Reflexión Actividad 4.3

Cristóbal Alberto Escamilla Sada - A00827074

El propósito de la actividad realizada fue encontrar las conexiones de las direcciones IP en el archivo de entrada y almacenarlas en un grafo. Esta es una estructura de datos en la que no hay restricciones sobre las conexiones y/o direcciones entre los nodos. En esta actividad pudimos simular el análisis de una red infectada mediante las conexiones entre dichas direcciones. Se leyó el archivo de entrada conteniendo más de 100,000 renglones de información y se extrajeron las direcciones IP así como las conexiones entre las mismas para posteriormente almacenarlas en una lista de adyacencia perteneciente a un grafo. Los datos también se almacenaron en un unordered_map— parte de std conteniendo la dirección como llave y un par como elemento. El par contiene el índice de la dirección así como su fan-out. Al fan-out también se le conoce como outdegree y este señala el número de conexiones que salen del nodo. Esta función de lectura tiene una complejidad lineal ya que n (número de renglones) cantidad de veces. También se realizó la función determinaFanOut cuyo propósito es desplegar el fan-out de la dirección ingresada por el usuario. Esta es de complejidad constante ya que utiliza el unordered_map usando la dirección IP como índice. Finalmente se realizó la función bootMaster cuyo propósito es desplegar la dirección IP del bootmaster. Esta dirección es la que contiene el mayor número de fan-outs. Por lo tanto en el caso de una red infectada se puede suponer que la dirección del bootmaster es la del origen de la infección. Esto dado que el bootmaster tiene el mayor número de conexiones saliendo hacia otras direcciones. Esta función tiene una complejidad lineal ya que itera sobre el unordered_map para encontrar el nodo.

Un problema de esta naturaleza requiere un sistema de seguridad para ser resuelto. A través de una búsqueda de internet encontré que los grafos son altamente utilizados en los sistemas de seguridad. A estos grafos se les conoce como *Attack Graphs*. Por definición los attack graphs son estructuras que modelan todas las posibilidades de atacar una red. Hay dos tipos de attack graphs que son los más utilizados en la industria de la ciberseguridad. El primer tipo es un grafo dirigido en el que los nodos representan el estado de una red mientras sus aristas representan *exploits* que pueden cambiar el estado de la red. Un exploit se define como un código que se aprovecha de una vulnerabilidad en el software para provocar malignidades (Exploit, definición y características, 2020). El segundo tipo de attack graph también es un grafo dirigido en el que los nodos representan la pre- o post-condición de un exploit mientras que sus aristas representan las consecuencias en caso de que las condiciones para habilitar el exploit se cumplan (What is Attack Graphs, 2020).

Los attack graphs se pueden ordenar, clasificar o agrupar para de esta manera hacer un análisis profundo de las vulnerabilidades de una red. Todo esto para prevenir un ataque cibernético. Estos ataques son sumamente comunes hoy en día y existen compañías dedicadas únicamente a supervisar las redes para prevenir dichos ataques. En estos tipos de ataques se frecuenta el robo de información así como la manipulación del sistema infectado. Es algo sumamente peligroso y costoso. Un ejemplo de un ataque cibernético es la infección de una red bancaria en la que el *hacker* redirecciona las conexiones para recibir transferencias de dinero destinado a alguien más. Esto causa grandes pérdidas. No solamente podría uno robar dinero sino que también robar información confidencial. Hay una infinidad de consecuencias negativas que pueden ocurrir a raíz de un ataque de esta naturaleza. Precisamente para evitar este tipo de ataques, se pueden obtener recomendaciones sobre cuál es el camino más vulnerable en una red y por

lo mismo dónde colocar sistemas de detección de intrusos. Todo esto mediante el análisis de los grafos de ataque. Los *hackers* se dedican a infectar estas redes por lo que buscan nuevas maneras de hacerlo cada día. Por esto mismo las empresas de ciberseguridad usan una basta cantidad de grafos para proteger a sus clientes (Shandilya et al., 2014).

Los problemas de esta naturaleza son bastante comunes en nuestra vida cotidiana. Es por esto que me parece sumamente interesante este proyecto ya que estamos aplicando nuestros aprendizajes sobre las estructuras de datos en casos de la vida real. Yo no sabía absolutamente nada sobre la ciberseguridad. Sin embargo, con lo que acabo de investigar me quedo sorprendido, intrigado y motivado a seguir aprendiendo sobre este tema tan interesante.

Referencias

Exploit, definición y características - Panda Security. (2020). Retrieved November 21, 2020, from Pandasecurity.com website: https://www.pandasecurity.com/es/security-info/exploit/

Shandilya, V., Simmons, C. B., & Shiva, S. (2014). Use of Attack Graphs in Security Systems. *Journal of Computer Networks and Communications*, *2014*, 1–13. https://doi.org/10.1155/2014/818957

What is Attack Graphs | IGI Global. (2020). Retrieved November 21, 2020, from Igi-global.com website: https://www.igi-global.com/dictionary/attack-graphs/1745