Resumen de seguridad en redes

Tema I: autenticación de mensajes (firma digital)

Definición con mis palabras:
La autenticación de mensajes mediante firma digital garantiza que un mensaje proviene realmente del remitente (autenticidad) y no ha sido alterado (integridad). Se logra usando algoritmos criptográficos y claves privadas/públicas. Solo el emisor puede firmar, y cualquier receptor puede verificar la firma.

Crítica a los métodos de firma es:
se mezclan con frecuencia dos funciones distintas, la autentificación y la confidencialidad, a veces se requiere únicamente la autenticación y no la confidencialidad

Cómo funciona:
Este esquema usa una función hash unidireccional que convierte un texto plano (de cualquier tamaño) en una cadena de bits de longitud fija llamada resumen de mensaje (message digest). Esta función tiene 4 propiedades importantes:

1. Dado P, es fácil de calcular MD(P)
2. Dado MD(P), es en efecto imposible de encontrar P
3. Dado P, nadie puede encontrar P' de tal manera que MD(P') = MD(P)
4. Un cambio en la entrada de incluso 1 bit produce una salida muy diferente

Explicando lo anterior:

- *P representa el mensaje original o texto plano*
- *MD(P) significa el resumen de mensaje generado a partir del texto usando la función hash*
- *P' es un mensaje diferente al original pero con la misma longitud o parecida*

Para lograr esto, la función hash debe tener al menos 128 bits y truncar bits cuidadosamente. Esto permite ahorrar tiempo y espacio al encriptar y enviar mensajes, porque solo se transmite el resumen en lugar del texto completo.

Algoritmo de hash seguro:
SHA-1 (Secure Hash Algorithm 1) es una función de resumen de mensaje desarrollada por el NIST en 1993. Procesa bloques de 512 bits y produce un hash de 160 bits. Opera truncando bits de manera compleja para que cada bit de salida dependa de cada bit de entrada.
Cuando Alice envía a Bob un mensaje firmado (pero no secreto), aplica SHA-1 al mensaje para generar un hash de 160 bits. Luego, firma ese hash con su clave privada RSA y lo envía junto con el mensaje. Bob calcula el hash por su cuenta y usa la clave pública de Alice para verificar la firma. Si los hashes coinciden, el mensaje es válido y no ha sido alterado.
Este esquema se usa cuando la integridad del mensaje es importante, aunque su contenido no sea secreto. Es eficiente y permite detectar modificaciones con alta probabilidad.
Se han creado versiones más seguras de SHA-1: SHA-2, que incluye SHA-224, SHA-256, SHA-384 y SHA-512. La más usada hoy es SHA-256, que genera un hash de 256 bits y es resistente a colisiones y ataques de preimagen. Se usa en firmas digitales, certificados SSL y verificación de integridad de datos.

MD5 – Resumen simplificado:

MD5 (Rivest, 1992) rellena el mensaje hasta 448 bits (módulo 512), luego añade la longitud original como un entero de 64 bits, para obtener un total múltiplo de 512 bits. Cada bloque de 512 bits se mezcla con un búfer de 128 bits usando una tabla derivada del seno, para evitar sospechas de puertas traseras. Este proceso continúa hasta procesar todo el mensaje. El contenido final del búfer es el resumen de mensaje (hash).

explicado de mejor manera lo anterior: MD5 es un método que convierte un mensaje en un código único de 128 bits. Primero, rellena el mensaje para que tenga el tamaño adecuado, luego agrega su longitud original. Después, divide el mensaje en partes de 512 bits y las procesa con un búfer (una memoria temporal) usando una tabla basada en el seno para evitar que se detecten patrones. Al final, el contenido del búfer es el resultado del hash.

Tras años de uso, se descubrió que MD5 permite generar colisiones (mensajes distintos con el mismo hash), lo cual invalida su seguridad. Por eso, MD5 está considerado obsoleto y no debe usarse en nuevos sistemas, aunque todavía se encuentra en algunos ya existentes.

Certificados – Resumen simplificado:

Una solución inicial para distribuir claves públicas era usar un centro de distribución de claves (KDC) siempre en línea, pero esto no escala y puede fallar, afectando toda la seguridad.

Por eso se usa otro enfoque: las Autoridades de Certificación (CA), que certifican claves públicas en lugar de distribuirlas directamente. La CA firma la clave pública con su propia clave privada, generando un certificado.

Por ejemplo, Bob va a una CA con su clave pública y una identificación oficial. La CA verifica su identidad y le entrega un certificado firmado (con hash SHA-1 firmado). Bob puede publicar este certificado (por ejemplo, en su web) para que cualquiera lo use.

