Graph Mining for Insider Threat Detection

Larry Holder
School of Electrical Engineering and Computer Science
Washington State University
Contact: holder@wsu.edu

Download materials at https://eecs.wsu.edu/~holder/cyser

Outline

- Insider threats
- Graph mining
  - Pattern learning
  - Anomaly detection
- Insider threat detection

Insider Threats

Compromise Trends 2015 to 2021

COMPROMISE TOTALS	2021	2020	2019	2018	2017	2016	2015
Number of Compromises	1,862	1,108	1,279	1,175	1,506	1,099	785
Number of Victims	293,927,708	310,116,907	883,558,186	2,227,849,622	1,825,413,935	2,541,070,438	318,276,407

ITRC 2021 Data Breach Report
idtheftcenter.org

- Scenarios
  - Violations of system security policy by an authorized user
  - Malicious exploitation, theft, or destruction of data
  - Compromise of networks, communications, or other IT resources
- Reality
  - 35% percent of the security breaches in 2021 came from insiders[1]
- How can we detect if an employee is
  - Planning to harm our organization, or
  - Leak sensitive information?

[1] ITRC 2021 Business Aftermath Report, idtheftcenter.org

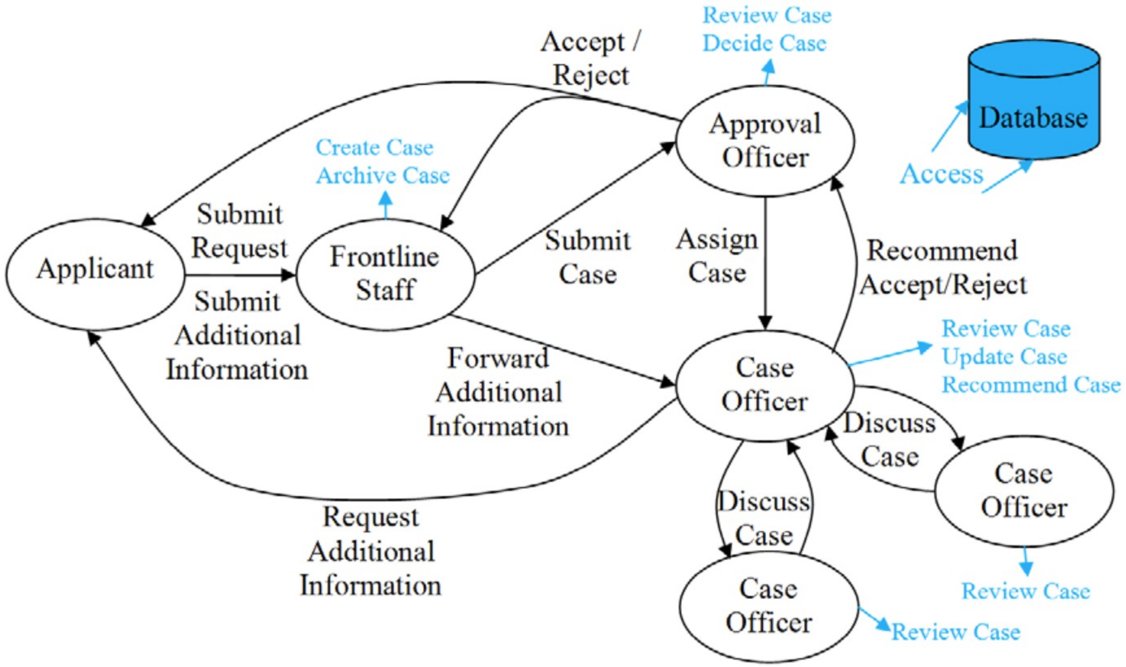
Insider Threat Detection: Existing Approaches

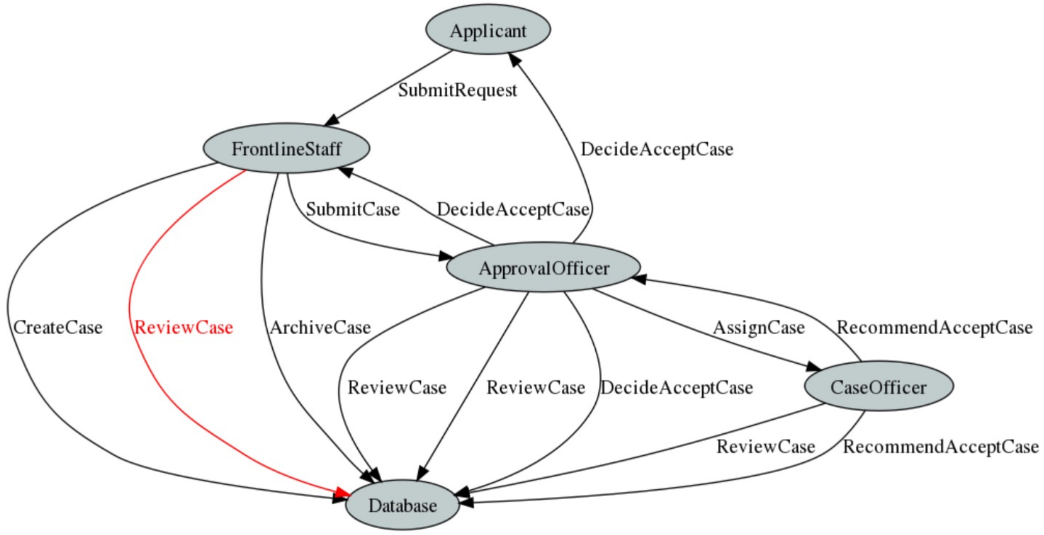
- Monitor/filter all external interactions
  - Not all threatening interactions show up externally
- Build behavioral profiles based on simple attributes and rules
  - Need training data and/or predefined rules
- Analyze people and movements
  - Statistical approaches not designed to handle relationships and structure

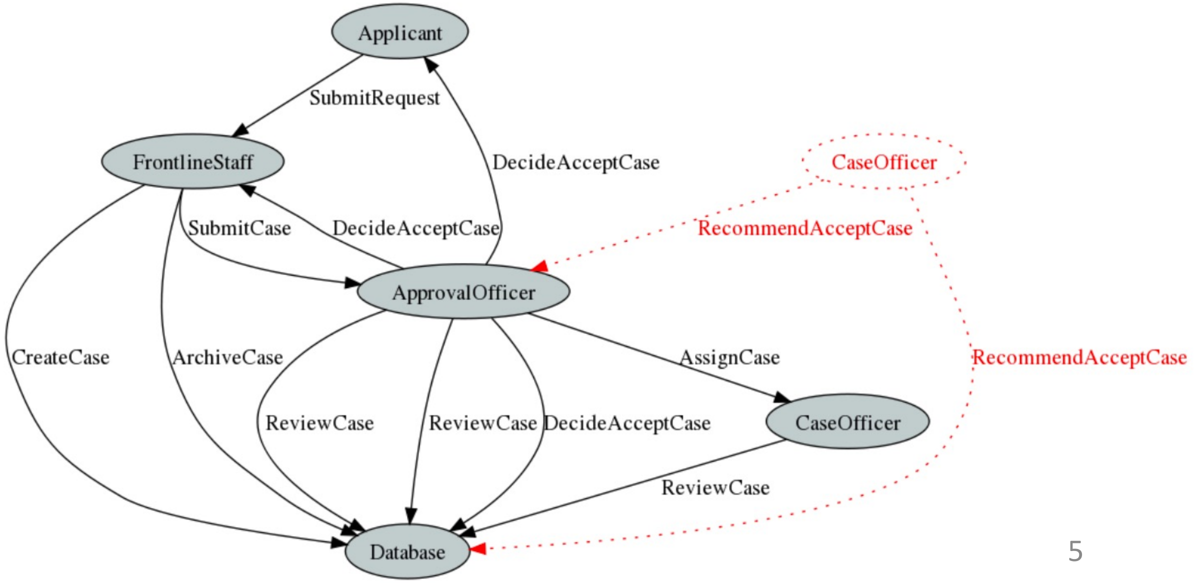
R. A. Alsowail and T. Al-Shehari, "Empirical Detection Techniques of Insider Threat Incidents," in *IEEE Access*, vol. 8, pp. 78385-78402, 2020, doi: 10.1109/ACCESS.2020.2989739.

