El Ataque de los Bots

ITESM CEM | Programación de estructuras de datos y algoritmos fundamentales
A01750150 Hortencia Alejandra Ramírez Vázquez
A01745037 Gala Flores García

Clases y Métodos

```
a01750150:~/environment/tc1031/ProyectoIntegrador $ ./main bitacora1.txt

loading...

****************************************************************
1) LAS CINCO DIRECCIONES IP CON MÁS ACCESOS ILEGALES
124.98.4.152 con 9 accesos ilegales
145.24.21.210 con 7 accesos ilegales
12.198.179.180 con 6 accesos ilegales
179.78.228.48 con 6 accesos ilegales
11.181.68.12 con 6 accesos ilegales
****************************************************************
2) FRECUENCIA DE LOS DISTINTOS MENSAJES DE ERROR
Failed password for illegal user guest: 45
Failed password for admin: 14
Failed password for illegal user root: 22
Illegal user: 19
****************************************************************
3) NUMERO DE PUERTOS DISTINTOS ATACADOS EN TOTAL: 79
****************************************************************
4) SEMANA EN LA QUE SE DETECTARON MAYOR CANTIDAD DE ACCESOS ILEGALES
Semana del 11-17 de mayo de 2020
Es la semana 20 del año 2020
```

Clase Bitacora

Obtener los datos que se encuentran en las bitácoras. Se tienen 3 funciones: 1) operator: ordenar los días de la bitácora (pregunta 4) 2) getSemana: define de qué día a que día es una semana del año 2020 por mes.

Clase Direcciones

Obtiene las direccines Ip, con un contador que permite guardar el número de veces que aparece la la dirección Ip en cada una de las bitácoras analizadas. Esta es una clase que sirve de manera auxiliar en la pregunta 1.

Lectura de Datos

1.Argumento en la lectura del Archivo
El programa recibe como argumento el nombre del archivo que se va a analizar mediante la llamada al programa, es decir, el primer argumento es el nombre del programa (main) y como segundo argumento el nombre del archivo de bitacora que se analizará (bitacora1.txt). En caso de que se reciba más de estos dos argumentos, el programa muestra un mensaje de error. La forma en la que se espera que se espera que sea esta llamada al programa es: **./main bitacora1.txt**

2.Separación de la lectura del archivo
La lectura del archivo se hace usando las funciones pertenecientes a la librería <fstream> de c++ para hacer lectura de archivo, mediante la clase ifstream, se declaro un objeto de este tipo, y con los métodos is_open(), en caso de que el archivo no existe se manda un mensaje de error. A través del método getline, se va leyendo línea por línea que va separando y guardando en variables diferentes los distintos elementos de cada bitacora, identificandolos ya sea por espacios (' ') o por dos puntos (:).

3.Creación de vectores
Conforme se va leyendo cada línea se van guardando cada uno de los valores en diferentes vectores para cada uno de los elementos de la bitácora.

4.Clase Bitácora
Una vez que se tienen los vectores de cada uno de los elementos, se crearon objetos de tipo bitácora que fuera asignando cada uno de los elementos y mediante un vector de tipo bitácora se fue asignando una por una a este vector.

Pregunta 1

5 Direcciones IPs con más accesos ilegales

Se seleccionó la estructura de hashtables, ya que debido a sus características esta estructura se pudo utilizar como contador, de tal forma que las key fueran las direcciones Ip y el value el contador del número de veces que fue atacada la dirección.
Se utilizó la clase hashtables y la función realizada en la práctica 5 *contains_key.* Esta función regresa true si el valor se encuentra en la hashtable. Se obtienen las direcciones IPs de la clase Bitácora y con contains_key checa si el valor se encuentra, en caso de encontrarse dicha durección obtiene el valor actual asignado e introduce en la hashtable la dirección y su valor (contador) actualizado. En caso de que no se encuentre en la hashtable, la dirección ip es agregada a la hashtable y el valor de 1, además de que la dirección ip se va guardando en un vector de direcciones.
Haciendo uso de la clase Direcciones, se recorre cada una de las direcciones almacenadas en la hashtable y se crean objetos de este tipo para después irlos guardando en un vector.
Se hace un ordenamiento del vector de mayor a menor, usando la función sort de la librería <algorithm>.

