Introducción a la ciberseguridad

Clase Práctica

Introducción: Los Pilares de la Seguridad Digital

La **Ciberseguridad** es la práctica de defender ordenadores, redes, sistemas y datos de ataques maliciosos. Para un futuro profesional de informática, es crucial entender que la seguridad se basa en una combinación de estrategias:

1. **Protección (Criptografía):** Hacer que los datos sean incomprensibles.
2. **Ocultamiento (Esteganografía):** Hacer que los datos sean indetectables.
3. **Modelo de la CIA:** La sigla representa Confidencialidad, Integridad y Disponibilidad. Su objetivo es guiar a los profesionales para identificar, priorizar y proteger los activos de información de una organización

Parte 1: Protección de la Información (Criptografía)

La criptografía es la ciencia que estudia los métodos para proteger la información y las comunicaciones. En el campo de la ciberseguridad, la criptografía garantiza que los datos sean incomprensibles para personas o sistemas no autorizados. Su objetivo principal es asegurar los tres pilares de la seguridad de la información: la confidencialidad, la integridad y la autenticidad.

1. Mecanismos de Cifrado (Confidencialidad)

El cifrado transforma el mensaje original (texto plano) en un formato ilegible (texto cifrado) mediante algoritmos, garantizando la Confidencialidad.

TIPOS DE CIFRADO

Cifrado Simétrico: Utiliza la misma clave para cifrar y descifrar el mensaje. Es extremadamente rápido, por lo que es ideal para cifrar grandes volúmenes de datos. Un ejemplo común es el algoritmo AES.

Cifrado Asimétrico: Utiliza un par de claves distintas: una clave pública para cifrar y una clave privada para descifrar. La clave pública se puede compartir libremente, mientras que la privada se mantiene en secreto. ***Un ejemplo es el algoritmo RSA.*** Es más lento que el cifrado simétrico, pero es fundamental para la autenticación y para el intercambio seguro de claves.

- *El algoritmo **RSA** es un sistema de cifrado asimétrico ampliamente utilizado en la criptografía moderna. Su nombre proviene de las iniciales de sus creadores: Ron **R**ivest, Adi **S**hamir y Leonard **A**dleman.*

2. Integridad y Autenticidad

Integridad (Funciones Hash): Permite verificar que el mensaje no ha sido alterado desde que fue enviado. Una función hash crea un resumen de longitud fija del mensaje. Si un solo bit del mensaje cambia, el hash resultante será totalmente diferente.

- **Autenticidad** (Firmas Digitales): Combina el cifrado asimétrico y el hashing para probar que el remitente es quien dice ser. El remitente cifra el hash del mensaje con su clave privada, y el receptor lo verifica con la clave pública del remitente.

Parte 2: Sistemas Criptográficos Integrales

Los sistemas de seguridad de alto nivel combinan técnicas simétricas y asimétricas para ofrecer velocidad y seguridad.

1. SSL/TLS: Canales Seguros en la Web

El protocolo SSL (Secure Sockets Layer) y su sucesor, TLS (Transport Layer Security), crean un canal de comunicación seguro (HTTPS).

Proceso Híbrido: Emplean criptografía asimétrica para la Autenticación y el Intercambio de Claves.
Cifrado de Sesión: El resto de la comunicación utiliza cifrado simétrico con una clave de sesión secreta, garantizando la confidencialidad y la integridad del tráfico a alta velocidad

2. PGP/GPG: Confianza Descentralizada

PGP (Pretty Good Privacy) y GPG (GNU Privacy Guard) se usan para cifrar y firmar correos electrónicos y archivos.

Modelo: Representan una seguridad distribuida basada en la "Web de Confianza" (Web of Trust)
Mecanismo Híbrido: Utilizan un algoritmo simétrico rápido para cifrar el mensaje y luego cifran esa clave simétrica con la clave pública del destinatario (asimétrico), lo que asegura la confidencialidad, autenticidad e integridad.

Parte 3: Ocultamiento, Desarrollo Seguro y Referencias Clave

3.1. Ocultamiento de la Información: Esteganografía

El Ocultamiento es la técnica que busca esconder la existencia misma de un mensaje o comunicación, lo cual complementa al cifrado al evitar la sospecha.