El objetivo principal de un certificado es enlazar una clave pública con una entidad (persona, empresa, etc.). Los certificados no son secretos.

Además, un certificado puede enlazar una clave pública con un atributo, como "mayor de 18 años". El receptor puede comprobar ese atributo cifrando un número aleatorio con la clave pública; si el dueño lo puede descifrar, prueba que posee la clave privada.

También se usan en sistemas distribuidos: un certificado puede indicar qué métodos puede ejecutar un cliente sobre un objeto (como un mapa de permisos). Esto permite controlar el acceso sin necesidad de conocer la identidad del usuario.

Explicando de mejor manera lo anterior: Antes se usaba un centro de distribución de claves (KDC) para compartir claves públicas, pero no era escalable ni seguro. Por eso ahora se usan las Autoridades de Certificación (CA), que verifican identidades y firman claves públicas con su propia clave privada, creando un certificado.

Por ejemplo, Bob lleva su clave pública e iden tificación a una CA. Esta verifica su identidad y le da un certificado firmado. Bob puede publicarlo para que otros usen su clave de forma confiable.

Los certificados no son secretos y sirven para vincular una clave pública con una persona o atributo (como ser mayor de 18). También pueden usarse para definir permisos en sistemas distribuidos, sin saber quién es el usuario.

Criptografía cuántica – Resumen simplificado:

La criptografía cuántica se basa en las leyes de la física. En concreto se basa en los principios de la mecánica cuántica

- las partículas son inherentemente inciertas: pueden existir en varios lugares al mismo tiempo
- los fotones se pueden medir aleatoriamente en posiciones binarias: se pueden configurar para que los espines puedan serie como una contraparte binaria para los unos y ceros de los sistemas computacionales clásicos
- no se puede medir un sistema cuántico sin alterarlo:
- las partículas pueden clonarse parcialmente, pero nunca totalmente:

El término espines se refiere a una propiedad cuántica de las partículas, como los fotones, que se utiliza para codificar información. Son una forma de describir el estado cuántico de una partícula, y en este contexto, se aprovechan para transmitir claves criptográficas de manera segura.

¿Cómo se relacionan los espines con la criptografía cuántica?

- polarización de fotones: los fotones pueden tener diferentes estados de polarización (vertical, horizontal, diagonal, etc) estos pueden interpretarse como bits de información (0 y1)
- Codificación de claves: En protocolos como BB84, los espines o polarizaciones de los fotones se utilizan para codificar claves criptográficas. Si alguien intenta interceptar los fotones, el acto de medirlos altera su estado debido al principio de incertidumbre cuántica, lo que permite detectar la intrusión.
- seguridad basada en física: en vez de basarse en matemáticas, se basa en física

Protocolo BB84 – Resumen estructurado

El BB84 es un protocolo de criptografía cuántica creado por Charles Bennett y Gilles Brassard en 1984. Utiliza polarización de fotones para transmitir información de forma segura mediante alfabetos cuánticos aleatorios. Es el primer protocolo práctico de distribución de claves cuánticas (QKD, Quantum Key Distribution).

Funcionamiento del protocolo BB84:

1. Generación y envío de cúbits:
    a. El emisor (Alice) elige un bit (0 o 1) y un alfabeto cuántico (rectilíneo o diagonal) al azar.
    b. Codifica ese bit en un fotón con la polarización correspondiente.
    c. Envía los cúbits uno a uno por un canal cuántico.
2. Medición por el receptor:
    a. El receptor (Bob) mide cada cúbit recibido usando alfabetos aleatorios, ya que no conoce el original.
    b. La medición colapsa el estado cuántico a un valor clásico (0 o 1).
3. Comunicación clásica:
    a. Una vez cerrada la transmisión cuántica, Alice y Bob usan un canal clásico para comparar los alfabetos usados.
    b. Se descartan los bits en los que usaron alfabetos distintos.
    c. El resto forma la clave secreta compartida.

Seguridad del protocolo BB84:

- Imposibilidad de clonar cúbits y el colapso al medir impiden que un atacante lea la clave sin ser detectado.

- Si un espía (Eve) intercepta un cúbit, colapsará su estado cuántico y generará errores que Alice y Bob detectarán.
- En un ataque de tipo "Man-in-the-Middle" cuántico (interceptar y reenviar):
    - Eve mide el cúbit, lo reemplaza por uno nuevo y lo envía a Bob.
    - Bob lo recibe y lo mide, pero puede ser incorrecto.
    - Los errores revelarán la presencia del espía si el número de cúbits afectados es significativo.

