servidores) son forzados a direccionar sus comunicaciones a través del circuito *proxy*, no directamente al destino proyectado. De ésta manera el *proxy* obtiene un paquete direccionado a él, y cambia la dirección para representar el destino deseado (interno). Este procedimiento no es tan eficiente como el anterior, a pesar de no haber mucha diferencia (solamente se está modificando una parte del encabezado de cada paquete). La mayor ventaja que posee este esquema es que esconde las direcciones de IP reales, que es una información valiosa en caso de que alguien esté intentando acceder al sistema.
- Aplicación *proxy*: la tercera aproximación es conocida como aplicación *proxy*. Para cada aplicación, existe un *proxy* que conoce los protocolos y datos de la primera, e intercepta toda la información que va dirigida a la misma. Este tipo de *firewalls* puede ejecutar chequeos más complejos y específicos a las aplicaciones, en comparación con el filtro de paquetes.

Características de un firewall

Los *firewalls* contribuyen a la seguridad perimetral de un sistema basado en red: separan claramente la parte interna y externa del mismo, estableciendo chequeos en el perímetro de la red interna, para evitar accesos indebidos. La ventaja más importante de estos sistemas es que imponen un único punto de ingreso al sistema, en donde se realizan auditorias de acciones y se unifican controles de acceso. Opcionalmente, se obtienen estadísticas referentes al tráfico entrante y saliente de la conexión. Algunas consideraciones referentes a estos sistemas son:

- Cache de contenido: ya que todo el tráfico de información pasa a través del *firewall*, es el lugar ideal para verificar cual es el contenido accedido mas frecuentemente, y almacenar el mismo en una memoria que permita un acceso más rápido. A pesar de que ésta no es una característica tradicional de los *firewalls*, es cada vez más importante.
- Prueba de *firewalls*: para probar el sistema de *firewall*, se deben entender los conceptos de seguridad, TCP/IP, y estar familiarizado con varios tipos de ataque mediante IP, como negativas de

servicio, *spoofing*, robo de información, etc. Además, es crítico probar correctamente el *firewall*, ya que una pequeña falla en su configuración puede dejar hoyos de seguridad en la red a proteger. Existen productos especiales que facilitan estas pruebas.

- *Logging* y alertas: una característica importante de una solución de *firewall* es permitir un registro de eventos, determinar cuando ciertas acciones son apropiadas, y notificar a la autoridad correspondiente. No será beneficioso para el sistema descubrir que el mismo ha sido atacado y esto no se indica por varias semanas. También es importante que los archivos de registro de actividades sean por igual seguros y accesibles, lo que parece ser opuesto, ya que esta información es uno de los objetivos preferidos de quienes atacan el sistema y tratan de cubrir sus actividades, y al mismo tiempo, debe ser fácilmente accesible para la gente que corresponde. La seguridad de los registros del *firewall* debe ser considerada adecuadamente para evitar su acceso indiscriminado.
- Administración: existe una sobrecarga en la operación y administración del sistema, al incorporar un *firewall*, que deben ser consideradas, y deben realizarse en forma segura.
- Legal: se debe tener en cuenta el aspecto legal al configurar una solución basada en *firewall*.

Ataques que no se evitan con un firewall

Los siguientes son ataques que no se evitan con la imposición de un *firewall* en la empresa.

- El concepto de *firewall* asume que las personas que pueden causar daño se encuentran en su totalidad en el exterior de la red, lo cual es a menudo una suposición equivocada. La mayor parte de incidentes que causan daño en un sistema informático cumplen su objetivo gracias a la ayuda de personal interno a la red: el 90% del trabajo de los denominados *hackers* consiste en un trabajo de ingeniería humana, es decir, establecer contacto con personal perteneciente a la empresa que constituye su objetivo, para así obtener luego su ayuda.
- Un *firewall* no controla nada que sucede luego que un usuario ha sido autenticado correctamente y accede al interior del perímetro de seguridad. Por ejemplo, un empleado disconforme puede acceder a la red desde Internet, obtiene el control de una computadora interna y realiza acciones maliciosas desde esta posición; a pesar de esto, el *firewall* realizó correctamente su trabajo, asegurando que es un usuario autenticado.
- Un *firewall* no puede, obviamente, proteger un sistema de ataques que no pasan a través de él mismo: muchas organizaciones se preocupan por la fuga de datos a través de una conexión de su red con Internet, olvidando que este escape de información puede producirse por otros medios (almacenamientos magnéticos, impresiones, etc.) que no son contemplados por un *firewall* y que ni siquiera involucran a Internet.
- Si la organización no posee un esquema coherente de políticas de seguridad, especificaciones de cómo debe realizarse el acceso al sistema desde Internet, una implementación de un *firewall* tendrá problemas seguramente: no alcanza con colocar uno de estos sistemas de mas alto nivel si se tienen varios hoyos en la red, a través de los cuales los ataques pueden sucederse libremente sin intervención del *firewall*. Este último

debe ser parte de una arquitectura de seguridad corporativa totalmente consistente. Las políticas que gobiernan su funcionamiento deben ser realísticas, y reflejar el nivel de seguridad existente en toda la red.

- Un *firewall* no puede proteger de ataques de personal mal intencionado o ineficiente que pertenecen al entorno de la empresa. Un espía dentro de la organización puede exportar información a través de un teléfono o un fax, o algún otro medio no contemplado por la protección del *firewall*. Los usuarios ingenuos pueden revelar información en un entorno diferente al de la empresa (reuniones sociales, etc.).
- Es bastante simple encapsular código en HTTP, SMTP y otros protocolos de tal manera que evite ser detectado por la protección del *firewall*, por lo que deben establecerse controles adicionales en las aplicaciones, servidores o incluso clientes.
- Un *firewall* no puede proteger eficazmente en contra de virus informáticos. Existen muchas maneras de codificar archivos binarios a través de una red, y demasiadas variantes de virus para buscar por todas ellas. Un *firewall* no puede reemplazar una seguridad a conciencia de parte de los usuarios. En general, es muy dificultoso hacer que un *firewall* proteja contra código que es enviado por correo electrónico o copiado a una máquina interna y es ejecutado allí. Una mejor solución para una organización preocupada por el ataque de virus, es implementar un amplio control de ellos en cada computadora vulnerable (detectores de virus que se ejecutan periódicamente), lo que protegerá en contra de virus que ingresen vía discos, módem, e Internet, mientras que el *firewall*, en el mejor de los casos, protegerá sólo contra aquellos que ingresen vía Internet (a pesar que un cierto número de vendedores de firewalls ofrecen protección en contra de virus, generalmente son útiles en plataformas muy específicas, como sistemas *Windows-Intel*).
- Si el perímetro de seguridad ha sido quebrado, entonces una persona puede sortear todas las prohibiciones impuestas por el *firewall*. Por ejemplo, una persona conecta una computadora interna a una línea de teléfono, otorgando una conexión al exterior que rompe el perímetro establecido.

Conclusión

Un sistema de *firewall* está preparado para proteger una red interna o corporativa de ataques externos provenientes de Internet, no para proteger a los datos de su manipulación indebida por los usuarios, ni tampoco de asegurar que las reglas del negocio se cumplen cuando se realizan transacciones dentro de la red. Para estos casos, donde generalmente los errores (voluntarios o no) tienen la participación de personas que no constituyen un ataque en su manera más pura, si no que es mas bien fraude o engaños, es necesario imponer controles de otro tipo, ya sea en aplicaciones, servidores o demás.

Un *firewall* es parte fundamental en el esquema de seguridad de una empresa, ya que de no contar con esta protección, será un sitio atrayente para quienes desean realizar ataques desde Internet. Si sólo se brinda un sistema para evitar fraudes o engaños en las transacciones, se deja abierta la

posibilidad que alguien ingrese al sistema libremente desde Internet, coloque programas de su propiedad e inutilice el resto.

A pesar de ser necesario, un *firewall* por si sólo no soluciona los problemas planteados si el nivel de seguridad que se desea alcanzar es de niveles superiores:

- Confidencialidad: Una implementación de *firewall* no asegura la confidencialidad de los datos. Si un usuario pasó la barrera que impone, tiene acceso al sistema en su conjunto. La información no está codificada de ninguna manera, existe en su estado natural. Por otro lado, la misión del *firewall* no es asegurar la confidencialidad de los datos, es evitar ataques. Como una funcionalidad adicional, algunos tipos de *firewall* brindan los datos encriptados, en una comunicación *firewall-a-firewall*, donde el origen los encripta y el destino los desencripta, para obtenerlos en su forma original.
- Integridad: El *firewall* ayuda a mantener la integridad de la información ya que evita intromisiones inadecuadas, y que un usuario externo acceda libremente a la base de datos desde Internet y realice modificaciones. No puede impedir que los datos sean modificados desde el interior de la red, para lo que deben establecerse otros controles. Tampoco impide que un usuario que ha ingresado correctamente a la red interna (desde el exterior) realice modificaciones que afecten la integridad de los datos.
- Disponibilidad: La disponibilidad de la información puede verse afectada, según la configuración del firewall. El mismo puede impedir la salida de, por ejemplo, video o audio. Los datos, en formato texto, generalmente no ven afectada su disponibilidad (más aún cuando el firewall solamente realiza filtro de paquetes, lo que no afecta la conexión ni el ancho de banda disponible).
- Autenticación: Todo usuario que a traviesa la protección impuesta por el firewall, debe ser reconocido por éste para que obtenga acceso a la red interna. Usualmente, el usuario indica su identificador y una palabra clave. Si ambos se corresponden, el acceso solicitado es permitido. Si en el sistema existe otro mecanismo de autenticación (por ejemplo, certificados digitales) los usuarios deberán acceder al sistema sin indicarle al *firewall* algún dato, y serán autenticados mas adelante. En este caso, la autenticación no es parte de la funcionalidad del firewall sino de las aplicaciones que él protege.
- No-repudiación: No existe un mecanismo basado en *firewalls* para evitar la repudiación de transacciones válidas. Un *firewall* no asegura protección del negocio, sino del sistema en cuanto ataques.

Sistemas de detección de intrusos

Los sistemas de detección de intrusos (*IDS, Intrusion detección system*) complementan la funcionalidad de un *firewall*: son los encargados de monitorear, analizar y detectar problemas en la red interna. Las características que debería tener un IDS perfecto (a pesar que no existe tal sistema) serían:

- Notifica de un ataque exitoso (o potencialmente exitoso) en progreso.

- No comete equivocaciones en estas notificaciones. Esto significa que no produce falsos positivos ni falsos negativos.
- Lo hace rápidamente (en un tiempo inferior a 1')
- Debe brindar un diagnostico completo del ataque. No sirve que tenga una eficacia del 100% en la detección de ataques sino es capaz de brindar información sobre el mismo.
- También debe brindar algunas recomendaciones de cómo bloquear el ataque, como por ejemplo, deshabilitar el dispositivo que está siendo atacado.

Uno de los mayores inconvenientes que debe solucionar para obtener un IDS ideal es evitar los falsos positivos y falsos negativos. Los primeros se dan cuando el sistema produce una alerta innecesaria, por ejemplo, una nueva aplicación es ingresada al sistema y produce un patrón de tráfico no reconocido por el IDS. Los falsos negativos implican la situación opuesta, es decir, el sistema está siendo atacado pero el IDS no lo detecta.

Palabra clave

El esquema de protección mediante palabra clave es el siguiente: cada usuario posee un nombre de usuario y una palabra clave asociados, y en el momento de ingresar al sistema debe indicar ambos. El sistema valida si estos datos se corresponden, con lo cual el usuario obtiene acceso al sistema o recibe un error en caso que la validación indique que uno (o ambos) de los datos es incorrecto y es rechazado.

El usuario puede obtener la clave mediante dos procedimientos:

- Al ingresar al sistema por primera vez, el mismo le solicita que indique cual será la palabra clave que usará el usuario de aquí en adelante.
- La empresa asigna, inicialmente, a cada usuario una palabra clave. Esta es notificada al usuario correspondiente quién ingresa al sistema. Opcionalmente, el usuario puede cambiar la clave.

Se debe tener en cuenta que la palabra clave (denominada *password*) es información privada de cada usuario, y que no debe ser compartida. Se pueden seguir algunas recomendaciones para restringir la elección de una clave adecuada por parte de usuarios, tales como:

- Debe ser de al menos seis (6) caracteres de longitud.
- Debe contener al menos un carácter alfabético y al menos un carácter numérico.
- Debe ser significativamente distinto de claves anteriores.
- No puede ser el igual al identificador de usuario.
- No puede comenzar ni finalizar con las iniciales de la persona que es la poseedora del identificador de usuario.
- No puede incluir alguno de los nombres o el apellido de la mencionada persona.
- No debería ser información que se pueda relacionar con la persona, tal como número telefónico, número de documento o licencia, dirección, sobrenombre, etc.

Debe contarse con una adecuada capacitación a los usuarios finales, que, si bien es mínima, es necesaria: deben concientizarse del valor que significa su palabra clave, la necesidad de no compartirla y de mantenerla en secreto, como seleccionarla, etc. Por parte de la empresa, debe obligarse a que las claves sean modificadas periódicamente, que cumplan con requisitos mencionados anteriormente, etc.

Si contemplamos los distintos aspectos que inciden sobre una política de seguridad, tenemos que este esquema, basado en el uso de palabras claves como única protección, vemos que se puede adoptar para aquellos casos en los cuales los requerimientos de seguridad son mínimos y la información transmitida no constituye un peligro en caso de caer en manos equivocadas. Por ejemplo, constituye una opción válida a ser implementada en un entorno de Intranet donde cada empleado de la empresa tiene acceso a sus datos básicos, a información brindada por la compañía, o consultas varias que no implican información delicada.

En cambio, al pensar en un mayor grado de seguridad, el esquema no es tan apropiado, y se deben tener en cuenta los siguientes puntos:

- Confidencialidad: Sólo la persona poseedora de una determinada clave puede acceder a la información. La confidencialidad depende de cuan secreta es una determinada clave, ya que si es conocida por alguien que no debiera poder acceder a cierta parte del sistema,

nada le impediría hacerlo. La información puede ser interceptada fácilmente, una vez que el usuario la ha solicitado, por otras personas.

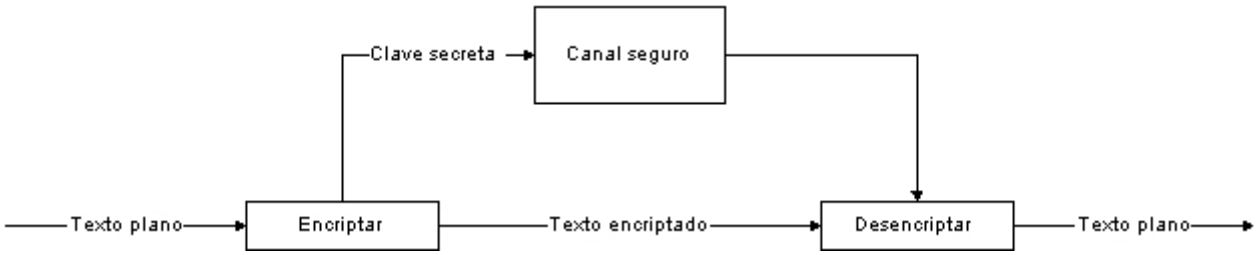
- Integridad: La integridad de la información no se ve afectada en general por el acceso al sistema mediante el uso de palabras clave. En el caso en que se permitan modificaciones a través del mismo, no deberán realizarse sobre información crítica, ya que este esquema no brinda un grado de seguridad apropiado para ello (no existe un esquema de autenticación realmente eficaz para realizar modificaciones, la información puede ser alterada en su camino, etc.)
- Disponibilidad: La disponibilidad de información no se ve afectada por el uso de claves, ya que las mismas son directamente comprobadas por el sistema en el momento del acceso al mismo por parte del usuario. La información sigue estando disponible a los usuarios, sólo que ahora deben ingresar la clave correspondiente para determinar a qué parte de la misma tienen acceso los usuarios.
- Autenticación: Un usuario solo podrá ingresar al sistema si conoce la palabra clave que está relacionada con su identificador de usuario. Por lo tanto, si un usuario ingresa al sistema con una clave correcta, se puede aceptar que es quién dice ser, sin mas comprobaciones. Esto es válido para un nivel de seguridad relativamente bajo, ya que en otro caso debe tenerse en cuenta que es bastante probable que una persona obtenga la clave de otra, ya sea por descuido o por uso de programas especiales: puede crearse un sistema con apariencia similar al original, pero lo que hace es capturar las palabras claves para luego comunicarlas a alguien. Otro problema podría darse si alguien está en complicidad con personas del entorno de la empresa, quienes les comunican las claves.
- Facilidad de uso: El usuario debe ingresar su identificador de usuario y su clave al ingresar al sistema, sin mas cambios. El mayor cambio radica en el cuidado que debe tener el usuario al usar su clave, mantenerla en secreto, recordarla, etc. Por lo tanto, éste esquema presenta una gran facilidad de uso, sin mayores inconvenientes para los usuarios finales del sistema.

Encriptación

Encriptación es el proceso mediante el cual se toma un mensaje legible, conocido como *texto plano*, se le aplica una función de criptografía, conocida como *cipher*, y que produce *texto codificado*, o *ciphertext*. La función de criptografía toma dos parámetros: una clave y un contenido: la encriptación del contenido varía de acuerdo a la clave. El receptor del mensaje encriptado lo envía junto a la clave a un determinado *cipher* y así obtiene el mensaje original en texto plano. Existen diferentes esquemas de encriptación.

Criptografía de clave simétrica (clave secreta)

Conceptualmente, el algoritmo de encriptación más fácil de comprender es aquel que usa la misma clave para encriptar y desencriptar. De esta manera, la confidencialidad del mensaje es un resultado directo de la confidencialidad de la clave.



Las dos partes involucradas deben establecer, de alguna forma, la clave, y no divulgarla (en caso contrario, el poseedor de la clave puede desencriptar el mensaje). Hay algoritmos que utilizan este tipo de encriptación, entre ellos IDEA, DES y 3DES.

DES son las siglas de *Data Encryption Standard*, o Encripción Estándar de Datos. Utiliza criptografía de clave simétrica para encriptar los datos, con claves de 56 bits de longitud y cuyo tamaño no puede ser modificado. En el momento de su creación, en la mitad de la década de 1970, era considerado muy seguro. Actualmente, debido a los avances de la computación, la velocidad de procesamiento y el abaratamiento de los recursos, no se considera suficientemente fuerte para aplicaciones críticas.

Las aplicaciones que requieren altos niveles de seguridad utilizan una versión mas poderosa del algoritmo DES, denominada Triple DES o 3DES: en esta versión se seleccionan dos claves (de 56 bits) y los datos son encriptados tres veces mediante DES. La primera vez, utilizando la primera clave seleccionada, la segunda haciendo uso de la otra clave, y por último se codifica el resultado con la clave obtenida en primer lugar, nuevamente. Este proceso crea datos encriptados que son imposibles que quebrar con las técnicas y el poder de la computación actuales, mientras se mantiene compatibilidad con DES. A pesar de estas ventajas, esta técnica plantea problemas a futuro muy importantes: encriptar tres veces consecutivas un conjunto de datos antes de transmitirlos consume mucho tiempo de CPU, y mientras que hoy la encripción de datos parece ser mas la excepción que la reglar, en un futuro esto será al revés. Sobre todo si se tiene en cuenta el crecimiento del uso de Internet, y dispositivos como teléfonos celulares, se nota que estos aparatos necesitarán una comunicación segura y a la vez un

protocolo de encriptación con un mejor desempeño, que utilice menos recursos. En estos casos, 3DES no es una solución viable.