Graph-Based Insider Threat Detection Example

Government ID Request Processing

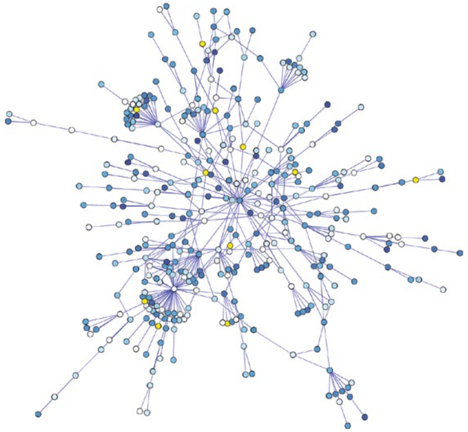




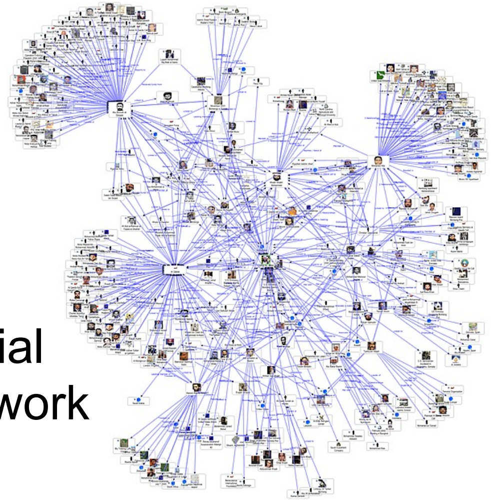


Graph Mining

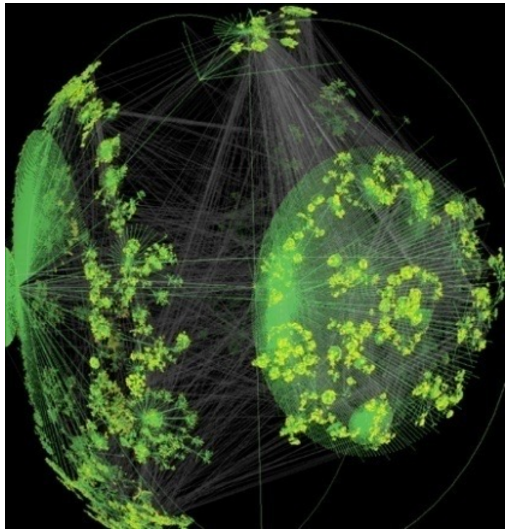
Graphs



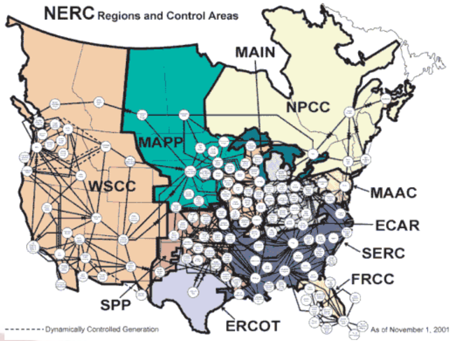
Protein-protein Interaction



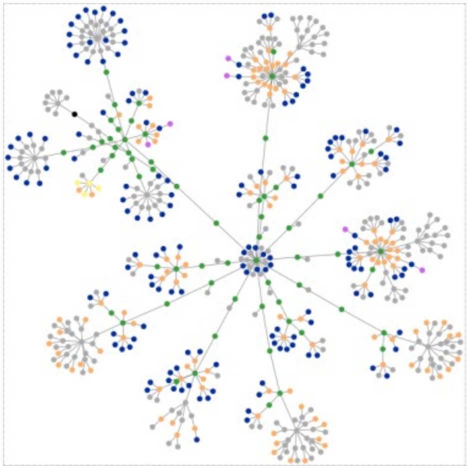
Social Network



Internet



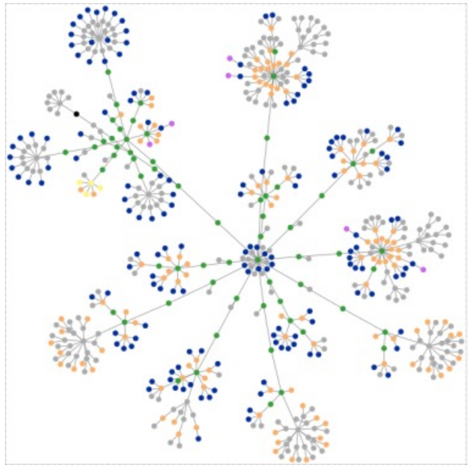
Power Grid



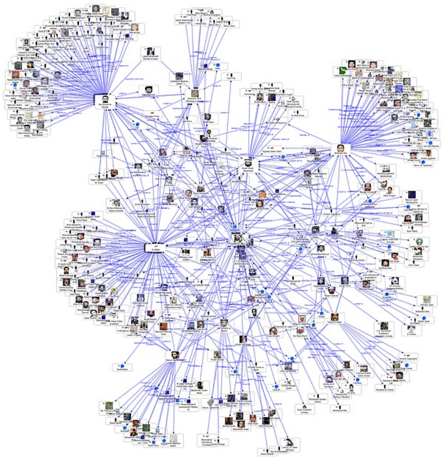
Web

Some Graph Statistics

- Web
  - 2B websites, 1T hyperlinks
  - 250K new websites per day



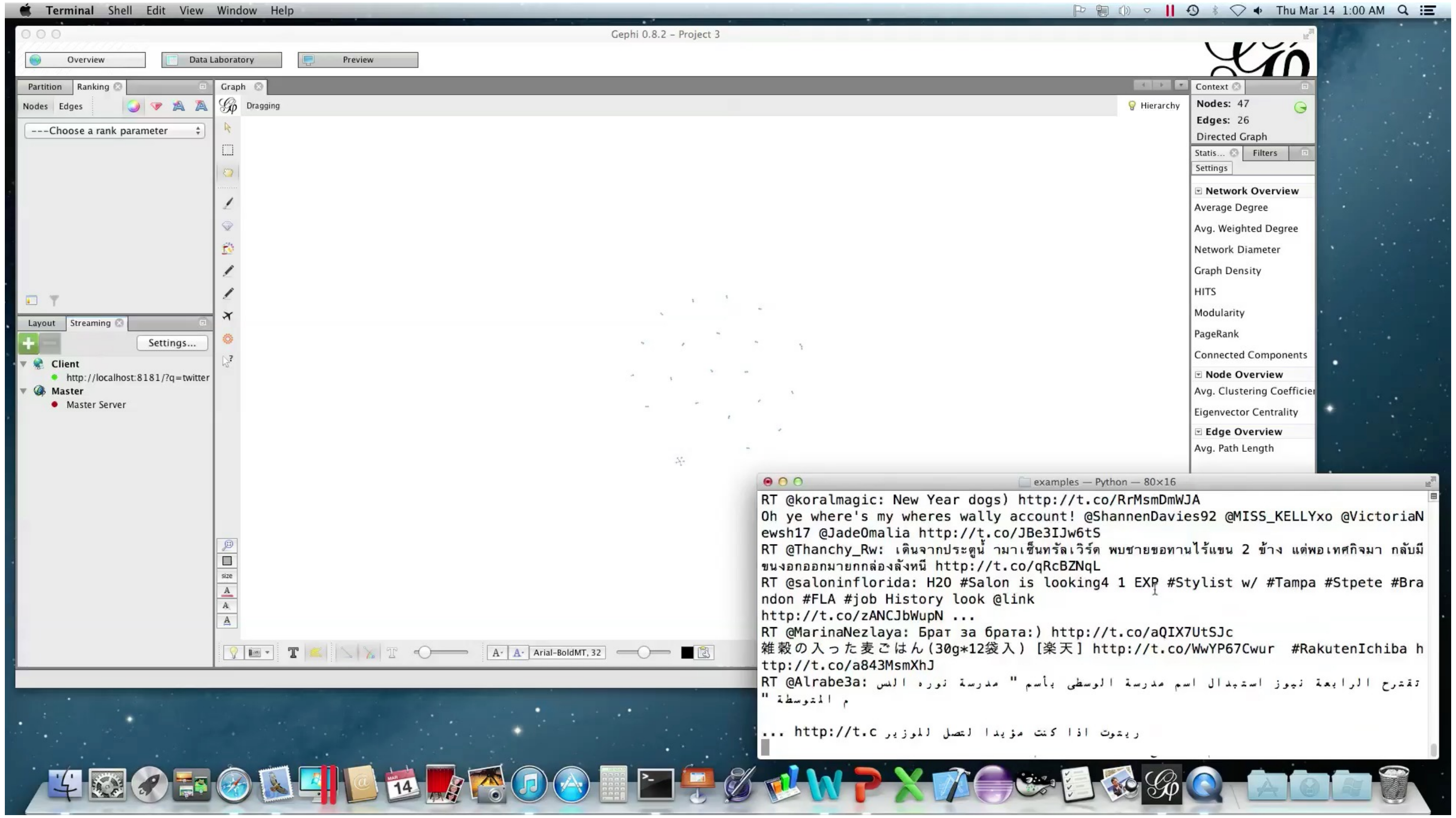
- Facebook
  - 2.9B active users links
  - 30B content pieces per month



Example: Twitter

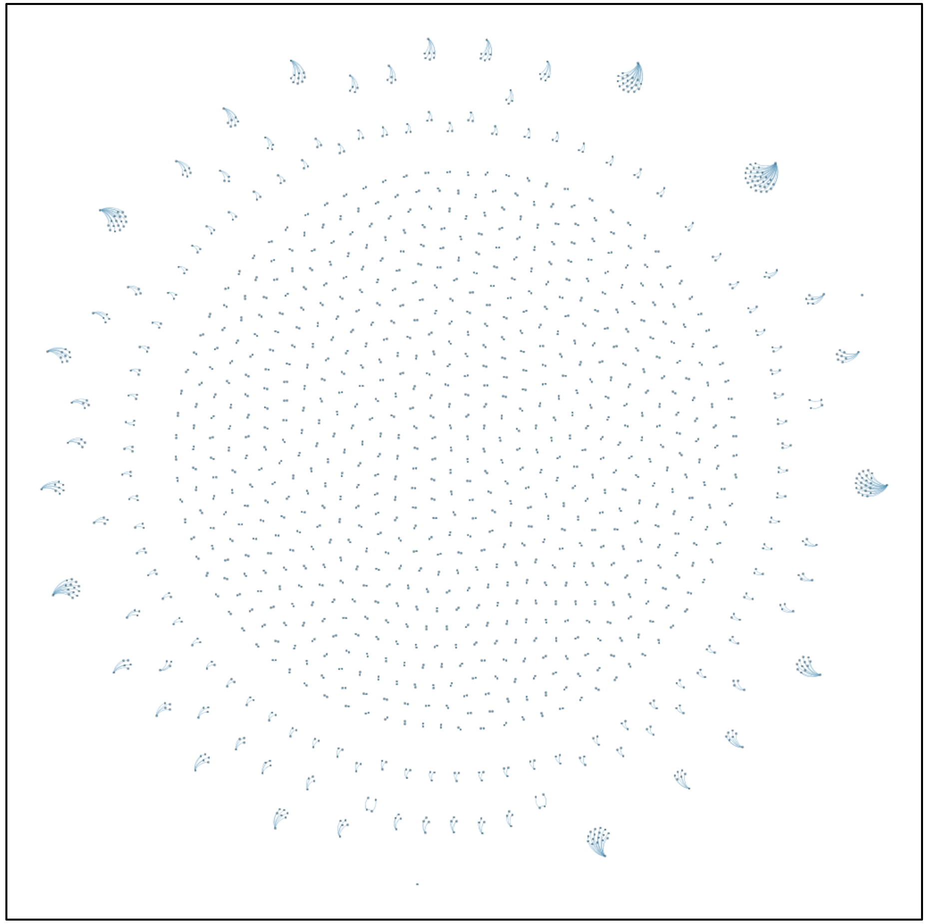
- 350M users
- 6K tweets per second

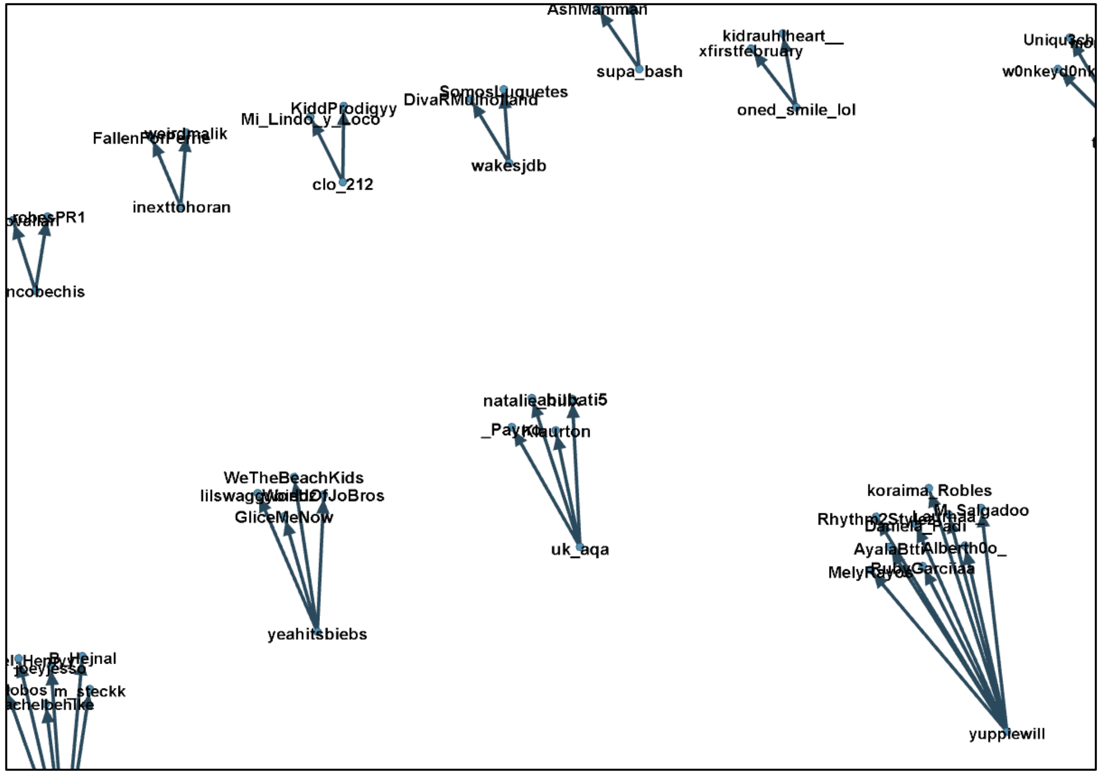
- Patterns?
- Anomalies?



