Tecnológico de Monterrey

A01713396 - Santiago Martin del Campo Soler

Reflexión de la Actividad 5.2 – Uso de Códigos Hash.

En esta actividad 5.2, se trabajó con una bitácora de tráfico de red que registra las conexiones entre distintas direcciones IP. A diferencia de las actividades anteriores a la actividad 4.3, donde la información se analizaba de forma cronológica, en este caso el enfoque se centró en estudiar las relaciones existentes entre las IPs, ya que una misma dirección podía comunicarse con múltiples destinos.

Para analizar esta conexión entre las distintas direcciones IP leímos el archivo “bitacoraGrafos.txt”, la bitácora mencionada anteriormente, posteriormente almacenamos los datos en una estructura de grafos, esta misma basada en listas de adyacencia.

Cada dirección IP fue representada mediante un Nodo, una conexión como una arista dirigida desde la dirección IP de origen hasta la IP de destino.

Esta forma de representación permitió organizar y analizar la información de manera más clara y eficiente, facilitando el estudio de las relaciones entre los nodos, algo que no habría sido posible de forma óptima utilizando únicamente arreglos o listas lineales.

Importancia y eficiencia de las tablas hash en el análisis de bitácoras

El uso de tablas hash en esta actividad fue fundamental para manejar de forma eficiente la gran cantidad de direcciones IP presentes en la bitácora. Dado que el análisis requiere consultar repetidamente información asociada a cada IP, como el número de conexiones salientes y entrantes, una tabla hash permite acceder a estos datos en tiempo constante promedio O(1), evitando recorridos lineales costosos sobre el grafo.

Las operaciones principales de la tabla hash, como la inserción y la búsqueda, tienen una complejidad promedio de O(1), lo cual representa una ventaja importante frente a otras estructuras más simples como listas o arreglos. Sin embargo, esta eficiencia depende directamente del número de colisiones generadas.

A medida que el número de colisiones aumenta, especialmente cuando la tabla se acerca a su capacidad máxima, la complejidad de las operaciones puede verse afectada, dejando de ser constante y acercándose a O(n) en el peor de los casos. Por esta razón, fue importante analizar cómo iba variando el número de colisiones al modificar el tamaño de la tabla hash, ya que un mal dimensionamiento del hash puede provocar una disminución significativa en el rendimiento del sistema.

Importancia del Uso de Tablas hash en este caso

El uso de tablas hash en este tipo de problemas resulta altamente eficiente siempre que se controle el factor de carga y el manejo de colisiones. Esta estructura permitió complementar el análisis realizado con grafos, facilitando consultas rápidas y mejorando el desempeño general del programa al trabajar con grandes cantidades de información.

En el contexto de la ciberseguridad, está estructura de datos puede resultar clave para analizar posibles amenazas, debido a que puede identificar direcciones IP específicas que cuentan con actividad sospechosa.

Reflexión final

El uso de tablas hash ha sido una novedad dentro de esta actividad, nos permite complementar a los grafos que utilizamos en la actividad pasada, mas sin embargo, en conjunto funcionan bastante bien.

En mi caso se me dificulto implementar y entender del todo los grafos y tablas hashes, estos últimos dos temas los he considerado bastantes complicados de entender, pero con la ayuda de mis compañeros y una pequeña investigación pude comprender de una manera sencilla el uso de grafos y su importancia.

Referencias Bibliográficas:

Bose, A., & Bose, A. (2025, 14 febrero). *Malware Detection: How to detect and remove malware ?* Seceon Inc. https://seceon.com/malware-detection-how-to-detect-and-remove-malware/

GeeksforGeeks. (2025, 23 julio). Implementation of Graph in C++. GeeksforGeeks. https://www.geeksforgeeks.org/cpp/implementation-of-graph-in-cpp/

GeeksforGeeks. (2026, 21 enero). Dijkstra's algorithm. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/dijkstras-shortest-path-algorithm-greedy-algo-7/

Linkurious. (2026, 8 enero). Cybersecurity graph visualization: Assessing vulnerabilities. Linkurious. https://linkurious.com/blog/graph-data-visualisation-cyber-security-threats-analysis/

PuppyGraph. (2025, 21 septiembre). PuppyGraph | Query your relational data as a graph. no ETL. https://www.puppygraph.com/blog/graphs-for-cybersecurity#why-do-you-need-graphs-for-cybersecurity

GeeksforGeeks. (2025c, diciembre 22). *Heap sort*. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/heap-sort/