ScanSphere

An Advanced Network Mapper and Analysis Tool

Dhivahari Vivekanandasarma, Turgerel Amgalanbaatar, Hosna Zulali, Katerina Walter, and Kulprawee Prayoonsuk
<table>
<thead>
<tr>
<th>Context</th>
<th>Goal</th>
<th>Goal</th>
</tr>
</thead>
</table>
| Network scanners and vulnerability analysis tools are invaluable in discovering fingerprinting data about network hosts.  
  - Host OS  
  - Open/Closed Ports  
  - IoT Devices | Network mapper tool that generates a high-level map of a network.  
  - Hosts are clustered according to their fingerprint similarity.  
  - Graph Embedding techniques to tackle dimensionality. | This tool provides network administrators:  
  - Holistic view of the network  
  - Assists them in real-time security decisions regarding threat management and analytics. |
High Level Overview
Implementation
Implementation

Data Collection
- Using Python’s *Nmap* library to scan network
- Collecting information using IP addresses
- Prepares raw nmap data for database transfer to MongoDB server

Graph Construction
- Sorting the raw Nmap data into nodes and edges
- Generating Host nodes, Attribute nodes, and edges in MongoDB server
- Using Python’s *NetworkX* library to construct a directed, weighted graph

Graph Embedding Methods (GEM)
- Using graph embedding techniques (GEM library) on constructed graph
- Structural Deep Network Embedding (SDNE)
- Node2Vec
Implementation

**Clustering**
- Using Elbow Method to determine the optimal number of clusters (called $k$) with the dimensionally reduced data
- Implemented K-Means clustering
- Plotted the generated clusters

**GUI**
- Using Python’s `Tkinter` library to build the tool’s GUI
- Inputs:
  - IP Address
  - Filter attributes
- Outputs:
  - Clustered embedded Graph
Data Collection

- Uses Python’s `python-nmap` library to manipulate nmap scan results
- *Python-nmap* offers:
  - Port Scanning
  - Host Name
  - OS Family
  - OS Generation
  - OS Vendor
- ScanSphere has three scanning options for the network:
  - Scan a single IP Address
  - Scan a range of IP Addresses
  - Scan several specific IP Addresses
ScanData.nmapData

COLLECTION SIZE: 6.09KB  TOTAL DOCUMENTS: 42  INDEXES TOTAL SIZE: 36KB

Find  Indexes  Aggregation  SearchBETA

FILTER {"filter": "example"}

QUERY RESULTS 1-20 OF MANY

_id: ObjectId("5e9290407eac9d185e57c12a2")
scanID: 2020-04-11T23:51:16.884+00:00
hostIP: "192.168.86.20"

scanData: Object
  host_name: "android-8946dec95592ecfa.1an"
  host_state: "up"
  os_type: "specialized"
  os_vend: "VMware"
  os_fam: "Player"
  os_gen: "XP"
Graph Construction

- Uses the raw Nmap data collected to create nodes and edges
- Two types of nodes:
  - Host: Contains each scanned host
  - Attributes: Contains each attribute_key/value pair collected in the scan
- Creating edges:
  - Contains the edge between host node and attribute node with the appropriate weight
- Uses Python’s *NetworkX* library with the addition of nodes and edges to construct a directed and weighted graph
nmapData

_id: ObjectId("5e9290407eac9d18e57c22e2")
scanID: 2020-04-11T23:51:16.884+00:00
hostIP: "192.168.85.20"

scanData: Object
  host_name: "android-8946dec95592ecfa.lan"
  host_state: "up"
  os_type: "specialized"
  os_vend: "VMware"
  os_fam: "Player"
  os_gen: "XP"

nodes

_id: ObjectId("5e9b793820306980aa626859")
num_ID: 3345
node_type: "H"
ref_ID: ObjectId("5e9290407eac9d18e57c22e3")

attributes

_id: ObjectId("5e97ee5c9e9bb106c5fc12b1")
att_key: "tcp_open"

att_val: Array
 0: 135
 1: 137
 2: 139
 3: 443

dges

_id: ObjectId("5e9b795020306980aa626882")
node1: 3344
node2: 2552
weight: 1
Graph Embedding Methods (GEM)

- Using Python’s GEM library to embed the graph using SDNE and node2vec:
  - Lower Dimensionality
  - Directed
  - Weighted
- Structural Deep Network Embedding (SDNE)
  - Captures highly non-linear structures by using first and second order proximities
  - No random walks
- Node2Vec
  - Takes random walks while investigating neighboring nodes
  - Utilizes BFS and DFS as parameters to decide next node walk
Structural Deep Network Embedding (SDNE)
node2vec
Clustering

- Using Python’s *sklearn* library for K-Means Clustering
- Elbow method:
  - K-means clustering on our dataset
  - Determine a range of k values
  - From k, different clusters are generated
- Each dataset value will belong to a cluster
Elbow Method (SDNE)
Elbow Method (node2vec)
K-Means (SDNE)
K-Means
(node2vec)
SEE DEMO VIDEO