**High-level problem definition**

"How to win the battle against terrorism"

**Detailed problem definition**

What vulnerabilities exist in the terrorist relations network and how can we exploit this to break it apart?

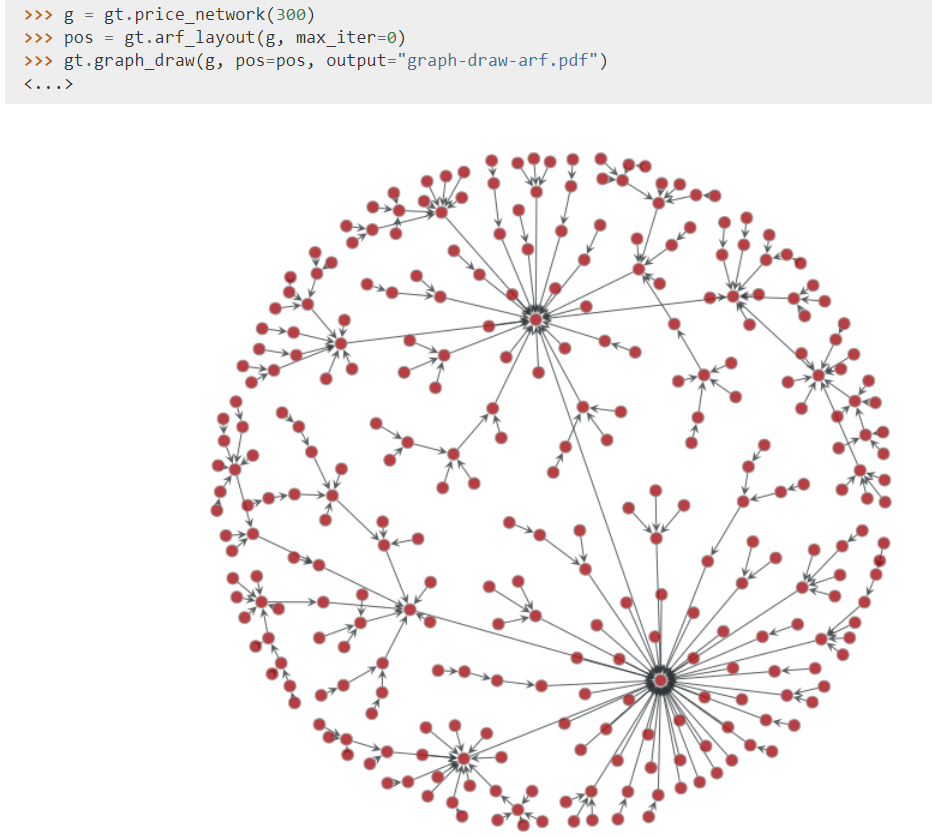
**Some ideas of how to define vulnerability (different colors are different tasks):**

* **Number of communities** (in our largest component) and **their connectivity** with each other.
* **Minimum number of links to cut** to disconnect these communities.
* Evaluating **smoothness of randomly placed Dirac impulses** across the entire network (smoothness implies well communicating network)
* Invert the network first so that nodes represent terrorists and edges relations. Compute **probability transition matrix** to reveal **which terrorists have the highest probabilities**; perhaps they are the most important people in the network in the sense that everyone else is most likely to end up “meeting” these terrorists. What happens when we remove (assassinate) these terrorists?
* **Betweenness centrality:** number of shortest paths that pass through a node. Invert the network first. Identify the **top 5 (or top 10) nodes with the highest betweenness centrality**. What happens when we remove these terrorists?
* Invert the network first. Can we **classify the label (relation) of the edges**? If we can, it means we can accurately identify the type of relations between any new terrorist who enters the network with the rest.
* Invert the network first. **Generate feature vectors for each node (terrorist)** which includes things like degree, betweenness centrality, relations, etc. and conduct **clustering** to identify communities in this way. Does removing a few of these clusters make our network random (and thus effectively destroy its effectiveness)?

\*First, we need to invert our network (first step, data pre-processing step)

**Visualization ideas:**

Python graph\_tool ARF layout to represent our inverted network:



**Important reminder:**

We want to have something to hit every box in the data science process below. Some examples include:

**Data processing**: network inversion

**Clean dataset**: not sure yet…

**Exploratory data analysis**: computing network properties (betweenness centrality, #communities, smoothness, etc.)

**Models/algorithms**: clustering task

**Data product**: “So what are the vulnerable points of the network?”, “How will these inform decisions or actions for fighting terrorism?”

