Mario Frías Piña - A01782559
Juan Pablo Cruz - A01783208
Ma. Fernanda Cortés Lozano - A01026613

Actividad Integradora 1

Reflexión Evidencia Final

Durante el desarrollo de la primera evidencia, trabajamos de la mano con cadenas de caracteres, encontrándolas en diferentes formatos y objetos de comunicación. Se estableció un escenario donde se envía una serie de datos. Sin embargo, existen personas malintencionadas que interceptan estos mismos datos y los modifican con el objetivo de tomar control del dispositivo de destino.

Nuestro trabajo es identificar estos archivos maliciosos dentro de los archivos de transmisión para aplicar parámetros de seguridad y evitar cualquier ataque a nuestro dispositivo. Para lograrlo, hemos implementado ciertos algoritmos de manejo de cadenas de texto para identificar, manipular y neutralizar código malicioso dentro de los archivos de transmisión. Esta práctica permite que el usuario se sienta seguro ante cualquier peligro desconocido, simplificando la función de las defensas o protocolos que tiene ante estas mismas amenazas.

El primer método o algoritmo que utilizamos fue el de Knuth-Morris-Pratt (KMP) para determinar si existía algún código de dudosa procedencia. Sin embargo, también exploramos la misma solución con el algoritmo de la función Z. Ambos son eficaces para la búsqueda de subcadenas, lo que es esencial en la detección de patrones específicos de código que podrían indicar una intromisión.

El algoritmo KMP es particularmente útil por su eficiencia, ya que evita la revisión redundante de caracteres que ya han sido comparados. Esto acelera el proceso de detección, lo que es vital cuando se manejan grandes volúmenes de datos en tiempo real. Por otro lado, la función Z aporta su capacidad para preprocesar la cadena de texto y determinar la longitud de los segmentos que coinciden con el prefijo más largo de la cadena. Esto puede ayudar en la rápida localización de estructuras sospechosas que podrían estar escondidas en los datos.

Además, aplicamos el algoritmo de Manacher para identificar secuencias palindrómicas, que son comúnmente utilizadas en código malicioso para evadir la detección. A pesar de que el uso de memoria dinámica puede sugerir un mayor control y flexibilidad, reconocemos la

importancia de las buenas prácticas de programación, optando por contenedores automáticos como std::string y std::vector en C++ para un manejo de memoria más seguro y eficiente.

También implementamos la técnica del Longest Common Substring (LCS) para encontrar similitudes entre archivos de transmisión y bases de datos de códigos conocidos como maliciosos. La utilización de una matriz de programación dinámica permite rastrear estas coincidencias de manera eficaz, mientras que la estructura LCS_Result que hemos definido para devolver los resultados, refuerza la claridad y la estructura del código.

La reflexión sobre estos procesos nos lleva a entender que, en la seguridad informática, es crucial contar con algoritmos rápidos y eficientes. La habilidad para adaptar y combinar diferentes métodos matemáticos y de ciencia de la computación es fundamental para mantener un paso adelante de las amenazas cibernéticas. Además, es imperativo implementar una actualización constante de los patrones de búsqueda basados en las tendencias emergentes de malware, lo que requiere un análisis constante y un aprendizaje automático efectivo.

Finalmente, estos algoritmos son solo una parte de un sistema de defensa robusto. La seguridad cibernética es un campo de batalla dinámico, donde las medidas preventivas y reactivas deben evolucionar continuamente para enfrentar amenazas cada vez más sofisticadas. Por ello, la educación y la conciencia sobre las mejores prácticas de seguridad deben ser una prioridad para todos los usuarios, complementando así la tecnología de detección y prevención con un comportamiento prudente en el ciberespacio.