

# **TEMA 4**

## **RED TEAM**

Metodología: Diseñar ataques específicos, explotar vulnerabilidades con herramientas como Kali Linux, Metasploit y Nmap.

Objetivo: Reportar debilidades al equipo defensivo (Blue Team).

### **Distribuciones Linux:**

-Kali Linux: Auditoría y pruebas de seguridad.

-Parrot Security OS: enfocado en pruebas de penetración y análisis forense.

### **Herramientas específicas:**

-Nmap/Zenmap: escaneo de puertos y servicios

-Wireshark: Análisis de tráfico para red o sniffing.

-Metasploit: Framework para pruebas de penetración

-John the Ripper: crackeo de contraseñas

etc,...

## **Ciberincidentes y Taxonomía**

Definición: Sucesos que afectan la confidencialidad, integridad o disponibilidad de la información.

### Clasificación:

Nivel crítico: amenazas avanzadas como Advanced Persistent Threats.

Nivel muy alto: Distribución de Malware, sabotaje, robo,....

Nivel medio: DoS, DDoS, pérdida de datos, phishing.

Nivel bajo: Spam, análisis de paquetes, sniffing,....

### Gestión:

Equipos CERT/CSIRT especializados en recolectar, analizar y dar respuesta a posibles ciberincidentes o incidentes de seguridad.

### Repositorios como:

*MITRE ATT&CK*: ofrece conjunto de tácticas y técnicas de ataque.

*CVE MITRE* con registros de vulnerabilidades ya conocidas.

*CWE MITRE*: lista de tipos de componentes software y hardware vulnerables, una CWE esta asociada a una o varias CVE.

*NVD* con registros de vulnerabilidades ya conocidas.

*NVD de NIST*: ofrece APIS para conectar y descargar vulnerabilidades registradas en el tiempo.

## **BLUE TEAM**

Metodología: Evaluar riesgos, diseñar estrategias de mitigación y monitorear el sistema.

Objetivo: Proteger la infraestructura frente a los ataques detectados por el Red Team.

## Seguridad en email: PGP y S/MIME:

**PGP(Pretty Good privacy):** *autenticidad, integridad y confidencialidad.*

*Se usa más a nivel personal.*

- Firma digital.
- Encriptación.
- Integra criptografía de clave pública y privada.
- Operaciones de compresión(ZIP).
- Modelos de clave: Private-Key ring(users) y Public-Key ring(otros users).
- Compatibilidad de emails.

**S/MIME (Secure/Multipurpose Internet Mail Extension):**

*Se usa más a nivel comercial.*

- Firma digital.
- Encriptación.
- Utiliza certificados de clave pública en formato X.509v3
- Sigue el modelo de PKI híbrido.

**Telnet (puerto 23):** facilita el acceso remoto a otros sistemas sobre TCP, de forma que el terminal local aparenta ser el terminal del sistema remoto.  
protocolo de texto plano vulnerable a sniffing.

**FTP(p20/21):** permite la transferencia de ficheros entre diferentes recursos remotos.  
protocolo de texto plano vulnerable a sniffing.

**SSH(Secure Shell(puerto 22):**

- alternativa cifrada que utiliza criptografía pública.
- Funciona como túnel seguro para transferencia de datos(SFTP) y copia segura(SCP).
- SSH2 permite ser usado para transferir ficheros como alternativa a FTP conocido como SFTP y SCP.

**SSH1 Y SSH2 son incompatibles:**

- SSH1: tripleDES, Blowfish para cifrado, RSA para autenticación.
- SSH2: AES, Blowfish, Twofish, 3DES, CAST,... para cifrado, MAC integridad, DH intercambio de claves segura, y RSA o DSA para autenticación.