Protocolo B92 – Resumen estructurado

El protocolo B92 fue desarrollado por Charles Bennett en 1992 como una simplificación del BB84. Está diseñado para la distribución segura de claves cuánticas entre dos partes (Alice y Bob), utilizando solo dos estados cuánticos no ortogonales.

Características principales:

1. Uso de estados no ortogonales:
    a. Solo se utilizan dos estados cuánticos que no pueden distinguirse perfectamente entre sí. Ejemplo típico:
        i. Bit 0 → fotón con polarización vertical (0°)
        ii. Bit 1 → fotón con polarización diagonal (45°)

2. Detección de un espía:
    a. Un atacante (Eve) no puede medir sin alterar el estado del fotón (principio de incertidumbre cuántica).
    b. Esto provoca errores detectables por Alice y Bob, lo que indica una posible intrusión.
3. Simetría simplificada:
    a. Bob utiliza una sola base de medición, a diferencia del BB84 que alterna entre dos bases.
    b. Esto reduce la complejidad del proceso de decodificación.
4. Autenticación del canal clásico:
    a. Se requiere un canal clásico autenticado para evitar ataques del tipo MITM (Man-in-the-Middle).

Ventajas del protocolo B92:

- Estructura más simple y eficiente que BB84.
- Usa menos recursos físicos (solo dos estados cuánticos).
- Fácil de implementar con hardware básico de polarización.

Desventajas del protocolo B92:

- Menor redundancia → lo hace más vulnerable a errores del canal cuántico.
- Mayor sensibilidad al ruido y la pérdida de fotones.
- Puede ser menos robusto ante ataques sofisticados comparado con BB84.

Tema II: Aplicaciones de seguridad en redes

Tipo de aplicaciones de seguridad en redes:

- Aplicaciones de autenticación: APP de autenticación seguro y fácil de verificación de identidad que funciona mediante la generación de códigos numéricos que los usuarios introducen junto con sus credenciales para acceder a una cuenta.

- Software antivirus y antimalware: Detecta y elimina software malicioso, como virus y malware, que pueden infectar dispositivos y comprometer la red.
- Firewalls: Son programas de software o dispositivos de hardware que controlan el tráfico de red entrante y saliente, bloqueando o permitiendo el acceso según reglas predefinidas.
- Sistemas de detección y prevención de intrusiones (IDS/IPS): Monitorean la red en busca de actividad sospechosa y pueden tomar medidas para prevenir o bloquear ataques.
- Análisis de red: Evalúa la red en busca de vulnerabilidades y configura las políticas de seguridad para protegerla.

tipo de aplicaciones de seguridad en redes informáticas:

- soluciones de seguridad avanzadas (XDR): Ofrecen una visibilidad integral de la red, detectan amenazas avanzadas y automatizan la respuesta a ellas.
- Gestión de vulnerabilidades:Identifica y gestiona vulnerabilidades en la red y sistemas para reducir el riesgo de ataques.
- redes privadas virtuales (VPN): Cifran el tráfico de red y enmascaran la dirección IP para proteger la privacidad y seguridad de la comunicación.
- Gestión de acceso a la red (NAC): Controla el acceso a la red, asegurando que solo los dispositivos autorizados puedan conectarse.
- Gestor de contraseñas: sirven para almacenar nuestras credenciales (usuarios, contraseñas, sitios web a los que corresponden, etc.) en una base de datos cifrada mediante una contraseña maestra.

tipos de métodos de autenticación:

- autenticación basada en contraseñas
- autenticación basada en certificados
- autenticación biométrica
- Autenticación basada en tokens: En la autenticación basada en tokens, tanto el dispositivo como el sistema generan un nuevo número singular llamado PIN temporal de un solo uso (TOTP) cada x segundos. Si los números coinciden, el sistema comprueba que el usuario tiene el dispositivo.
- contraseñas de un solo uso:
- notificaciones push:
- autenticación multifactor (MFA): Una de las mejores formas de reducir el riesgo de vulneración de cuentas es requerir dos o más métodos de autenticación, entre los que se pueden incluir cualquiera de los enumerados anteriormente. Un procedimiento recomendado eficaz es requerir dos de los siguientes métodos:
    - Algo que el usuario conozca, normalmente una contraseña.
    - Algo que el usuario posea, como un dispositivo de confianza que no se pueda duplicar fácilmente; por ejemplo, un teléfono o token de hardware.
    - Algo que el usuario sea, como una huella dactilar o escáner facial.

autenticación PPP: La autenticación PPP (Point-to-Point Protocol) es un proceso que verifica la identidad de los emisores de llamadas que intentan establecer un enlace PPP con un equipo. Este proceso es opcional, pero cuando se utiliza, asegura que solo los usuarios autorizados puedan establecer conexiones a la red. La autenticación PPP no proporciona confidencialidad de datos, por lo que para cuestiones de confidencialidad se deben utilizar métodos de cifrado adicionales.

