Anomaly Detection in Bipartite Networks

Mohammed Eslami, Ph.D
George Zheng, Hamed Eramian, Georgiy Levchuk
Disclaimers

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA).

The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.
Outline

Goal: Formulate cyber logs as bipartite graphs and an analytical workflow that use graph features to highlight events of interest to a cyber analyst.

• Overview of Network Defense
• Cyber Data Represented as Bipartite Graphs
• Graph Analytical Components, Features, and Workflow for Cyber Security
• Scalability and Examples
• Conclusion/Next Steps
The Challenge of Network Defense

Rapid identification of network anomalies in billions of records across a heterogeneous logs.

Enterprise Netflow and log data:
12 billion events per day,
1 TB per day of communications
>60,000 employees,
>570,000 users

Moving beyond State of the Art:
Rule-based signatures \(\rightarrow\) Adaptive behavior detection
Stateless single IP analyses \(\rightarrow\) Context based decisions
Manual analysis \(\rightarrow\) Guided automation

Source: Deason, L. et. al. Scalable Temporal Analytics to Detect Automation and Coordination. Flocon 2017
The Challenge of Network Defense

Rapid identification of network anomalies in billions of records across a heterogeneous logs.

**Enterprise Netflow and log data:**
12 billion events per day,
1 TB per day of communications
>60,000 employees,
>570,000 users

**Moving beyond State of the Art:**
- Rule-based signatures
- Stateless single IP analyses
- Manual analysis
- Adaptive behavior detection
- Context-based decision making
- Guided automation

Source: Deason, L. et. al. Scalable Temporal Analytics to Detect Automation and Coordination. Flocon 2017
Types of Bipartite Graphs from Enterprise Networks

Bipartite graphs, graphs that have edges only between two distinct entity types, provide an opportunity to capture the relationships between entities within and across types but pose a unique set of challenges in their storage, scalable analysis, and interpretability.

**IP-IP Graphs**
- Netflow records – edges only between internal/external IPs

**Client-Server Graphs**
- DNS, HTTP, SMTP, etc. logs – edges only between client IP and server IP
- 1,000
- 1
- 100
- 10
- 10

**User-Service Graph**
- RDP, SMB, Kerberos, etc. logs – edges only between users and services used
- SMB
- Kerberos

User 1
User 2
User 3
Bipartite Graph Analysis for Enterprise Scale Network Defense

Analytical suite infers relationships between similar entities, scales to billions of records, and provides rapid situational awareness to SOC analyst.

Data
Sensor 1 ... Sensor N

Data

Analytics

Directional Graph Projections

Community Detection

First Order Graph Features

Analytical Contributions

1. A novel method to capture and represent similarity between network entities

2. A scalable method to compute directional graph projections for enterprise scale networks

3. A mapping between extracted graph features and relevant cyber use cases to an analyst

Analytics: Directional Graph Projections

Traditional Graph Projections

Directed Graph Projections

Nuances introduced by different graph weights and different destination nodes are ignored.

Asymmetric similarity measure can capture difference in usage of uncommon servers between clients.
Analytics: Community Detection

Identify communities within a network that are more connected to each other than other parts of the network.

\[
\Delta Q = \left[ \frac{\Sigma_{in} + k_{i,in}}{2m} \right] - \left[ \frac{\Sigma_{tot} + k_i}{2m} \right]^2 - \left[ \frac{\Sigma_{in}}{2m} \right]^2 - \left[ \frac{\Sigma_{tot}}{2m} \right]^2 - \left[ \frac{k_i}{2m} \right]^2
\]

- \( \Sigma_{in} \) is the sum of the weights of the links inside C
- \( \Sigma_{tot} \) is the sum of the weights of the links incident to nodes in C
- \( k_i \) is the sum of the weights of the links incident to node i
- \( k_{i,in} \) is the sum of the weights of the links from i to nodes in C
- m is the sum of the weights of all the links in the network.

## Analytics: Interpretable First Order Graph Features

<table>
<thead>
<tr>
<th>Graph Feature</th>
<th>Cyber Story</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Degree</td>
<td># of requests made, # of services used, …</td>
</tr>
<tr>
<td>Raw Weighted Degree</td>
<td>Amount I’m using a specific service</td>
</tr>
<tr>
<td>Projected Degree</td>
<td># of entities that I think I am similar to because we use a common service</td>
</tr>
<tr>
<td>Projected Community Size</td>
<td># of entities I am actually similar to</td>
</tr>
<tr>
<td>Projected Page Rank</td>
<td>My “significance” as compared to other entities (ex. Admins will use more services than clients)</td>
</tr>
</tbody>
</table>
## Cyber Use Case to Graph Feature Mapping

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Cyber Use Case Description</th>
<th>Features</th>
</tr>
</thead>
</table>
| **User → Service**   | Infer user roles           | **Admins** = High projected degree, community size, and page rank  
                       |                            | **Non-admins** = High projected degree but small community size and page-rank |
| **Client → Server**  | Infer similarities between groups of clients | **Typical Client Systems** = High community size, projected degree, and low page rank |
| **Internal IP → External IP** | Identifying firewalls, VPNs, or other network access points from flow data | **Firewalls** = High raw degree, weighted degree, projected degree, community size, and page rank  
                       |                            | **VPNs** = High raw degree, weighted degree, projected degree, but small community size, and page rank |
Modularization and integration identifies cyber use cases from graph feature mappings and also provides flexibility to identify anomalies within and across derived communities.
Scaling Directional Graph Projections

Message passing algorithms on graph data structure allows for custom asymmetric similarity measure and scales to $O(e)$. 

![Graph Diagram]
Technology Base

Graph Analytic Workflow

10 executors
4GB driver memory
3GB executor memory

45x Dell servers, 17.28 TB RAM, and 2.304 PB HDFS Storage

Run-time (secs) Vs Number of Records

- Cartesian
- GraphX
Use Case: Netflow from Edge of Network

*k-Nearest Neighbor* anomaly detection on graph projection features highlighted the single client in the largest community that made communication with a particular DNS server an anomalous number of times.
Graph projections onto username from Remote Desktop Protocol (RDP) logs highlights that communities of users that login from the same IP have multiple aliases.

Bipartite graph: username $\rightarrow$ source ip

Single user identified within a community with an alias of no resemblance to other users.

{$jsmith, johns, johnsmith, \ldots, adamb$}
Conclusions and Next Steps

Conclusions
1. A novel method to capture and represent similarity between network entities
2. A scalable method to compute directional graph projections for enterprise scale networks
3. A method to rapidly visualize, identify, and interpret anomalies from cyber logs using graph features

Next Steps:
1. Identify more relevant and concrete cyber use cases for improvements and expansions on various similarity metrics and graph features.
2. We would like to extend our work to better account for temporally evolving graphs to identify significant events that occur on a network at a particular time.