## SSH: Secure Shell (v2)

- SSHv2 / SSH-2 sigue el siguiente funcionamiento general:
  - **Capa de aplicación:**
    - Gestiona la autenticación del cliente haciendo uso de un usuario/contraseña o aplicando criptografía de clave pública
  - **Capa de transporte:**
    - Gestiona e intercambia las claves iniciales para el cifrado
    - Establece los modos de cifrado y de comprensión, donde es el servidor quien realmente indica el método a aplicar y el cliente selecciona el más conveniente para él
  - **Capa de red:**
    - Establece una “conexión directa” entre el cliente-servidor y redirige el tráfico entre estos puntos de conexión
      - Modo túnel (en base a cifrado simétrico negociado en la capa de transporte)
- SSH-2 / SSHv2:
  - Facilita de forma segura la transferencia de datos y gestión de dispositivos remotos
  - Garantiza el “*tunneling*” entre puntos (P2P: *Peer-to-Peer*), es decir, todas las sesiones y comunicaciones se realizan en un túnel cifrado
  - Mitiga o evita ataques específicos:
    - Man-in-the-Middle
- **SFTP (FTP sobre SSH) # FTPS (FTP sobre TLS)**
  - Funcionamiento de SFTP:
    - Se puede basar de diferentes modos de autenticación para conectar con el servidor SFTP:
      - **Modo básico:** usuario y contraseña
      - **Modo avanzado:** usando las claves públicas de SSH, previamente generadas, y compartiendo dichas claves públicas con el servidor SFTP
        - » De esta forma, cuando el cliente quiere establecer conectividad con el sistema remoto, el proceso software del cliente tendrá que transmitir su clave pública al servidor para su autenticación

## Cifrado de Disco y Archivos:

### Herramientas de OpenSource:

- TrueCrypt/VeraCrypt*: cifrado de discos duros.  
(veracrypt incluye nº de iteraciones para el cifrado)
- DiskCryptor*: similar a trueCrypt pero incluye dispositivos externos.
- OpenStego/OpenPuff*: técnicas de esteganografía para ocultar información en imágenes o multimedia.

# SEGURIDAD EN PAGOS ELECTRÓNICOS

## **Sistemas de pago electrónicos:**

- Permiten realizar pagos en redes abiertas como Internet, mientras que la transferencia de valor real es garantizada por bancos a través de sus redes cerradas, consideradas más seguras.
- Los actores principales incluyen compradores, vendedores, bancos emisores y adquirentes, y pasarelas de pago.

## **Clasificación de sistemas de pago electrónicos:**

- Según **momento de contacto con el banco:**
  - **On-line:** El vendedor verifica la validez del pago antes de enviar el producto.
  - **Off-line:** El pago es validado y depositado después de completada la transacción.
- Según **momento de deducción del dinero:**
  - **Pre-pago:** El comprador paga antes de adquirir el producto (e.g., tarjetas telefónicas).
  - **Pago instantáneo:** El dinero se deduce al realizar la compra (e.g., tarjetas de débito).
  - **Post-pago:** El banco asegura el pago al vendedor, pero el comprador ve el cargo después.
- Según **cantidad implicada:**
  - **Macropagos:** Más de 10 euros.
  - **Pagos medios:** Entre 1 y 10 euros.
  - **Micropagos:** Menos de 1 euro.

## **Problemas de seguridad en pagos electrónicos**

- **Principales amenazas:**
  - Escuchas ilegales (sniffing).
  - Suplantación de identidad (cliente o vendedor).
  - Generación o modificación de datos falsos.
  - Etc...
- **Soluciones implementadas en los protocolos:**
  - Criptografía.
  - Mecanismos de autenticación y autorización.
  - Firmas y certificados digitales.
  - Certificados digitales
  - Etc...

## Protocolo SSL y su evolución

- **SSL (Secure Sockets Layer):**
  - Creado en 1994 por Netscape para proteger comunicaciones.
  - Usaba criptografía híbrida:
    - **Asimétrica** (RSA o Diffie-Hellman) para autenticación y claves de sesión.
    - **Simétrica** (e.g., DES, RC4) para cifrado de datos.
  - SSL también asegura la integridad de los datos mediante MAC y una clave secreta para dicha MAC.
  - Problemas:
    - No protege al comprador del vendedor.
    - Solo protege las transacciones entre dos puntos.
    - No hay mecanismos de autenticación de tarjetas.
    - No incluye mecanismos de facturación ni gestión de recibos.