Protocolos de autenticación en PPP:

PPP admite dos protocolos principales para autenticar a los usuarios:

1. PAP (Password Authentication Protocol)

- Funcionamiento: Similar al comando login de UNIX.
- Proceso: El emisor (cliente) envía su nombre de usuario y contraseña en texto plano al autenticador.
- Autenticación: Unidireccional (solo el cliente se autentica ante el servidor).
- Seguridad:
    - Muy baja: las credenciales viajan sin cifrar.
    - Vulnerable a ataques de escucha (sniffing).
- Uso recomendado: Solo en redes seguras y controladas.

2. CHAP (Challenge Handshake Authentication Protocol)

- Funcionamiento: Basado en el método de desafío y respuesta.
- Proceso:
    - El autenticador envía un desafío aleatorio con un ID.
    - El cliente genera una respuesta mediante un hash del ID, el desafío y su secreto compartido.
    - El autenticador verifica que la respuesta sea válida.
- Autenticación: Puede repetirse periódicamente durante la conexión.
- Seguridad:
    - Mucho más segura que PAP.
    - Las contraseñas no se transmiten directamente.
    - Protege contra ataques de reproducción.

<u>Base de datos de autenticación (secrets)</u>

Ambos protocolos utilizan una base de datos que contiene:

- Nombres de usuario
- Contraseñas o secretos compartidos
- Reglas de acceso (quién puede conectarse y desde dónde)

Comparación rápida: PAP vs CHAP

Característica	PAP	CHAP
Seguridad	Baja	Alta
Método de autenticación	Contraseña directa	Desafío / Respuesta
Cifrado de credenciales	No	Sí
Autenticación continua	Solo al inicio	Periódica
Protección contra ataques	Vulnerable	Alta protección
Uso recomendado	Solo en redes seguras	Para redes con mayor riesgo

IPsec (Internet Protocol Security)

IPsec es un conjunto de protocolos criptográficos que protegen las comunicaciones en la capa IP, proporcionando confidencialidad, integridad y autenticación. Se define principalmente en los RFC 2407 y RFC 2408.

Características generales de IPsec

- IP significa "Internet Protocol" y sec, "seguro".
- Protege datos a nivel de red (capa 3 del modelo OSI).
- Utiliza asociaciones de seguridad (SA) para establecer comunicaciones seguras entre hosts.
- Las SA pueden configurarse de forma manual o dinámica.
- IPsec admite dos modos de operación:
    - Modo de Transporte
    - Modo Túnel

Modos de seguridad de IPsec

1. Modo Túnel

- Utilizado principalmente entre enrutadores.
- Encapsula y cifra el paquete IP completo (encabezado + carga útil).
- Agrega un nuevo encabezado IP para su enrutamiento.
- Ideal para conexiones VPN entre redes a través de Internet.

2. Modo de Transporte

- Utilizado generalmente entre hosts finales.
- Solo cifra la carga útil, manteniendo el encabezado IP original sin cifrar.
- Permite que los routers intermedios conozcan el destino del paquete.
- Más eficiente, pero menos confidencial que el modo túnel.

Administración de claves en IPsec

Clave Manual

- La configuración se hace manualmente en ambos extremos.
- Apropiado solo para redes pequeñas y estables.
- Difícil de escalar y propenso a errores o compromisos si la clave se transmite insegura.

Clave Automática (IKE - Internet Key Exchange)

- Protocolo que automatiza la generación y negociación de claves y SA.
- Usa:
    - Clave previamente compartida (pre-shared key)
    - Certificados digitales

Intercambio de claves: Diffie-Hellman

- Permite generar un valor secreto compartido entre dos partes sin enviarlo por la red.
- Puede implementarse en software o hardware.
- Base fundamental para la negociación de claves en IKE.

Protocolos de Seguridad IPsec

1. AH (Authentication Header)

- Proporciona:
  - Autenticación de origen
  - Integridad de datos
- No cifra los datos.
- Utiliza HMAC con algoritmos hash (MD5 o SHA).

2. ESP (Encapsulating Security Payload)

- Proporciona:
  - Cifrado de datos
  - Autenticación de origen (opcional)
- Puede cifrar toda la carga útil y/o autenticarla.
- Algoritmos soportados:
  - Cifrado: DES, Triple DES, AES
  - Autenticación: MD5, SHA

ACL (Listas de Control de Acceso IP)

Una ACL es una serie de comandos que determinan si un router reenvía o descarta paquetes según la información en su encabezado.