Link A → B when B retweets tweet from A (A followed-by B)

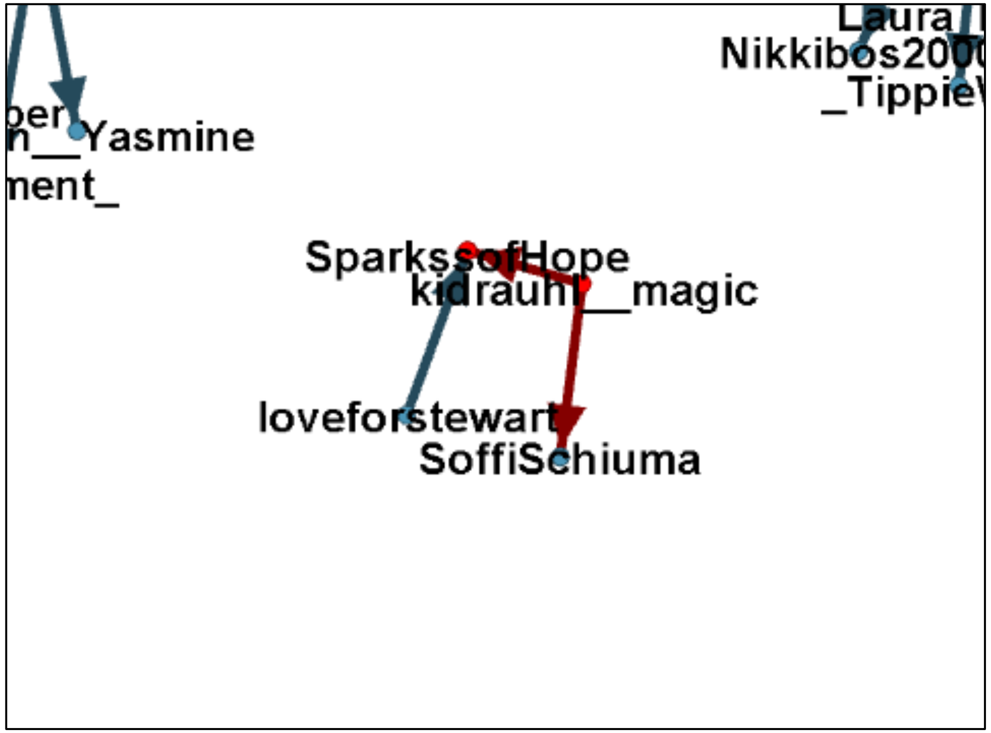
Patterns in Twitter Graph



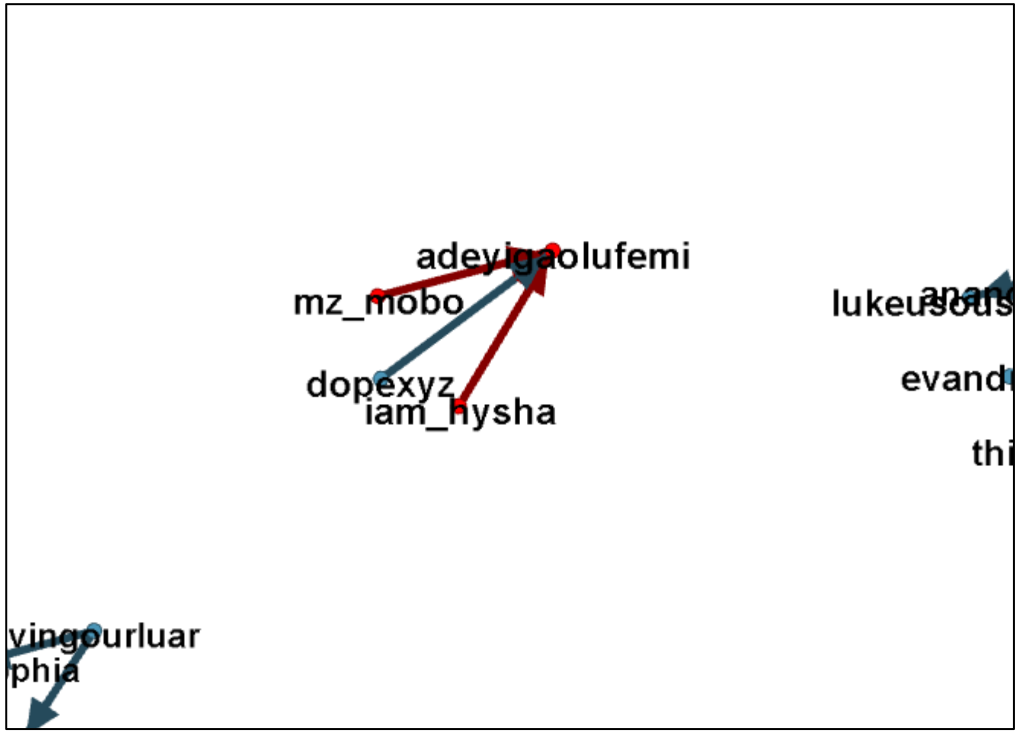


Many different users follow one user.

Anomalies in Twitter Graph



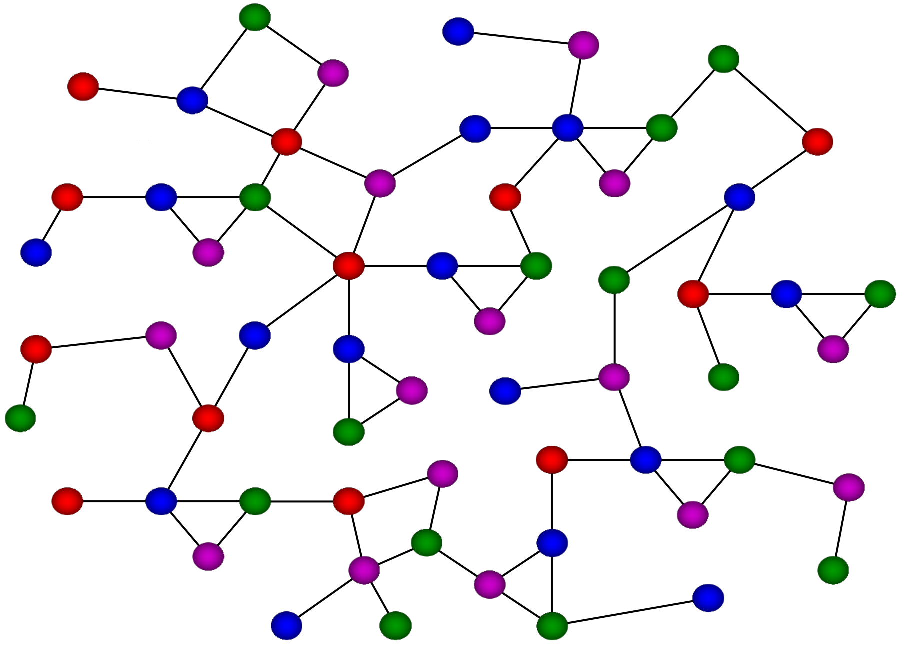
User with 2 followers, only 1 of which follows another user.



3 users followed by same user, but no one else.

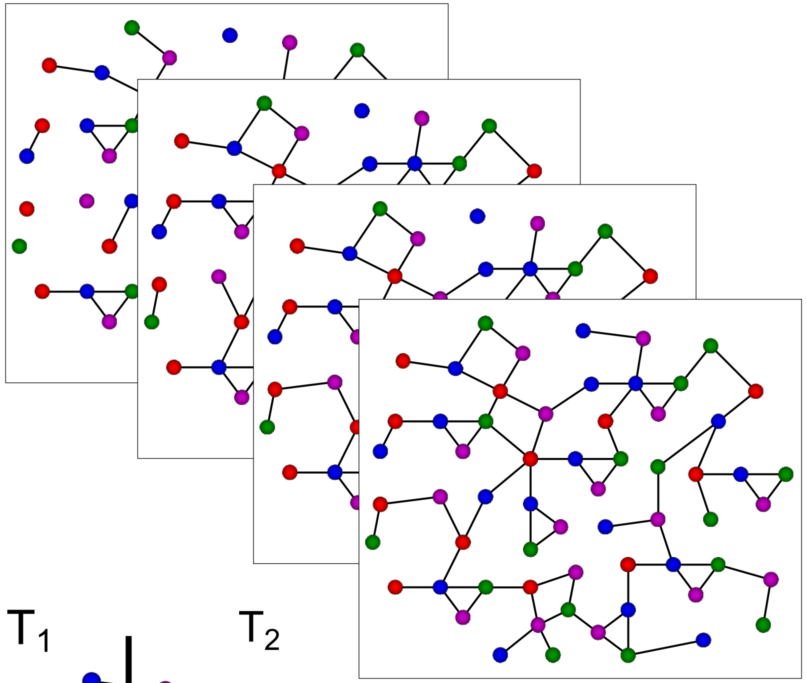
Definitions: Static Graph

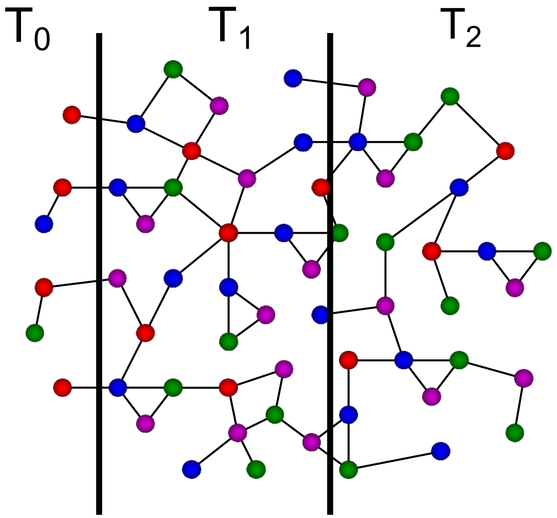
- Static graph
  - Set of nodes and links, each with attributes
- Pattern
  - Commonly-occurring subset of nodes, links, attributes
- Anomaly
  - Unexpected deviation to normative pattern
- Noise
  - Expected deviation to normative pattern
- Outlier
  - Unexpected subset of nodes, links, attributes



Definitions: Dynamic Graph

- Dynamic graph
  - Ordered sequence of static graph snapshots
  - Initial graph plus ordered sequence of changes (”stream”)
    - Add/change/delete nodes, edges, attributes
- Dynamic pattern
  - Static patterns plus temporal ordering
- Dynamic anomaly
  - Static anomalies plus temporal ordering



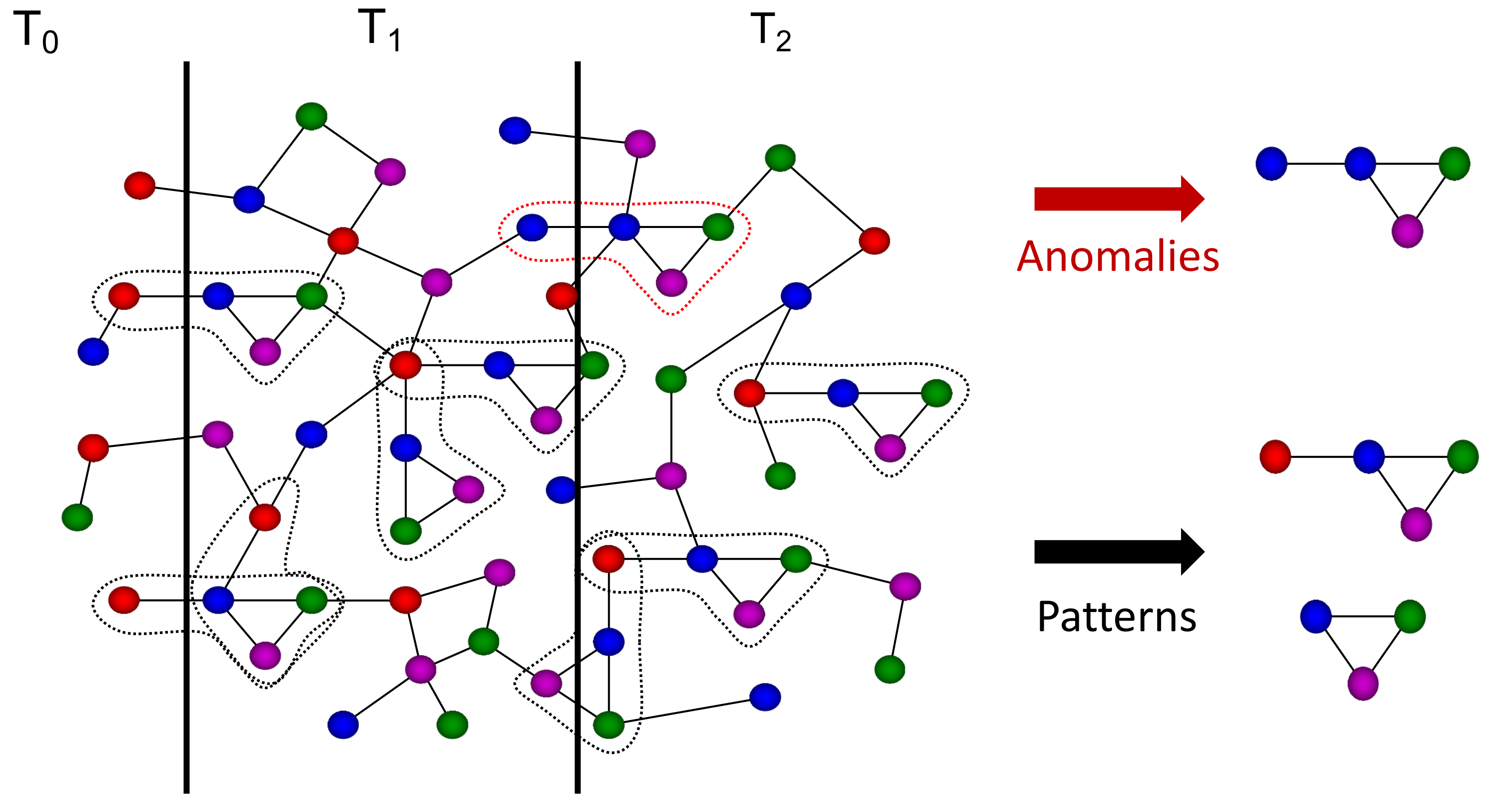


Graph "Mining"