## Protocolo SET (Secure Electronic Transaction)

- **Desarrollado por VISA y Mastercard (1996)** para reducir fraudes y garantizar pagos en las transacciones electrónicas basadas en tarjetas de crédito.
- **Características clave:**
  - Uso obligatorio de certificados X.509 para todas las entidades.
  - Confidencialidad, autenticación, integridad y no-repudio (autorización de pago).
  - Privacidad: El vendedor no conoce los datos de la tarjeta del cliente, y el banco no conoce los detalles del pedido.
- **Desventajas:**
  - Depende de los algoritmos que ofrece el protocolo.
  - Gestión compleja de certificados digitales.
- **Pasos del protocolo:**
  1. Petición del producto.
  2. Intercambio de certificados.
  3. Envío de información de pedido y pago.
  4. Autorización del pago por el banco emisor.
  5. Confirmación del pago.
  6. Compensación hacia el vendedor.
- **Firma dual:**
  - Divide información en:
    - **Payment Information (PI):** Datos de pago para el banco.
    - **Order Information (OI):** Detalles del pedido para el vendedor.
  - Ofrece privacidad al cliente y garantiza no-repudio

## CyberCash:

- Utiliza una pasarela privada para gestionar pagos electrónicos.
- Autenticación de entidades y cifrado de datos.

### Desventaja:

- Problema de privacidad: La pasarela puede analizar los hábitos del cliente.
- Usa DES.

**iKP (i-Key Protocol):**

- Desarrollado por IBM, incluye variantes (1KP, 2KP, 3KP) según el número de entidades certificadas.
- Uso de criptografía de clave pública para autenticación.

**Las desventajas de uso de 1KP son:**

- el cliente se autentica utilizando sólo un número de tarjeta de crédito y, opcionalmente, un PIN, en lugar de firmas digitales
- el vendedor no se autentica ni ante el cliente ni ante al banco
- ni el vendedor ni el cliente proporcionan evidencias de intervención en la transacción.

**Micropagos (Millicent):**

- Protocolo para transacciones de bajo valor.
- Basado en criptografía simétrica y cupones electrónicos (scrips).
- Incluye un bróker que media entre clientes y vendedores para reducir costos.

# PRIVACIDAD DE LOS USUARIOS EN APLICACIONES

## Conceptos Generales

- **Definición de privacidad:**
  - Derecho a proteger, controlar y decidir sobre el acceso y uso de información personal (identidad, localización, rutinas, etc.).
  - **Diferencia entre privacidad y confidencialidad:**
    - Privacidad: Relacionada con la persona.
    - Confidencialidad: Relacionada con los datos.
- **Amenazas comunes:**
  - **Rastreo de actividad en la red:** Mediante análisis de tráfico.
  - **Análisis pasivo:** Observación de datos cifrados, como cabeceras o patrones de paquetes.
- **Propiedades fundamentales de la privacidad:**
  - **No vinculabilidad** (unlinkability): Imposibilidad de relacionar entidades o mensajes.
  - **No observabilidad** (unobservability): Imposibilidad de rastrear mensajes o identificar a los emisores.
- **Enfoques complementarios:**
  - **Enfoque legislativo:** limitan practicas abusivas de empresas.
  - **Enfoque tecnológico:** mecanismos de preservación de la privacidad

## Anonimato y Técnicas Relacionadas

- **Definición:** Estado en el que un individuo no puede ser identificado entre un grupo.
- **Técnicas de anonimato:**
  - **Pseudónimos:** Sustituyen la identidad real, pero pueden ser vinculados a largo plazo.
  - **Anonimato rastreable:** Permite revelar la identidad en casos justificados.
  - **Anonimato no rastreable:** Garantiza que no se pueda identificar al usuario.
  - **Anonimato no rastreable y no vinculante:** Evita la vinculación entre acciones del mismo usuario.
- **Técnicas avanzadas:**
  - **Ofuscación:** Generalización o supresión de datos para ocultar información sensible.
  - **Esquemas avanzados de firma digital.**
  - **Protocolos de enrutado y criptografía:** Ocultan direcciones de red y trazas de paquetes.

## Esquemas Avanzados de Firma Digital

- **Extensiones de la firma digital tradicional:**
  - **Firma ciega:** Permite que un firmante valide un mensaje sin conocer su contenido (útil para voto electrónico).
  - **Firma de grupo:** Cualquier miembro de un grupo puede firmar en nombre del grupo. Un administrador puede revelar al firmante en caso de disputas.
  - **Firma de anillo:** Ofrece anonimato total, ya que ni siquiera el administrador puede identificar al firmante.