Funciones principales de una ACL:

- **Limitar el tráfico de red** para mejorar el rendimiento. Por ejemplo, si no se permite tráfico de video, una ACL puede bloquearlo para reducir la carga.

- **Controlar el flujo de tráfico**, como restringir actualizaciones de routing para asegurar que provienen de fuentes conocidas.

- **Proporcionar seguridad básica**, permitiendo que ciertos hosts accedan a partes específicas de la red, mientras se bloquea el acceso a otros.

- **Filtrar tráfico por tipo**, como permitir correo electrónico pero bloquear Telnet.

- **Restringir el acceso a servicios**, permitiendo o denegando el uso de archivos como FTP o HTTP.

Filtrado de paquetes

Una ACL es una lista secuencial de instrucciones permit (permitir) o deny (denegar), llamadas ACE (Access Control Entries). Cuando un paquete atraviesa una interfaz con ACL, el router compara su información con cada ACE, en orden, hasta encontrar una coincidencia. Este proceso se llama **filtrado de paquetes**.

El filtrado se aplica al analizar los paquetes entrantes y salientes y decidir si se transfieren o descartan, con base en criterios específicos. Puede realizarse en la **capa 3** (red) o **capa 4** (transporte) del modelo OSI.

Diferencia entre ACL y firewall

Ambos controlan el acceso a recursos de red, pero:

- **ACL**: Lista de reglas en routers o switches, que permiten o deniegan tráfico basado en criterios simples.

- **Firewall**: Dispositivo especializado en seguridad, que analiza y filtra el tráfico según reglas más avanzadas y específicas.

Tema III: Autenticación de Aplicaciones

Un Kerberos es un sistema o enrutador que proporciona una puerta de enlace entre los usuarios e Internet. Por lo tanto, ayuda a evitar que los ciberatacantes ingresen a una red privada. Es un servidor denominado “intermediario”, porque está entre los usuarios finales y las páginas web que visitan en línea.
En nuestro mundo, Kerberos es el protocolo de autenticación de red informática desarrollado inicialmente en la década de 1980 por científicos informáticos del MIT. La idea detrás de Kerberos es autenticar a los usuarios y evitar que las contraseñas se envíen por Internet.

Kerberos utiliza criptografía de clave simétrica y un centro de distribución de claves (KDC) para autenticar y verificar las identidades de los usuarios. Un KDC implica tres aspectos:

1. Un servidor de concesión de tickets (Ticket Granting Server - TGS) que conecta al usuario con el servidor de servicio (SS)

2. Una base de datos de Kerberos que almacena la contraseña y la identificación de todos los usuarios verificados

3. Un servidor de autenticación (AS) que realiza la autenticación inicial

Durante la autenticación, Kerberos almacena el ticket específico para cada sesión en el dispositivo del usuario final. En lugar de una contraseña, un servicio consciente de Kerberos busca este ticket.

La autenticación Kerberos es un proceso de varios pasos que consta de los siguientes

componentes:

1. El cliente que inicia la necesidad de una solicitud de servicio en nombre del usuario

2. El servidor, que aloja el servicio al que el usuario necesita acceso

3. AS, que realiza la autenticación del cliente. Si la autenticación se realiza sin problemas, se emite al cliente un ticket de concesión de tickets (TGT) o un token de autenticación de usuario, que es prueba de que el cliente ha sido autenticado.

4. El KDC y sus tres componentes: el AS, el TGS y la base de datos de Kerberos

5. La aplicación TGS que emite los tickets de servicio

El flujo del protocolo de Kerberos incluye tres claves secretas: hash cliente/usuario, clave secreta TGS y clave secreta SS.

1. **Solicitud de autenticación inicial del cliente:** El cliente inicia sesión y solicita un TGT al KDC de Kerberos.
2. **Verificación de credenciales del cliente:** El KDC verifica al cliente y al TGS en la base de datos. Si ambos están, el AS genera una clave de sesión (SK1) cifrada con la clave secreta del usuario y un TGT que incluye datos del cliente. El TGT es cifrado con la clave secreta del TGS.
3. **Descifrado de mensajes:** El cliente usa su clave secreta para obtener el TGT y SK1, luego genera un autenticador para validar al TGS.
4. **Solicitud de acceso mediante el TGT:** El cliente envía al TGS el TGT y el autenticador.
5. **Creación de ticket para el servidor de archivos:** El TGS descifra el TGT, valida al cliente y la vigencia del ticket. Si es correcto, genera una nueva clave de sesión (SK2) y un ticket de servicio cifrado con la clave secreta del servidor. SK2 y el ticket de servicio son cifrados con SK1 y enviados al cliente.
6. **Autenticación mediante el ticket de archivo:** El cliente usa SK1 para extraer SK2 y genera un nuevo autenticador cifrado con SK2, que luego envía al servidor junto al ticket de servicio.
7. **Descifrado y autenticación del servidor objetivo:** El servidor descifra el ticket con su clave secreta y valida el autenticador usando SK2. Si todo coincide, el servidor responde confirmando la autenticación mutua.

