

Las estructuras de datos tipo árbol, como el heap utilizado en los códigos anteriores, son fundamentales en situaciones donde se necesita organizar y acceder a datos de manera eficiente. En el contexto de los códigos proporcionados, el uso de un heap es esencial para ordenar los registros de una bitácora por IP y realizar operaciones de inserción y eliminación eficientes.

En todo el código usamos algoritmos de Ordenamiento:

- Bubble sort:  $O(n^2)$
- Insertion Sort:  $O(n^2)$
- Merge Sort:  $O(n \log n)$
- Quick Sort: peor caso  $O(n^2)$  y su mejor caso  $O(n \log n)$

Algoritmos de búsqueda:

- Búsqueda Lineal:  $O(n)$
- Búsqueda Binaria:  $O(\log n)$
- Búsqueda en árboles binarios de búsqueda (BST): En promedio  $O(\log n)$ , pero en el peor caso podría ser de  $O(n)$  si el árbol no está balanceado.

La eficiencia del uso de estructuras de datos tipo árbol radica en su capacidad para manejar grandes conjuntos de datos de manera rápida y optimizada. En el caso del heap, por ejemplo, la inserción y eliminación de elementos se realizan en tiempo logarítmico en relación con el número de elementos en la estructura, lo que lo hace especialmente útil para aplicaciones que requieren acceso rápido y eficiente a los datos.

En cuanto a la determinación de si una red está infectada o no utilizando la aplicación proporcionada, se pueden seguir varios pasos:

Primero, haciendo un análisis de la bitácora utilizando la función que ordena los registros de la bitácora por IP, se pueden identificar las direcciones IP que están accediendo al sistema. Después se puede hacer un conteo de accesos por IP e identificación de patrones, si una dirección IP está accediendo repetidamente a áreas sensibles del sistema o realizando solicitudes anormales, podría ser un indicio de actividad maliciosa.

Al final podemos hacer una comparación con bases de datos de amenazas conocidas. Se puede integrar la aplicación con bases de datos de amenazas conocidas para comparar las direcciones IP detectadas con listas negras de direcciones IP conocidas por estar asociadas con actividades maliciosas.

Referencias:

Binary Heap. (2014, November). GeeksforGeeks; GeeksforGeeks.

<https://www.geeksforgeeks.org/binary-heap/>

ALGORITMIA ALGO+ - Algoritmos y Estructuras de Datos. (2024). Algoritmia.net.

<http://www.algoritmia.net/articles.php?id=17>

Ordenación, D., & Búsqueda, Y. (n.d.).

<https://www.mheducation.es/bcv/guide/capitulo/8448198441.pdf>