Reflexión

Esta actividad consistió en utilizar una estructura de grafo para manejar una bitácora con la información de las IP de las que provenían los registros de ataque a otras direcciones IP. La actividad nos pidió después de leer del archivo “bitácora.txt”, que almacenaba tanto IPs como los registros de ataques, almacenar la información obtenida en una lista de adyacencia. Después, ya generada la lista de adyacencia era necesario determinar el fan-out de cada nodo y establecer qué nodos eran los de mayor fan-out, los cuales contendrían las direcciones IP del boot master de los ataques.

Empezamos a plantear esta actividad con la lectura del documento “bitacora.txt”. En el main leemos la primera línea del documento “bitacora.txt” que contiene n y m, que son el numero de nodos y el número de arcos respectivamente, que son almacenados en variables. Después creamos un vector de vectores de enteros para formar la lista de adyacencias, llamada listAdj, de tamaño n. También creamos nodos_ip, que es un unordered map que recibe como clave el string de dirección IP atacante y como dato un par formado por el número de nodo en el que se encuentra en la lista de adyacencia y el número de outdegree de dicho nodo que corresponde al fan-out del IP atacante.

Luego llamamos a la función leerArchivo, que recibe como parámetros listAdj, nodos_ip, n y m. En leerArchivo lo que se hace es abrir “bitacora.txt” y guardar los valores de n y m. Después en un ciclo for, lee las siguientes n líneas de la bitácora que contienen los n IPs. El string de la IP se almacena en el unordered map como la clave de un nodo para facilitar su manejo, mientras que el dato al que apunta se compone de un par llamado ip en el que se le asigna el número de ciclo como su número de nodo y se le da un outdegree de 0.

En otro for de 0 a m se leen las siguientes m líneas contienen los m registros con los que se formaran los arcos del grafo. Se leen del archivo el mes, día, hora, IP atacante e IP atacado en sus respectivas variables, siendo el IP atacante ipi y el atacado ipi2. Los valores de ipi e ipi2 se toman hasta el puerto y se usan como índices para acceder al nodo que almacene dicha IP en el unordered map. Estos nodos se asignan a las variables de numNodoSale y numNodoEntra. En la lista de adyacencia se añade al vector en el índice numNodoSale el valor de numNodoEntra para ir completando la lista de adyacencia.

Finalmente se imprimen los nodos con mayores fan-out con sus respectivas IP de boot master desde las que se efectúa el ataque.

En este problema podemos apreciar la utilidad del uso de estructuras como los grafos, que son bastante versátiles y fáciles de utilizar para representar redes de nodos conectados, en este caso las direcciones IP que ejecuten ataques a otras direcciones IP. Esta estructura acompañada de otros algoritmos, o en este caso también de otra estructura como el unordered_map, puede lograr formar una lista de adyacencia que contenga las conexiones con la información necesaria para representar y almacenar una red de nodos apropiadamente y así manejar grandes cantidades de datos de manera eficiente permitiendo gran escalabilidad y un fácil manejo de los datos.