(PGP Privacidad Bastante Buena, del inglés Pretty Good Privacy), es un programa de seguridad que se utiliza para cifrar y descifrar correos electrónicos y autenticar mensajes de correo electrónico a través de firmas digitales y cifrado de archivos, es ampliamente utilizado hoy en día.

PGP fue uno de los primeros software de criptografía de clave pública que estuvo disponible de manera gratuita. Originalmente, se utilizaba para permitir que los usuarios individuales se comunicaran en servidores informáticos del sistema de tablero de anuncios. Luego, fue estandarizado y admitido por otras aplicaciones como el correo electrónico.

PGP combina criptografía, compresión de datos y técnicas de hash. Utiliza un sistema de **clave pública y privada**, donde cada usuario tiene una clave pública (conocida por otros) y una clave privada (sólo conocida por él).

El mensaje se **cifra con la clave pública del destinatario** y **solo puede ser descifrado con su clave privada**.

PGP usa tanto **criptografía simétrica (clave privada)** como **asimétrica (clave pública/privada)** para proteger datos que se transmiten por redes.

Hay dos versiones de clave pública de PGP:

RSA (Rivest-Shamir-Adleman):
Es uno de los primeros criptosistemas de clave pública. Cifra una clave corta generada con el algoritmo IDEA, y usa MD5 para crear un hash. Se basa en dos números primos grandes y, aunque es muy seguro, es lento y no se usa para cifrar grandes volúmenes de datos.

Diffie-Hellman:
Permite que dos usuarios generen una clave compartida a través de un canal inseguro. Usa el algoritmo CAST para cifrar y SHA-1 para crear un código hash. Se enfoca en el intercambio seguro de claves.

Usos del cifrado PGP:

Cifrado de correos electrónicos:

PGP se utiliza para cifrar mensajes de correo electrónico. Su uso ha aumentado significativamente frente a las organizaciones y agencias gubernamentales que recopilan datos de usuarios, ya que las personas buscan mantener su información personal y confidencial en privado.

Verificación de firma digital:
La encriptación PGP se puede utilizar para la verificación por correo electrónico. Por ejemplo, si un destinatario de correo electrónico no está seguro de la identidad de las personas que le envían un correo electrónico, puede usar una firma digital junto con PGP para verificar su identidad. Como resultado, el destinatario sabrá si algún carácter del mensaje ha sido modificado en tránsito.

Cifrado de archivos:
El algoritmo que utiliza PGP, que generalmente es el algoritmo RSA, se considera en gran medida irrompible, lo que lo hace ideal para cifrar archivos. Es particularmente eficaz cuando se utiliza con una herramienta de detección de amenazas y respuesta. El software de cifrado de archivos permite a los usuarios cifrar todos sus archivos mientras eliminan la complejidad del proceso de cifrado-descifrado.

Ventajas del cifrado PGP

- La mayor ventaja del cifrado de PGP es que el algoritmo es indescifrable.
- Es ampliamente utilizado por personas que necesitan proteger sus comunicaciones privadas.

- Se considera un método líder para mejorar la seguridad en la nube.
- PGP hace imposible que un pirata informático, estados-nación u organismos gubernamentales penetren en archivos o correos electrónicos protegidos.
- (Nota: hubo fallas en algunas implementaciones, como la vulnerabilidad EFAIL en OpenPGP y S/MIME).

Desventajas del cifrado PGP

1. Complejidad de uso: No es fácil de usar. Cifrar datos y archivos lleva tiempo y esfuerzo. Se necesita capacitación.
2. Administración de claves: Requiere entender completamente el sistema para evitar errores, pérdida o corrupción de claves.
3. Falta de anonimato: No anonimiza mensajes; se puede rastrear a remitentes y destinatarios. La línea de asunto no se cifra.
4. Compatibilidad: Solo funciona si remitente y destinatario usan la misma versión del software.