## Firma de grupo

- Un esquema de firma de grupo debe satisfacer las siguientes **propiedades iniciales** para cumplir la condición de “anonimato rastreable”:
  - sólo los miembros del grupo pueden firmar mensajes de forma correcta (*infalsificable*)
  - a excepción del administrador del grupo nadie puede descubrir:
    - qué miembro del grupo ha firmado el mensaje (*anonimato*)
    - si dos firmas han sido emitidas por el mismo miembro del grupo (*no-vinculación*)
  - los miembros no pueden evitar la **apertura de la firma** por parte del administrador, ni firmar por otro

## Protocolos Criptográficos y de Enrutado

Estos protocolos protegen las comunicaciones frente a observadores externos. Los métodos incluyen:

### *a. Uso de proxies:*

- Los servidores proxy actúan como intermediarios para ocultar la dirección IP del emisor.
- **Limitación:** El proxy puede ser un punto de fallo si no es confiable.

### *b. Uso de mezcladores (mixers):*

- Almacenan y mezclan mensajes antes de enviarlos, ocultando la relación entre emisor y receptor.
- **Limitación:** Introducen latencia y pueden ser vulnerables si el mezclador es comprometido.

### *c. Enrutado por capas (Onion Routing):*

- Los mensajes se cifran en múltiples capas, que se descifran progresivamente al pasar por cada nodo.
- **Ejemplo avanzado: TOR (The Onion Router):**
  - Ofrece anonimato mediante rutas aleatorias y cifrado.
  - Limita ataques de correlación aplicando guardianes de entrada.
  - Forward secrecy: va cambiando los nodos cada 10 minutos.
  - **Limitación:** Velocidad lenta y no garantiza privacidad de datos fuera de la red.



*d. Basados en creación de grupos:*

- **Crowds:** Los nodos enrutan mensajes de forma aleatoria entre compañeros del grupo.
- **Hordes:** Variante más rápida, pero menos segura, ya que transmite respuestas por difusión (broadcast).

	Arquitectura	Latencia
Proxy	Centralizada	Baja
Mezcladores		Alta
Enrutado de cebolla		Muy alta
Tor		Media-baja
Crowds	Distribuida	Media-baja

# TEMA 5

## Introducción

- La expansión de la web en los años 90 incrementó los riesgos de seguridad:
  - **Lado del cliente:** Troyanos, suplantación de identidad.
  - **Lado del servidor:** Ataques de denegación de servicio, robo de información.
  - **Información en tránsito:** Escuchas ilegales y modificación de mensajes.
- **Principales amenazas:**
  - **Confidencialidad:** Escuchas no autorizadas (sniffing).
  - **Integridad:** Modificación de datos en tránsito.
  - **Disponibilidad:** Ataques de denegación de servicio (DoS).
  - **Autenticación:** Suplantación de identidad o datos falsificados.

## Seguridad en la Capa de Transporte

### SSL(Secure Sockets Layer)

- **SSL (Secure Sockets Layer):**
  - Creado por Netscape en 1994 para proporcionar seguridad en la capa de transporte.
  - Asegura confidencialidad, integridad y autenticación mediante criptografía híbrida:
    - **Clave pública:** Para autenticación e intercambio de claves.
    - **Clave simétrica:** Para cifrado de datos.
  - NO proporciona servicio de no repudio.
- **Evolución hacia TLS (Transport Layer Security):**
  - **1999 (TLS 1.0):** Estandarización por el IETF.
  - **TLS 1.2:** Introduce AES-GCM y SHA-256 para mayor seguridad.
  - **TLS 1.3 (2018):** Mejora rendimiento (handshake más corto) y refuerza seguridad (Perfect Forward Secrecy).
- **Principales características de SSL/TLS:**
  - **Independencia de la capa de aplicación:** Puede usarse con múltiples protocolos (HTTP, FTP, Telnet, etc.).
  - **Doble funcionalidad:**
    1. **Establecer una conexión segura:** Autenticación mutua entre cliente y servidor.
    2. **Transmitir datos de forma segura:** Cifrados y protegidos contra manipulación.