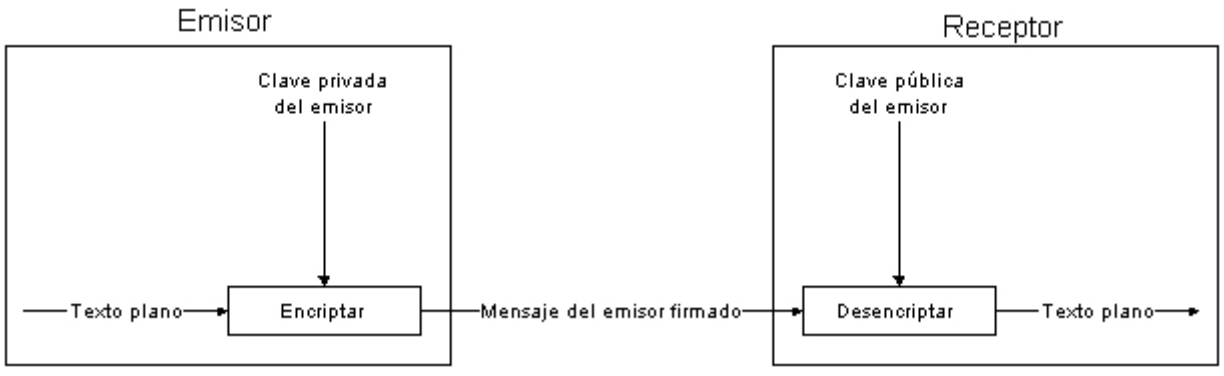
AES (*Advanced Encryption Estándar*) utiliza también criptografía simétrica, pero las claves pueden ser de 128, 192, o inclusive 256 bits, lo que brinda un espacio posible de claves mucho mayor y por ende reduce las posibilidades de ser quebrado. Además, requiere solamente un paso para encriptar datos y ha sido diseñado para soportar una amplia variedad de dispositivos, ser sumamente rápido e inquebrantable. La principal diferencia entre Triple-DES y AES no es la seguridad, sino la mejora en el rendimiento y un mejor uso de los recursos por parte de éste último.

Conclusión

Si bien es ampliamente utilizado, el esquema de criptografía simétrica no brinda una solución completa a los problemas de seguridad que se detectan en sistemas abiertos. La clave puede ser interceptada en el momento en que se da a conocer, con lo que otras personas tendrían acceso a la información, o incluso pueden llegar a realizar transacciones personificando a otras, etc. También se debe tener un esquema que admita varias claves, ya que en caso contrario (contar con una clave única) se brinda el acceso a toda la información a todas las personas poseedoras de tal clave; la administración de claves múltiples en este caso conlleva una carga considerable de trabajo, si se tiene en cuenta el amplio alcance de las redes de tipo Intranets o Extranets. Finalmente, no es posible implementar un sistema que evite la repudiación de transacciones válidas.

Criptografía de clave asimétrica (clave pública y clave privada)

Este esquema utiliza dos claves: una es mantenida privada y la otra puede hacerse pública, por lo que estos algoritmos son llamados de clave pública. Estas dos claves trabajan juntas: si los datos son encriptados con la clave pública, sólo la clave privada que se corresponde con ella puede desencriptarlos. Recíprocamente, si los datos son encriptados con la clave privada, sólo la clave pública correspondiente permite su desencriptado.



Existe una ventaja en este esquema, que es la siguiente: como la clave pública puede ser dada a conocer, cualquiera puede enviar un mensaje encriptado con la misma al publicador de la clave, que sólo él puede desencriptar (con la clave privada): el emisor y el receptor no necesitan conocerse previamente.

Cada una de las claves usadas en este esquema es un número grande, como por ejemplo de 1024 bits (alrededor de 300 dígitos decimales), expresada en un párrafo con un formato especial, muy difícil de comprender a simple vista. Un ejemplo es el siguiente:

```
-----BEGIN PUBLIC KEY BLOCK-----
Version:2.6.3i
mQCNAzGvwGAAAAEEAMQXI06gfdoZzy2Ngdqua6Zf6q4Bfdotc8qGHk9RncuEHSBf
2DrqYrkVmn6cANJp/HdBkJH39LcKybOGbxiahmjVnngPp+PzvX8+Wi7kQ5NP267S
0JIituePxuklEQ5pqywHw8yxtOGIqLjkJtb/pRvZyiC0Cyw1bjnbPFHw2SetAAUR
tCZSb2JpbiBXaGl0dGxlIDxmaXJzdHByQG96ZW1haWwuY29tLmF1PokAlQMFEDGv
wGE52zxR8NknrQEBbV0D/1gJSldscj2bFJ0uD9LOY+LSTj71yxdONZ3cycPZ+3zp
ShCNcsqNAGvHXDtqcGQrNrxHmYqnKBaJ/+46n/FSkDnt/bvEAb105m+6T5oTK8h+
MaaVuvdcphwKfIPQbIoI6LcmtwSd0cyBBndp+O+02x0xhcd2Qx7Gni7J+fz8mm0y
=Ysjn
-----END PUBLIC KEY BLOCK-----
```

Ambas claves son generadas en la misma operación, y a pesar de estar relacionadas, una no puede ser deducida de la otra.

Una parte importante en este esquema es el algoritmo de *hash*, que funciona de la siguiente manera: toma una cantidad de información y la procesa en una forma compleja para emitir un texto de longitud fija que representa un número muy grande, representado como unas pocas líneas de texto. Un buen algoritmo de hash producirá una salida completamente diferente ante la mínima modificación en sus datos de entrada, por lo que será imposible crear dos salidas iguales para distintos textos.

Un documento firmado digitalmente consta de un texto y de una firma, que permite corroborar la autenticidad del primero. El resultado de aplicar un algoritmo de *hash* al texto del documento es la firma, y se adjunta al final del mismo. Para corroborar la autenticidad, se le aplica el mismo algoritmo de *hash* al texto y se compara el resultado con la firma adjunta: si son iguales, el texto no ha sido modificado y está probada su autenticidad.

Para firmar digitalmente un documento en una forma mas confiable, se siguen los siguientes pasos:

1. Se obtiene el resultado del algoritmo de *hash* para dicho documento.
2. Se encripta el texto *hash* obtenido en el paso anterior, con la clave privada.
3. Se envía, publica o almacena el texto con el texto *hash* encriptado (firma). Si estos dos están juntos en un mismo documento, se tiene un documento firmado digitalmente.
4. Cualquier persona conocedora de la clave pública puede probar que la versión del documento es la original (versión firmada), haciendo un *hash* y comparándolo con la versión desencriptada (con la clave pública) de la firma del documento. Cualquier cambio en el documento original, causará que su versión *hash* encriptada no se corresponda con el mismo.

Combinación de esquema simétrico y asimétrico

La criptografía basada en clave pública depende de operaciones matemáticas que son intensivas, lo cual significa que no puede ser utilizada eficientemente para encriptar datos en la manera que muchas aplicaciones requieren. Por otro lado, los sistemas simétricos tienen un rendimiento excelente para ser utilizado en aplicaciones *on-line*.

En la inmensa mayoría de los sistemas basados en clave pública, la criptografía asimétrica es utilizada solamente en el paso inicial de autenticación, para intercambiar la clave secreta. Esta clave secreta puede ser usada por un breve período de tiempo, y luego descartada, lo que hace que la encriptación sea mucho más difícil de quebrar.

La diferencia entre los términos clave privada y secreta está dada porque la primera (usada en un esquema asimétrico) solo es conocida por su poseedor, mientras que la segunda (esquema simétrico) es “un secreto” compartido por las dos partes que intervienen en una conexión.

Kerberos

Kerberos se basa el sistema de autenticación por secreto compartido: si un secreto es compartido solamente por dos partes, entonces cada una de ellas se autentica ante la otra dando pruebas del conocimiento del mismo. En la práctica, el secreto compartido es la clave de sesión o *session key*, que se utiliza para encriptar los mensajes de la comunicación por ambas partes, en un esquema de encriptación simétrica.

Kerberos es un servicio de autenticación distribuido que permite a un proceso (cliente) probar su identidad a un servidor sin enviar datos a través de la red que permitan a un atacante impersonarlo. Opcionalmente provee integridad y confidencialidad para el intercambio de datos entre el cliente y el servidor

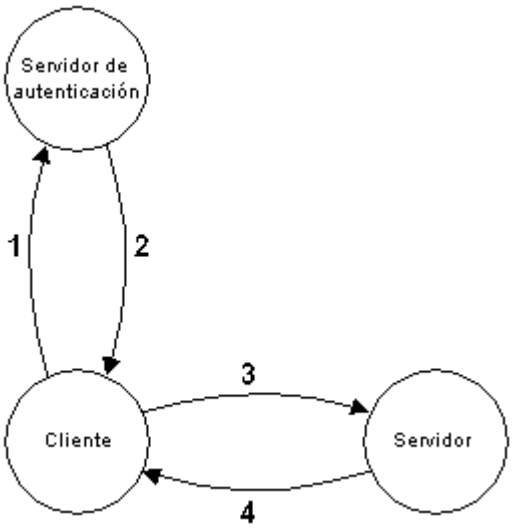
El sistema de autenticación de Kerberos usa una serie de mensajes encriptados para probar la identidad de un cliente (denominado también *principal*) ante un servidor (*verificador*). Para cumplir su objetivo, el cliente tiene conocimiento de una clave de encriptación conocida solamente por el usuario y por un servidor de autenticación. En Kerberos, esta clave es derivada y debe ser pensada como una password, una palabra clave. Similarmente, cada servidor de aplicaciones comparte con el servidor de autenticación su clave de encriptación. La encriptación de datos se hace con un esquema simétrico (se utiliza una clave tanto para encriptar el texto plano como para obtener su forma original a partir del texto encriptado). Como un mecanismo para proveer integridad, cada mensaje posee un código de redundancia que permite detectar modificaciones al texto cifrado (se genera el código de redundancia para el texto plano obtenido y debe concordar con el del texto cifrado recibido).

El ticket de Kerberos

El cliente y el servidor inicialmente no comparten una clave para encriptar sus datos. Cuando un cliente debe autenticarse ante un servidor, confía en un servidor de autenticación para generar una nueva clave de encriptación y distribuirla seguramente a ambas partes. Esta clave de encriptación se conoce como *clave de sesión* (o *session key*), y para darla a conocer al verificador, se utiliza el *ticket* de Kerberos.

El ticket de Kerberos es un certificado emitido por un servidor de autenticación. Además de otra información, el ticket contiene la clave de sesión que será usada para la autenticación del principal ante el verificador, el nombre del principal para el que la clave fue emitida, y un tiempo de expiración luego del cual la clave de sesión es invalida. El ticket no es enviado directamente al verificador, sino que es enviado al cliente quien sí lo envía al verificador como parte de un pedido. Como está encriptado con la clave del servidor (conocida solamente por el servidor de autenticación y el verificador), no es posible que el cliente modifique el ticket sin que esto sea detectado por el verificador.

La figura a continuación muestra el intercambio de mensajes mas simple en Kerberos. Los mensajes **1** y **2** indican el pedido del ticket al servidor de autenticación, mientras que **3** y **4** reflejan el pedido y la respuesta entre el cliente y el servidor o verificador. En el pedido al servidor de autenticación (mensaje **1**), el cliente envía su identidad reclamada, el nombre del verificador, un tiempo de expiración para el ticket pedido, y un número aleatorio usado para relacionar los mensajes de pedido y respuesta de autenticación. En respuesta, el servidor de autenticación retorna la clave de sesión, el



tiempo de expiración asignado, el número aleatorio del pedido, el nombre del verificador y demás información, toda encriptada con la clave del cliente registrada con este servidor de autenticación, junto con un ticket conteniendo información similar.

Cuando el cliente realiza el pedido al servidor verificador (mensaje **3**), además de incluir el ticket en el mismo, existe un llamado *autenticador*, encriptado con la clave de sesión, que consta de: hora actual, código de redundancia (*checksum*), y parámetros de encriptación.

Cuando el verificador recibe un pedido, desencripta el ticket (utilizando la clave compartida entre él y el servidor de autenticación) y entonces obtiene la clave de sesión. Con esta última decodifica el autenticador, y si el *checksum* obtenido concuerda con el recibido junto con el texto codificado, entonces el verificador puede asumir que el autenticador a sido generado por el principal que figura en el ticket y para quien la clave de sesión fue emitida. Para evitar que el mensaje de pedido sea interceptado por un atacante y éste tome el lugar del usuario original, el verificador también valida el *timestamp* para asegurarse que el autenticador es fresco: el mismo debe estar dentro de una ventana (típicamente de 5 minutos) y en ese período no debe haberse visto.

En este punto el cliente se ha autenticado ante un servidor, si se requiere la operación inversa, puede incluirse información que permita la autenticación del servidor ante el cliente en el mensaje de respuesta.

Tickets adicionales

El cliente necesita un ticket y una clave de sesión nuevos para cada verificador con el cual se conecta. En lugar de solicitar una clave al usuario ante cada solicitud de un ticket, el sistema puede debe soportar que esta operación ocurra una sola vez, y las siguientes autenticaciones ocurrirán automáticamente. La opción válida, para evitar almacenar las claves de usuario en el servidor de autenticación, es almacenar los tickets junto con las claves de encriptación (par denominado *credencial*) que serán válidas durante un período limitado de tiempo.

Para su uso con Kerberos, una aplicación debe ser *kerberizada*: esto es, modificarse para adaptarse al esquema de autenticación del protocolo. Esta es la parte mas difícil de instalar Kerberos. Existen versiones kerberizadas de las aplicaciones mas populares, como telnet, POP, etc.

Infraestructura de clave pública

Introducción

En el mundo físico, las protecciones en contra de fraudes en las transacciones comerciales toman las mas diversas formas, desde identificación por fotografías, huellas digitales, firmas escritas o simplemente transacciones personales, donde las partes intervinientes se conocen previamente. En Internet, donde el usuario permanece relativamente anónimo, saber quien es cada una de las partes es una tarea difícil de llevar a cabo.

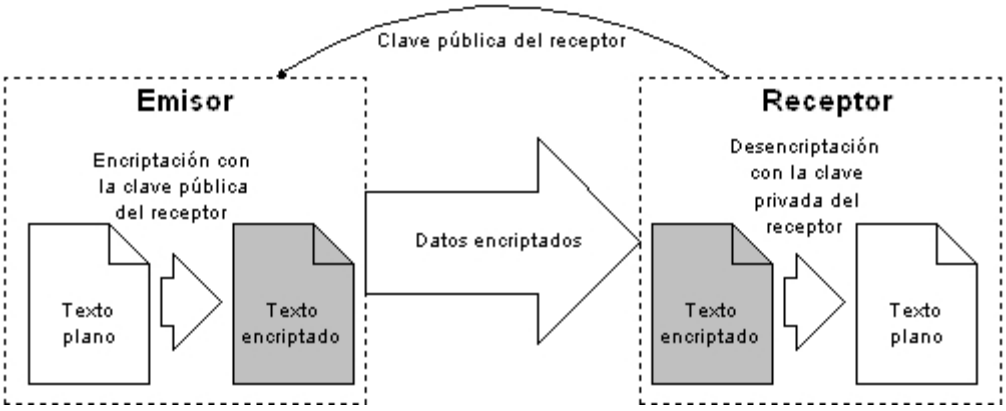
En los últimos quince años se ha desarrollado un conjunto de tecnologías que permiten cumplir con los objetivos relacionados con la seguridad, como privacidad, integridad, facilidad de uso, evitar la no repudiación, y lograr la autenticación, dentro de un ambiente de red como puede ser Internet. Ampliamente llamadas infraestructura de clave pública (*PKI, public key infraestructure*), permiten que las organizaciones usen redes abiertas, tales como Intranets y Extranets basadas en TCP/IP, y repliquen (e incluso mejoren) mecanismos para asegurar seguridad de transacciones en el mundo real. Una muestra de cómo se obtienen mejoras son las firmas digitales, que no sólo aseguran el origen de un documento, si no que impiden que el mismo sea reconocido como original si ha padecido una modificación durante su trayecto.

La base de PKI es la criptografía de clave pública. Ambas partes que intervienen en un intercambio de datos, mediante una red basada en TCP/IP, deben dar a conocer su clave pública entre ellas. Cuando una de estas partes envía datos, encripta los mismos con la clave pública de la otra parte, y luego los envía. De esta manera, sólo la segunda parte puede conocer los datos originales, que sólo pueden ser desencriptados solamente con la clave privada correspondiente.

De ésta manera, si alguien intercepta los datos durante su trayecto, no puede conocer su contenido porque sólo el receptor correspondiente puede desencriptarlos (con su clave privada, y suponiendo que la misma no se ha dado a conocer). También se detecta si los datos son modificados antes de llegar a su destino, por que la firma digital de los mismos será inválida.

Certificados digitales

En el mundo real existen las llamadas “jerarquías de confianza”, que determinan quién tiene acceso a qué. En esencia, PKI brinda la tecnología para manejar estas jerarquías y relaciones en el mundo virtual. Para esto, se introducen los *certificados digitales*: el equivalente electrónico de las pruebas físicas de identidad, tales como pasaportes, documentos de identidad, etc., y son los encargados de identificar a los usuarios a través de una red. Constituyen el elemento esencial del esquema de PKI.



Existen varios usos para un certificado digital determinado, como pueden ser:

- Establecimiento de comunicaciones seguras en Internet, Extranets e Intranets.
- Autenticación de clientes y servidores de Internet.
- Encriptación y firma de correo electrónico.
- Firmar código ejecutable habilitado para la descarga en Internet.
- Verificar el origen y la integridad de documentos y/o código ejecutable.

Un certificado digital es el enlace entre la identidad del dueño del certificado y un par de claves (una pública y la otra privada) que son utilizadas para encriptar datos y firmar digitalmente documentos, utilizando criptografía asimétrica. Físicamente, es un archivo binario que puede ser mantenido en cualquier medio de almacenamiento físico, o dentro del contexto de un navegador de Internet.

Cada certificado contiene al menos la siguiente información:

- Clave pública del dueño.
- Nombre o alias del dueño.
- Fecha de expiración del certificado.
- Número de serie del certificado.
- Nombre de la entidad que entregó el certificado.
- Firma digital de la entidad que entregó el certificado.

Además, cada certificado puede contener información variada suministrada por el usuario, como ser su dirección de correo electrónico, código postal, género, edad, etc.

Autoridad de certificación

Cuando se recibe un mensaje o documento firmado digitalmente, es necesario obtener la clave pública para comprobar su veracidad. La misma puede estar adjunta al mensaje, o obtenerse de un centro de distribución (servicios de directorio), o solicitarse a terceros. En estos casos, se tiene que establecer si la clave pública obtenida realmente pertenece a quién se supone, ya que si no nunca se podrá comprobar la firma de un documento.

En este punto debe establecerse una entidad en la cual tanto el emisor como el receptor del mensaje firmado admiten como confiable, cuya misión sea autenticar la veracidad de la clave pública a usar. Esta entidad, es una *autoridad* para ambos.

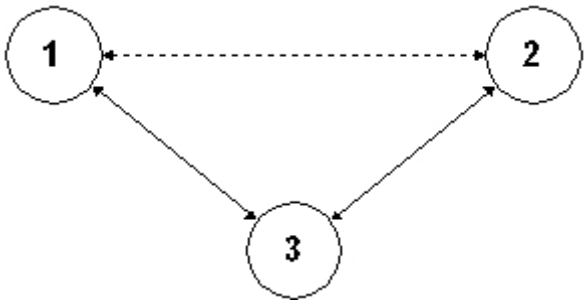
Los certificados digitales son administrados, otorgados y autenticados por una tercera parte confiable denominada autoridad de certificación (*CA, Certificate Authority*). Cada vez que la misma otorga un certificado digital (también conocidos como ID digitales o identificador digital), verifica que el destinatario del mismo no presente una falsa identidad, es decir, testifica la verdadera identidad de la persona.