S/MIME (Secure/Multipurpose Internet Mail Extensions)

- Es un estándar del IETF que permite cifrar y firmar correos electrónicos usando criptografía de clave pública.
- Garantiza confidencialidad, autenticidad e integridad de los mensajes, sin importar el destinatario o su ubicación.
- Amplía el alcance del correo electrónico al permitir el envío de múltiples tipos de archivos adjuntos (gracias a MIME).
- Riesgos de seguridad: al permitir adjuntar archivos de varios tipos, los atacantes pueden enviar scripts o ejecutables, que usuarios desprevenidos podrían abrir.

Un archivo PEM (Privacy-Enhanced Mail) es un formato de archivo estándar que se utiliza ampliamente para almacenar y transmitir claves criptográficas, certificados y otros datos seguros. Se asocia comúnmente con el estándar de certificados X.509 y se utiliza en diversas aplicaciones criptográficas, particularmente en el contexto de los certificados SSL/TLS y la infraestructura de clave pública (PKI).

Características del archivo PEM

- Formato de codificación: Los archivos PEM utilizan codificación Base64, lo que permite que los datos binarios se representan en formato de texto ASCII.
- Privacidad y seguridad: Los archivos PEM pueden almacenar información criptográfica confidencial de forma segura. Pueden contener claves privadas, claves públicas, certificados X.509, Solicitudes de firma de certificados (CSR) y otros objetos criptográficos
- Flexible y extensible: PEM es un formato flexible que puede acomodar diferentes tipos de datos. Admite múltiples tipos de datos criptográficos, incluidas claves, certificados e información diversa, como listas de revocación de certificados (CRL) y autoridades de certificación (CA).

Beneficios:

Compatibilidad: Los archivos PEM son ampliamente soportados por varias bibliotecas, herramientas y aplicaciones criptográficas. Se pueden utilizar con diferentes lenguajes de programación y sistemas operativos, lo que los hace altamente interoperables.

Entendible por humanos: El formato de texto ASCII de los archivos PEM los hace fácilmente legibles por los humanos, simplificando el proceso de visualización, edición y uso compartido de datos criptográficos.

Transportabilidad: Los archivos PEM se pueden transferir y compartir fácilmente a través de correo electrónico, protocolos de transferencia de archivos y otros medios. El formato basado en texto garantiza que se puedan transmitir sin problemas de corrupción binaria.

Casos de uso de archivos PEM

Se usan para almacenar certificados SSL/TLS, claves privadas, certificados de autoridades de certificación (CA), firma de código y cifrado de correos electrónicos, garantizando seguridad y autenticidad.

Encabezado y pie de página

Los archivos PEM incluyen encabezados y pies que indican el tipo de datos, como certificados o claves privadas, facilitando su identificación y uso seguro.

Tema IV: Aplicaciones de seguridad en redes

IPsec:

La IETF sabía que faltaba seguridad en Internet, pero no era fácil agregarla. Muchos expertos querían que el cifrado e integridad fueran de extremo a extremo (en la capa de aplicación), ya que así se detectan alteraciones, incluso del sistema operativo.

Sin embargo, esto implicaba modificar todas las aplicaciones. Una alternativa fue poner la seguridad en la capa de transporte o entre la capa de aplicación y transporte, manteniendo el enfoque extremo a extremo sin cambiar los programas.

Otra postura decía que los usuarios no entienden la seguridad ni quieren modificar sus programas, por lo que debía hacerse en la capa de red, sin que ellos participen. Esta idea ganó apoyo y se creó un estándar: IPsec.

IPsec, descrito en los RFC 2401, 2402 y 2406, protege a nivel IP. No todos quieren cifrado (consume recursos), así que se permite un algoritmo nulo (RFC 2410). IPsec es un marco

flexible con múltiples servicios (confidencialidad, integridad y protección contra repeticiones), algoritmos (para adaptarse si uno se rompe) y niveles de granularidad (proteger desde una sola conexión hasta todo un canal entre routers).

Aunque IPsec opera en la capa IP, es orientado a conexión. Cada conexión segura se llama SA (Security Association) y tiene un identificador. Se necesita una SA por dirección. Los identificadores se usan para encontrar las claves y otra información al recibir un paquete.

IPsec cifra y autentica los datos IP. Se creó en los 90 para proteger datos en redes públicas. Por ejemplo, se usa con VPNs para que usuarios accedan a redes privadas de forma segura.

En resumen: IPsec es un conjunto de reglas que protege los datos cuando viajan por Internet.

Sirve para que la información no sea leída ni modificada por nadie mientras va de un punto a otro.

Lo hace usando cifrado, verificación de identidad y protección contra ataques.

Aunque trabaja a nivel de red (IP), asegura conexiones completas entre dos dispositivos. Se usa, por ejemplo, en VPNs para conectarte de forma segura a redes privadas.

