Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus CEM (Edo. de México)

Reflexión Actividad Integradora

Juan Antonio Figueroa Rodríguez A01369043

Programación de estructuras de datos y algoritmos fundamentales.

Grupo 850.

17 de abril de 2024.

De acuerdo a la situación problema planteada, se requiere almacenar información pertinente para detectar accesos maliciosos a través de redes de robots. A continuación, se presentan algunas estructuras de datos que pueden ser útiles en este contexto:

- Listas Enlazadas (Linked Lists): Una lista enlazada es una estructura de datos lineal donde cada elemento, llamado nodo, contiene un valor y una referencia al siguiente nodo. En este caso, una lista enlazada podría ser utilizada para almacenar la información recopilada sobre los accesos maliciosos. Sin embargo, debido a que la búsqueda eficiente es un requisito importante, una lista enlazada simple no sería la mejor opción, ya que la búsqueda requeriría recorrer todos los nodos de la lista hasta encontrar el acceso deseado.
- Listas Doblemente Enlazadas (Doubly Linked Lists): Una lista doblemente enlazada es similar a una lista enlazada, pero cada nodo también contiene una referencia al nodo anterior. Esto permite una búsqueda más eficiente, ya que se puede recorrer la lista en ambas direcciones. En este caso, una lista doblemente enlazada podría ser preferible sobre una lista enlazada simple, ya que permitiría una búsqueda más rápida de la información relevante.

Para determinar si un acceso es malicioso, se pueden utilizar diferentes técnicas y algoritmos de detección de intrusiones. Algunas posibles estrategias incluyen:

- Análisis de comportamiento: Se puede monitorear el comportamiento de los accesos y buscar patrones o actividades sospechosas. Por ejemplo, si un acceso intenta realizar múltiples solicitudes en un corto período de tiempo o si intenta acceder a recursos no autorizados, podría ser considerado como malicioso.
- Filtrado de direcciones IP: Se pueden utilizar listas negras o listas de direcciones IP conocidas por ser maliciosas para bloquear o restringir el acceso desde esas direcciones.
- Análisis de firmas: Se pueden utilizar bases de datos de firmas conocidas de ataques o malware para identificar accesos maliciosos que coincidan con esas firmas.

Para distinguir los accesos "reales" de los maliciosos, se pueden utilizar diferentes técnicas de autenticación y autorización. Algunas posibles métodos incluyen:

- Autenticación de usuarios: Se puede requerir que los usuarios proporcionen credenciales válidas, como un nombre de usuario y una contraseña, para acceder a los recursos.
- Control de acceso basado en roles: Se pueden asignar diferentes niveles de acceso a los usuarios según su rol o nivel de autorización. Esto permite restringir el acceso a ciertos recursos solo a usuarios autorizados.
- Verificación de integridad de datos: Se pueden utilizar técnicas de verificación de integridad, como firmas digitales o hashes, para asegurarse de que los datos no hayan sido modificados o manipulados de manera maliciosa.

Para realizar grupos de información que caractericen estos accesos, se pueden utilizar diferentes criterios de clasificación. Algunas posibles características que se pueden utilizar para agrupar la información incluyen:

- Tipo de acceso: Se pueden agrupar los accesos según el tipo de actividad que están realizando, como solicitudes de archivos, intentos de inicio de sesión, etc.
- Origen del acceso: Se pueden agrupar los accesos según su origen, como direcciones IP o ubicaciones geográficas.
- Patrones de comportamiento: Se pueden agrupar los accesos según patrones de comportamiento similares, como accesos que ocurren en momentos específicos del día o que siguen una secuencia de acciones similar.

En cuanto a la eficiencia del uso de diferentes estructuras de datos lineales, es importante considerar la complejidad computacional de las operaciones básicas de cada estructura. En el caso de una Doubly Linked List, las operaciones básicas tienen las siguientes complejidades:

- Inserción: O(1) en el mejor caso (al inicio o al final de la lista), O(n) en el peor caso (cuando se inserta en una posición específica).
- Borrado: O(1) en el mejor caso (cuando se borra al inicio o al final de la lista), O(n) en el peor caso (cuando se borra en una posición específica).
- Búsqueda: O(n) en el peor caso, ya que se debe recorrer la lista enlazada para encontrar el elemento deseado.

La elección de una Doubly Linked List sobre una Linked List simple se justifica por la capacidad de realizar búsquedas más eficientes en ambas direcciones. Aunque la

complejidad de las operaciones básicas es similar, la capacidad de búsqueda más rápida puede ser crucial en la detección oportuna de accesos maliciosos.

En cuanto a los algoritmos de ordenamiento, es necesario tener en cuenta que no se mencionan en la pregunta original. Por lo tanto, no se puede proporcionar información sobre la comparación de algoritmos de ordenamiento o su complejidad temporal.