Todos los certificados digitales son firmados digitalmente usando la clave privada de la CA que los emitió, y un certificado de la autoridad (llamado *root certificate*) es ampliamente distribuido en paquetes de software para permitir que las personas identifiquen los certificados legítimamente emitidos por ésta autoridad de certificación. Si la CA mantiene una buena protección de su clave privada, es virtualmente imposible falsificar un certificado.

Existen dos formas de relacionarse con una CA: una es interactuar con una CA comercial (como por ejemplo *VeriSign*), y la otra es implementar un servidor propio de certificados como CA, que atienda a las necesidades particulares y brinde los servicios de certificación dentro de una empresa. En la segunda opción, la empresa en sí tiene la tarea de autenticar los usuarios y certificar que son quienes dicen ser en el momento de recibir un certificado. La elección entre estas dos opciones también depende de la relación que la empresa tenga con sus clientes.

Los servicios llevados a cabo por una autoridad de certificación (que pueden variar de una a otra) son los siguientes:

- Otorgar y renovar certificados.
- Autenticar las identidades de individuos y organizaciones.
- Verificar los registros de los individuos y las organizaciones.
- Publicar y mantener una lista de revocación de certificados (*CRL, certificate revocation list*) con los certificados que han sido revocados por la CA.
- Manejar problemas legales y responsabilidades relacionadas con la seguridad.



En una comunicación entre dos partes, donde cada una posee un certificado digital correcto (es decir, con su correspondiente clave pública y privada) una tercera persona puede interponerse entre ambas e impersonar a una de ellas (tomar su lugar). Por ejemplo: se supone que **1** quiere comunicarse con **2**, ambos poseen su correspondiente certificado digital y ambos publican sus respectivas claves públicas en un lugar accesible para **3**, pensando que las mismas no necesitan protección. Si **3** intercepta el mensaje donde **1** envía su clave pública a **2**, y reemplaza la misma con su propia clave pública, entonces todos los mensajes que **2** envíe a **1** serán legibles por **3**, luego encriptados con la clave pública de **1** y enviados a su destino, por lo que **1** no se percata del incidente.

La clave de este problema es que, para que el uso de las claves públicas sea totalmente seguro, debe existir una forma mediante la cual se verifique la identidad de la persona que posee la correspondiente clave privada. Obviamente, si el primer mensaje (el intercambio de claves públicas entre los participantes de una conexión) es firmado digitalmente, se está en la misma situación. Para solucionar este problema, existen dos soluciones posibles:

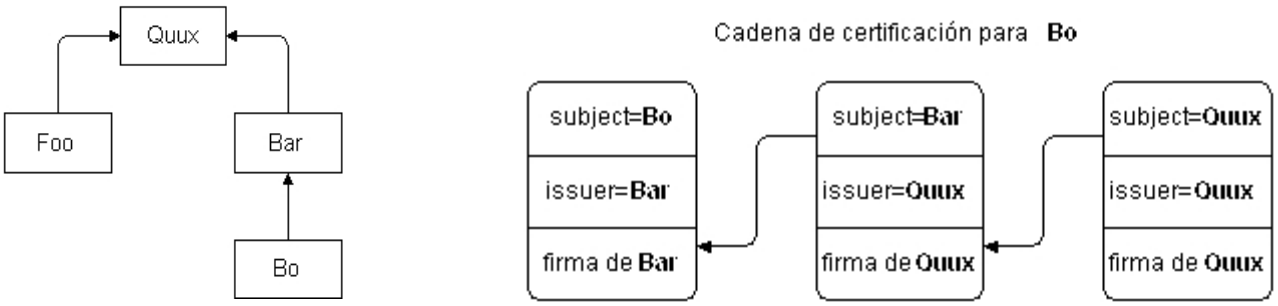
1. Intercambiar claves públicas inicialmente solamente con personas conocidas (personalmente, dentro de lo posible), para evitar intervenciones de terceros en las comunicaciones. Esta solución es la adoptada por un sistema denominado *Pretty Good Privacy, PGP*.
2. Usar una jerarquía de autoridades confiables. Este modelo es el utilizado por el modelo de certificados digitales x.509.

En el primero de los casos, los dos participantes intercambian sus claves públicas personalmente, por correo electrónico, telefónicamente, o algún método similar que les asegure la confidencialidad de la clave. Luego de esto, ellos pueden comunicarse mediante el intercambio seguro de mensajes. Si una tercera persona desea incorporarse a la conexión, debe conocer a uno de los dos participantes iniciales, este lo presentará ante la otra parte, que al confiar en éste último, acepta como válido al nuevo integrante. De esta manera se forma una comunidad de usuarios donde todos son confiables, y cada uno de ellos confía en la clave pública de cada uno del resto.

El segundo, el modelo X.509, establece una rígida jerarquía de autoridades. En el caso más simple, sólo existe una autoridad, cuya clave pública es conocida. Cuando se inicia el intercambio de mensajes, cada parte envía, en un certificado digital, su clave pública. Este certificado digital está firmado digitalmente con la clave privada de la autoridad, que ambos reconocen como confiable (la autoridad es la que emite el certificado, y es la encargada de firmarlo correctamente). De esta manera, el certificado no puede ser alterado durante su trayecto y, gracias a que la clave de la autoridad es conocida, las dos partes pueden comprobar su autenticidad. De esta manera, se implementa un sistema de verificación distribuida de las credenciales (certificados digitales) en donde no es necesario conectarse directamente con la autoridad que ha emitido tal certificado para verificar su autenticidad.

En el mundo real, existen varias autoridades cuyas claves públicas (contenidas en certificados firmados por ellas mismas) son entregados con navegadores de Internet, como *Netscape Communicator* o *Microsoft Internet Explorer*. Muchas compañías también mantienen sus propias autoridades de certificación de tal manera que pueden expedir certificados, lo cual funciona muy bien si tales certificados son utilizados para acceder dentro del alcance de la propia compañía. Para extender el alcance de éstos certificados, la autoridad de certificación local puede ser validada (o reconocida) por una de las autoridades mas conocidas, formando así un árbol de autoridades confiables.

Por ejemplo: el certificado que posee **Bo**, emitido por **Bar**, será reconocido por **Foo**, ya que bar ha sido reconocida por una autoridad que **Foo** considera confiable. Esto se manifiesta como una cadena de certificación, comenzando con el certificado de **Bo**, que está firmado digitalmente por **Bar**, y el certificado de **Bar** a su vez está firmado por **Quux**. Al ser una autoridad de certificación de raíz (*root authority*) entonces firma su propio certificado. Este último es el certificado que será distribuido con software, como por ejemplo un navegador de Internet.



De esta manera se establecen las jerarquías de confianza, y se expande el alcance (sitios a los cuales puede acceder) de un certificado, al ser reconocido por otra autoridad de mayor difusión.

Listas de revocación

Un certificado digital puede ser invalidado antes que expire su correspondiente período de validez. Existen razones por las cuales un certificado, como credencial de seguridad, puede tornarse desconfiable aún antes de su vencimiento estipulado previamente, que incluyen motivos como el compromiso de la clave privada del mismo, descubrir que ha sido obtenido mediante un fraude, cambiar el estado, el poseedor no existe mas dentro del ámbito establecido, etc. Además de asegurar la confiabilidad de los certificados habilitados, la revocación ayuda a mantener la integridad de un esquema de PKI dentro de una empresa.

Al ser PKI un esquema de verificación distribuido de certificados digitales, se tiene la necesidad de distribuir la información de revocación de dichos certificados a todas las entidades (desde personas hasta computadoras y aplicaciones) que intervienen en su uso, para determinar la validez de los mismos. Esta información, junto con el momento en que se hace accesible, varía de acuerdo a las aplicaciones y a la implementación del chequeo de revocación de certificados.

La responsabilidad de revocar un certificado y publicar la información sobre estas inhabilitaciones recae sobre la autoridad de certificación, ya que es la encargada de administrar los

certificados digitales, y es el punto en común que ellos tienen. Por supuesto, una autoridad sólo puede revocar un certificado que ella misma ha emitido, no puede interferir con la tarea de las otras autoridades.

Para la publicación de la información conteniendo datos sobre los certificados revocados, se utiliza un estándar llamado listas de revocación (*CRL, certificate revocation list*) que es un documento creado, mantenido y distribuido por la autoridad de certificación, que consta de una lista con los certificados revocados por la misma (cada vez que un certificado es considerado inválido, se agrega el mismo a esta lista). Un sistema recibe esta lista de la CA periódicamente mediante una conexión de red, y mantiene una copia local para verificar la autenticidad de los certificados usados dentro de él.

Algunos sistemas de emisión de certificados permiten realizar una revocación temporaria de un certificado, para luego habilitarlo nuevamente. Esto permite suspender el certificado durante un lapso de tiempo, sin necesidad de revocarlo definitivamente y luego emitir uno nuevo (esta operatoria es conocida como *certificate hold*).

Planificación

Un tema relacionado con CRL es la frecuencia con la cual la misma será emitida (publicada) por la CA. Si este período es largo, certificados inhabilitados en un momento dado serán reconocidos como válidos hasta una nueva publicación de la lista de revocación. En cambio, si es un período corto, obligará a los distintos interesados a establecer conexiones a menudo con la autoridad de certificación para obtener la mencionada CRL.

Existe un denominado plazo de validez, establecido para cada CRL, y es el período durante el cual un verificador de certificados considera válida la CRL correspondiente. Una vez que este plazo ha vencido, el sistema debe solicitar una nueva CRL a la autoridad correspondiente. Por lo tanto, hay que establecer la diferencia entre el período de publicación de cada CRL (que hace referencia a la frecuencia con la cual la autoridad emitirá la lista de revocación) y el período de validez de la misma. Si este último es, por ejemplo, el doble que el primero, un sistema de verificación de certificados tomará una CRL de cada dos publicadas.

La complejidad total se incrementa si el sistema admite certificados de varias autoridades diferentes: por ejemplo, debe consultar las CRLs de cada una de ellas para validar un certificado. Aquí reside unos de los puntos mas criticados de esta arquitectura: es complicado crear un sistema que requiere revocación en tiempo real para un escenario de comercio electrónico que implique varias autoridades de certificación.

Para evitar el crecimiento indefinido de las listas de revocación, los certificados poseen una fecha límite de validez, mas allá de la cual es rechazado automáticamente, sin necesidad de consultar con la lista de revocación pertinente.

Ventajas de PKI respecto de Kerberos

- Con PKI se elimina la necesidad de tener un repositorio de claves centralizado (servidor de autenticación, posee una clave para cada entidad que autentica), manteniéndolas en cada cliente.

- Se necesita una clave de encriptación para cada comunicación, lo cual implica conectarse con el servidor de autenticación muy a menudo. PKI provee un soporte mas natural para autenticación ante múltiples recipientes.
- En Kerberos es difícil implementar un sistema que solucione el problema de la no-repudiación, dado que la clave de encriptación es conocida por ambas partes, el cliente puede suponer que el mensaje que origina una transacción ha sido generado por la otra parte en perjuicio de él. En cambio, con PKI, la clave de encriptación del cliente es conocida solamente por él (es privada, no secreta) y nadie puede generar mensajes en su lugar.
- PKI forma una abstracción del mundo real, donde el usuario debe indicar sus credenciales antes de realizar una transacción. De esta manera el usuario ve incrementada su sensación de seguridad y se establece un paralelismo entre el mundo real y el virtual.
- La verdadera fortaleza de Kerberos se hace notoria cuando es utilizado en un ambiente controlado. En cambio, para ser usado Internet, aparecen problemas de seguridad relacionados con la clave de encriptación basada en password y el repositorio común de éstas (servidor de autenticación).

Conclusión

PKI es un sistema que utiliza criptografía asimétrica, de clave pública, junto con el esquema de certificados digitales, para cumplir con los objetivos de seguridad de una empresa. Los certificados digitales son el equivalente digital de pruebas físicas de identidad, como pasaportes, licencia de conducir, documento de identidad, etc. Entre sus ventajas hay que destacar que está universalmente aceptado como un estándar para la seguridad en Internet, y brinda una plataforma común para la solución de todos los problemas de seguridad que se presentan en un entorno de red, ya sea Internet, Extranet o Intranet, trasladando en algunos casos mecanismos del mundo real al electrónico.

Las cuatro partes que componen un sistema de este tipo son:

1. Una autoridad de certificación, que es la encargada de emitir y administrar los certificados digitales. En lo posible, debe ser ampliamente reconocida.
2. Un servicio de directorio, que será útil para la publicación de certificados y claves públicas.
3. Usuarios en un paradigma cliente/servidor.
4. Servicios que se prestan a estos usuarios por parte de una empresa.

También hay que resaltar que los aspectos técnicos de la arquitectura se encuentran estandarizados (norma X.509), y permite la interoperabilidad de distintos productos de inclusive distintos vendedores.

Pero también existen desventajas: la más notoria implica las listas de revocación, ya que en un sistema de comercio electrónico la cancelación de un certificado debe ser en tiempo real y no se admite el retraso inherente al uso de CRLs como único método.

Esta infraestructura es aceptada casi unánimente como el medio adecuado para solucionar los problemas de seguridad informática.

Protocolos de seguridad

La familia de protocolos TCP/IP (*Transmision Control Protocol/Internet Protocol*) controla el transporte y direccionamiento de datos a través de Internet. Otros, como HTTP (*Hypertext Transport Protocol*) y LDAP (*Lightweight Directory Access Protocol*) funcionan encima de TCP/IP, es decir, todos ellos lo utilizan para ejecutar aplicaciones tales como mostrar páginas web o servicios de correo electrónico. Estos protocolos no garantizan la confidencialidad de los datos, que son transmitidos en formato plano, haciendo que se puedan interceptar e interpretar fácilmente.

Los protocolos de seguridad se encuentran implementados como una capa adicional sobre algunos de estos protocolos, incrementando su nivel de seguridad para cubrir aspectos relacionados con la autenticación, la confidencialidad de los datos y su integridad.

Protocolo SSL

El protocolo SSL (*Secure Socket Layer*) fue originalmente desarrollado por Netscape, pero ha sido aceptado universalmente en entornos de Internet para establecer comunicaciones autenticadas y encriptadas entre clientes y servidores.

El protocolo SSL se ubica por encima del TCP/IP, pero por debajo de los protocolos de aplicación, como el HTTP, permitiendo que un servidor y un cliente que lo utilizan se autentiquen uno al otro, y permite a ambos establecer una sesión encriptada. SSL hace uso de un protocolo orientado a conexión, como TCP, ya que si los fragmentos de mensajes son recibidos en un orden distinto al que fueron enviados, el receptor no podrá decodificarlos (cada uno de estos fragmentos no es desencriptable independientemente).

Las características aportadas por SSL son las siguientes:

- Autenticación del servidor: permite a un usuario confirmar la identidad de un servidor. Software basado en SSL puede determinar la validez de la identidad de un servidor, para luego operar con él.
- Autenticación del cliente: permite que un servidor confirme la identidad de un usuario, mediante técnicas similares a las utilizadas en el punto anterior
- Conexión encriptada: toda la información enviada entre el cliente y el servidor es encriptada por el software que la envía, y desencriptada por el que la recibe, brindando así un alto grado de confidencialidad. Además, los datos enviados a través de una conexión SSL están protegidos con un mecanismo para evitar lo que se llama *tampering* (alteración de los datos durante su tránsito)

El protocolo SSL incluye dos subprotocolos:

1. SSL Record Protocol: define el formato utilizado para transmitir los datos.
2. SSL Handshake Protocol: involucra el uso del protocolo anterior para intercambiar una serie de mensajes cuando un servidor y un cliente establecen una conexión SSL.

El intercambio de mensajes entre dos partes mediante el protocolo SSL ha sido diseñado para facilitar las siguientes acciones:

- Autenticar al servidor ante el cliente.
- Permitir al servidor y al cliente la selección de algoritmos de criptografía que ambos soporten.
- Opcionalmente, autenticar el cliente ante el servidor.
- Uso de encriptación mediante clave pública, para generar secretos compartidos.
- Establecer sesiones encriptadas.

SSL Record Protocol

Este protocolo toma un mensaje de la aplicación a ser transmitido, fragmenta estos datos en bloques manipulables, opcionalmente comprime estos bloques, aplica un MAC (siglas que provienen de *message authentication code*, código de autenticación del mensaje), encripta, adiciona un encabezado, y transmite el resultado en un segmento de TCP. Los datos recibidos son desencriptados, verificados, descomprimidos, y rearmados para ser entregados a una capa de mayor nivel. En un mayor detalle, los pasos son los siguientes:

1. Fragmentación: cada mensaje de las capas superiores es fragmentado en bloques de 2^{14} bytes (16.384 bytes) o menos.
2. Compresión: es aplicada opcionalmente. La compresión debe ser sin pérdida y no debe incrementar el tamaño del bloque en más de 1.024 bytes (se espera que la compresión disminuya el tamaño de los datos, sin embargo, para bloques pequeños, es posible que por convenciones de formato que el algoritmo de compresión produzca como salida un tamaño de datos mayor a su entrada). En la versión 3.0 de SSL (equivalente a TSL) no hay un algoritmo de compresión especificado, por lo que el valor de dicha característica es **null**.
3. Agregado del código de autenticación del mensaje, o MAC: para este propósito, una clave compartida secreta es utilizada. El cálculo del MAC se define como sigue:

```
hash(MAC_write_secret || pad_2 ||
    hash(MAC_write_secret || pad_1 || seq_num || SSLCompressed.Type
        || SSLCompressed.length || SSLCompressed.fragment ))
```

donde los valores son los siguientes:

Nombre	Valor
`||`	Concatenación
`MAC_write_secret`	Clave secreta compartida
`hash`	Algoritmo de criptografía hash, MD5 o SHA-1
`pad_1`	El byte 0x36 (0011 0110) repetido 48 veces (384 bits) para MD5 y 40 veces (320 bits) para SHA-1
`pad_2`	El byte 0x5C (0101 1100) repetido 48 veces para MD5 y 40 veces para SHA-1

`seq_num`	El número de secuencia para este mensaje
`SSLCompressed.type`	El protocolo de nivel mas alto usado para procesar este fragmento
`SSLCompressed.length`	La longitud del fragmento comprimido
`SSLCompressed.fragment`	El fragmento comprimido (si la compresión no es usada, el fragmento de texto plano)

4. Encriptación: el mensaje comprimido y el MAC son codificados utilizando encriptación simétrica., y este proceso no debe incrementar el tamaño del bloque en mas de 1.024 bytes. Los algoritmos de encriptación permitidos son los siguientes:

Algoritmo	Tamaño de la clave
IDEA	128
RC2-40	40
DES-40	40
DES	56
3DES	168
Fortezza	80
RC4-40	40
RC4-128	128

5. Encabezado: el paso final del proceso es la adición de un encabezado (*header*), que consiste en los siguientes campos:
    - **Tipo de contenido (8 bits):** el protocolo de capa mas alta utilizado para procesar el fragmento.
    - **Mayor versión (8 bits):** indica la mayor versión de SSL en uso. Para SSL Versión 3, el valor es 3.
    - **Mínima versión (8 bits):** indica la menor versión en uso. Para SSL Versión 3, el valor es 0.
    - **Longitud comprimida (16 bits):** la longitud en bytes del fragmento en texto plano (o fragmento comprimido si se ha utilizado compresión).