- Degree
- Diameter
- Centrality
- Shortest path
- Cycles/tours
- Min spanning tree
- Traversals/search
- Connectivity
- Cliques

- Clustering
- Partitioning
- Subgraph matching
- Frequent subgraphs
- Motifs
- <u>Pattern learning</u>
- <u>Anomaly detection</u>
- Link Prediction
- Dynamics

Methods: Patterns and Anomalies



Main heuristic: Compression (~ graph zip/unzip)

Pattern Learning

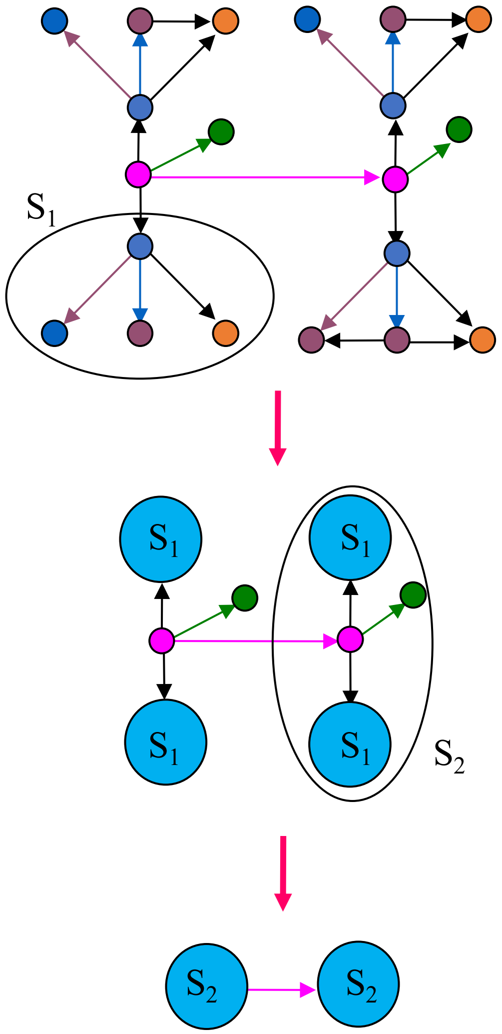
Methods: Pattern Learning

- Graph compression and the minimum description length (MDL) principle
- Given graph G, find pattern S maximizing compression of G

$$\min_{S}(DL(S)+DL(G\,|\,S))$$

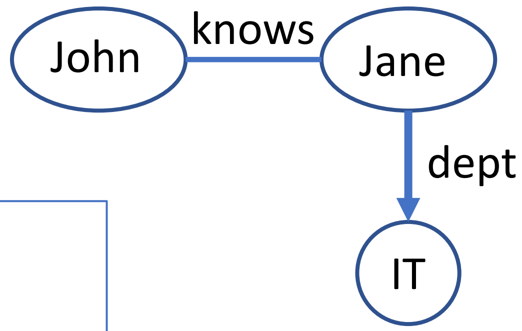
where description length DL(G) is the minimum number of bits needed to represent G

SUBDUE: http://ailab.wsu.edu/subdue
Python: https://github.com/holderlb/Subdue
C version: https://github.com/holderlb/CSubdue



Exercise 1: Use Subdue to find patterns

- Input graph format
  - Graph node or vertex: ‘v <n> label’
    - Where <n> is vertex number
  - Edge: ‘e <n1> <n2> label’
    - Directed: ’d <n1> <n2> label’
    - Undirected: ‘u <n1> <n2> label’
    - ‘e’ assumed directed by default
  - Labels quoted if contain whitespace or special characters
- Example




```
v 1 John
v 2 Jane
v 3 IT
u 1 2 knows
d 2 3 dept
```

Exercise 1 (continued)

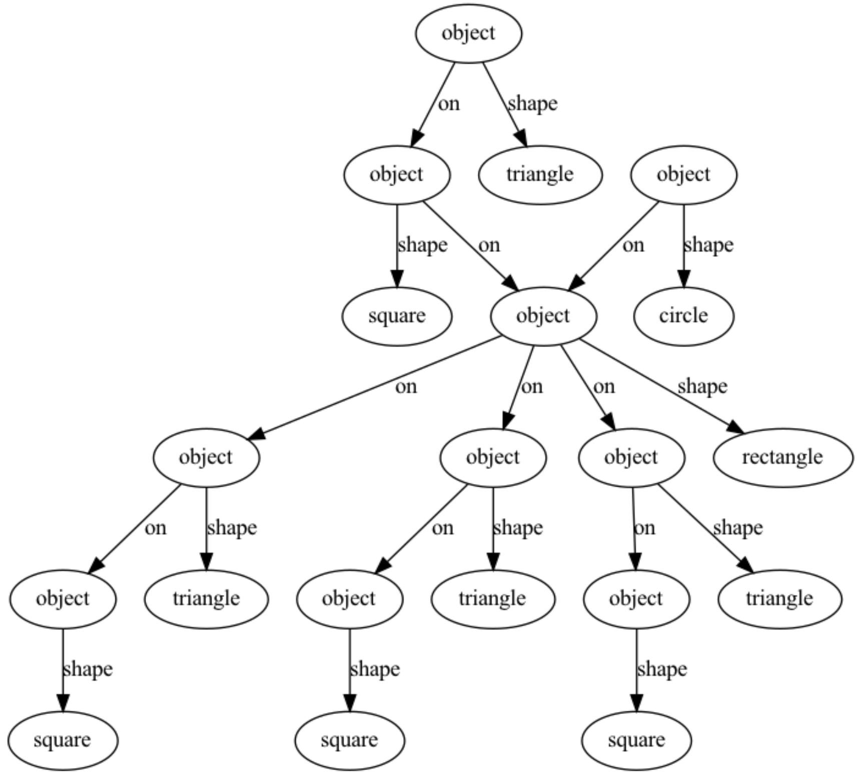
- Download CSubdue.zip
- unzip CSubdue.zip
- cd CSubdue/graphs
- ls
- more sample.g (type ‘q’ to quit)
- cd ../src
- make
- make install
- cd ..
- bin/subdue graphs/sample.g

```
    /\
   / 1\           __
  +----+         /  \
  |    |        | 9  |
  | 5  |         \  /
+-+----+----------------+
|                       |
|          10           |
+-----------------------+
    /\     /\     /\
   / 2\   / 3\   / 4\
  +----+ +----+ +----+
  |    | |    | |    |
  | 6  | | 7  | | 8  |
  +----+ +----+ +----+
```

```
Best 3 substructures:

(1) Substructure: value = 1.86819, pos instances = 4, neg instances = 0
  Graph(4v,3e):
    v 1 object
    v 2 object
    v 3 triangle
    v 4 square
    d 1 3 shape
    d 2 4 shape
    d 1 2 on

(2) Substructure: value = 1.37785, pos instances = 4, neg instances = 0
  Graph(3v,2e):
    v 1 object
    v 2 object
    v 3 square
    d 2 3 shape
    d 1 2 on

(3) Substructure: value = 1.37219, pos instances = 4, neg instances = 0
  Graph(3v,2e):
    v 1 object
    v 2 object
    v 3 triangle
    d 1 3 shape
    d 1 2 on
```

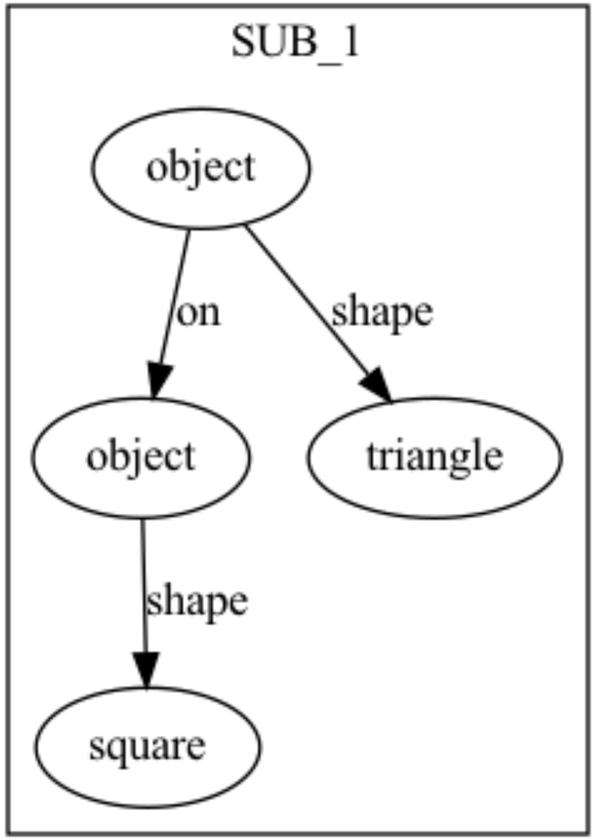
Exercise 1 (cont.): Visualize graph

- Download and install Graphviz (dot, sfdp)
  - AWS: sudo yum install graphviz
- bin/graph2dot graphs/sample.g sample.dot
- dot -Tpng sample.dot > sample.png
- Open sample.png in image viewer or navigate to sample.png file and double-click



Exercise 1 (cont.): Visualize Pattern

- bin/subdue -out subs.g graphs/sample.g
- bin/subs2dot subs.g subs.dot
- dot -Tpng subs.dot > subs.png
- Open subs.png in image viewer or navigate to subs.png file and double-click

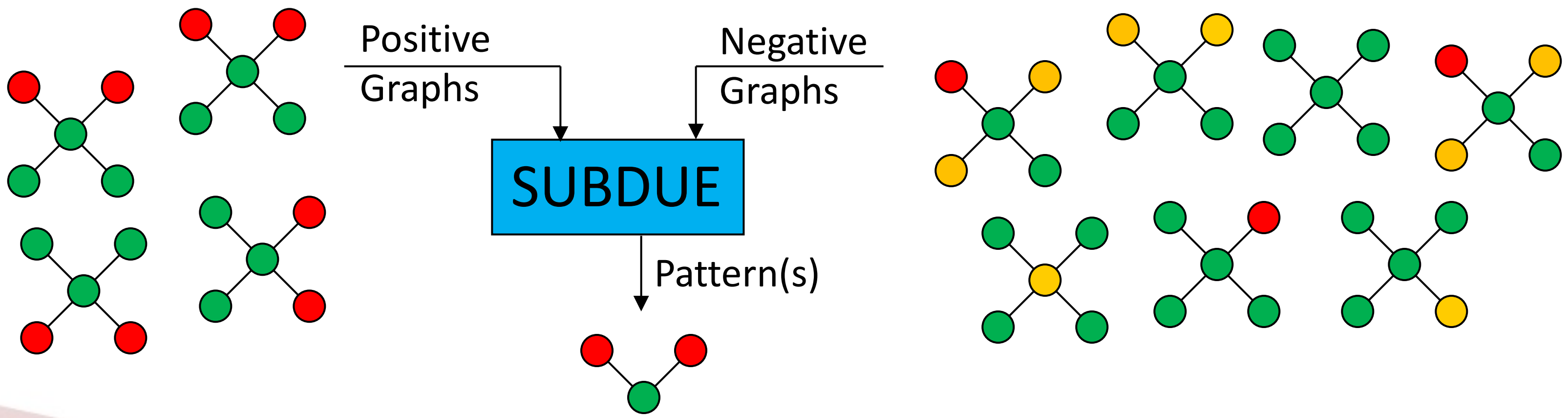


Supervised Learning

- Given positive graph G+ and negative graph G-
- Find pattern S that compresses G+ but not G-

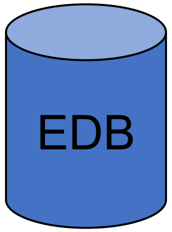
$$\min_{S} \frac{size(G^{+} \mid S)}{size(G^{-} \mid S)}$$