## Detalles del Protocolo SSL/TLS

- **Dos conceptos:**
  - Sesión SSL: asociación entre el cliente y el servidor en la que se negocian los parámetros de seguridad para todas las conexiones de esa sesión.
  - Conexión SSL: realización de la transmisión de datos entre el cliente y el servidor, protegida criptográficamente según lo negociado en la sesión.
- **Estructura por subcapas:**
  - **SSL Record Protocol:** asegura que los datos de la capa de aplicación se procesen y transmitan de forma segura mediante las siguientes etapas:
    - Fragmentación:** Divide los datos en bloques manejables.
    - Compresión:** Opcional, reduce el tamaño de los datos.
    - Añadir MAC:** Garantiza la integridad de los datos.
    - Cifrado:** Protege la confidencialidad de los datos.
    - Añadir una cabecera SSL Record:** Indica detalles como el tipo de protocolo.
    - Fragmentación:** Los bloques tienen un tamaño máximo de 16,384 bytes.
    - Reensamblado:** En el destino, los datos son descifrados, descomprimidos y reensamblados antes de ser entregados a la capa de aplicación.
  - **Subprotocolos principales:**
    - **SSL Handshake Protocol:** Negociación de parámetros de seguridad (versiones, algoritmos, claves).  
**Es la parte más compleja de SSL** porque permite al servidor y al cliente:
      - **autenticarse** mutuamente
      - **negociar** un algoritmo de cifrado y una función MAC
      - así como las claves a usar para **proteger los datos del SSL record**.Por lo tanto, cada mensaje tiene 3 campos:
      - **Tipo** (1 byte): indica uno de 10 posibles mensajes (ver siguiente tabla)
      - **Longitud** (3 bytes): longitud del mensaje en bytes
      - **Contenido** (≥ 0 bytes): parámetros asociados con el mensaje (ver también siguiente tabla)
    - **SSL Change Cipher Spec Protocol:** Activa el algoritmo de cifrado negociado.
    - **SSL Alert Protocol:** Intercambia alertas (por ejemplo, errores críticos o advertencias).  
Cada mensaje de este protocolo consta de 2 bytes.  
-El primer byte toma el valor 1 (warning) o 2 (fatal) para informar de la severidad del mensaje
      - si el nivel es fatal, SSL termina la conexión de forma inmediata
      - otras conexiones de la misma sesión pueden continuar pero no se producen nuevas conexiones dentro de la misma sesión
      - El segundo byte contiene un código que indica la alerta específica
    - ejemplos: unexpected\_message, bad\_record\_mac, decompression\_failure, illegal\_parameter, ...

- **SSL Application Data Protocol:** es el propio protocolo de la capa de aplicación (ej: HTTP) y alimenta al SSL Record Protocol.

- **Proceso de intercambio de claves (Handshake):**

1.-Cliente envía un `ClientHello` con parámetros iniciales (algoritmos soportados, número aleatorio, la versión del protocolo, método de compresión).

2.-Servidor responde con un `ServerHello`, certificados(opcional) y parámetros necesarios para gestionar la clave secreta(puede solicitarle certificado al cliente).

3. El cliente envía certificado si se le pidió, y los parámetros de seguridad necesarios para computar la clave de sesión.

4.Ambas partes acuerdan una **clave de sesión compartida** para cifrar la sesión.

**Necesitamos:**

- clave de sesión
- clave para el MAC
- IV para el modo de operación

**Paso 1:**

- generar números aleatorios (la **salt** para generar después los parámetros de seguridad comentados anteriormente),
- establecer el resto de parámetros de seguridad (ej. tipo de algoritmos de intercambio de clave),
- enviar toda esta información a la otra parte