    El tipo de contenido no hace distinción alguna entre las aplicaciones que pueden utilizar SSL (por ejemplo, HTTP), ya que el contenido de los datos creados por esas aplicaciones es indiferente para SSL. Los valores para el tipo de contenido que han sido definidos son los siguientes:
    - **Change Cipher Spec Protocol:** es uno de los tres protocolos específicos de SSL que usan SSL Record Protocol, y es el mas simple. Este protocolo consiste en un único mensaje, el cual está constituido por un solo bit con el valor 1. El único

propósito de este mensaje es que el estado pendiente sea copiado como estado actual, que actualiza el cifrado utilizado en esta conexión.

- **Alert Protocol:** es usado para comunicar alertas propias de SSL entre los pares de una conexión. De igual manera que otras aplicaciones que usan SSL, mensajes de alerta son comprimidos y encriptados, y especificados en el estado actual. Cada mensaje de este protocolo consiste de 2 bytes: el primero de ellos toma el valor de alerta (*warning*, 1) o fatal (2) para indicar la gravedad del mensaje. Si el nivel es fatal, SSL termina la conexión inmediatamente, otras conexiones en la misma sesión pueden continuar pero no se pueden establecer otras nuevas. El segundo byte contiene un código que indica el alerta específico. Las alertas fatales, según están definidas en la especificación de SSL, son las siguientes:

Alerta	Significado
`unexpected_message`	Un mensaje inapropiado fue recibido.
`bad_record_mac`	Un MAC incorrecto fue recibido.
`decompression_failure`	La función de descompresión ha recibido una entrada inapropiada (por ejemplo, no es posible descomprimir porque se excede el máximo permitido).
`handshake_failure`	El que envía no ha podido negociar un conjunto aceptable de parámetros de seguridad dadas las opciones disponibles.
`illegal_parameter`	Un campo en el mensaje estuvo fuera de rango o es inconsistente con otros campos.

Las respuestas a las distintas alertas son los siguientes:

Respuesta	Significado
`close_notify`	Notifica que el emisor no enviará mas mensajes dentro de esta conexión. Cada parte debe enviar una alerta de este tipo antes de cerrar quien escribe en una conexión.
`no_certificate`	Debe ser enviada en respuesta a un pedido de certificado, si no hay certificado disponible.
`bad_certificate`	Un certificado recibido es inválido.
`unsupported_certificate`	El tipo de certificado recibido no es soportado.
`certificate_revoked`	Un certificado ha sido revocado por su

	firmante.
`certificate_expired`	Un certificado ha expirado.
`certificate_unkown`	Alguna otra cuestión, durante el procesamiento del certificado, ha hecho que el mismo sea inaceptable.

Handshake Protocol

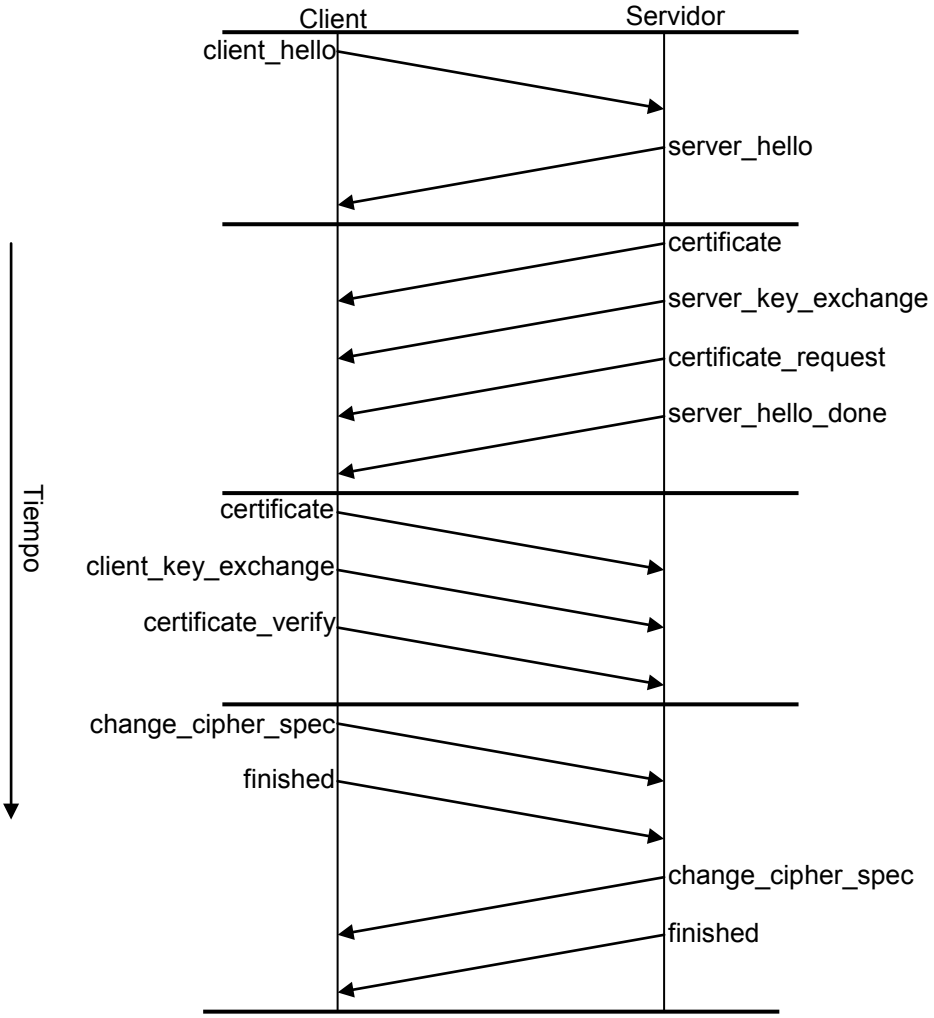
Es la parte más compleja de SSL. Este protocolo permite al servidor y al cliente autenticarse uno al otro y negociar los algoritmos de encriptación y MAC, y las claves de criptografía a ser usadas en un registro de SSL. Este protocolo es usado antes de la transmisión de cualquier paquete de datos sea transmitido.

El protocolo de Handshake consiste en una serie de intercambio de mensajes entre el cliente y el servidor. Cada mensaje tiene 3 campos:

- **Type (1 byte):** indica uno de los 10 mensajes.
- **Length (3 bytes):** la longitud en bytes del mensaje.
- **Content (mas de 1 byte):** los parámetros asociados con el mensaje.

Tipo de mensaje	Parámetros
`hello_request`	Null.
`client_hello`	Version, random, session id, cipher suite, compression method.
`server_hello`	Version, random, session id, cipher suite, compression method.
`Certificate`	Cadena de certificados X.509v3.
`server_key_exchange`	Parameters, signature.
`certificate_request`	Type, authorities.
`server_done`	Null
`certificate_verify`	Signature
`client_key_exchange`	Parameters, signature
`Finished`	Valor hash

El siguiente esquema muestra el intercambio necesario para establecer una conexión lógica entre un cliente y un servidor, que puede ser visto en 4 fases.



Fase 1: Establecer las características de seguridad.

En esta fase se inicia una conexión lógica y se establecen las características asociadas con la misma. El intercambio es iniciado por el cliente, quien envía un mensaje `client_hello` con los siguientes parámetros:

- `Version`: la versión de SSL más alta comprendida por el cliente.
- Random: una estructura de random generada por el cliente, consistente en un *timestamp* de 32 bits y 28 bytes generados por un generador randómico seguro de números. Estos valores son usados durante el intercambio de claves para prevenir ataques.
- `Session ID`: un identificador de sesión de longitud variable. Un valor distinto de cero indica que el cliente desea modificar el parámetro de una conexión existente o crear una nueva dentro de la misma sesión. Un valor de cero indica que el cliente desea establecer una nueva conexión en una nueva sesión.
- `CipherSuite`: es una lista de las combinaciones de algoritmos de criptografía que el cliente soporta, en orden decreciente de preferencia. Cada elemento de la lista define un algoritmo de intercambio de claves y una especificación de criptografía.

- `Compression method`: es una lista de los algoritmos de compresión que el cliente soporta.

Luego de enviar `client_hello`, el cliente espera por el mensaje `server_hello`, que contiene los mismos parámetros que el primero. Para el mensaje `server_hello`, se aplican las siguientes convenciones: el campo `version` contiene la versión mas baja sugerida por el cliente y la mas alta soportada por el servidor; `random` es generado por el servidor en forma totalmente independiente del cliente; si `SessionID` del cliente es un valor distinto a cero, el mismo valor es usado por el servidor, en caso contrario contiene el valor del nuevo identificador de sesión; `CipherSuite` contiene un solo valor, seleccionado de los propuestos por el cliente; de igual manera el campo `compression`.

El primer elemento de cada valor del parámetro `CipherSuite` es el método de intercambio de claves (mediante el cual las claves para encriptación convencional y MAC son intercambiadas). Los métodos soportados son los siguientes:

- **RSA:** la clave secreta es encriptada con la clave pública RSA de quien recibe.
- **Fixed Diffie-Hellman:** Es un intercambio de claves en el cual el certificado del servidor contiene los parámetros públicos firmados por la autoridad de certificación (*CA, certificate authority*). El cliente provee su parámetros de clave pública en un certificado (si la autenticación del cliente es requerida) o en un mensaje de intercambio de claves.
- **Ephemeral Diffie-Hellman:** esta técnica es utilizada para crear claves secretas temporarias, usadas sólo una vez.
- **Anonymous Diffie-Hellman:** usa el algoritmo básico de Diffie-Hellman sin autenticación. Cada parte envía sus parámetros al otro, sin autenticación. Esta aproximación es vulnerable a ataques en el medio de ambas partes intervinientes en la conexión.

Fase 2: Autenticación del servidor e intercambio de claves.

El servidor comienza esta fase enviando su certificado, si el necesita ser autenticado: el mensaje contiene uno o una cadena de certificados X.509. El mensaje `certificate` es requerido para cualquier intercambio acordado de clave, excepto *Anonymous Diffie-Hellman*. Luego, el mensaje `server_key_ exchange` debe ser enviado, si es requerido. El siguiente paso es, si el servidor no es anónimo, requerir un certificado del cliente con el mensaje `certificate_request`, que tiene dos parámetros: el primero es el tipo de certificado solicitado (`certificate_type`) y el segundo es una lista de autoridades aceptadas (`certificate_authorities`). El mensaje final de esta fase, y siempre requerido, es `server_done`, que no tiene parámetros y luego de lo cual el servidor esperará por mensajes del cliente.

Fase 3: Autenticación del cliente e intercambio de claves.

Luego de recibir el mensaje `server_done`, el cliente debe verificar que el servidor ha provisto un certificado válido, si es requerido, y comprobar que los parámetros de `server_hello` sean aceptables. Si todo esto es satisfactorio, el cliente envía uno o mas mensajes al servidor.

Si el servidor ha solicitado un certificado, el cliente comienza esta fase enviando un mensaje `certificate`. Si no hay un certificado disponible, entonces el cliente envía en su lugar una alerta `no_certificate`. El siguiente paso es enviar el mensaje `client_key_exchange`, cuyo contenido depende del tipo de intercambio de claves a realizar. Finalmente, en esta fase el cliente debe enviar un mensaje `certificate_verify` para proveer una validación explícita del certificado del cliente, que solo se envía como siguiente a un certificado de cliente que tiene propiedades de firmado (parámetros propios de *Fixed Diffie-Hellman*).

Fase 4: Finalización.

Esta fase completa el establecimiento de una conexión segura. El cliente envía el mensaje `change_cipher_spec`, adecua sus condiciones de criptografía, e inmediatamente envía `finished` sobre los algoritmos, claves y secretos. Como respuesta a estos dos mensajes, el servidor envía sus mensajes `change_cipher_spec` y `finished`. En este punto el protocolo *Handshake* es completo y el cliente y el servidor pueden comenzar a intercambiar datos.

Internet Assigned Numbers Authority (IANA) ha reservado el port número 443 para conexiones con HTTP sobre SSL, y HTTPS es el nombre del protocolo resultante.

Protocolo TLS

Existen otras opciones de encriptación de datos, pero el protocolo SSL es un desarrollo aceptado casi universalmente, por sus características.

El protocolo TLS (*Transport Layer Security*) es una iniciativa de estandarización del protocolo SSL, cuya versión actual es muy similar a la versión 3 de SSL, cuyas características se han expuesto anteriormente.

Protocolo PCT

El protocolo PCT (*Private Comminications Technology*) es un derivado de SSL 2.0, realizada por Microsoft, que contiene algunas leves mejoras.

Rendimiento

Un servidor que utiliza el protocolo SSL (o su similar, TSL) se ve notablemente recargado en el procesamiento, ya que esto implica realizar operaciones matemáticas complejas de tal manera que la encriptación realizada sea la adecuada. Obviamente, se puede pensar en soluciones que disminuyan esta sobrecarga, como agregar procesadores, memoria o inclusive adicionar servidores para distribuir el trabajo. Pero incluir hardware de propósito general no mejora en gran medida la situación, es una solución a corto plazo: es común que una empresa dedicada al comercio electrónico posea un crecimiento exponencial de su tráfico en ciertos períodos de tiempo.

Los llamados *aceleradores de criptografía* atacan el problema en su esencia. Mientras que los procesadores de propósito general actuales no pueden manejar eficientemente el número de operaciones aritméticas de punto flotante necesarias para la encriptación de datos, los mencionados aceleradores son diseñados específicamente para ejecutar la clase de operaciones matemáticas realizadas durante las transacciones seguras, haciendo que las mismas se realicen hasta en un 90% más rápido cuando se agrega este tipo de hardware a un servidor. Además de la mejora en velocidad de ejecución, se tiene que es mucho mas fácil y efectivo en cuanto a costos, colocar aceleradores de criptografía a un conjunto de servidores, que incrementar el número de éstos últimos.

Los aceleradores de criptografía no sólo mejoran el rendimiento del servidor, sino que también contribuye a un aumento de la estabilidad del servidor: si los procesadores del mismo se encuentran sobrecargados, el sistema completo es notoriamente inestable. Al reducir esta carga, se obtiene el mencionado aumento de estabilidad.

Cuando algunas de las funciones de seguridad pasan de estar centralizadas en el sistema operativo del servidor, o una aplicación, a un acelerador de criptografía basado en hardware el sistema en su totalidad se torna mas seguro, ya que este hardware puede proteger por sí mismo claves y datos sensitivos, al mismo tiempo que se asegura un correcto funcionamiento de los algoritmos.

Productos

Intel Corp. planea liberar un procesador de 64 bits (llamado Itanium) que, según la propia compañía, realiza operaciones de SSL unas 10 veces más rápido que los procesadores anteriores, ya que contiene dos unidades de punto flotante. Empresas como IBM, HP, Novell y Microsoft planean liberar versiones *beta* de sus sistemas operativos y herramientas adecuadas a este nuevo producto.

Sun Microsystems ha indicado que la próxima versión de su CPU (Ultrasparc III) tendrá instrucciones RISC para hacer el procesamiento de la criptografía basada en clave pública mas eficiente.

HTTP Seguro (HTTPS)

HTTP seguro (HTTPS) es un protocolo seguro orientado a comunicaciones mediante el intercambio de mensajes que ha sido diseñado para permitir la coexistencia del modelo de mensajes de HTTP y ser fácilmente integrado con aplicaciones que utilizan este último mecanismo. De esta manera, HTTPS provee los mecanismos de seguridad necesarios para el amplio rango de usuarios (incluso potenciales) de Internet. El objetivo de diseño de HTTPS es brindar un protocolo flexible que brinde una alta ortogonalidad, en cuanto a sus modos de operación, manejo de claves, algoritmos de criptografía y formatos de encapsulado de información, mientras que se preservan las características de implementación y el modelo de transacción de HTTP.

HTTPS no requiere el uso de certificados de clave pública en el cliente (o directamente claves públicas), ya que soporta un esquema simétrico de claves en sus operaciones. Esto es significativo, ya que significa que transacciones espontáneas pueden ocurrir sin que los usuarios individuales necesiten tener una clave pública establecida. HTTPS está habilitado para tomar ventaja de la infraestructura de certificados digitales, aunque puede prescindir de ella.

Operación

Existe una completa flexibilidad de algoritmos de criptografía, modos y parámetros. El cliente y el servidor deben acordar un modo de operación y encriptación de la información para establecer una conexión segura. La creación de un mensaje en HTTPS puede ser vista como una función que toma tres parámetros:

1. El mensaje en su forma original. Puede ser un mensaje de HTTP u otro dato.
2. Las preferencias del receptor en cuanto a criptografía y manejo de claves.
3. Las preferencias del emisor en cuanto a criptografía y manejo de claves.

Este mecanismo puede llegar a necesitar intervención del usuario: si por ejemplo existen varias claves disponibles para firmar el mensaje, de las cuales el usuario debe seleccionar una de ellas.

Para recuperar un mensaje de HTTPS, el receptor necesita leer los encabezados del mismo para descubrir las transformaciones (procesos de encriptación) que fueron realizadas al mensaje original, y luego removerlas utilizando una combinación de claves y algoritmos acordados entre ambas partes intervinientes en la conexión. Análogamente a la composición de un mensaje de HTTPS, la recuperación puede ser vista como una función que toma cuatro parámetros:

1. El mensaje HTTPS.
2. Las preferencias de criptografía y claves del emisor establecidas para este documento.
3. Las preferencias actuales del emisor.
4. Las opciones de criptografía previamente establecidas por el emisor.

Protección de los mensajes

Las opciones para proteger de los mensajes son firma, autenticación y encriptación de los mismos. Cualquier combinación de éstas es válida (incluyendo la opción de no proteger un mensaje de manera alguna).

Existen múltiples mecanismos para soportar el manejo de claves, incluyendo las del estilo de palabras claves ingresadas manualmente y clave pública. La diversificación de estos mecanismos permite el envío de mensajes confidenciales a aquellos que no poseen clave pública.

Firma

Si la opción de firma digital es aplicada, un certificado apropiado debe ser adjuntado al mensaje (posiblemente con su correspondiente cadena de certificación) o el emisor puede esperar que el receptor lo obtenga independientemente (de igual manera con la cadena de certificación).

Encriptación e intercambio de claves

HTTPS soporta dos mecanismos de encriptación:

- Mediante el uso de clave pública: el mecanismo de encriptación simétrico es transferido encriptado bajo la clave pública del receptor.
- Mediante el uso de claves acordadas previamente en una forma externa al propio protocolo. La encriptación se lleva a cabo con las claves acordadas.

Integridad del mensaje y autenticación del emisor

HTTPS provee un medio para verificar la integridad de un mensaje y autenticidad del emisor, mediante el cálculo de un código de autenticación del mensaje (*MAC, Message Authentication Code*), como hash codificado sobre el documento utilizando una clave secreta compartida (que puede ser acordada con anterioridad en diversas maneras, como por ejemplo personalmente). Esta técnica requiere o el uso de clave pública o encriptación.

Este mecanismo es útil en casos donde es necesario permitir a las partes identificarse mutuamente para transacciones confiables (sin proveer un mecanismo realmente eficiente para solucionar la no repudiación) , lo que permite solucionar varias necesidades de autenticación (como por ejemplo, control de acceso) con un mecanismo liviano y escalable.

Demás opciones

El protocolo también provee un control de la frescura de la conexión, para evitar que un mensaje sea interceptado y enviado mas tarde, personificando otro usuario. Esta opción requiere de un acuerdo de tiempos de reloj entre las partes.