- Separate graphs using XP and XN in Subdue input file
- bin/subdue graphs/groups.g



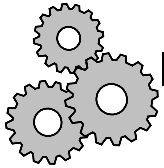
DARPA / AFRL EAGLE Project

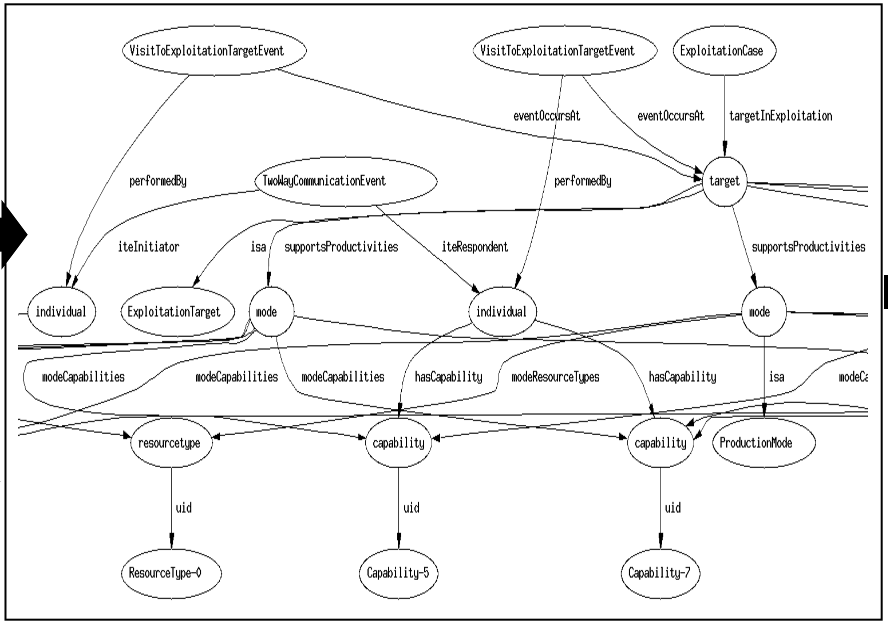
Evidence Assessment, Grouping, Linking and Evaluation (EAGLE) Program



Evidence DB (EDB) contains simulated data on threat and non-threat activity
- Persons, targets, capabilities, resources, transfers, and communications

Convert EDB to SUBDUE graph format

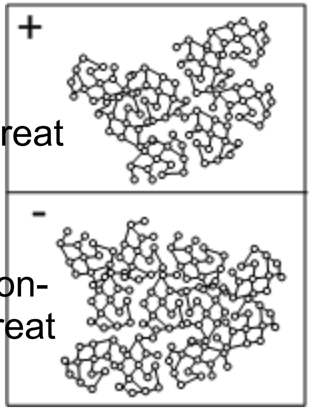




Positive & negative examples

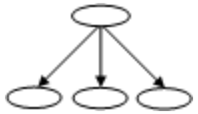
Threat

Non-threat



SUBDUE

Patterns

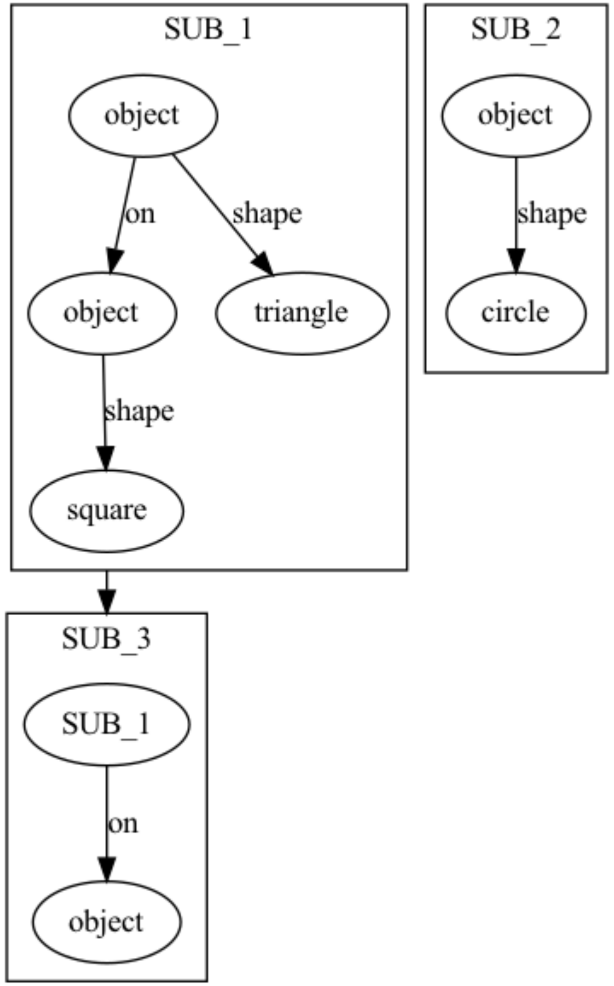


Evaluate (to Link Discovery)

Results	Examples	Entities	Relations	Accuracy	Time
Events	308	533,196	630,733	80%	86 min
Groups	84	457,209	597,163	85%	813 min

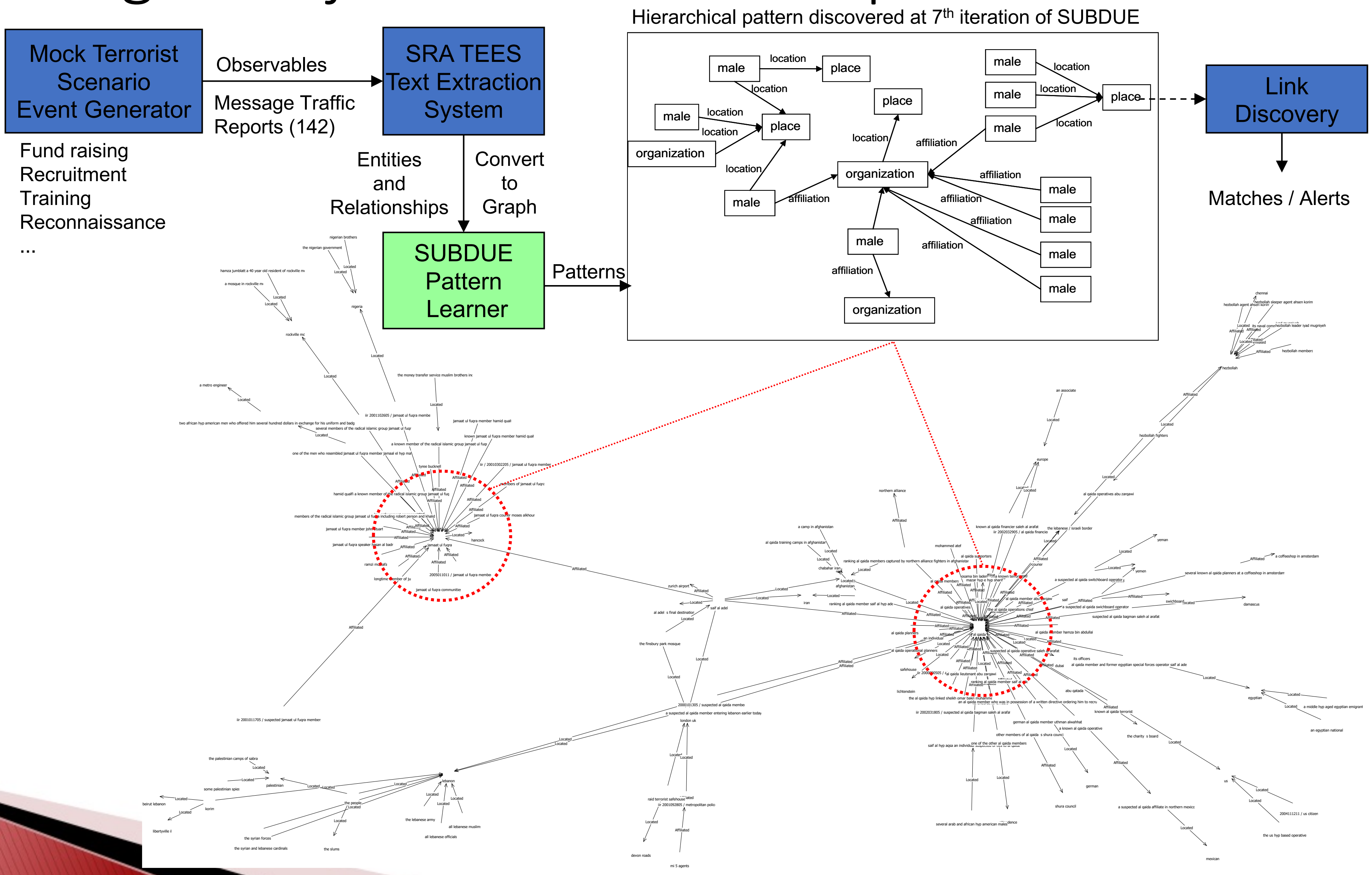
Hierarchical Pattern Learning

- Use iterative process on input graph G
  - Repeat
    - Find best pattern S in graph G
    - Add S to hierarchy
    - G = G compressed with S
  - Until no more compression




```
bin/subdue -iterations 3 –out subs.g graphs/sample.g
bin/subs2dot subs.g subs.dot
dot –Tpng subs.dot > subs.png
```

DHS Insight Project: Terrorist Group Data

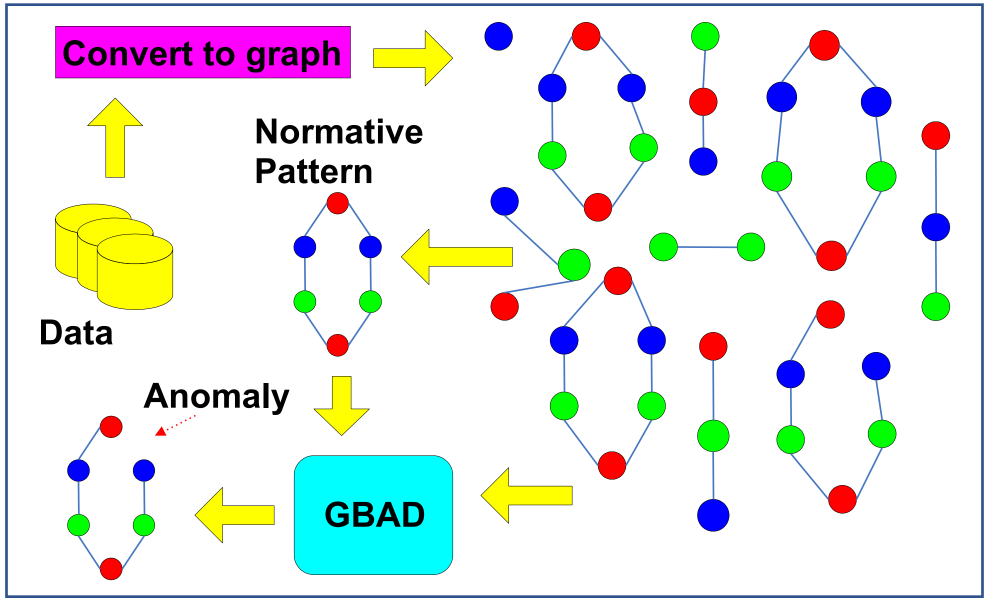


Anomaly Detection

Anomaly Detection

- Given normative pattern S in graph G
- Find pattern S' such that
  - $d(S, S') < T_1$
  - $P(S' \mid G, S) < T_2$

  - d = graph edit distance
  - T_1 & T_2 user input



Graph-Based Anomaly Detection (GBAD)