**Paso 2:** crear y enviar la semilla  
“**pre-shared master key**” mediante  
`ClientKeyExchange`

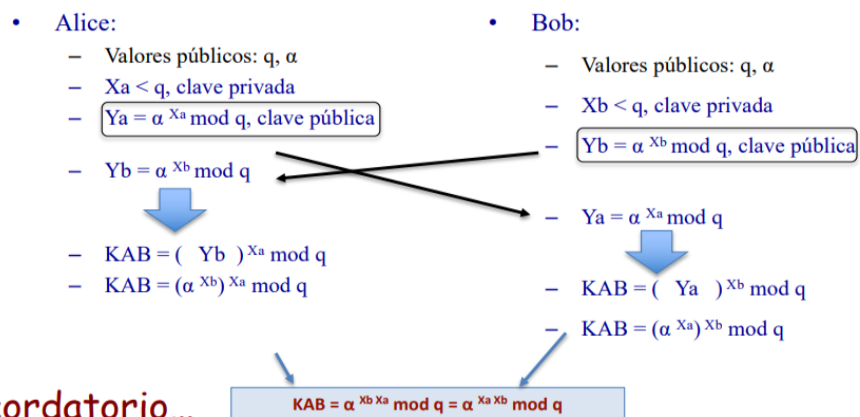
ServerHelloDone

**Paso 3:** continuar con los objetivos del paso 2, pero esta vez usando como semilla el master key

[CertificateVerify]

## SSL - Secure Sockets Layer - intercambio de claves

- Tanto SSL como TLS usan también **DHE (Diffie Helman efímero)**:
  - Sin embargo, antes de entrar en el DHE, recordemos cómo funciona DH



- Perfect Forward Secrecy (PFS):**
  - Introducido con Diffie-Hellman efímero (DHE) y curvas elípticas (ECDHE).
  - Garantiza que las claves de sesión no puedan ser descifradas incluso si la clave privada del servidor es comprometida.

## TLS 1.2

### Cambios significativos introducidos en TLS 1.2:

- Cálculo de claves:** Emplea SHA-256 en lugar de MD5 y SHA-1, que eran utilizados en versiones anteriores.
- Cipher Suites:** Introduce AES-GCM y AES-CCM como modos de cifrado autenticado (AEAD). Elimina algoritmos débiles como DES e IDEA.
- Introduce el concepto de **"Authenticated Encryption with additional data"**
- Soporte para curvas elípticas:** Utiliza ECDHE para mejorar la seguridad en el intercambio de claves.
- Extensiones en los mensajes ClientHello y ServerHello:** Permite detallar certificados, autorizaciones y parámetros de seguridad.

# TLS 1.3


## Principales mejoras respecto a TLS 1.2:

- **Rendimiento:** Reduce el tiempo de handshake a un solo Round-Trip Time (RTT).
- **Seguridad mejorada:** Elimina algoritmos inseguros como CBC y MD5.
- **0-RTT para reconexiones:** Permite conexiones más rápidas reutilizando credenciales previas.
- **Perfect Forward Secrecy (PFS):** Obliga a utilizar claves efímeras (DHE o ECDHE).
- **Simplificación:** Reducción de modos de operación, limitándose a AEAD (GCM, CCM).

## Comparativa entre TLS 1.2 y TLS 1.3

Aspecto	TLS 1.2	TLS 1.3
Handshake RTT	2	1
Algoritmos inseguros	Algunos permitidos (CBC, RC4)	Eliminados
Reconexión rápida	No soportada	0-RTT con PSK
Intercambio de claves	RSA, DHE	Solo DHE/ECDHE (PFS obligatorio)

### SSL Handshake Protocol

Mensajes	Parámetros asociados con los contenidos (contenido)	Tipo
Hello_request	null	Tipo = 0 Lo solicita el servidor para renegociar una sesión
Client_hello	Versión, random_cliente, sesión ID, cipher suite, método de compresión	Tipo = 1
Server_hello	Versión, random_servidor, sesión ID, cipher suite, método de compresión	Tipo = 2
Certificate	Cadena de certificados X.509	Tipo = 11
Server_key_exchange	Parámetros de seguridad necesarios para computar la clave de sesión (ej. usando DH como seed inicial) y <b>firma del hash(cliente_random + servidor_random + parámetros de seguridad)</b>	Tipo = 12
Certificate_request	Tipo de certificados y autoridades	Tipo = 13
Server_hello_done	Null	Tipo = 14
Client_key_exchange	El pre-shared master key cifrado con RSA, o añade los parámetros de DH/Fortezza para computar el master key en cada una de las partes	Tipo = 16
Certificate_verify	Se aplica cuando se solicita el certificado. Consiste en <b>firmar el hash(master key + hash(todos los mensajes intercambiados hasta el momento))</b>	Tipo = 15
Finished 	Es el <b>cifrado</b> del hash(master_key + hash(hash(todos los mensajes	Tipo = 20

## DTLS (Datagram Transport Layer Security)

**DTLS** es una adaptación de TLS para protocolos que operan sobre **UDP**, garantizando seguridad en entornos donde se requiere baja latencia (como IoT o videoconferencias).