Esteganografía: Esconder información sensible dentro de un archivo portador que parece inofensivo (imágenes, audio, video)
Mecanismo LSB (Least Significant Bit): Una técnica común que modifica los bits menos significativos de una imagen, permitiendo que el mensaje oculto viaje sin levantar sospechas, ya que el cambio es indetectable para el ojo humano
Anonimización: Uso de redes (ej. Tor) que enrutan el tráfico a través de múltiples nodos para dificultar el rastreo del origen de la comunicación.

2. DevSecOps y la Regla de Oro para Credenciales

DevSecOps: significa Development, Security, and Operations (Desarrollo, Seguridad y Operaciones)

Es la filosofía que busca **integrar la seguridad** como una responsabilidad compartida en **cada etapa** del ciclo de desarrollo de *software* . **DevSecOps** representa la evolución de la cultura de desarrollo de software DevOps (Development + Operations) al integrar la Seguridad (Sec) como un componente esencial desde el inicio de todo el ciclo de vida del desarrollo.

En lugar de ser un proceso de revisión tardío y separado, la seguridad se convierte en una responsabilidad compartida entre los equipos de desarrollo, operaciones y seguridad.

Su principio rector es el ***Shift Left***: mover las prácticas y pruebas de seguridad a las **primeras etapas** del desarrollo.

2. La Regla de Oro para la Protección de Información Sensible en Bases de Datos

Regla de Oro para Contraseñas: Las credenciales de usuario deben ser almacenadas utilizando **Hashing Salteado, Fuerte y con *Key Stretching.***

La regla fundamental e innegociable para proteger la información sensible en cualquier base de datos (DB) se resume en una máxima: **Nunca almacenes contraseñas en texto plano, y nunca confíes únicamente en el cifrado de campo.**

Las credenciales de usuario (contraseñas) deben ser almacenadas utilizando *Hashing* Salteado, Fuerte y con *Key Stretching* (Estiramiento de Clave) para hacer la reversión computacionalmente prohibitiva.

***Hashing* Fuerte:** Utiliza algoritmos lentos (como **Argon2** o **bcrypt**)
***Salting* (Salazón):** Se añade una cadena aleatoria y **única** ("la sal") antes de aplicar el *hash*, lo que inutiliza ataques de *rainbow tables*
Key Stretching: El proceso de *hashing* se repite miles de veces (iteraciones) para hacer que los ataques de fuerza bruta sean computacionalmente prohibitivos.

Argon2

Es el algoritmo de *hashing* de contraseñas más moderno, reconocido por el *Password Hashing Competition* (PHC) en 2015. Su principal objetivo es maximizar el costo que representa para un atacante intentar descifrar contraseñas mediante fuerza bruta.

Características Claves: Intensidad de Memoria (*Memory Hardness*): Argon2 está diseñado para requerir una gran cantidad de memoria RAM para su ejecución. Esto lo hace particularmente resistente a los ataques que utilizan **GPUs** (Unidades de Procesamiento Gráfico) y **ASICs** (Circuitos Integrados de Aplicación Específica). Los atacantes que utilizan estos *hardware* avanzados para probar miles de millones de *hashes* por segundo se ven limitados por el costo y la latencia de la memoria, no solo por el tiempo de CPU.**Factor de Paralelismo:** Permite aprovechar los sistemas multi-núcleo al utilizar múltiples *threads* para calcular el *hash*. Esto lo hace eficiente en el servidor que lo implementa, mientras sigue siendo costoso para el atacante en un entorno distribuido.**Versatilidad:** Ofrece tres variantes diferentes (*Argon2d*, *Argon2i*, *Argon2id*) para optimizar su uso según el nivel de resistencia deseado contra ataques de *side-channel* o de fuerza bruta. **Argon2id** es generalmente la versión recomendada por combinar la seguridad de las otras dos.

Posibilita la configuración de Parámetros Configurables:

Iteraciones (Time Cost): El número de veces que el algoritmo repite su proceso.