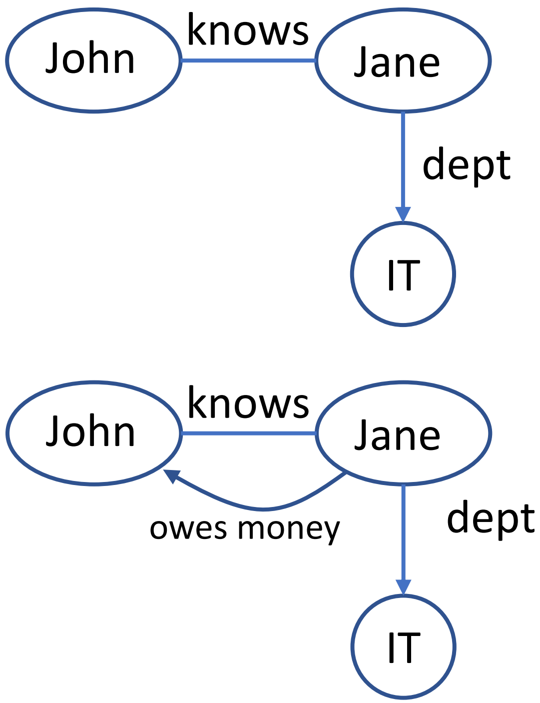
GBAD: http://www.gbad.info

Exercise 2: Use GBAD to find anomalies

- Input graph format
  - Each separate graph must start with 'XP # <n>', where <n> is the example number
  - Graph node or vertex: 'v <n> "label"'
    - <n> is the vertex number
  - Edge: 'e <n1> <n2> "label"'
    - Directed: "d # # label"
    - Undirected: "u # # label"
    - "e" assumed directed by default
    - <n1> and <n2> are the vertex numbers connected by the edge
  - All labels must be in quotes

- Example

```
XP # 1
v 1 "John"
v 2 "Jane"
v 3 "IT"
u 1 2 "knows"
d 2 3 "dept"
XP # 2
v 1 "John"
v 2 "Jane"
v 3 "IT"
u 1 2 "knows"
d 2 3 "dept"
d 2 1 "owes money"
```



Exercise 2 (continued)

- Download GBAD.zip
- unzip GBAD.zip
- cd gbad-tool-kit_4.0/graphs
- ls
- more prob_example.g (type 'q' to quit)
- cd ../gbad-mdl_4.0/src
- make
- make install
- cd ..
- bin/gbad -all 0.5 ../graphs/prob_example.g > output.txt

```
XP # 5
v 1 "1"
v 2 "2"
v 3 "3"
v 4 "4"
v 5 "5"
u 1 2 "e"
u 1 3 "e"
u 1 4 "e"
u 3 5 "e"
XP # 6
v 1 "1"
v 2 "2"
v 3 "3"
v 4 "4"
v 5 "5"
v 6 "V"
u 1 2 "e"
u 1 3 "e"
u 1 4 "e"
u 3 5 "e"
u 4 6 "e"
```

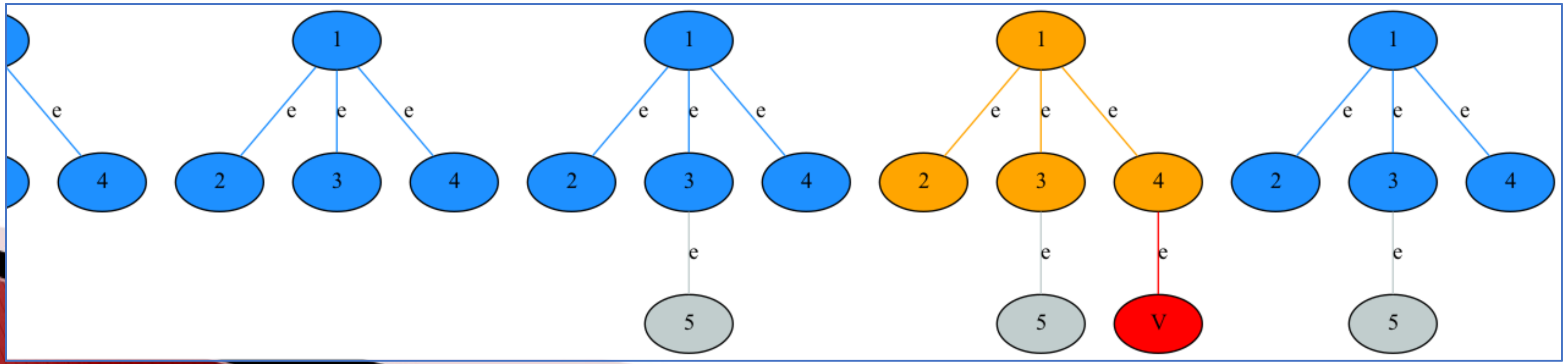
Exercise 2 (continued)

- more output.txt

```
Normative Pattern (1):
Substructure: value = 2.80952, instances = 7
  Graph(4v,3e):
    v 1 "1"
    v 2 "2"
    v 3 "3"
    v 4 "4"
    u 1 2 "e"
    u 1 3 "e"
    u 1 4 "e"

Discovering anomalous substructure instances...
5 initial substructures
9 substructures being considered
23 substructures being considered
37 substructures being considered
47 substructures being considered
50 substructures being considered

Anomalous Instance(s):

 from example 6:
    v 22 "1"
    v 23 "2"
    v 24 "3"
    v 25 "4"
    v 27 "V" <-- anomaly (original vertex: 6 , in original example 6)
    u 22 23 "e"
    u 22 24 "e"
    u 22 25 "e"
    u 25 27 "e" <-- anomaly (original edge vertices: 4 -- 6, in original example 6)
    (anomalous value = 2.000000 )
```

Exercise 2 (cont.): Visualize Anomaly

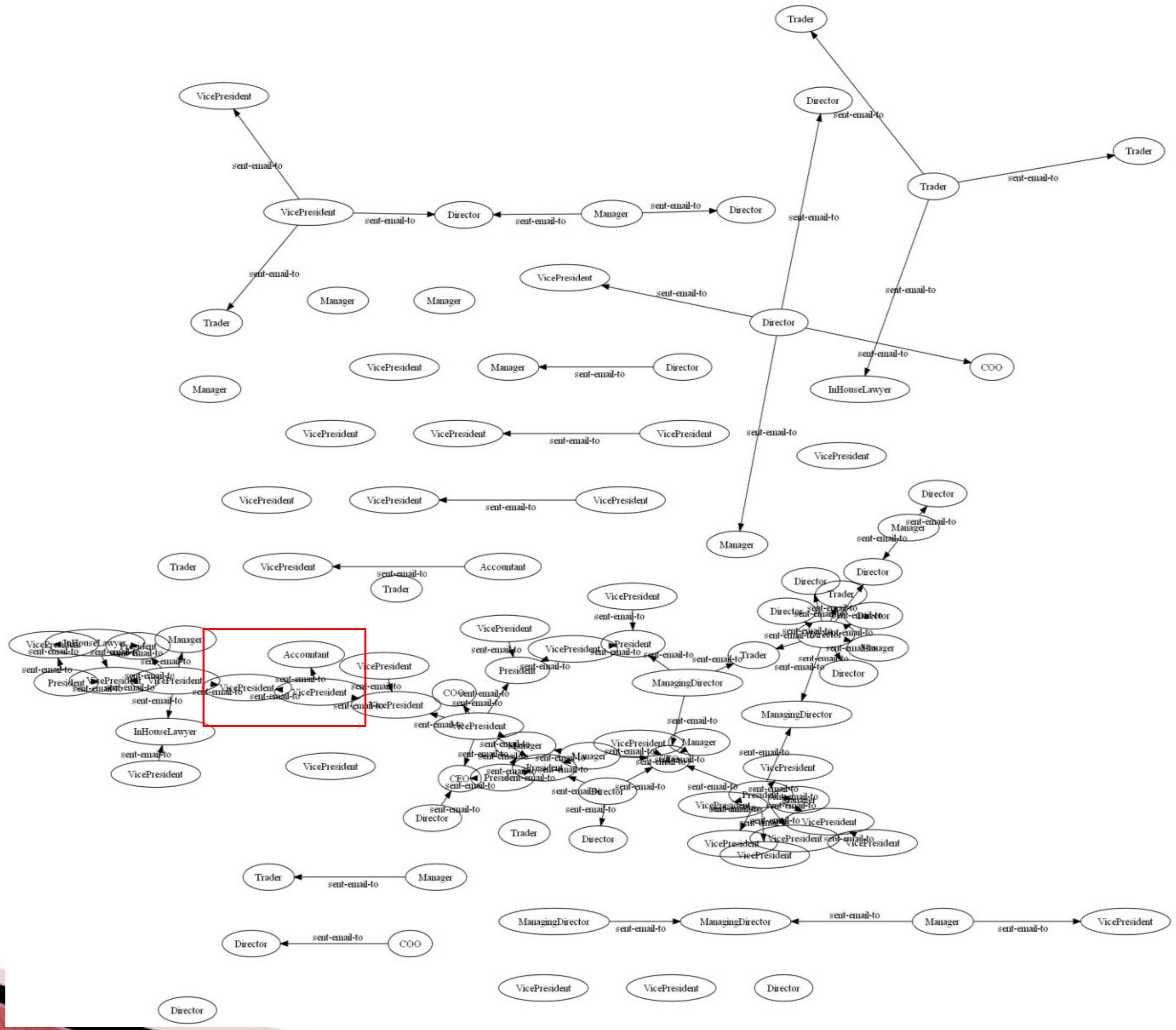
- Download and install Graphviz (dot)
  - AWS: sudo yum install graphviz (Ex. 1)
- bin/gbad -all 0.5 -dot output.dot ../graphs/prob_example.g
- dot -Tpng output.dot > output.png
- Open output.png in image viewer or navigate to output.png file and double-click
  - Normative pattern in blue
  - Anomalies in red and orange
  - Non-anomalous differences from normative pattern in gray



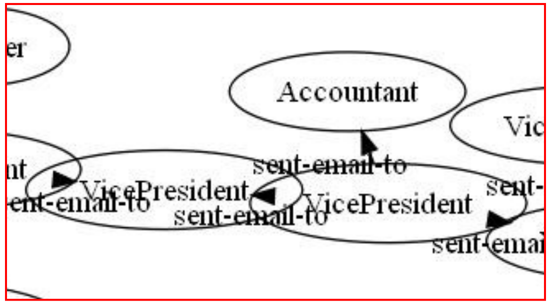
GBAD on Enron Email Data

- https://www.cs.cmu.edu/~enron/
- Data contains emails made among Enron employees in 2001
- Enron collapsed in October 2001
- Graph representation
  - Node for each employee labeled with position
  - Edge for each email between employees
- One graph per day in October 2001

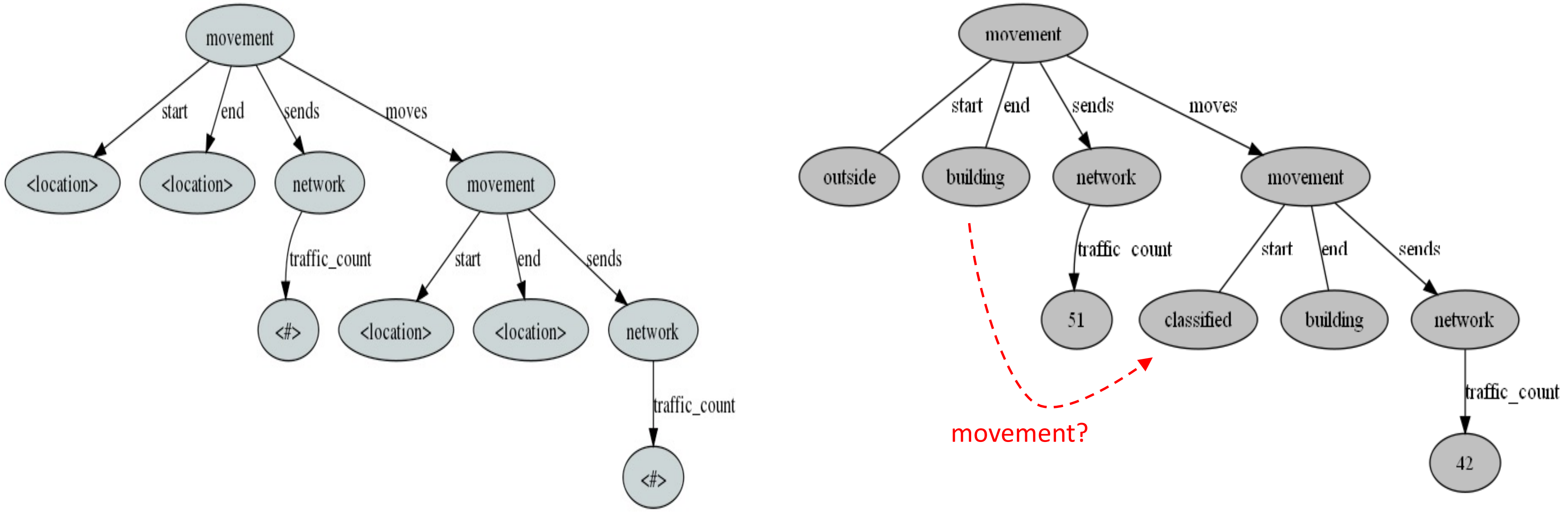
GBAD on Enron Email Data



- Graph for October 24,2001
- Normative pattern: VP emails VP
- Anomaly: VP emails Accountant
- Accountant is Wanda Curry
  - Later identified as a whistleblower