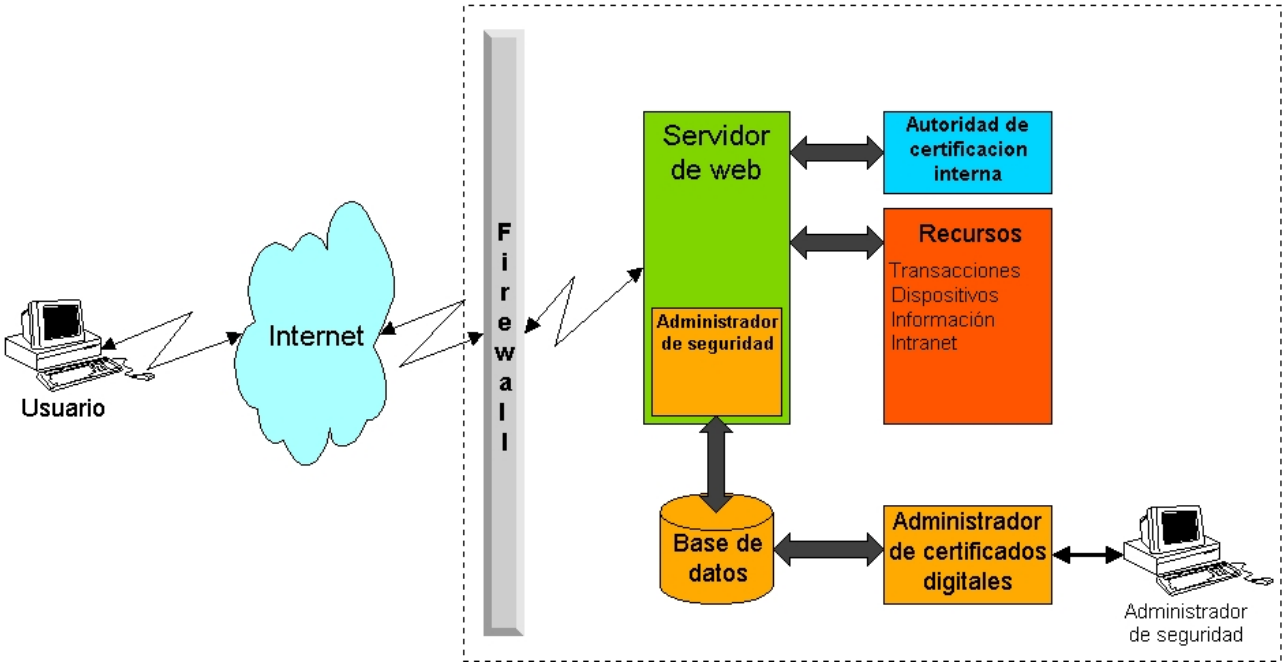
Combinación de protocolo SSL y HTTPS

Los protocolos HTTPS y SSL pueden ser combinados para obtener un mayor grado de seguridad. Al encontrarse en distintos niveles, pueden utilizarse ambos al mismo tiempo, lo que producirá que SSL encripte nuevamente (posiblemente con otros algoritmos o conjuntos de claves) los mensajes ya encriptados en su momento por HTTPS. De esta manera, aún si se lograra decodificar un mensaje de HTTPS, el mismo estaría nuevamente codificado.

Implementación

A continuación se darán los detalles de la implementación de un sistema cuyo objetivo es brindar acceso seguro a un sitio de Internet mediante el uso de certificados digitales, implementando un esquema de clave pública (PKI), y permitiendo establecer restricciones y cambio de permisos en tiempo real para las credenciales digitales, sin necesidad de acceder a las listas de revocación de la autoridad de certificación.

Arquitectura



El sistema consta de dos módulos que actúan sobre una base de datos común. Estos son:

Administrador de Seguridad de Internet (ASI)

Su función es brindar una interface para que el personal de seguridad de la empresa pueda establecer permisos y restricciones para cada certificado, en tiempo real, con lo que se tiene un sistema capaz de suspender provisoriamente cada uno de ellos sin necesidad de consultar las listas de revocación correspondientes. Esta herramienta interactúa sobre una base de datos encriptada, para evitar que sea manipulada mediante otros caminos.

Agente de seguridad

Es la herramienta encargada de controlar el acceso. Ante cada intento, que debe hacerse con un certificado digital, comprueba los datos del mismo en la base de datos, decidiendo en función de ellos si es aceptado o rechazado. Esta validación se hará ante cada pedido del usuario, por lo que cada página web que solicite, habrá sido validada.

Este módulo interactúa directamente con el servidor de web funcionando como un supervisor de datos, validando cada página que se entrega al usuario final. No posee una interface de usuario.

Funcionamiento

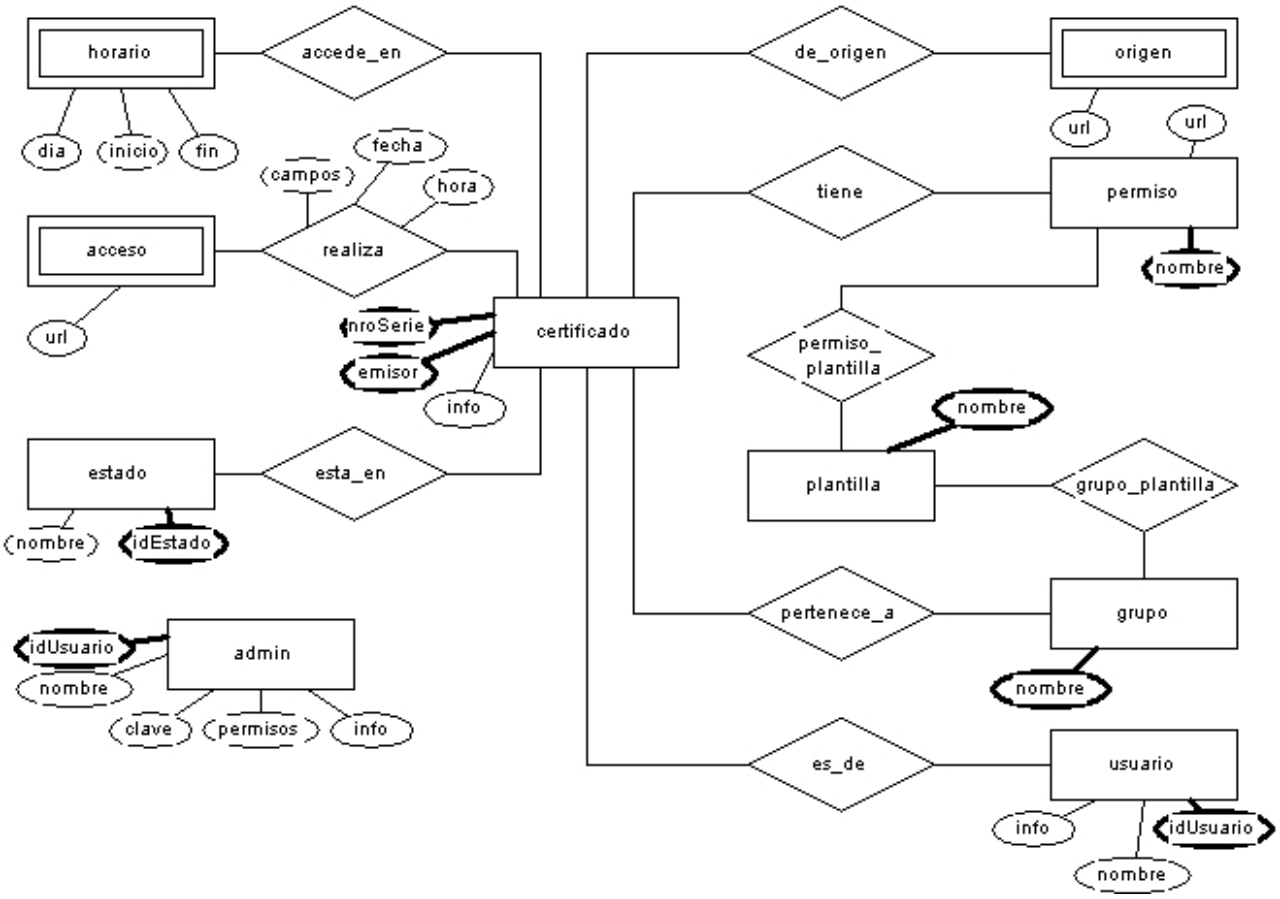
1. Como primer paso, un usuario debe solicitar un certificado digital a la empresa, trámite que puede ser realizado a través de Internet o algún otro medio. Debe existir una forma de corroborar que el solicitante es quien dice ser, para lo cual existen varios mecanismos y que a su vez dependen del nivel de seguridad deseado:
    a. puede solicitarse el número de tarjeta de crédito, a partir del cual se puede obtener el resto de los datos.
    b. puede hacerse personalmente.
    c. puede establecerse una infraestructura mediante la cual cada persona que desea probar su identidad puede concurrir a un notario o escribano (que ha sido autenticado personalmente) el cual, con su correspondiente certificado digital, accede al sitio y asegura la identidad del solicitante. En muchos casos se implementa una variante de este método, mediante la cual la autoridad de certificación posee oficinas donde se autentica a las personas que solicitan certificados digitales: esto es lo indicado para un alto grado de confiabilidad.
2. El usuario registra su certificado accediendo a una dirección de Internet especialmente establecida para tal fin. El usuario no está habilitado para realizar operatoria alguna, solamente permite que el sistema registre los datos de su certificado en la base de datos para su posterior habilitación por los administradores de seguridad. En este punto es importante destacar que el certificado debe provenir de una autoridad de certificación declarada como confiable.
   Este paso puede ser automatizado y realizarse automáticamente en el paso 1, durante la emisión del certificado.
3. El administrador de seguridad chequea la lista de certificados desde el `Administrador de Seguridad de Internet`, y determina cual es el rol que le corresponde a cada uno de ellos, estableciendo permisos y restricciones (esta operación puede ser llevada a cabo en forma grupal, para varios certificados a la vez, para acelerar la tarea). Para establecer el rol correspondiente, es necesario la intervención de otras áreas de la empresa, ajenas a seguridad.
4. Una vez que el certificado ha sido habilitado, y se han establecido los permisos y restricciones correspondientes, el usuario poseedor del mismo puede ser notificado (por correo electrónico, por ejemplo), o simplemente puede esperar un tiempo prudencial y luego intentar acceder. La primera opción es la preferible.
5. El usuario puede operar con su certificado. Antes de llevar a cabo un pedido del usuario, el `Agente de Seguridad` verifica el estado del certificado que está siendo utilizado en la sesión establecida, y en caso de detectar que ha sido suspendido o inhabilitado, se rechaza la solicitud. Cada acceso es registrado en la base de datos en la sección de auditoria.

Si el usuario ya posee un certificado digital, debe obviarse el punto 1. para que la empresa lo acepte, la autoridad que lo emitió debe estar entre las declaradas como confiables por la empresa, en caso contrario será rechazado (para lo cual se debe presentar una opción no automatizada del paso 2, para el registro del certificado, de tal forma que el usuario pueda hacerlo desde Internet).

Si el usuario detecta alguna anomalía con su certificado, como por ejemplo que alguien ha obtenido su clave privada, puede solicitar su inhabilitación parcial o total, hasta que el problema haya sido solucionado (tal vez mediante la entrega de un nuevo certificado).

El usuario debe conectarse desde un navegador de Internet utilizando el protocolo HTTPS, ya que desde el servidor se requiere el establecimiento de una sesión de SSL para la comunicación, lo que permite asegurar la autenticación de ambas partes y la confidencialidad de los datos que se transmiten. Los intentos de conexión que no sean realizados bajo estas normas serán rechazados. Obviamente queda excluida de esta condición la solicitud de certificado digital a través de Internet, para lo cual el usuario se conecta simplemente usando protocolo HTTP sobre TCP/IP.

Base de datos



La base de datos almacenará los datos de cada certificado, junto con los permisos y restricciones de seguridad, e información de auditoria. Si bien la seguridad de la base de datos es un tema no abarcado por esta tesis, se piensa en datos encriptados para evitar que sean manipulados por algún otro camino (aún a pesar de una reducción en la performance, ya que los datos deben ser decodificados para leer y codificados para ser almacenados).

Algunas restricciones de los datos son:

Un usuario puede poseer varios certificados, pero cada uno de ellos pertenecerá a un único usuario.

Cada certificado es identificado por una clave formada por su emisor y su numero de serie (por norma X.509, el número de serie debe ser único para cada certificado entregado por un determinado emisor). Un certificado se encuentra en un estado determinado en cada momento. Un certificado puede pertenecer a mas de un grupo (tabla `pertenece`)

Las restricciones de acceso se encuentran en las tablas `de_origen` (valida las direcciones IP desde donde puede provenir la solicitud con el uso del certificado), `horario` (que restringe el horario en que puede usarse un certificado para acceder) y `tiene` (que especifica a qué URLs puede acceder).

La información de auditoria se encuentra en la tabla `acceso`, para cada página web que contenga campos se almacena el contenido de los mismos, como una prueba que impide la repudiación de una transacción por parte del usuario.

La tabla `admin` posee información sobre los usuarios habilitados para utilizar el Administrador de Seguridad de Internet y restricciones de uso.

La `plantilla` permite agrupar un conjunto de permisos y aplicar los mismos a un grupo de certificados, simplifica esta tarea ya que evita que se haga individualmente.

Tablas

Tabla	*Atributo*		*Descripción*
certificado	emisor	Varchar(50)	Emisor del certificado
	nroSerie	Varchar(40)	Número de serie del certificado
	idUsuario	Varchar(20)	Usuario asociado al certificado
	estado	Tinyint	Estado actual
	info	Varchar(255)	Observaciones
estado	idEstado	Tinyint	Identificador de estado
	nombre	Varchar(20)	Descripción del estado
de_origen	emisor	Varchar(50)	Datos que identifican un certificado
	nroSerie	Varchar(40)	
	IP	Varchar(15)	Dirección IP de origen habilitada para el certificado
grupo	nombre	Varchar(20)	Descripción del grupo de certificados
	info	Varchar(255)	Observaciones
pertenece	emisor	Varchar(50)	Datos que identifican un certificado
	nroSerie	Varchar(40)	
	grupo	Varchar(20)	Grupo al cual pertenece el certificado
horario	dia	Tinyint	Datos que indican los momentos válidos para ingresar con el certificado
	horaInicio	Date/time	
	horaFin	Date/time	
	emisor	Varchar(50)	Datos que identifican un certificado
	nroSerie	Varchar(40)	
usuario	idUsuario	Varchar(20)	Identificador de usuario
	nombre	Varchar(50)	Nombre
	info	Varchar(255)	Observaciones
acceso	emisor	Varchar(50)	Datos que identifican un certificado
	nroSerie	Varchar(40)	
	fecha	Date/time	Fecha y hora en la que se realizó el acceso
	url	Varchar(255)	URL accedida
	campos	Varchar(255)	Valores ingresados en la transacción (si es que hay)
tiene	permiso	Varchar(20)	Nombre del permiso al cual puede acceder
	emisor	Varchar(50)	Datos que identifican un certificado
	nroSerie	Varchar(40)	
permiso	nombre	Varchar(20)	Nombre del permiso
	url	Varchar(255)	URL asociada
plantilla	nombre	Varchar(20)	Nombre de la plantilla
grupo_plantilla	grupo	Varchar(20)	Nombre del grupo
	plantilla	Varchar(20)	Nombre de la plantilla
permiso_plantilla	permiso	Varchar(20)	Nombre del permiso
	plantilla	Varchar(20)	Nombre de la plantilla
admin	idUsuario	Varchar(20)	Identificador de usuario
	nombre	Varchar(50)	Nombre
	clave	Varchar(15)	Clave
	permisos	Varchar(20)	Conjunto de permisos
	info	Varchar(255)	Observaciones

Descripción de los tipos de datos:

- `Varchar(n)`: cadena de caracteres de longitud variable, pero siempre menor o igual a n.
- `Tinyint`: entero chico (byte)
- `Date/time`: hora y día.

Enumeración de los estados posibles para un certificado:

1. **Ingresado**: el certificado está ingresado en la base de datos, pero aún no está habilitado para ser utilizado.
2. **Habilitado**: el certificado está registrado en la base de datos y ya puede ser usado, con las restricciones indicadas.
3. **Suspendido**: el certificado está momentáneamente suspendido, a la espera de ser habilitado nuevamente o inhabilitado definitivamente.
4. **Revocado** o **Inhabilitado**: el certificado ha sido inhabilitado definitivamente

Encriptación de la base de datos

La base de datos almacena, además de los datos asociados a cada certificado, el registro de las transacciones que lleva a cabo cada usuario (o cada certificado, si suponemos que un usuario puede tener mas de un certificado asociado para llevar a cabo varios roles) cada una de ellas con su completo conjunto de datos y fecha. Este registro permite evitar la repudiación de transacciones llevadas a cabo por los usuarios, demostrando que la misma fue llevada a cabo, y en qué fecha.

Para evitar complicidad de personal perteneciente al ámbito de la empresa, los datos almacenados en esta base deberían ser codificados por una clave especial, desconocida para ellos (puede ser establecida por la organización que desarrolla el sistema de seguridad, o por mitades). La información es codificada antes de ser almacenada, y decodificada en el momento de su extracción. Como ya se ha dicho anteriormente, se sacrifica performance en pos de la seguridad.

La encriptación de la base de datos está fuera del alcance de este trabajo.

Selección de las herramientas.

Todas las herramientas y demás elementos utilizados en la implementación del trabajo son de Microsoft Corp. , por razones de rapidez y familiaridad con estos productos por parte de quien realiza el desarrollo en el momento de comenzar el mismo.

Actualmente las decisiones serian muy diferentes, tomando como base tecnología Java y desarrollos *open source*, tales como los siguientes:

- Sistema operativo: Linux, por razones de costo, confiabilidad y seguridad. Aunque el desarrollo debería ser multiplataforma.
- Web server: Apache/Tomcat (es decir, podría basarse en servlets y solamente se necesitaría un contenedor, como Tomcat)
- Autoridad de certificación: Netscape Certificate Services o alternativamente puede usarse OpenSSL como generador de claves y certificados (sin ser totalmente una autoridad de certificación).
- Lenguaje de desarrollo: se mantendría Visual Basic para el Administrador Visual de Certificados Digitales pero el Agente de Seguridad sería un filtro incorporado al servidor de web realizado en Java.

Sería interesante desarrollar un sistema basado en certificados digitales utilizando JAAS (*Java Authentication and Authorization Services*) escribiendo módulos de login que permitan chequear las restricciones impuestas a un certificado digital ante cada pedido. Esto puede hacerse con relativa facilidad en un entorno con un servidor de aplicaciones compatible con JAAS, por ejemplo JBoss (servidor de aplicaciones *open source,* www.jboss.org): alcanza con escribir el *login module* correspondiente, establecer un dominio de seguridad de tal modo que cada EJB que se desea asegurar esté contenido en él y relacionar este dominio de seguridad con el módulo de *login* escrito. Otra facilidad que brinda este *application server* es la de poder cambiar la encripción: pueden reemplazarse las existentes por clases mas sofisticadas de SSL o incluso incorporar una encripción ad-hoc, dependiendo de las necesidades del negocio. De esta manera se abre un abanico de posibilidades para continuar con este trabajo en el mundo Java.

En este caso las implementaciones deberían basarse en distintos estándares para asegurar la portabilidad y de esta manera independizarse del servidor de Internet, el contenedor de aplicaciones e incluso el sistema operativo usado. Algunos estándares: J2EE, JAAS, JSSE.