- Introduce un número de secuencia explícito para controlar la entrega de paquetes no ordenados o perdidos.
- Última versión: DTLS 1.2 (2012).

# SEGURIDAD EN REDES TCP/IP

## Seguridad en la Capa de Internet

- **Informe de la Internet Architecture Board (IAB, 1994):**
  - Recomendó incorporar **cifrado** y **autenticación** en la arquitectura de Internet, especialmente en IPv6.
- **IPSec (RFC 4301):**
  - Introduce especificaciones de seguridad en la capa de Internet, aplicables tanto a IPv6 como a IPv4.
  - **Beneficios:**
    - Protección transparente para todas las aplicaciones sin necesidad de cambios en ellas.
    - Aplicable a escenarios como redes empresariales, intranets, extranets y comercio electrónico.
- **Propiedades de IPSec:**
  - **Autenticación:** Verifica el origen de los mensajes.
  - **Integridad:** Garantiza que los datos no han sido alterados.
  - **Confidencialidad:** Cifra los datos para evitar accesos no autorizados.

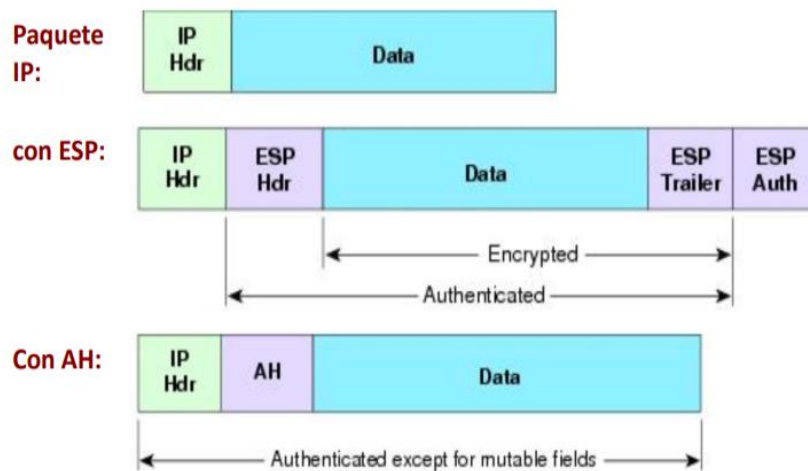
->usando: MAC, cifrado, algoritmos para el intercambio de clave.
- **Limitaciones de IPSec:**
  - No proporciona no-repudio (como TLS/SSL).
  - No protege completamente frente a ataques DoS, aunque puede mitigar ataques de repetición.

## Protocolos de IPSec

- **ESP (Encapsulating Security Payload):**
  - Cifra el contenido del paquete.
  - Garantiza confidencialidad, y autenticación e integridad opcional.
  - Usa un número de secuencia para evitar ataques de repetición.
- **AH (Authentication Header):**
  - Garantiza la integridad y autenticación del paquete.
  - También incluye un número de secuencia para mitigar ataques de repetición.
- **IKE (Internet Key Exchange):**
  - Gestiona las claves de cifrado y establece asociaciones de seguridad (SAs), específico para generar y distribuir claves para ESP y AH.
  - Usa certificados X.509 y algoritmos como Diffie-Hellman para el intercambio de claves (autentica la identidad del sistema remoto)