Memoria (Memory Cost): La cantidad de RAM que requiere el proceso.

Grado de Paralelismo (Parallelism Cost): El número de hilos de ejecución que utiliza.

bcrypt

Fue diseñado por Niels Provos y David Mazières en 1999 como una adaptación del algoritmo de cifrado **Blowfish**. Fue uno de los primeros algoritmos de *hashing* adaptativos y, a pesar de su antigüedad, sigue siendo una opción muy segura y de uso extendido.

Características Clave

Intensidad de Cómputo (*Work Factor*): El diseño de bcrypt permite aumentar intencionalmente el tiempo que tarda en calcular un *hash*. Lo logra mediante la repetición del proceso de *hashing* un número exponencialmente configurable de veces.**Resistencia a la Fuerza Bruta:** Su principal innovación fue ser **"adaptativo"**. Esto significa que, a medida que la capacidad de cómputo (velocidad de los procesadores) de los atacantes aumenta con el tiempo, la cantidad de repeticiones puede incrementarse, manteniendo constante el tiempo de *hashing* deseado (ej. 500 milisegundos), lo que fuerza al atacante a gastar más tiempo y energía. **Uso de Sal (*Salt*):** bcrypt incorpora automáticamente una **sal aleatoria** y la almacena dentro del *hash* resultante, lo que elimina la necesidad de que el desarrollador la gestione por separado.

Parámetros Configurables

El factor de seguridad de bcrypt se controla principalmente mediante un parámetro:

Costo (Cost Factor o Work Factor): Es un logaritmo que determina el número de iteraciones. Un costo de 10 significa que el *hash* se calcula 210 veces. Este factor se aumenta periódicamente para compensar el avance del *hardware*.

3. Recursos Clave para Profundizar

1. **Dr. Jorge Ramió Aguirre. (2006). *Libro Electrónico de Seguridad Informática y Criptografía Versión 4.1.***

Explica que la **Criptografía** es la base matemática y tecnológica sobre la cual se asienta la seguridad digital. Aborda la necesidad de proteger los sistemas de información, redes y datos de ataques, sentando las bases para entender las vulnerabilidades y las técnicas de defensa. El libro funge como un recurso para comprender los **mecanismos de seguridad** (cifrado, *hashing*, firmas) antes de que puedan ser aplicados en entornos reales.

Sirve como el primer paso para entender qué es la **seguridad perimetral**, la **seguridad de las comunicaciones** y la **protección de la información** a nivel conceptual.

2. Dan Boneh and Victor Shoup. (2017). *A Graduate Course in Applied Cryptography*

Este libro se posiciona como una **fuente avanzada y de diseño de protocolos**.

Su foco es la **Criptografía Aplicada**, lo que implica ir más allá de los algoritmos y centrarse en cómo estos se integran para resolver problemas de seguridad en sistemas operativos y aplicaciones. Aborda el diseño seguro de **primitivas criptográficas** y cómo se deben construir protocolos (como SSL/TLS o sistemas de autenticación) que sean resistentes a los ataques computacionales avanzados.

Proporciona el marco teórico riguroso necesario para que los futuros investigadores y diseñadores de *software* no solo utilicen herramientas de ciberseguridad, sino que también las **diseñen y evalúen su seguridad** formalmente.

3. CrackStation. *Salted Password Hashing - Doing it Right*.

Este artículo se posiciona como una **guía de mejores prácticas operacionales**.

Posicionamiento: Es un recurso fundamentalmente **práctico y de defensa activa**. Explica que la seguridad de un sistema depende en gran medida de la protección de las credenciales de usuario. Su objetivo es exponer las fallas de las implementaciones sencillas de *hashing* y demostrar la necesidad crítica de utilizar **"salting"** (añadir una cadena aleatoria única a la contraseña antes de aplicar el *hash*) para mitigar ataques masivos como los de *rainbow tables*.

Aporte a la Ciberseguridad: Se centra en la **seguridad de las aplicaciones** y en el **desarrollo de *software* seguro** (DevSecOps), proporcionando una regla de oro para la protección de la información sensible en bases de datos.