GBAD on VAST 2009 Challenge: Embassy Leak

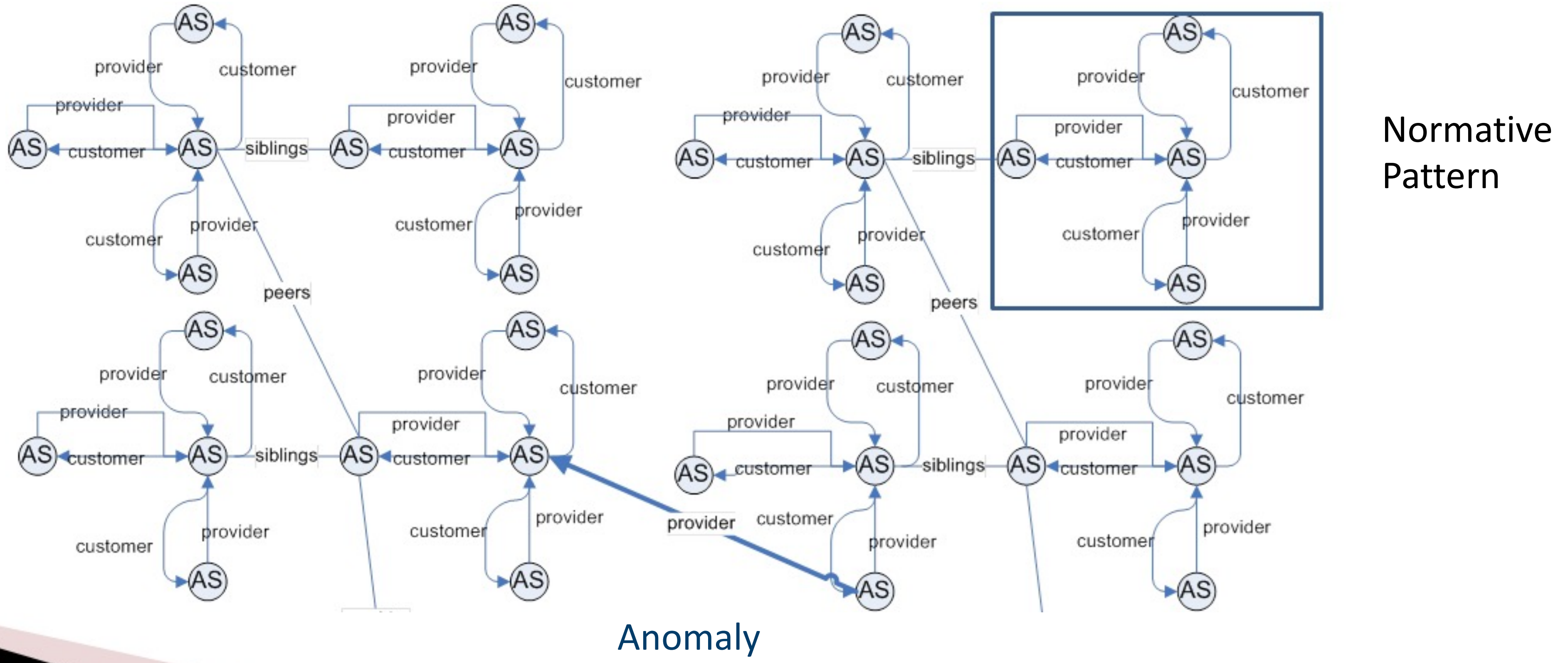


Left: Graph topology of movement and activity. Right: Anomalous structure in the graph indicating unrecorded entry into classified area.

http://visualdata.wustl.edu/varepository/VAST Challenge 2009/challenges/MC1 - Badge and Network Traffic/

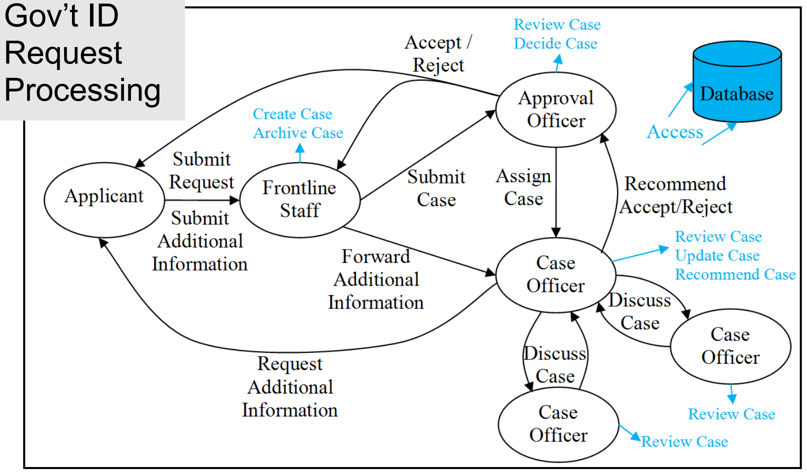
GBAD on CAIDA Network Topology

- Cooperative Association for Internet Data Analysis (www.caida.org)



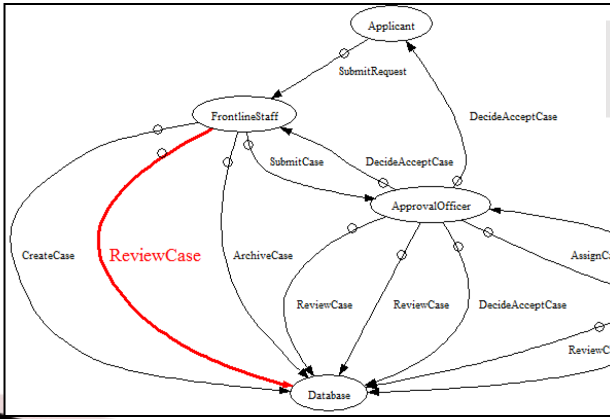
DHS CyberSecurity R&D Program: Insider Threat Detection using Graphs

Gov't ID Request Processing

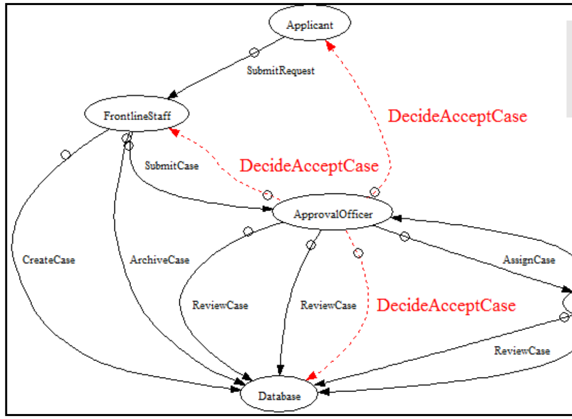


<u>Insider Threat Scenarios (CERT Insider Threat Documents)</u>

1. Frontline staff reviews case (invasion of privacy).
2. Frontline staff submits case directly to a case officer (bypassing the approval officer).
3. Frontline staff recommends or decides case.
4. Approval officer reverses accept/reject recommendation from assigned case officer.
5. Unassigned case officer updates or recommends case.
6. Applicant communicates with approval officer or case officer.
7. Unassigned case officer communicates with applicant.
8. Database access from an external source or after hours.



GBAD on Scenario 1

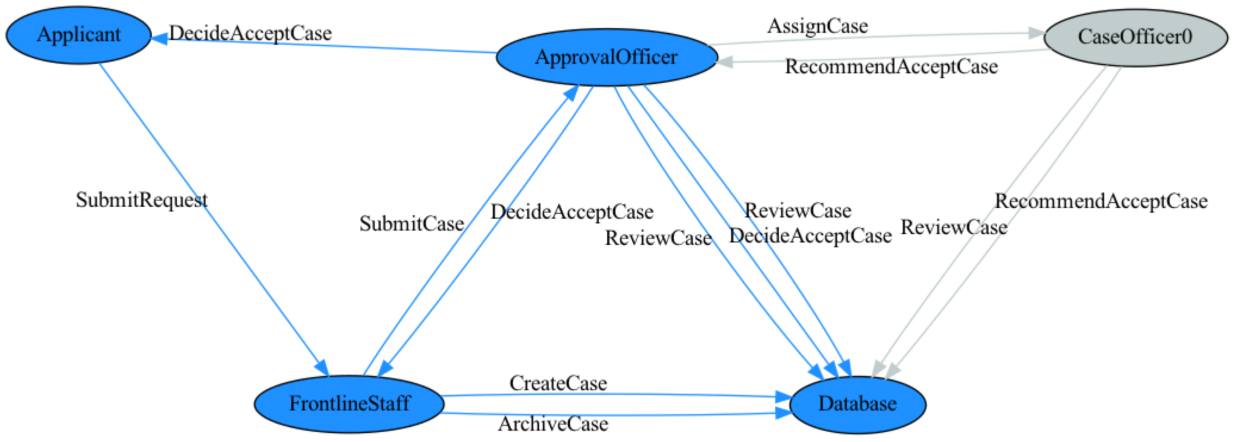


GBAD on Scenario 4

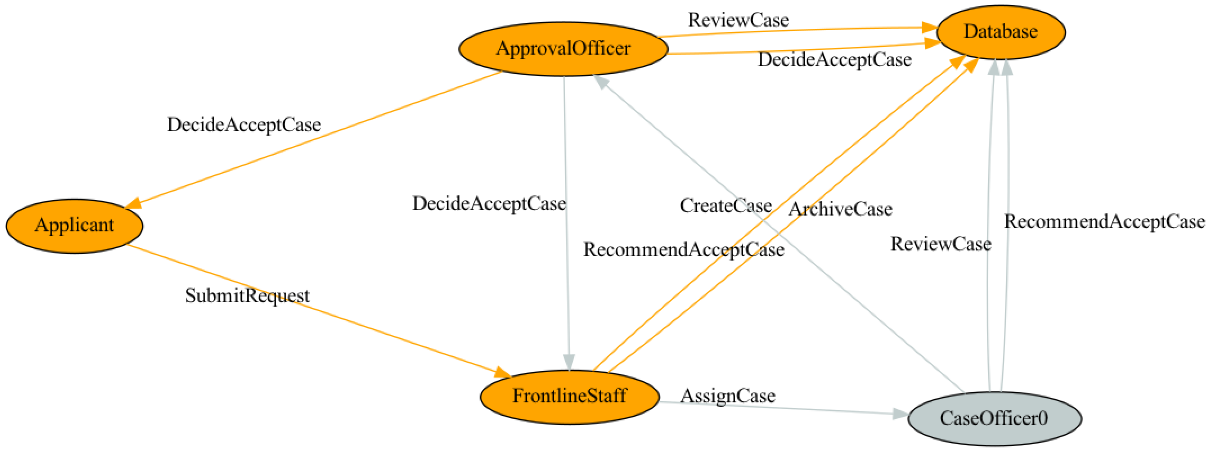
- 1000 cases
- Multiple normative patterns
- 1-3 anomalies
- No false positives

Exercise 3: GBAD on Gov't ID Processing

- Scenario #2: Frontline staff submits case directly to a case officer (bypassing the approval officer).



