Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Ingeniería En Ciencias de la Computación

Materia:

Modelo de Redes

Trabajo:

Documentación Simulador Enlace Datos MD5

Maestra/o:

Dr. Iván Olmos Pineda

Alumno:

- Ruiz Lozano Paulo Cesar

Introducción

El programa SilumadorMD5, hace una simulación a nivel de la capa de enlace de datos con un programa Cliente y otro Servidor. El Servidor recibe peticiones de “loggin”, busca si el usuario existe en la base de datos, genera un mensaje aleatorio y mezcla el mensaje con la contraseña generando un MD5 que se va a comparar con el MD5 del Cliente, si son iguales entonces se establece una conexión. Todos los mensajes que se envían entre cliente y servidor son encriptados en Base64.

Funcionamiento del programa

El programa necesita ser compilador bajo java 11 o superior, se puede usar un IDE como NetBeans, pero en la carpeta del programa o en el repositorio https://github.com/Fatake/SilumadorMD5 existen 3 archivos de Shell:

1. Compila-sj: Lanza las instrucciones javac -d class Servidor.java y javac -d class Cliente.java
2. Ser.sh: Lanza la instrucción java -cp class Servidor.
3. Cli.sh: Lanza la instrucción java -cp class Cliente localhost 9999.

La ejecución del programa Cliente requiere de 2 parámetros, dirección IP del servidor que por defecto es localhost(127.0.0.1) y el puerto (9999).

El servido inicia y genera varios hilos para cada petición que existe, el hilo que atiende peticiones siempre está escuchando peticiones, el paso de mensajes ente cliente y servidor se envía como una cadena con la siguiente estructura: Cadena mensaje = “código,valor”. Los códigos que se envían son:

- us,usuario: El cliente recibe un mensaje del usuario a buscar, comprueba si existe o no, si existe procede a enviar un mensaje aleatorio, si no envía in no existe.
- un,null: El cliente recibe un código un en caso de que el usuario no exista.
- ms,mensajeAleatorio: El cliente recibe un mensaje aleatorio del servidor, el cliente procede a mezclarlo con la contraseña proporcionada con el usuario y genera un MD5 que envía al servidor.
- md,MD5Usuario: El servidor recibe un md5 del usuario y procede a compararlo con el md5 que el genera con el mensaje aleatorio y contraseña dentro del servidor. De ser correcto el servidor envía un código de cn de conectado donde ese establece la

"conexión" con el cliente. Caso contrario envía un código nn de no conectado.

- cn,Conectado: El cliente recibe un código cn si la contraseña es exitosa.
- nn,NoConectado: El cliente recibe este código si la contraseña es incorrecta.
- fn,null: El servidor recibe este código para fin de conexión.

Desarrollo

El programa está creado de las siguientes clases:

- Usuario: Clase que almacena la información del usuario, nombre y contraseña
- Servidor: Clase que maneja peticiones de "login" comparando si existe el usuario y comparando MD5.
- GestorPeticion: Subclase de Servidor que administra peticiones.
- Mezclador: Clase que mezcla un texto aleatorio con una contraseña.
- MD5: Clase que genera un md5 de una cadena de texto.
- LectorArchivo: Clase que lee la "base de datos" de usuarios y almacena en una lista ligada.
- Cliente: Clase que realiza peticiones de "login" al servidor, pidiendo un usuario y contraseña.

Clases

Usuario

Atributos

- Nombre: atributo cadena privado que almacena el nombre del usuario.
- Password: atributo cadena privado que almacena la contraseña del usuario

Constructores

- No tiene.

Métodos

- getName(): Retorna una cadena el nombre del usuario.
- getPass(): Retorna una cadena la contraseña del usuario.
- toString(): Retorna en cadena la información del usuario.