## Modos de IPSec

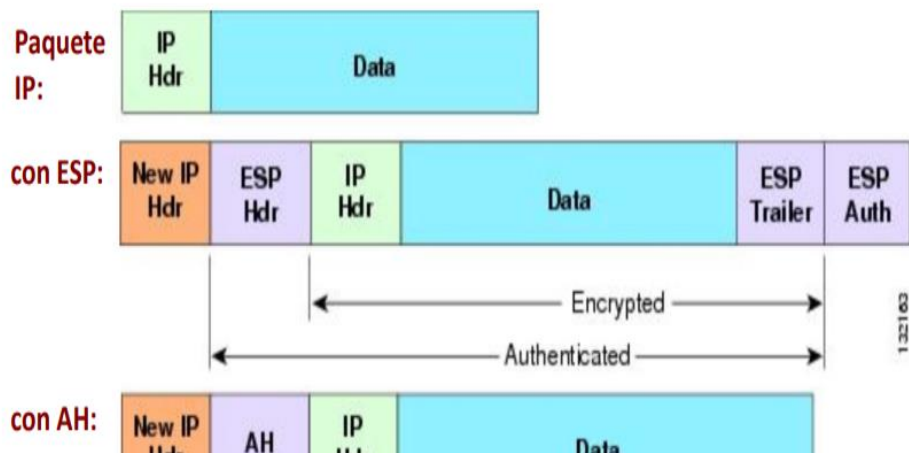
- **Modo Transporte:**
  - Protege únicamente la carga útil (payload) del paquete IP.
  - Adecuado para comunicaciones entre dos hosts.
  - Permite cifrado, autenticación o ambos en el payload.
  - Ejemplo: **Visible la IP de origen y destino finales.**





- **Modo Túnel:**

- Protege todo el paquete IP, incluyendo su cabecera.
- Adecuado para comunicaciones entre gateways o routers.
- Encapsula el paquete IP original dentro de otro paquete con una nueva cabecera IP.
- **Ventaja:** Oculta las IPs de origen y destino final, útil para VPNs.



## Asociaciones y Políticas de Seguridad

- **Asociaciones de Seguridad (SAs):**

- Para activar IPsec es necesario establecer previamente:
  - el origen y el destino de los paquetes IPsec
  - el modo de autenticación de los mensajes; p. ej. el HMAC
  - el algoritmo de cifrado; p. ej. AES o Blowfish
  - el índice de parámetro de seguridad (SPI - Security Parameter Index)
    - núm. de 32 bits único para cada asociación definida para ESP/AH
  - un número de secuencia única (Sequence Number) de paquetes para controlar los ataques replay
    - sólo se aceptan paquetes que tienen un número actual de secuencia o posterior, las anteriores se descartan
- Contienen parámetros como:
  - Algoritmos de cifrado y autenticación (AES, HMAC).
  - Claves.
  - Índices de parámetros de seguridad (SPI, Security Parameter Index).
- Gestionadas mediante bases de datos (SAD, Security Association Database).

- **Políticas de Seguridad (SPs):**

- Definen las reglas para proteger el tráfico (origen, destino, puertos, protocolos).
- Almacenadas en una base de datos de políticas de seguridad (SPD, Security Policy Database).

## Cabeceras AH y ESP

- **AH (Authentication Header):**
  - Proporciona integridad y autenticación.
  - Campos clave:
    - **Next Header:** Tipo de cabecera siguiente.
    - **SPI:** Identificador de la SA.
    - **Sequence Number:** Evita ataques de repetición.
    - **Authentication Data:** Valor MAC para integridad.
- **ESP (Encapsulating Security Payload):**
  - Cifra el payload y, opcionalmente, proporciona autenticación.
  - Campos clave:
    - **Encrypted Payload:** Datos cifrados.
    - **Padding:** Alinear datos al tamaño del bloque.
    - **Authentication Data:** Verifica integridad.

## Protocolo Internet Key Exchange (IKE)

- **Funciones de IKE:**
  - Negocia SAs para IPSec.
  - Autentica las partes de la comunicación.
  - Establece claves secretas.
- **Estructura de IKE:**
  - Basado en **ISAKMP (Internet Security Association and Key Management Protocol)**.
  - Usa:
    - Certificados X.509 para autenticación.
    - Diffie-Hellman para intercambio de claves.
- **Fases de IKE:**
  - **Fase 1:** Autenticación de las partes y establecimiento de una SA ISAKMP.
    - Modos:
      - **Agresivo:** Más rápido, pero expone la identidad en texto plano.
      - **Principal:** Más seguro, pero más lento.
  - **Fase 2:** Negociación de SAs y generación de claves de sesión para IPSec.