Sistema operativo

Nombre: Microsoft Windows 2000 Advanced Server

Características especiales:

- Facilidad de administración
- Seguridad
- El servidor de web está incluido en el sistema operativo lo que permite un alto rendimiento.
- También se incluye con el sistema una autoridad de certificación personalizable, lo que permite tomar el rol de emisor de certificados digitales.

Web server

Nombre: Microsoft Internet Information Server 5.0

Características especiales:

- Al estar integrado al sistema operativo, utiliza la misma base de datos de usuarios y cuentas, lo cual facilita la administración. También se utilizan un conjunto de herramientas del sistema operativo como Monitor de rendimiento, Visor de eventos, etc., que permiten lograr una administración unificada de ambos elementos.
- Es extensible, ya que soporta *Internet Server Aplication Programming Interface*, ISAPI, para extender la funcionalidad del servicio de http: es posible crear programas que preprocesen o postprocesen datos enviados a o desde el servidor, lo cual es fundamental para la arquitectura planteada: se puede colocar el Agente de seguridad como una extensión del servidor de Internet, analizando todos los pedidos que procesa.

Autoridad de certificación

Nombre: Microsoft Certificate Services

Características especiales:

- Incluido en el sistema operativo.
- Altamente personalizable.
- Accesible desde programas externos mediante interfaces establecidas.

Base de datos

Nombre: Microsoft SQL Server 7.5.

Características especiales:

- Facilidad de uso y experiencia en el uso de la herramienta.
- Rapidez en el desarrollo del esquema de base de datos.

Lenguaje de desarrollo

Agente de seguridad

Nombre: Microsoft Visual C++ 6.0

Características especiales:

- Rendimiento: al ser un elemento clave en el desarrollo propuesto, por el cual pasarán todos los pedidos que los usuarios realicen al servidor de web, su performance debe ser optima.
- Razones la elección de este lenguaje de programación para implementar el **Agente de seguridad** es consecuencia de la elección del servidor de Internet (*IIS*). La extensibilidad del mismo se logra mediante ISAPI (*Internet Server Aplication Program Interface*), y para lograr una implementación compatible es necesario hacerlo en C++.

Administrador de Seguridad de Internet

Nombre: Microsoft Visual Basic 6.0

Características especiales:

- Rapidez en el desarrollo.
- Creación de interfaces de usuario productivas y amigables en forma rápida y eficiente.

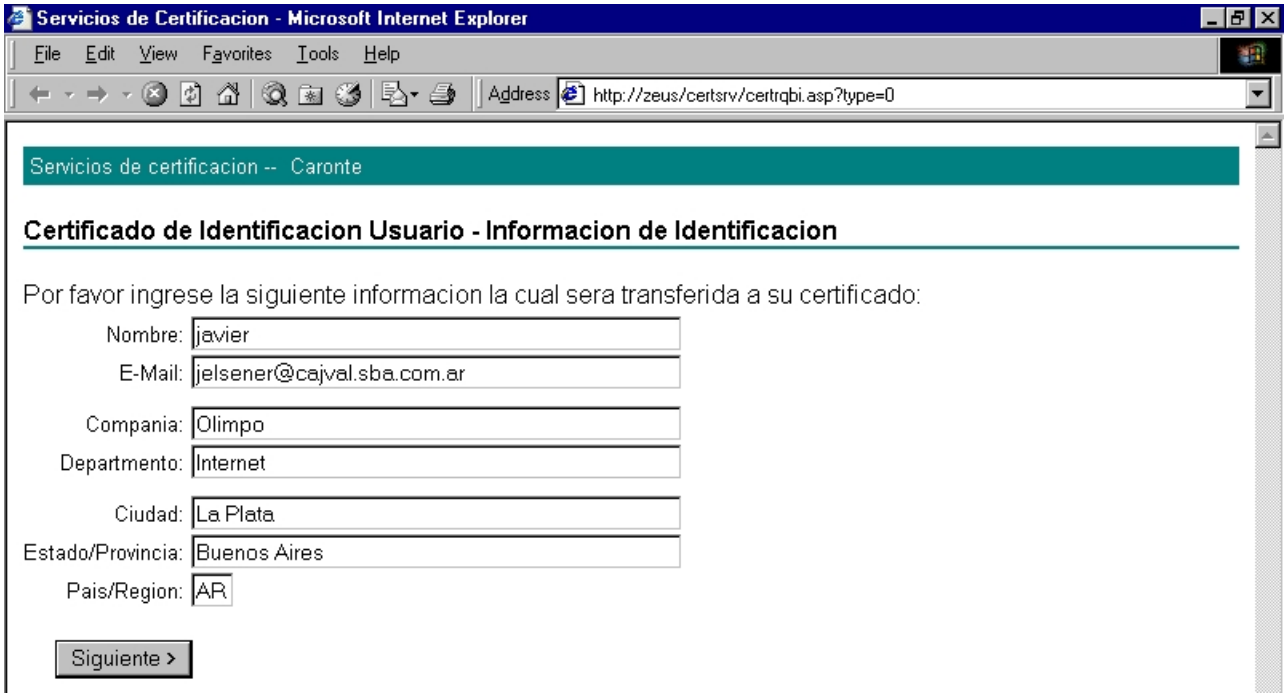
Autoridad de certificación

La autoridad de certificación puede ser un sistema interno a la empresa, puede establecerse como un esquema paralelo (involucrando otro negocio) o puede no proveerse y utilizar servicios de certificación externos; la selección debe supeditarse a un análisis de compatibilidad con los objetivos de la empresa, riesgos, costos y beneficios de cada opción. En este caso, debido a la facilidad de uso y confiabilidad de la autoridad de certificación usada, se simulará como un servicio mas de la empresa.

En este caso también han sido automatizadas varias funciones de la CA, para entregar al usuario final una interface de solicitud de certificados más confortable, sin tecnicismos u opciones inentendibles para personal no informático (opciones de encriptación, de seguridad, y de tipo de certificado, como por ejemplo).

La solicitud de certificados digitales se realiza mediante Internet siguiendo los siguientes pasos:

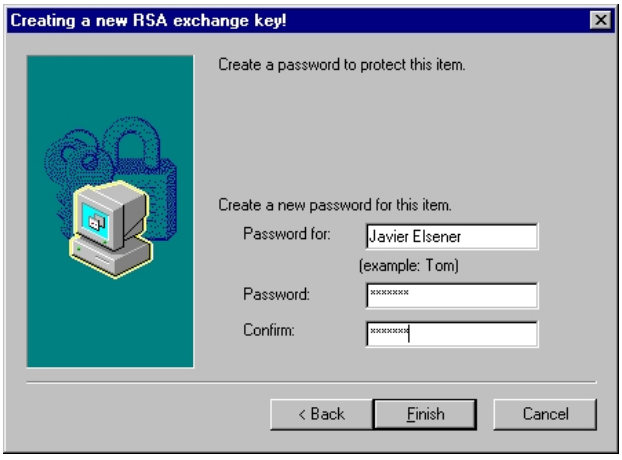
1. Solicitud:



El usuario debe acceder a un sitio particular donde indica sus datos. Al presionar “Siguiente”, accede a una pantalla donde debe establecer el nivel de seguridad a “Alto” (*High*), con lo que, como paso siguiente, debe indicar la clave secreta asociada al certificado y el nombre de referencia del mismo dentro del entorno del navegador de Internet para su posterior utilización. Estos pasos se reflejan en las siguientes ventanas:

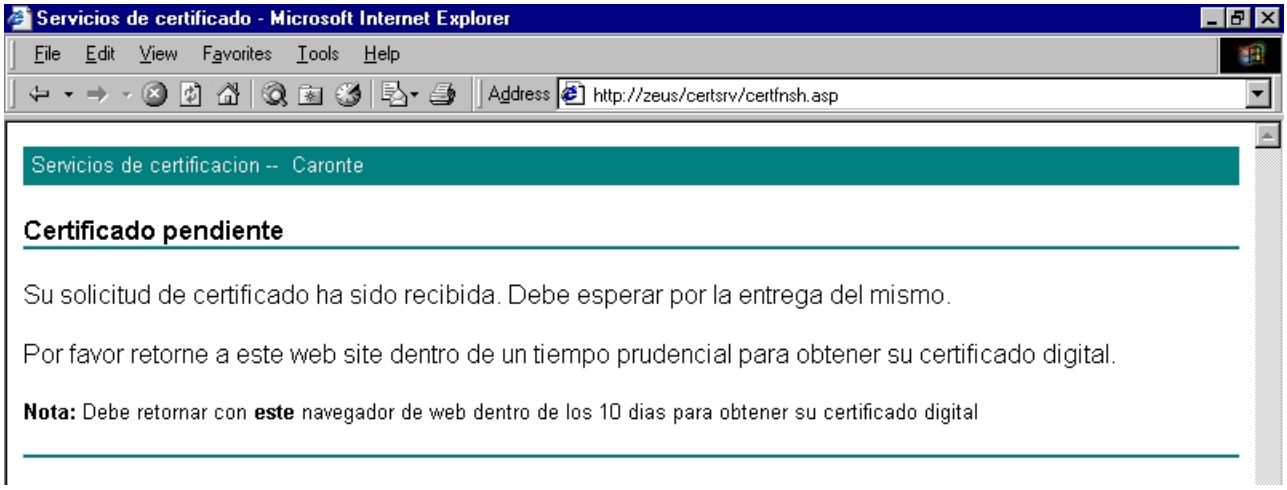
Es aquí donde el usuario debe presionar "Seleccionar nivel de seguridad" (*Set security Level...*") para establecerlo, en la siguiente pantalla, como "Alto" (*High*).

Establecimiento del nivel de seguridad para un certificado.



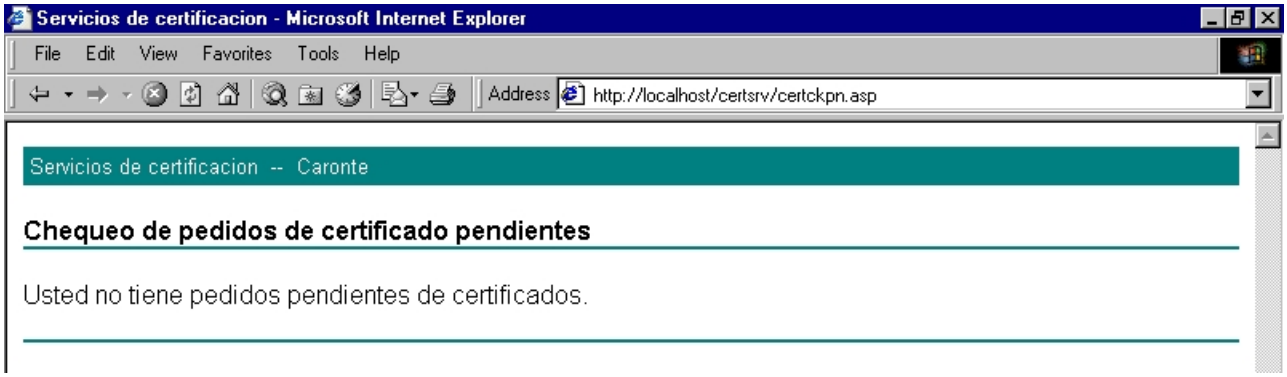
Establecimiento del nombre y la clave secreta del certificado digital solicitado.

Una vez completados todos los datos, el usuario recibe una notificación que indica el resultado de la operación. En este caso se muestra la aceptación de la misma, aunque puede ser negada por una diversidad de factores.



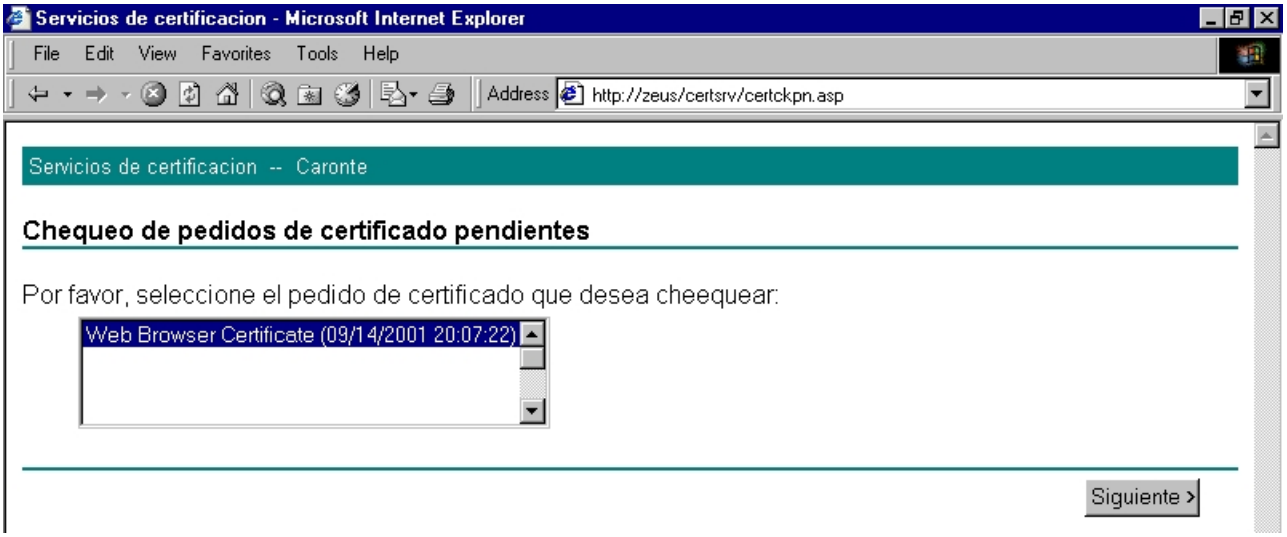
2. Consulta

El usuario debe acceder, dentro de un tiempo prudencial, al sitio de certificación para comprobar el estado de su solicitud de certificado. En caso de que la misma no haya sido procesada aún, recibe la siguiente indicación:

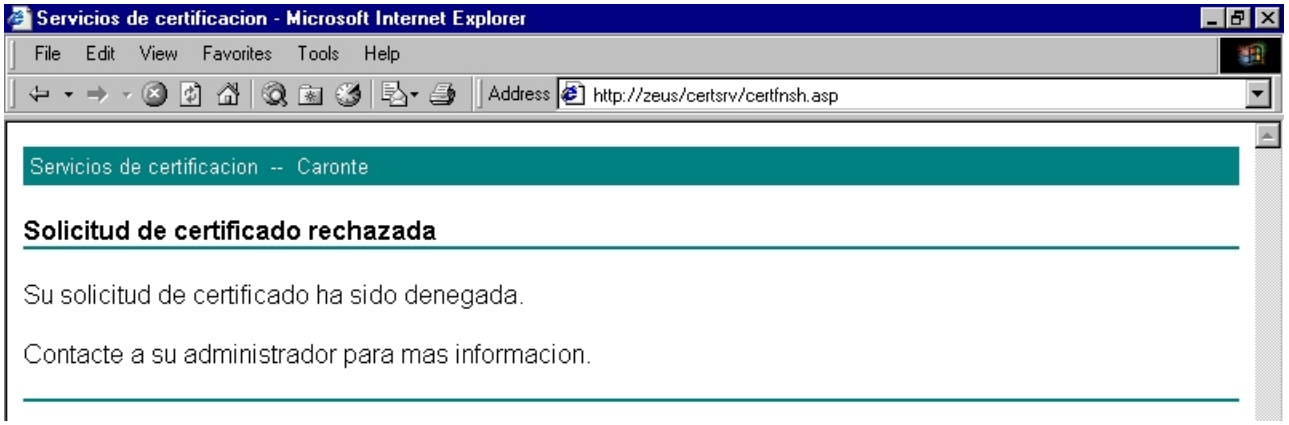


3. Procesamiento

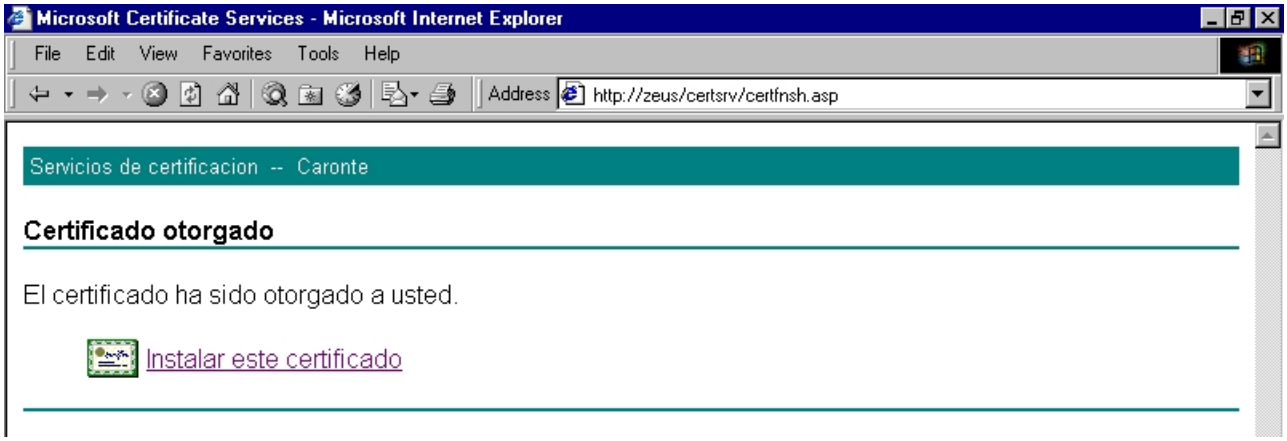
Cuando el pedido ha sido procesado, el usuario puede ser notificado vía correo electrónico, o simplemente puede chequear periódicamente el sitio de certificación. Aquí se muestra que el pedido (el usuario puede tener varios pedidos pendientes, por lo que se muestra una lista de los mismos para permitir la selección de uno de ellos) ha sido procesado:



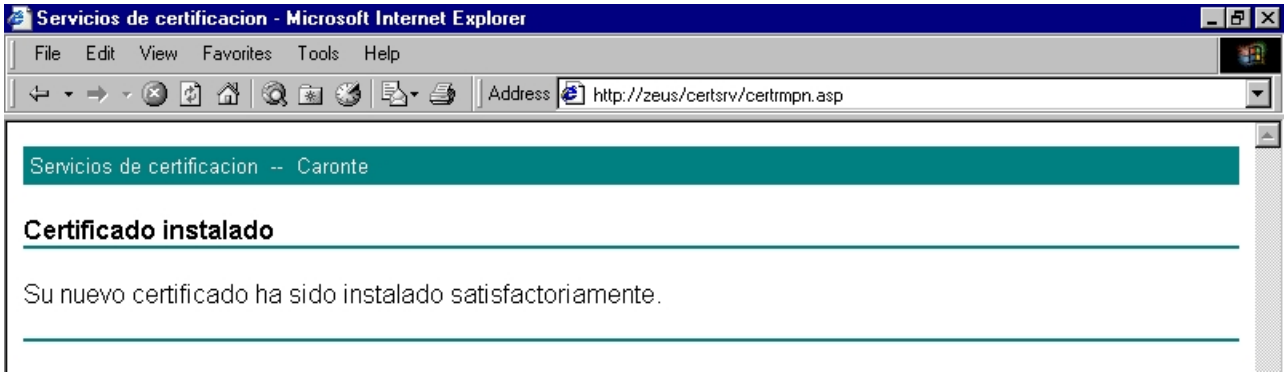
El usuario debe seleccionar el pedido para luego conocer el resultado del mismo. Existen dos posibilidades: ha sido rechazado o ha sido aceptado y entonces está disponible para ser instalado en el entorno del navegador de Internet usado. Para el primer caso la notificación sería:



En cambio, si el pedido ha sido procesado satisfactoriamente, el certificado es entregado al cliente (a través de un link) para su instalación en el navegador de Internet actual:

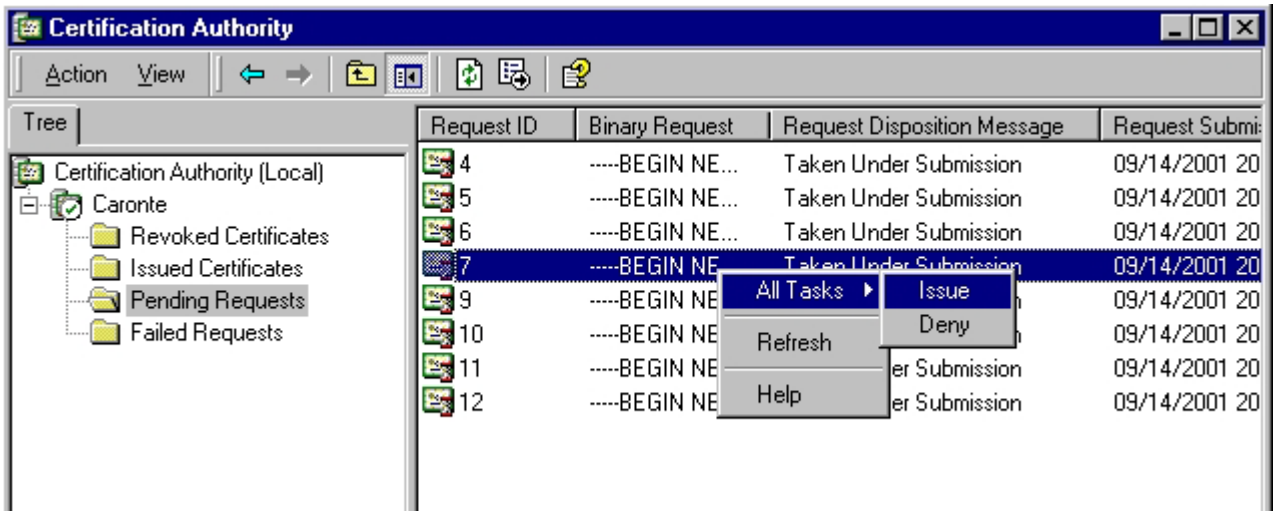


Haciendo click en el link “Instalar este certificado”, el certificado digital relacionado a la solicitud se instala en el entorno de seguridad del navegador de Internet que está siendo utilizado. El resultado de esta operación es indicado al usuario a través de un mensaje (aquí se muestra el resultado satisfactorio de la operación):



Procesamiento del certificado

Desde el lado del servidor, el personal responsable de la autoridad de certificación es el encargado de aceptar o rechazar las solicitudes de certificados digitales que se realizan. Para esto, la autoridad de certificación posee un listado de pedidos pendientes, sobre los cuales el personal puede actuar.



