Luis Cano Irigoyen – A00827178

Reflexión Individual

Actividad Integradora 1

Análisis y diseño de algoritmos avanzados, Gpo 8

Para esta actividad integradora nos basamos una situación problema real, siendo la intercepción y modificación con fines maliciosos de paquetes de información que van de un dispositivo a otro –incrustándoles scripts o pequeños programas–, para la cual nosotros generamos maneras para identificar estos diversos tipos de datos maliciosos ocultados en diferentes partes del flujo de bits de una transmisión.

La situación problema se basó en tres enfoques diferentes, para los cuales utilizamos algoritmos avanzados que pudieran identificar el código malicioso en cuestión:

1. Buscar líneas de texto/código específicas que se sabe que son maliciosas dentro de los archivos de transmisión.
   Para realizar este caso de manera optimizada utilizamos el algoritmo de búsqueda KMP, en el cual realizamos un arreglo LPS de los mayores valores prefijo sufijo para el texto malicioso que se desea buscar en la transmisión, con complejidad de O (longitud del texto malicioso), y luego realizamos la búsqueda de estos patrones en la transmisión para encontrar al texto, con complejidad de O (longitud de la transmisión).
   Nuestra solución imprime un booleano especificando si el texto fue encontrado en la transmisión o no, y en caso de serlo imprime el índice de en que punto de la transmisión se encontró.

2. Se asume que el código malicioso buscado siempre va a tener palíndromos de chars, por lo cual estos los buscamos en las transmisiones. Se debe encontrar el palíndromo más largo dentro de cada transmisión.
   Aquí implementamos el algoritmo de Manacher para *longest palindromic substring* en dónde se crea un arreglo auxiliar LPS de las longitudes máximas de palíndromos, con el cual se navega el texto de transmisión reposicionando las fronteras izquierda y derecha de cada palíndromo y guardando en el centro de cada palíndromo la longitud de este.
   Para evitar problemas de simetría en palíndromos de tamaño par agregamos un símbolo “$” entre cada carácter de la transmisión.
   Con el arreglo de LPS ya listo, simplemente se busca el mayor valor, y una vez encontrado regresamos el índice/posición (iniciando en 1) en donde inicia y termina el código "espejeado" más largo en cada transmisión.

3. Dos transmisiones son sospechadas de haber sido intervenidas y de traer el mismo código malicioso, para lo cual hay que analizar qué tan similares son dos archivos de transmisión, buscando el substring más largo común entre ellos.
   Para este caso realizamos un algoritmo de Longest Common Substring en el cual creamos una matriz de tamaño [longitud de transmisión1 + 1] [longitud de transmisión2 + 1] para almacenar las longitudes de los sufijos de substrings más largos. Esta matriz luego la construimos de manera bottom up, guardando los substrings comunes en la matriz, así como guardando la longitud y posición del substring común más largo cada vez que se encontraba uno mayor.
   Nuestra solución encuentra el substring más largo común entre las dos transmisiones, y como resultado imprimimos la posición inicial y final (iniciando en 1) del primer archivo de transmisión en donde se encuentra el substring.

Considero a la situación problema para el curso de *Análisis y Diseño de Algoritmos Avanzados* como un éxito, ya que esta nos permitió poner en práctica los conceptos de algoritmos avanzados vistos para crear soluciones a problemáticas similares a la realidad, adicionalmente al habernos llevado a aprender sobre otros algoritmos y estructuras los cuales no requerimos para solucionar el reto.