PUCE Virtual

DOCUMENTO DE PROFUNDIZACIÓN

Estructura de Datos

Laboratorio de Contenido 3: Algoritmo de Dijkstra aplicado al análisis de rutas de ataque

Laboratorio de Contenido 3: Algoritmo de Dijkstra aplicado al análisis de rutas de ataque

Explicación básica

Los grafos ponderados permiten representar redes donde cada conexión tiene un valor asociado. En ciberseguridad, este valor puede representar el nivel de dificultad que tendría un atacante para comprometer un sistema.

El algoritmo de Dijkstra permite calcular el camino con menor costo dentro de este tipo de grafos.

Objetivos

- Comprender el concepto de caminos mínimos.
- Identificar cómo funcionan los grafos ponderados.
- Aplicar el algoritmo de Dijkstra.
- Analizar rutas de ataque dentro de una red informática.

Caso práctico

Una empresa desea analizar su infraestructura tecnológica.

Los sistemas identificados son:

- Internet
- Firewall
- Servidor Web
- Servidor Aplicaciones
- Base de Datos

Cada conexión tiene un valor que representa la dificultad del ataque.

Problema

El equipo de seguridad desea identificar cuál sería el camino más sencillo para que un atacante llegue a la base de datos.

Aplicación paso a paso

Paso 1: Definir las conexiones

Internet → Firewall (3)

Firewall → Servidor Web (2)

Servidor Web → Servidor Aplicaciones (2)
Servidor Aplicaciones → Base de Datos (4)

Paso 2: Representar el grafo

```
grafo = {
"Internet":[("Firewall",3)],
"Firewall":[("ServidorWeb",2)],
"ServidorWeb":[("ServidorApp",2)],
"ServidorApp":[("BaseDatos",4)],
"BaseDatos":[]
}
```

Paso 3: Aplicar Dijkstra

```
import heapq

def dijkstra(inicio):
    distancias = {nodo:float("inf") for nodo in grafo}
    distancias[inicio] = 0

    cola = [(0,inicio)]

    while cola:
        distancia_actual, nodo = heapq.heappop(cola)

        for vecino,peso in grafo[nodo]:
            distancia = distancia_actual + peso

            if distancia < distancias[vecino]:
                distancias[vecino] = distancia
                heapq.heappush(cola,(distancia,vecino))

    return distancias

print(dijkstra("Internet"))
```

Conclusiones

El algoritmo de Dijkstra permite analizar rutas dentro de redes complejas y determinar cuáles son los caminos más eficientes o vulnerables dentro de una infraestructura tecnológica.

Este tipo de análisis es especialmente útil para identificar puntos críticos dentro de una red informática.