Normative Pattern

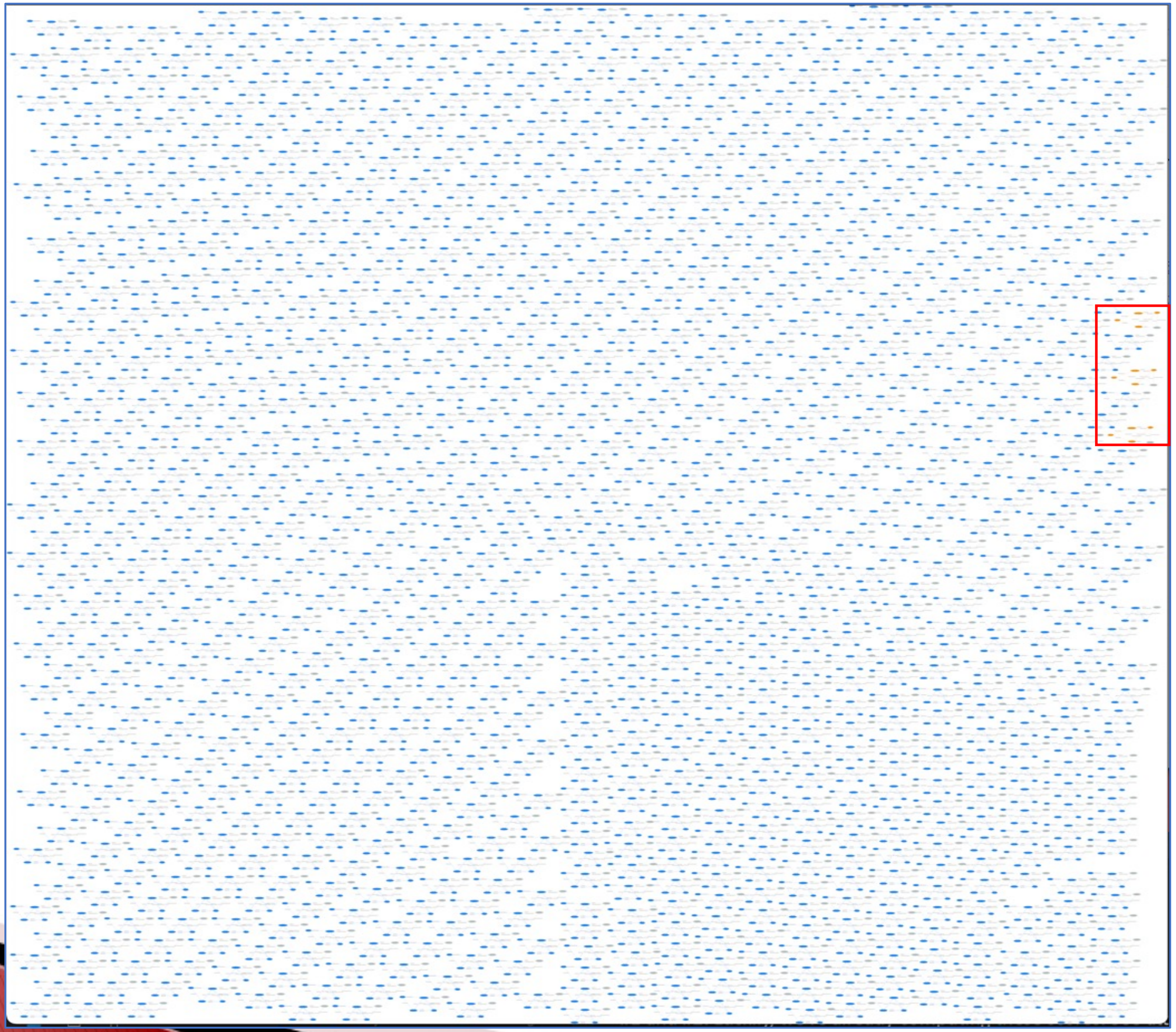


Anomaly

Exercise 3 (cont.)

- Download idprocess2.g (right-click and ‘Save Link As…’)
- cd gbad-tool-kit_4.0
- cp ~/Downloads/idprocess2.g graphs/.
- cd gbad-mdl_4.0
- bin/gbad -all 0.5 -dot idoutput.dot ../graphs/idprocess2.g (takes 9 min on AWS)
- sfdp -Tpng idoutput.dot > idoutput.png (takes 30 secs on AWS)
    - ‘sfdp’ used because faster and generates smaller files than ‘dot’

Exercise 3 (cont.)



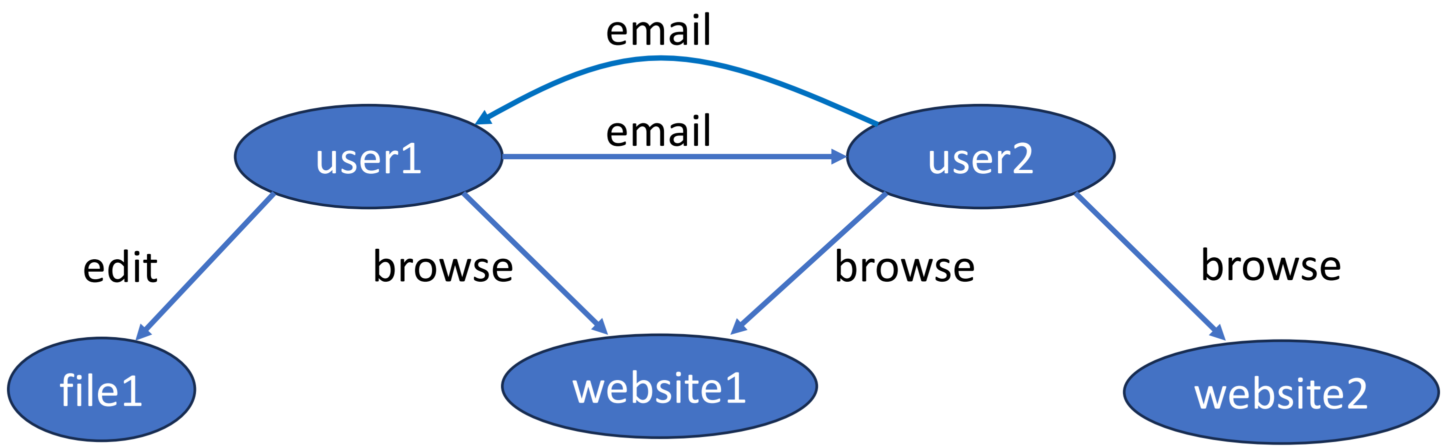
Recent Work: GHOSTS Simulator [Vincent Lombardi, Timothy Reidy]

- GHOSTS (General HOSTS) simulates an enterprise cyber environment using realistic models of user behavior
- Users can send email, browse the web, create/edit documents, run terminal commands
- All events logged
- GHOSTS user takes over a machine (VM in our case)
- Developed by CMU Software Engineering Institute
- https://github.com/cmu-sei/GHOSTS (ongoing project)
- ANIMATOR and SPECTRE: AI tools for more realistic users

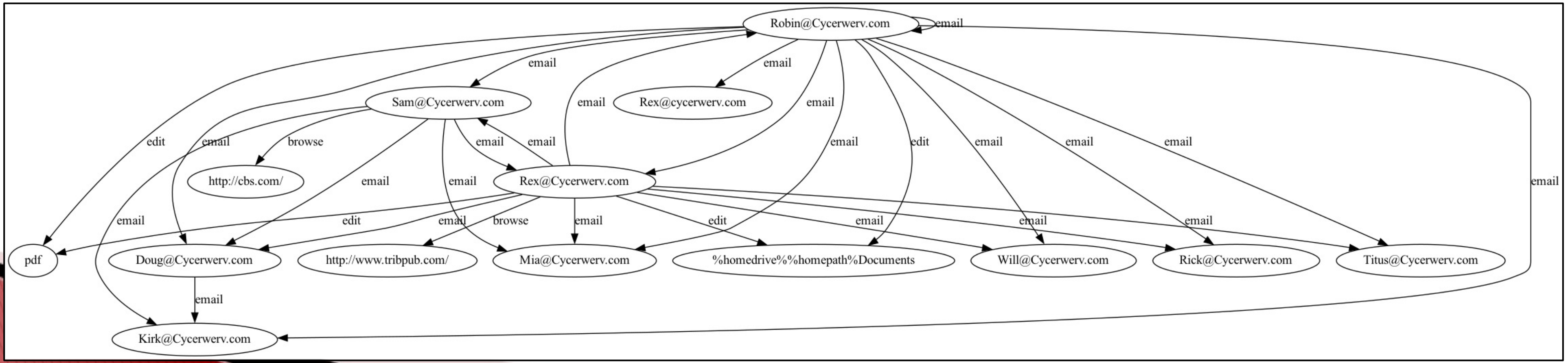
GHOSTS: Sample Data

machineId	createdUtc	handler	command	commandArg
2e1f64e2-ae9d-49c0-bcf7-40b7741bb053	2023-05-01T00:27:31	Outlook	email	Sending email from: Rex@Cycerwerv.com to: Doug@Cycerwerv.com cc: bcc:
2e1f64e2-ae9d-49c0-bcf7-40b7741bb053	2023-05-01T00:27:18	Outlook	email	Sending email from: Rex@Cycerwerv.com to: Doug@Cycerwerv.com cc: bcc:
2e1f64e2-ae9d-49c0-bcf7-40b7741bb053	2023-05-01T00:27:16	Excel	create	pdf
2e1f64e2-ae9d-49c0-bcf7-40b7741bb053	2023-05-01T00:27:16	Excel	create	%homedrive%%homepath%\Documents
2e1f64e2-ae9d-49c0-bcf7-40b7741bb053	2023-05-01T00:27:13	BrowserFirefox	browse	{"Uri":"https://www.nbcnews.com/politics/white-house/whcd-biden-speech-comedian-watch-stream-rcna81980","Category":null,"Method":"GET","Headers":null,"FormValues":null,"Body":null}
2e1f64e2-ae9d-49c0-bcf7-40b7741bb053	2023-05-01T00:27:05	Outlook	email	Sending email from: Rex@Cycerwerv.com to: Sam@Cycerwerv.com cc: bcc:
2e1f64e2-ae9d-49c0-bcf7-40b7741bb053	2023-05-01T00:27:04	BrowserFirefox	browse	{"Uri":"http://nbcnews.com/","Category":null,"Method":"GET","Headers":null,"FormValues":null,"Body":null}
2e1f64e2-ae9d-49c0-bcf7-40b7741bb053	2023-05-01T00:27:02	BrowserFirefox	browse	{"Uri":"http://www.tvguide.com/","Category":null,"Method":"GET","Headers":null,"FormValues":null,"Body":null}
2e1f64e2-ae9d-49c0-bcf7-40b7741bb053	2023-05-01T00:26:52	Outlook	email	Sending email from: Rex@Cycerwerv.com to: Kirk@Cycerwerv.com,Will@Cycerwerv.com cc: bcc:
2e1f64e2-ae9d-49c0-bcf7-40b7741bb053	2023-05-01T00:26:38	Outlook	email	Sending email from: Rex@Cycerwerv.com to: Kirk@Cycerwerv.com,Doug@Cycerwerv.com cc: bcc:

GHOSTS: Graph Representation



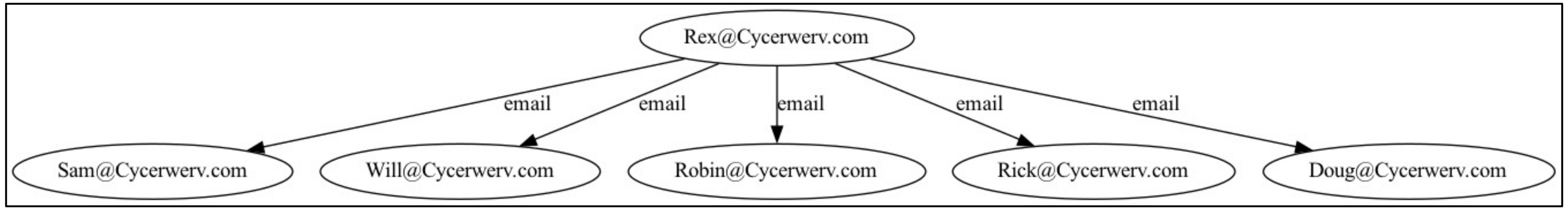
One hour's worth of data:



GHOSTS: Next Steps

- Define insider profiles
- Generate normal & insider data
- Run GBAD to look for anomalous behavior

Best normative pattern:



Conclusions

- Insider breaches continue to increase in number and cost
- Insider threat detection needs to take into account the relationships in the data
- Graph mining finds patterns and anomalies in the relationships
- Challenges
  - Representing data as a graph
  - Handling high volume and high velocity data
  - Fusing data from multiple sources