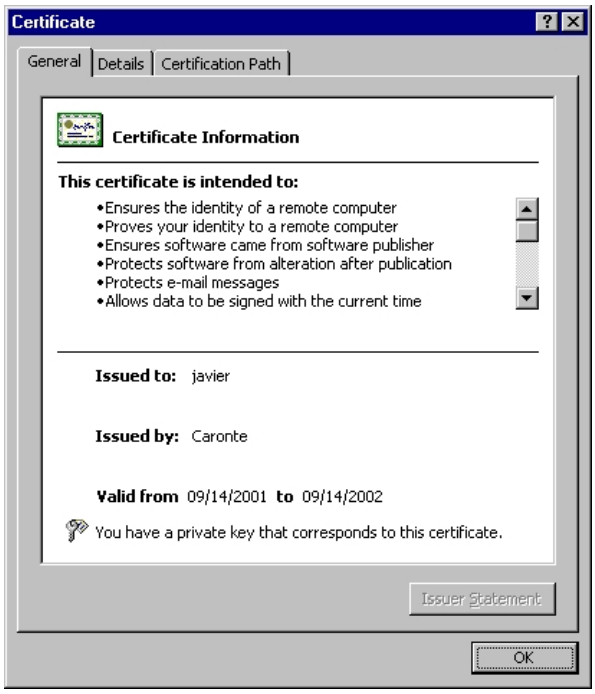
Automáticamente luego de tomar una acción sobre un pedido, el resultado está disponible mediante la interface web de la autoridad de certificación, permitiendo al usuario generador del pedido conocer el resultado del mismo. Del mismo modo se van actualizando las listas de la CA, pasando de "Pedidos pendientes" (*Pending requests*) a "Certificados otorgados (*Issued certificates*) o "Pedidos fallados" (*Failed requests*). Obviamente también se puede interactuar con los certificados otorgados para revocarlos y utilizar de esta manera el método de CRLs (listas de revocación) para manejar la seguridad de los mencionados certificados.

Nota: el nombre "Caronte" de la autoridad de certificación proviene de la mitología romana. Caronte era el barquero encargado de pasar los viajeros de una orilla a otra en los ríos del imperio de Plutón, el Ades.

Uso del certificado digital

El usuario puede corroborar la instalación del certificado digital llevada a cabo anteriormente, accediendo a él en los elementos de seguridad del navegador de Internet. El certificado se muestra visualmente en forma de credencial, con sus datos mas importantes (dependiendo del navegador de Internet) sin permitir la visualización de datos sensitivos (por ejemplo, si el certificado tiene una clave secreta asociada, esto solamente se indica a modo informativo, mientras que la clave no puede verse).

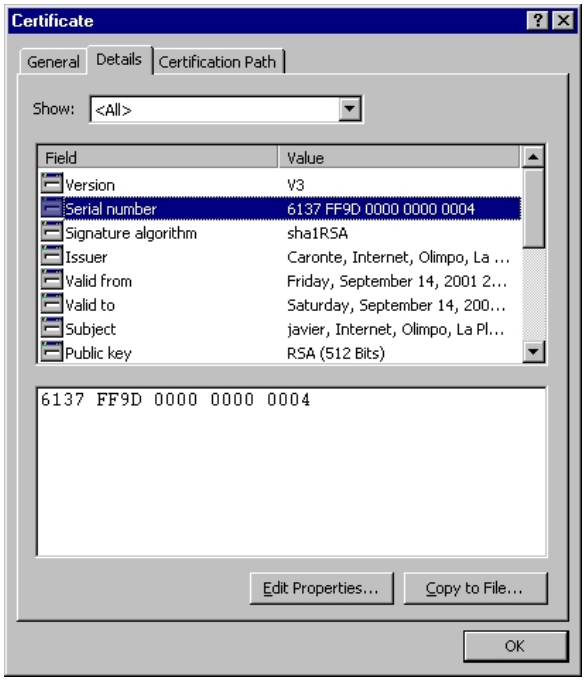
Normalmente, se obtiene una ventana donde se muestran los datos visibles del mismo, agrupados de la siguiente forma:

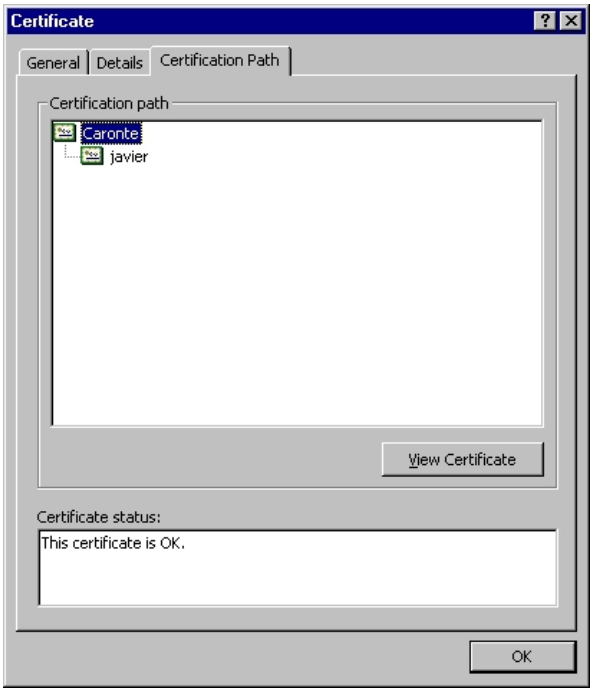


Primera pantalla de datos del certificado. Aquí se pueden ver los elementos generales del mismo, como por ejemplo: objetivos del certificado (utilidad, servicios para los que puede ser usado), destinatario, autoridad de certificación que lo emitió, período de validez. Adicionalmente se muestra una notificación que indica que se tiene asociada una clave secreta con el certificado.

En esta pantalla se pueden los valores para cada campo de la norma X.509 de certificados digitales (siendo un opcional el seleccionar los correspondientes a cual versión de la misma se listan). Entre los mas destacados se puede mencionar: número de serie, versión, algoritmo de firma, emisor, clave pública, información de CRL, etc.

Aquí también se presenta la oportunidad de exportar el certificado a un archivo físico de disco (Copiar a archivo, *Copy to file*), lo cual permite la portabilidad del certificado y utilizarlo en otra estación de trabajo.





Esta ventana muestra el "Camino de certificación" (*certification path*): en forma de árbol, indica la cadena de certificación de la cual depende el certificado. Si un certificado de alguna autoridad en la cadena no se encontrara o fuera inválido, el certificado en si mismo no es válido, y esto se indica gráficamente en este árbol (usualmente superponiendo una cruz roja sobre el nodo causante del problema).

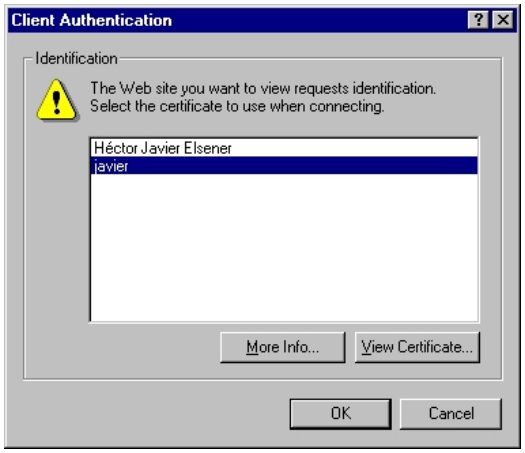
En la parte inferior se muestra una notificación del estado del certificado, su validez.

Registro del certificado en la base de datos

Luego de obtener el certificado, el usuario debe acceder a un sitio seguro utilizando la mencionada credencial. Esta operación puede ser automatizada, pero aquí la autoridad de certificación puede llegar a servir a otros fines con lo cual es conveniente no hacerlo.

Por lo tanto, el usuario interesado en ingresar al sitio seguro de la empresa debe primero registrar su certificado en la base de datos: esto lo lleva a cabo accediendo a una URL específica, e indicando el certificado que desea registrar. La función de esta URL es solamente leer los datos del certificado e ingresarlos en la base de datos, para que luego un administrador de seguridad habilite el mismo. Esto introduce un nivel de seguridad extra, ya que esta autorización no está relacionada con la autoridad de certificación y de las listas de revocación de la misma.

En el momento en que el usuario accede a esta URL de registro de certificados, el navegador presenta una lista de credenciales adecuadas (se requiere que se haga con un certificado digital, es un acceso seguro basado en SSL) de las cuales el usuario debe seleccionar cual es la que desea registrar en la base de datos para su posterior uso.



Ventana con lista de certificados digitales. En este caso se dispone de dos de ellos, de los cuales se selecciona el segundo para su uso.
Alternativamente, antes de ejecutar la selección, puede consultar el certificado para asegurarse de su finalidad.

Una vez indicado el certificado que se utilizará, el usuario pasa a la siguiente pantalla.

Para convalidar la credencial, el usuario debe indicar la clave secreta (que fue establecida por él en el momento de solicitar el certificado digital).
En este caso se presenta una opción para que el sistema operativo recuerde la contraseña y no sea requerida cuando se utilice nuevamente este certificado. Esta opción no es conveniente usarla, ya que si varias personas tienen acceso a la estación de trabajo, pueden usar el certificado aún sin conocer la clave secreta.

Estos pasos son necesarios en todo momento en que se usa el certificado, independientemente del destino elegido. A continuación se describe el acceso a la aplicación de lectura de certificados.

Luego de acceder, el usuario no debe ingresar ningún dato adicional y recibe una notificación del resultado del registro en la base de datos:

En el caso de no prosperar la operación de registro en la base de datos, el usuario recibe una notificación del motivo y la posibilidad de ejecutar nuevamente la acción.

En este instante la credencial se encuentra registrada en la base de datos en estado ”ingresada”, sin que aún sea posible operar con ella. El paso siguiente requiere de la intervención de personal administrativo encargado de habilitar para su uso al certificado digital en cuestión, o lo contrario, es decir, impedir su uso en el sitio asegurado. Mediante el “Administrador de Seguridad de Internet”, se puede realizar estas operaciones, sobre uno o varios certificados a la vez.

Uso del certificado para acceder a un sitio de Internet

Una vez más, el usuario puede ser notificado por correo electrónico o alguna otra vía directa, o simplemente puede chequear periódicamente el estado de su certificado (esta vez no con la autoridad de certificación, si no con un sitio especialmente establecido para ello y que refleje el estado de la base de datos).

En este caso la empresa provee una autoridad de certificación, con lo cual se convierte en una *proveedora* de certificados digitales: cualquier sitio ajeno a la misma que considere a la primera como “autoridad confiable” admitirá los usuarios registrados en ella y que posean una credencial válida. Estos sitios pueden basar su esquema de seguridad en las publicaciones periódicas de las listas de revocación de la autoridad de certificación, o bien implementar el esquema de protección en tiempo real que la empresa puede proveer como un producto adicional.

Servidor de web

Se configuró un servidor de web con los siguientes objetivos:

- Brindar una interface a través de la web para la solicitud, consulta y entrega de certificados digitales
- Colocar herramientas que permitan a los usuarios administrar sus certificados digitales.
- Colocar un sitio de prueba a ser accedido por los certificados digitales entregados y que será administrado por el agente de seguridad.

Por lo tanto, para brindar una interface web sobre la autoridad de certificación, fue necesario establecer un sitio sin restricciones de seguridad que sí necesitarían los otros sitios. Este primer caso, será accedido por http plano, sin ser necesario el uso del protocolo SSL o credenciales. La seguridad de este sitio podría incluir (en este caso no se implementó) una autenticación básica con usuario-palabra clave, previo registro del usuario y envío de la clave por correo electrónico.

En el segundo caso, se construyó otro sitio web al cual los usuarios accederán para realizar operaciones con sus certificados digitales. El acceso en este sitio es realizado utilizando SSL y certificados digitales (autenticación del cliente): el usuario ingresa con el certificado que será objeto de alguna operación con las herramientas que aquí se ofrecen. Este sitio está libre de la protección del agente de seguridad ya que las restricciones por éste impuestas llevarían a que no se puedan usar algunas herramientas (por ejemplo, no se podría acceder a la consulta del estado de un certificado que aún no ha sido ingresado en la base de datos). Las herramientas provistas son:

- Registro de certificado digital en la base de datos: el usuario ingresa su certificado a la base de datos para que luego pueda ser procesado por la empresa y eventualmente ser habilitado para su uso.
- Consulta de estado de certificado digital: el usuario consulta el estado de su certificado dentro de la empresa (si el mismo ha sido ingresado en la base de datos, si está habilitado o no, si ha sido suspendido). Este estado que el usuario consulta es distinto al estado propio del certificado, que involucra por ejemplo, su fecha de caducidad,.

En el tercer y último caso, se estableció un sitio de prueba con algunas páginas HTML, a modo de demostración del desarrollo completo. El acceso debe realizarse sobre SSL y con certificado digital, el cual será validado por el agente de seguridad en cada acceso que se realice, con las restricciones impuestas desde el Administrador de Seguridad de Internet.

En los tres casos presentados el nivel de seguridad establecido es distinto por lo que la configuración de cada uno de los sitios es distinta, a pesar que todos ellos forman parte del circuito de funcionamiento del desarrollo planteado. En la configuración de este servidor no se tuvieron en cuenta aspectos de performance, si no que se ha focalizado en los aspectos de seguridad del mismo.

Agente de seguridad

El Agente de seguridad es el componente encargado de comprobar en tiempo real el estado del certificado y demás restricciones de seguridad que puedan estar asociadas, permitiendo o rechazando accesos al servidor web, para un conjunto de sitios o URLs establecidas. Su funcionamiento es el siguiente:

1. En el momento de establecer la conexión, el usuario debe presentar su certificado digital.
2. En cada acceso (cada solicitud realizada al servidor de web) el agente de seguridad comprueba el estado del certificado en la base de datos, aceptando o rechazando el pedido del usuario en base a este chequeo. En este punto se pueden establecer otras restricciones que deberán ser satisfechas para permitir el acceso solicitado (estas restricciones son especificadas mediante el Administrador de Seguridad de Internet, mas adelante).
3. Al realizarse los chequeos en cada acceso (y no solamente en el momento de establecimiento de la sesión) se tiene un esquema que funciona en tiempo real. Si el estado del certificado es modificado, o nuevas restricciones de seguridad son impuestas (como las mencionadas anteriormente, direcciones IP, horarios, roles asociados al certificado / usuario, etc) en el próximo acceso (próximo *request* que el usuario efectué al servidor web) los chequeos se realizarán contra los nuevos datos.

En caso de rechazar el pedido del usuario, se cuenta con una lista configurable de errores “amigables” a mostrar al usuario. Esta lista consiste en una serie de pares de la forma error – URL de redireccionamiento, es decir que cuando se produzca un determinado error (por ejemplo, el certificado se encuentra momentáneamente suspendido) el Agente de seguridad redireccione la respuesta a una página HTML que contenga información útil para el usuario y no un error técnico poco comprensible por el mismo.

El Agente de seguridad está implementado como una extensión del servidor de web utilizado, como un filtro ISAPI (*Internet Server Application Program Interface*). Este filtro responde a cada pedido (*request*) HTTP que realiza el usuario, antes que el servidor de web, y tiene la posibilidad de rechazar cada uno de ellos en base a un examen de su contenido o, como en este caso, un análisis mas profundo del mismo

Dado que el Agente de seguridad interactúa con la base de datos para validar los accesos, fue necesario desarrollar una capa de acceso a datos para este componente. La misma fue implementada utilizando Visual Basic como lenguaje de desarrollo, con lo que se evitó la complejidad de desarrollarla en C++ y se completó con mayor rapidez. Al ser objetos ActiveX, esta capa de acceso a datos puede ser incorporada en cualquier lenguaje contenedor de estos, como C++, con un mínimo esfuerzo, y algunos componentes de ella son compartidos con el Administrador Visual de Certificados Digitales, por lo que fueron desarrollados sólo una vez. Otra ventaja de este desarrollo es que permite cambiar de base de datos sin necesidad de recodificar el Agente de Seguridad, basta solamente con implementar un nuevo módulo de acceso a datos (obviamente con la misma interface).

Registro de valores

Para evitar la no repudiación de transacciones válidas, se implementó un módulo de registro de valores.

Este módulo registra, para cada transacción que se desee, los valores indicados por el usuario en un formulario web, el certificado digital utilizado y el momento en el cual la acción se lleva a cabo. De esta manera se tiene un registro completo de las transacciones que un usuario ejecuta, asociadas a su certificado digital.

Si un usuario niega haber realizado una transacción, se emite un reporte con la fecha de tal acción, el certificado y los valores ingresados. En este instante se nota que el usuario debe aceptar la política de seguridad de la empresa, de tal manera que este reporte es irrefutable.

Este reporte no debe poder ser afectado internamente, y debe ser solamente accesible desde el sistema. Por lo tanto se almacenará encriptado en la base de datos.