Usos de IPSec:

- proporcionar datos al enrutador cuando se envíen datos a través de la red de internet pública
- cifrar los datos de la aplicación
- autenticar rápidamente los datos si proceden de un remitente conocido
- proteger los datos de la red estableciendo circuitos cifrados, llamados túneles IPSec, que cifran todos los datos enviados entre dos puntos de conexión

Las organizaciones usan IPSec para protegerse frente a los ataques de repetición. Un ataque de repetición, o ataque de hombre en el medio, es un acto de interceptación y alteración de la transmisión en curso mediante el enrutamiento de los datos a una computadora intermediaria.

¿Qué es el cifrado IPSec?

- Es una función que **protege los datos codificándolos** para que solo quien tenga la clave correcta pueda leerlos.
- Usa **clave de cifrado y descifrado**. Admite algoritmos como AES, Triple DES, ChaCha, DES-CBC.
- Comienza con **cifrado asimétrico** (más seguro para iniciar) y luego cambia a **simétrico** (más rápido para transmitir datos).

¿Cómo funciona?

1. La computadora revisa si debe usar IPSec según su política de seguridad.

2. Ambas computadoras acuerdan los parámetros de seguridad (algoritmos, claves, etc.).
3. Se transmiten datos cifrados y se valida que vienen de una fuente confiable.
4. Al terminar, se cierra la conexión IPSec.

¿Qué son los protocolos IPSec?

Son los que **envían datos de forma segura**. Cada paquete tiene:

- **Encabezado**: guía el paquete hasta su destino.
- **Carga útil**: el contenido real (los datos).
- **Tráiler**: marca el final del paquete.

Principales protocolos:

- **AH (Authentication Header)**: autentica y protege los datos de alteraciones.
- **ESP (Encapsulating Security Payload)**: cifra la carga útil (o todo el paquete), añade encabezado y tráiler.
- **IKE (Internet Key Exchange)**: establece la conexión segura y negocia claves y algoritmos.

¿Qué son los modos IPSec?

- **Modo Túnel**: **cifra todo el paquete** (encabezado + datos). Ideal para redes públicas como VPN.
- **Modo Transporte**: **cifra solo los datos**, deja el encabezado sin cifrar. Útil para redes privadas o confiables.

VPN (Redes Privadas Virtuales, del inglés Virtual Private Networks), que son redes superpuestas sobre redes públicas, pero con muchas propiedades de las redes privadas. Se llaman “virtuales” porque son sólo una ilusión, al igual que los circuitos virtuales no son reales ni la memoria virtual es real.

¿Para qué sirve una VPN?

Una VPN (Red Privada Virtual) tiene tres funciones principales:

1. **Privacidad:** Protege datos personales como contraseñas e historial de navegación, especialmente en redes públicas.

2. **Anonimato:** Oculta tu dirección IP, lo que evita que sitios web rastreen tu ubicación y actividad.

3. **Seguridad:** Usa cifrado para evitar accesos no autorizados y puede cerrar programas si detecta actividad sospechosa.

¿Cómo funciona una VPN?

Una VPN redirige tu conexión a través de un servidor remoto. Esto se logra mediante:

- **Túnel de datos:** Crea un canal seguro entre tu dispositivo y el servidor VPN, ocultando tu tráfico al proveedor de Internet.

- **Cifrado:** Protege los datos con protocolos como **IPSec**, que codifican la información y solo permiten leerla en el destino.

¿Por qué usar una VPN?

- **Seguridad en redes públicas:** Protege tu información en lugares como cafés o aeropuertos.

- **Privacidad en búsquedas:** Evita que proveedores de Internet vendan tu historial de navegación.

- **Acceso a contenido global:** Permite ver servicios de streaming aunque estés fuera de tu país.

- **Protección de identidad:** Evita que se rastreen tus actividades en línea, protegiendo tu derecho a expresarte libremente.

¿Cómo acceder a una VPN?

Tienes dos opciones:

1. **Proveedor de VPN:** Pagas una suscripción para usar una app o software VPN en cada dispositivo.

2. **VPN en el router:** Instalas la VPN en tu router para proteger todos los dispositivos conectados a tu red.

¿Cómo utilizan las empresas una VPN?

1. **VPN de sitio a sitio:** Conecta oficinas en distintas ubicaciones como si fueran una sola red interna.

2. **VPN de cliente:** Los usuarios se conectan desde sus dispositivos a la red empresarial con archivos de configuración proporcionados por la empresa.

3. **VPN SSL:** Permite el acceso remoto seguro desde un navegador web, ideal para equipos grandes sin necesidad de entregar dispositivos físicos.