Si hay direcciones IPs con la misma cantidad de accesos ilegales, solo mostrara aquellos 5 primeros que se encuentren en nuestro vector *direccionesCont.*

Pregunta 2

Frecuencia en los mensajes de error

En este caso se seleccionó la estructura de vectores ya que solo se tenían 4 diferentes mensajes de error además de ya conocerlos previamente al análisis del archivo, y ordenar los datos no sería complicado.
Se crearon vectores para cada razón de falla y asignando con un vector bitácora e ir llenándolos. Esto para tener el tamaño final de cada vector y haciendo uso de la función size() de la librería <vector> se obtuvo el tamaño de cada tipo de razón, y son estos valores lo que son mostrados.

Pregunta 3

Números de puertos distintos atacados en total

Se selecciono el uso de vectores para hacer el anásis de puertos, debido a que lo único que se nos pide es hacer el conteo total de puertos atacados, al no tener que analizarlos, los vectores nos ayudan a ir almacenando cada uno de los diferentes puertos que resultaría de hacer el conteo.
Mediante un vector vacío *puertos*, se fue llenando usando nuestra clase Bitacora para obtener los números de puertos. Compara este vector con los de bitácora y si encuentra otro igual, le suma 1 al contador. Solamente se agrega al vector aquellos que su contador es 0. Así mostramos el tamaño final del vector *puertos.*

Pregunta 4

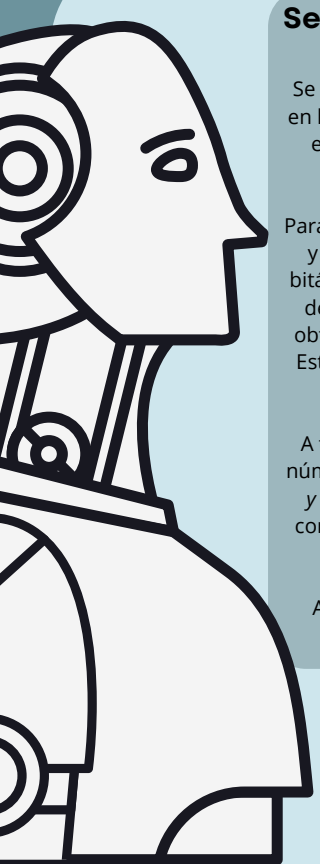
Semana en la que se detectaron una mayor cantidad de accesos ilegales

Se implementó las estructuras de hashtables para almacenamiento de cada una de las diferentes semanas que hay en la bitácora analizada, esto porque esta estructura debido a sus características nos permitió hace un contador que estuviera almacenado en la hashtable por cada semana, en dónde la key fue el número de semana y su value el contador de número de accesos ilegales.

Para hacer el análisis primero se transformo la palabra del mes a número del mes que se encontraban en la bitácora, y a través de la clase Bitácora se asignó a cada objeto como un atributo. Se realizó un análisis de cada una de las bitácoras y con ayuda de vectores se fueron almacenando los diferentes meses. Cada vector de cada mes se ordeno de manera ascendente de acuerdo al día, esto mediante la implementación de la función sort. Posteriormente se obtuvó la semana de cada bitácora haciendo uso del método getSemana() creado previamente en la clase Bitácora. Estas semanas fueron agregadas a un vector y después se ordeno de menor a mayor con la función sort, esto para un mejor manejo de los datos.

A través de hashtables en donde la key era el número de semana y el value el contador que nos permite revisar el número de accesos ilegales que hubo, se hace uso de la clase hashtables creada en clase, y del método *contains_key() y put()*, se recorre el vector de semanas y se va agregando a la hashtable en caso de no existir la semana, en caso contario solo le suma "1" al value de la key. Como último paso se obtiene la semana con mayor número de accesos ilegales.

Al finalizar se hace uso de la función nombreSemana que recibe el número de semana con más acesos ilegales.



Si hay direcciones semanas con la misma cantidad de accesos ilegales, solo mostrara aquella que el programa detecte primero.

Función nombreSemana()
Se le asigna de que día a qué día esta compuesta cada semana del año (53 semanas). De acuerdo al número de semana, despliega el mensaje con los días y el mes que es esa semana.