Administrador de Seguridad de Internet (ASI)

El Administrador de Seguridad de Internet es una interface visual que permite la administración de los certificados digitales, sus restricciones de seguridad y demás datos de una forma visual, altamente ágil, de manera de permitir la creación de políticas de seguridad y diversas restricciones. Interactúa con la base de datos en la cual se basará el Agente de Seguridad para procesar los pedidos.

Los datos que administra se pueden dividir en:

- Certificados digitales: constituyen las credenciales que permiten autenticar a los usuarios que ingresan al sistema desde Internet, utilizando un navegador (o un cliente mas inteligente que establece una conexión HTTPS). En la base de datos no se almacena el certificado completo, si no un conjunto de datos que permiten su identificación en el momento de acceso. El certificado completo es almacenado por la autoridad de certificación (en el servidor) y por el navegador (en el cliente).
- Grupos de certificados digitales: los certificados digitales pueden ser agrupados y establecer restricciones comunes a todo el grupo, agilizando de esta manera la administración de los mismos (no es necesario procesarlos individualmente).
- Usuarios: opcionalmente se puede almacenar información sobre el usuario identificado con un determinado certificado digital.
- Permisos: cada permiso constituye una URL que será protegida por el sistema.
- Plantillas: constituye un agrupamiento de permisos, para luego poder asociarlos a un certificado digital y de esta manera brindar acceso a una funcionalidad compuesta por varios permisos. La asignación de permisos en forma individual a un certificado también está contemplada.
- Errores: permite establecer una URL de redireccionamiento para un pedido rechazado, estableciendo una página HTML a retornar al usuario a modo informativo.

El Administrador de Seguridad de Internet en si misma una aplicación segura, con un módulo de administración para usuarios de la misma. Inicialmente se tiene un usuario “administrador” que cuenta con todos los privilegios, y es el encargado de crear nuevos usuarios del ASI con (posiblemente) funcionalidad restringida. En el inicio de la aplicación, se solicita la identificación de un usuario, y de acuerdo a esto la aplicación permitirá realizar el conjunto de funciones a las que el usuario tiene acceso. De esta manera se evita que cualquier usuario modifique la información relacionada con los certificados digitales.

Para esta aplicación también se desarrolló una capa de acceso a datos, en forma separada (utilizando Visual Basic como leguaje de la misma), con algunos componentes en común para la análoga del Agente de Seguridad.

Restricciones adicionales

Para cada certificado, el Administrador de Seguridad de Internet permite establecer un conjunto de restricciones adicionales que serán validadas por el Agente de Seguridad. Ellas son:

1. Dirección IP de origen: permite establecer, para cada certificado, un conjunto de direcciones de Internet en la forma de direcciones IP, desde las cuales el acceso será admitido. En el caso de no especificarse dirección alguna, no se validará el origen del pedido.

2. Permisos: similarmente al caso anterior, se puede establecer un conjunto de destinos (en forma de URLs) que serán habilitados para el certificado, y fuera de este conjunto, cualquier acceso validado será rechazado.
3. Esquema de horarios: permite establecer un cronograma de horarios y/o días en los cuales el acceso con un determinado certificado será aceptado, y fuera de este cronograma, rechazado. Como base para esta validación se toman la hora y fecha del servidor de web, evitando así ambigüedades.

Todas estas restricciones pueden ser especificadas individualmente para cada certificado o en forma global para el grupo, como un medio de facilitar la administración de certificados digitales. Si son indicadas para un grupo de certificados digitales, se puede seleccionar la forma de aplicar estas restricciones sobre los certificados asociados al grupo de dos formas:

a) agregar las restricciones indicadas en el grupo a las restricciones que cada certificado posee
b) reemplazar las restricciones de cada certificado perteneciente al grupo por las especificadas, eliminando cualquier restricción de cada certificado asociado al grupo que haya sido establecida anteriormente.

Además, se puede establecer el estado de certificados digitales en forma grupal, lo que permite, por ejemplo, habilitar todos los certificados de un grupo.

Integración con el Agente de Seguridad

El Administrador de Seguridad de Internet interactúa fuertemente con el Agente de Seguridad en los siguientes puntos:

- Restricciones: establece las restricciones que servirán al Agente de Seguridad para establecer el resultado de cada intento de acceso, ya sean restricciones adicionales o modificaciones sobre el estado del certificado digital.
- Errores: permite establecer una URL de redireccionamiento a ser usada cuando el acceso es denegado, y esta URL puede variar de acuerdo al motivo del rechazo.

Pruebas

Se realizaron pruebas de la implementación en su conjunto, desde la obtención del certificado digital hasta el acceso al sitio de prueba. Los resultados fueron satisfactorios, en cuanto se logró establecer restricciones en tiempo real al acceso, sin depender de las listas de revocación de la autoridad de certificación.

Luego de establecer nuevas restricciones sobre un certificado digital en particular, el siguiente acceso realizado será validado con estas nuevas restricciones.

Se realizaron pruebas sobre un conjunto adicional de restricciones, impuestas desde el Administrador de Seguridad de Internet. Ellas son:

- Restricciones de origen: se establecieron dos orígenes distintos para pedidos, indicando como válido solo uno de ellos. Los intentos de acceso desde el otro origen fueron rechazados.
- Restricciones de destino: Para un mismo certificado, se establecieron como válidos sólo algunos destinos del sitio de prueba desplegado, y los intentos de acceso a estas URLs fueron procesados y aceptados, mientras que el acceso a las demás direcciones fue negado sistemáticamente.,
- Restricciones horarias: estableciendo un cronograma válido de accesos para un certificado de prueba, y modificando repetidamente la hora y/o el día en el servidor de web, se pudo comprobar que esta restricción también era validada en forma correcta.

Además de pruebas sobre el acceso con validaciones del Agente de Seguridad, se realizaron pruebas sobre el Administrador de Seguridad de Internet, para obtener una herramienta que permita establecer las restricciones los mas ágilmente posible, y que además permita la administración de certificados digitales, grupos, permisos y demás en forma coherente y segura.

Resultados obtenidos

Luego de la implementación de los distintos componentes se obtuvieron los siguientes resultados:

- Se consiguió experiencia y conocimientos en el área de seguridad informática:
    - Comprensión de la problemática asociada y estudio de diversos niveles de seguridad (gracias al estudio realizado previo a la implementación).
    - Conocimiento de la infraestructura de clave pública.
    - Uso del protocolo SSL.
    - Administración de seguridad en un servidor de web.
    - Administración e interacción con certificados digitales, utilizando una autoridad de certificación (administración) y aplicaciones propias (interacción, individualización, etc.)

- Se consiguió establecer restricciones para certificados digitales en tiempo real, independientemente de las listas de revocación y/o otros mecanismos provistos por la autoridad de certificación. Un certificado digital puede ser habilitado o deshabilitado momentáneamente para un determinado destino o conjunto de ellos, y este estado será comprobado en el próximo acceso que se realice utilizando el certificado en cuestión, sin necesidad de interacción con las listas de revocación de la autoridad de certificación.

- Adicionalmente, se logró establecer un conjunto de comprobaciones que se llevan a cabo en cada acceso que incrementan el nivel de seguridad, como por ejemplo: la dirección IP de origen de la conexión, el horario de acceso, el destino solicitado. Todas estas restricciones pueden modificarse en cualquier momento y hacerse efectivas instantáneamente.

- Se logró establecer un método para evitar la no repudiación de transacciones válidas: para cada formulario completado por el usuario se registra en la base de datos los valores ingresados y el certificado utilizado en la conexión, además del momento en que la acción se lleva a cabo.

- Se ha establecido una base para trabajos futuros que requieran la inclusión de mecanismos de seguridad de PKI, seguramente relacionados con comercio electrónico, aunque no se descartan otros tipos de aplicaciones. Del estudio de seguridad realizado se puede seleccionar el nivel adecuado de seguridad, o utilizar la implementación de restricciones para certificados digitales en el lado del servidor de ser necesario.

Trabajos futuros

- Incorporar tecnología PKI y desarrollos realizados a trabajos relacionados con Internet y/o comercio electrónico, que requieran seguridad informática en su implementación.

- Tecnología Java: como se describió anteriormente en la sección ***Selección de las herramientas***, este desarrollo puede realizarse con tecnología Java tomando como base la API denominada JAAS (*Java Authentication and Authorization Service*) y herramientas *open source*, como servidor de web Apache (www.apache.org), servidor de aplicaciones JBoss (www.jboss.org), sistema operativo Linux y herramientas de generación de certificados y firma digital tal como OpenSSL. Adhiriendo a estándares establecidos, se tendrá una implementación multiplataforma, que además de independizarse del sistema operativo subyacente logrará independencia del servidor de web y de aplicaciones.

- Objetivos de negocio: analizar los objetivos de negocios que se pueden plantear al transformar a la empresa en una proveedora de certificados digitales y de servicios de seguridad.

- Automatizar totalmente la entrega de certificados

Bibliografía

- **Seguridad para Intranet e Internet**

Autor: Amoroso, Edward

Editorial Prentice Hall

ISBN 8489660662

Requisitos de seguridad y *firewalls*. Amenazas de seguridad. Protocolos de Internet.

- **PKI implementing and managing e-security**

Autor: Nash, Andrew

ISBN 0072131233

Editorial McGraw-Hill

Criptografía, infraestructura de clave pública, certificados digitales.

- **Cryptography and network security: Principles and practices.**

Autor: William Stallings, segunda edición

ISBN: 0130914290

Editorial Prentice Hall.

Páginas web y artículos consultados:

Seguridad en Internet

- **Security Basics**. PricewaterhouesCoopers. Introducción a la problemática de seguridad,
- **CSI, Computer Security Institute.** http://www.gocsi.com/
  - A comprehensive view of web security. Diversos aspectos de la seguridad en Internet.
  - Information protection fundamentals. Ampliación del concepto de seguridad, teniendo en cuenta factores humanos, físicos y empresariales. http://www.gocsi.com/.
  - Should you insure against Internet risk? Conveniencia de implementar un esquema seguro.
  - How to evaluate security technology. Métodos para comprobar el nivel de seguridad alcanzado.
  - Intranet security. Diferencias entre seguridad de Intranet y seguridad de LAN/WAN.
  - The risk of PKI. Inconvenientes de PKI.
  - How to build a corporate PKI. Implementación de PKI.
- **Information Security** http://www.infosecuritymag.com/
  - Security from end to end. Seguridad en e-mail. http://www.infosecuritymag.com/march2000/emailsec.htm.
  - PKI grows up. Incremento del uso de PKI. http://www.infosecuritymag.com/nov99/infosec2000/ford.htm

  - Mastering the fundamentals, Part 3. Requerimientos de seguridad: políticas, arquitectura. http://www.infosecuritymag.com/march2000/fundamentals3.htm
  - Web of worries. Análisis del nivel de seguridad necesario para cada caso. http://infosecuritymag.com/apr2000/websecurity.htm
  - Web democracy. Ventajas competitivas de implementar seguridad. http://www.infosecuritymag.com/may2000/ec_does_it.htm
- **Security issues in WWW.** Características de seguridad de distintos protocolos, navegadores, herramientas de seguridad. http://nsi.org/Library/Internet/security.htm
- **Delitos en Internet.** Distintas consideraciones legales de los delitos en Internet. http://www.onnet.es
- **Computer security and the law.** Gary S. Morris. Seguridad informática desde el punto legal. http://www.alw.nih.gov/security/first/papers/legal/cslaw.txt
- **The NCSA guide to enterprise security.** McGraw Hill (1996). Implementación de políticas de seguridad empresariales.
- **Identification, authentication and authorization on the WWW.** Descripción de distintos productos de seguridad informática. http://secinf.net/info/www/iaa/iaawww.shtml

Palabra clave

- **Center for academic computing password policy**. Policy password. Conceptos de palabra clave, recomendaciones, políticas. https://courseware.vt.edu/users/marchany/SANS2002/day5/psu%20password.htm
- **Password**. Técnicas para crear palabras clave efectivas. http://netiv.allegro.com.au/

Encriptación

- **Information Security**. http://www.infosecuritymag.com/
  - Fast... & secure. http://www.infosecuritymag.com/jan2000/cover.htm. Descripción de los aceleradores de criptografía, su funcionamiento y un caso de estudio.
  - Mastering the fundamentals, Part 1. Soluciones de problemas de seguridad utilizando encriptación. http://www.infosecuritymag.com/jan2000/fundamentals1.htm
  - SSL Crunch time. http://www.infosecuritymag.com/oct99/ssl.htm. Análisis del rendimiento del protocolo SSL.
- **The Secure HyperText Transfer Protocol.** E. Rescorla y A. Shiffman. Descripción de HTTPS, funcionamiento y características.
- **Introduction to SSL.** Descripción del protocolo SSL, funcionamiento, estableciemto de conexiones. http://developer.netscape.com/docs

Firewalls

- **MSDN. Web workshop: Fight fire with firewalls.** Introducción a firewalls, funcionamiento, características. http://msdn.microsoft.com/workshop/server/proxy/server072798.asp
- **Firewalls and Internet Security.** Introducción a los conceptos de firewall. http://www.soscorp.com
- **What is a firewall?** Funcionamiento de un firewall. http://www.real.com
- **General firewall white paper.** Discusión en general de los firewall para Windows NT. http://www.tec-ref.com
- **Firewalls FAQ.** Introducción a firewalls, seguridad perimetral, tipos de ataques, implementación, distintos protocolos. http://www.clark.net/pub/mjrpubs/fwfaq/ y http://www.interhack.net/pubs/fwfaq/.
- **Basic firewall designs.** Arquitectura de firewalls, seguridad perimetral, definiciones. http://netiv.allegro.com/CounterMeasures/LanServices/Firewalls/.
- **Internet firewall policy.** Análisis de las distintas opciones disponuibles en un firewall. Políticas.
- **Firewall?** Definiciones. http://www.clarkson.edu/~parrybj/firewall/index.html.
- **Information Security**. http://www.infosecuritymag.com/
  - Firewalls: Are we asking too much? Incorporación de nuevos servicios a un firewall manteniendo el nivel de seguridad. http://www.infosecuritymag.com/may99/cover.htm
  - Firewall futures. Discusión del mercado del firewall en el pasado, presente y futuro. http://www.infosecuritymag.com/may99/market.htm
  - Mastering the fundamentals, Part 2. Seguridad perimetral de una empresa, vulnerabilidades. http://www.infosecuritymag.com/feb2000/fundamentals2.htm
- **CSI, Computer Security Institute.** Intrusion detection. Introducción a los sitemas de detección de intrusos http://www.gocsi.com/.
- **An introduction to firewalls.** SOS Corporation. Introducción a firewalls, tipos, limitaciones. http://www.soscorp.com/SOS.html

Kerberos

- **Kerberos An authentication service for Computer Networks.** B. Cliffbord Neumann y Theodore Ts'o. Descripción de Kerberos, comparación con PKI. http://www.isi.edu/gost/publications/kerberos-neuman-tso.html
- **Kerberos FAQ.** Distintas cuastiones sobre Kerberos. http://www.mit.edu/kerberos/www
- **Kerberos RFC 1510.** Descripción de Kerberos en profundidad
- **Limitation of the Kerberos authentication system.** Steven M. Bellovin y Michael Merrit. Análisis y descripción de las limitaciones de Kerberos, debilidades y limitaciones. Ejemplos de ataques exitosos al mencionado sistema. https://cgi.cs.duke.edu/~geha/cps296/HyperNews1.10/get.cgi/Feb7.html?inline=1&nogifs
- **Windows 2000 Kerberos Authentication.** Descripción de Kerberos, interoperabilidad con Windows 2000. Paper obtenido de http://www.microsoft.com
- **Deploying Kerberos for Large organizations.** CyberSafe Corporation. Análisis del proceso completo de despliegue de Kerberos, problemas y soluciones, políticas, procedimientos, costos.

PKI – Certificados digitales

- **VeriSign** http://www.verisign.com
  - **Guide to securing Intranets and Extranets**. Verisign. Objetivos de seguridad para Extranets e Intranets, certificados digitales, PKI.
  - **Digital ID Introduction.** Introducción y descripción de identificadores digitales, firmas digitales, autenticación. http://www.verisign.com
- **Public Key Authentication Framework**. http://www.ozemail.com.au/~firstpr/crypto/pkaftute.htm. Criptografía de clave pública, PKI.
- **Microsoft Certificate Services.** Publicado por Microsoft Press. Descripció de los certificados digitales, uso de los mismos, autoridades, etc (capítulos 6 y 28).
- **Information security.** PKI: Be careful what ypu wish for...Costo de implementación de una solución de infraestructura de PKI. http://www.infosecurity.com/may2000/pki.htm
- **Public Key Infraestructure from VeriSign.** Factores de éxito en la implementación de PKI, enfoque correcto del mismo. http://www.verisign.com
- **Public Key Infraestructure.** Analogías de un esquema de PKI con el mundo real. http://www.id2.se/whitepapers
- **Information Security.** PKI: The mith, the magic and the reality. Problemas de la infraestructura de PKI, análisis de la conveniencia de su implementación. http://www.infosecurity.com/jun99/pki.htm
- **Windows 2000 Certificate Services.** Diseño y despliegue de una ionfraestructura de PKI basada en servicios de certificación de Windows 2000. http://www.microsoft.com/TechNet/win2000/2000cert.asp
- **Web security: putting a secure front end on your applications.** Keith Brown. Obtención de certificados digitales, revocación, cadenas de certificación. http://msdn.microsoft.com/msdnmag/issues/0600/websecure/default.aspx
- **Introduction to Public Key Cryptography.** Encriptación asimétrica utilizando claves públicas, autenticación mediante el uso de certificados digitales.
- **X.509v3 Certificate.** Petra Glökner. Descripción de certificados digitales de la norma X.509.

Contenido del CD

El CD adjunto contiene:

- Documento conteniendo la presente tesis y su presentación esquemática.
- Dentro de la carpeta BIN se encuentran los desarrollos realizados:
  - `bd`: contiene el script de creación de la base de datos (`tesis_bd.sql`). Para ser usada, debe existir un *DSN (Data Source Name)* llamado `DSN_TESIS` que referencie a la base de datos creada, con usuario `sa` y sin password.
  - `DataAccess`: contiene el desarrollo de la capa de datos, que debe ser instalada previa al uso del esquema de seguridad y luego de configurar la base de datos.
  - `ASI`: contiene el desarrollo de la aplicación denominada *Administrador de Seguridad de Internet.* Para su uso es necesario ejecutar `setup.exe` (ubicado en `instalar`) y previamente configurar la base de datos).
  - `FILTRO`: contiene el desarrollo identificado como *Agente de Seguridad*. Para su uso, debe ser incluido como **ISAPI Filter** en el sitio que se desee asegurar (dentro de un servidor web IIS). Este sitio debe ser configurado para requerir conexiones mediante SSL y requerir certificados digitales al cliente.
  - Aplicaciones utilitarias, que hacen al funcionamiento del esquema y que deben ser incluídas como directorios dentro del servidor de web. Estas son:
    - `LeerCertificado`, que permite ingresar un certificado en la base de datos desde una conexión web desde un navegador.
    - `EstadoCertificado`, que permite consultar desde un navegador el estado de un certificado digital.
    - `LOG`, un ejemplo sobre como implementar un mecanismo de control para evitar la no-repudiación.
  - `PRUEBA`, sitio de prueba con 3 niveles de acceso, para verificar el funcionamiento del sistema.
  - `certsvr`, Circuito completo de solicitud/entrega de los certificados (un conjunto de páginas *asp* que reemplazan a las páginas que por defecto incluye la autoridad de certificación elegida).