Documentación Simulador Enlace Datos MD5

Ruiz Lozano Paulo Cesar

Mezclador

Atributos

- No tiene

Constructores

- No tiene

Métodos

- Mezcla(): recibe un texto A que es el mensaje aleatorio, y un texto B, ingresa el texto b dentro del texto a separado por cada tamaño del texto B sobre 2.

Cliente

Atributos

- No tiene

Constructores

- No tiene

Métodos

- clearScreen(): Método auxiliar para limpiar pantalla.
- Menú(): método que maneja el menú de “login”.
- Encriptar(): Este método recibe un string y retorna un string en base 64 encriptado.
- Desencriptar: Este método recibe un string en base 64 y retorna el string origina.
- Main(): Método que gestiona la conexión al Servidor, enviendo y recibiendo los parámetros como, el usuario, recibiendo cadena de texto aleatorio, generar un md5 con la mezcla del texto aleatorio y la contraseña del usuario y enviando el md5 para generar o no una conexión de “login”.

MD5

Atributos

- INT_A: variable con el valor hexadecimal 0x67452301.
- INT_B: variable con el valor hexadecimal 0xEFCDAB89L.
- INT_C: variable con el valor hexadecimal 0x98BADCFEL.
- INT_D: variable con el valor hexadecimal 0x10325476.
- SHIFT_AMTS: arreglo con los valores 7, 12, 17, 22, 5, 9, 14, 20, 4,11, 16, 23, 6, 10, 15, 21.
- TABLE_T: Arreglo que tiene valores de una función sin().

Ruiz Lozano Paulo Cesar

Constructores

- No tiene

Métodos

- CalculaMD5(): recibe un arreglo de bytes que es el mensaje a general el MD5.
- toHexString(): recibe un arreglo de bytes a cambiar a String.
- getMD5(): Retorna el MD5 generado en forma de cadena.

Servidor

Atributos

- Usuarios: lista ligada de los usuarios que existen en la base de datos.

Constructores

- No tiene

Métodos

- Main(): Crea un Socket servidor y varios hilos para peticiones, también lee la base de datos de usuarios y los carga en ram.
- Encriptar(): Este método recibe un string y retorna un string en base 64 encriptado.
- Desencriptar: Este método recibe un string en base 64 y retorna el string origina.
- imprimeUsuarios(): Función que imprime todos los usuarios de la lista ligada.

GeneraPeticion

Esta clase es hija de la clase Thread o hilo

Atributos

- Usuarios: Lista ligada de usuarios.
- Entrada: Buffer lector.
- Salida: permite escribir al socket.
- Socket: Socket cliente.

Constructores

- Recibe un Socket cliente y una lista de usuarios.

Métodos

- Run(): Se encarga de leer y escribir en el buffer, es la transmisión de datos entre el cliente y servidor, compara si existe el usuario en el servidor, genera un texto aleatorio, mezcla el texto, envía el texto aleatorio, compara el MD5 del

cliente con el del servidor y permite o no una conexión si son iguales.

- buscaUsuario(): Función que busca un usuario y retorna el índice de este en el arreglo.
- generaTexto(): Función que genera un texto aleatorio. De 586 caracteres

LectorArchivo

Atributos

- ContendioArchivo: lista ligada que tiene todo el archivo

Constructores

- No tiene

Métodos

- leerArchivo(): función privada que lee un archivo pasado por parámetros y almacena el contendio en la variable contendioArchivo.
- procesaArchivo(): Genera una lista ligada de todos los usuarios de la “base de datos”.

Conclusión

Es importante conocer la capa de enlace de datos el como lleva la seguridad de la información de usuario al manejar datos críticos como lo es la contraseña. Es impensable que los sistemas de bancos envíen en texto plano o siquiera encriptado la contraseña del cliente por la internet, se requieren diferentes métodos para salvaguardar la integridad de la información del cliente. Este programa permite ver como se lleva acabo el proceso de generación de mensajes aleatorios y el uso de MD5 para comparar si el usuario colocó bien la contraseña sin necesidad de enviar esta por e medio.