Anomaly Extraction in Backbone Networks Using Association Rules

Daniela Brauckhoff\(^1\), Xenofontas Dimitropoulos\(^1\), Arno Wagner\(^1\), and Kavé Salamatian\(^2\)

1. ETH Zurich - 2. Lancaster University
Background

- Network misuse has become common these days. Probes, scanners, denial of service are a few of the most common types of network attacks.
- Anomaly detectors are used in combination with other intrusion detection systems as a last line of defence.
- Anomaly detectors have not found widespread usage mainly for two reasons:
  - Due to high dimensionality of data, training a classifier is often difficult and access to “normal” datasets is limited.
  - High rates of false positives could cause difficulties for the network admin while false negatives could be very costly.
Background - Anomaly Detection Process
Key Contributions

- Avoid the need for “normal” traffic in the training phase.

- Minimize the amount of information that is presented to the network admin and reducing false positive rates.
Methodology
Methodology - Cont.

- The authors rely on Netflow data for their analysis but methodology could be extended to support other features as well.
- A set of anomaly detectors (histogram based) provide metadata of anomaly.
- The union of flows matching the anomaly detectors are selected in the pre-filtering phase.
- A summary report is generated by running Frequent Itemset Mining algorithms on the selected flows.
Frequent Itemset Mining

- Given a set of items $I$ and a set of transactions $T$, where each transaction is a subset of $I$ the goal of a FIM algorithm is to find all subsets of $I$ that occur more than a predefined support value $s$ in the transaction set.
- Algorithm operates in an iterative fashion by finding i-frequent itemsets in each step and relying on them to find (i+1)-frequent itemsets.
Frequent Itemset Mining - Example

(a) Transactions

0: \{a, d, e\}
1: \{b, c, d\}
2: \{a, c, e\}
3: \{a, c, d, e\}
4: \{a, e\}
5: \{a, c, d\}
6: \{b, c\}
7: \{a, c, d, e\}
8: \{b, c, e\}
9: \{a, d, e\}

(b) Frequent item sets (with support)
(minimum support: \(s_{\text{min}} = 3\))

<table>
<thead>
<tr>
<th>0 items</th>
<th>1 item</th>
<th>2 items</th>
<th>3 items</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\emptyset): 10</td>
<td>{a}: 7</td>
<td>{a, c}: 4</td>
<td>{a, c, d}: 3</td>
</tr>
<tr>
<td></td>
<td>{b}: 3</td>
<td>{a, d}: 5</td>
<td>{a, c, e}: 3</td>
</tr>
<tr>
<td></td>
<td>{c}: 7</td>
<td>{a, e}: 6</td>
<td>{a, d, e}: 4</td>
</tr>
<tr>
<td></td>
<td>{d}: 6</td>
<td>{b, c}: 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>{e}: 7</td>
<td>{c, d}: 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>{c, e}: 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>{d, e}: 4</td>
<td></td>
</tr>
</tbody>
</table>
Frequent Itemset Mining - Lattice
Histogram Anomaly Detectors

- Histogram anomaly detectors rely on the difference between two distributions for detecting anomalies.
- Since the input data is Netflow records, the authors rely on n histogram detectors each one detecting anomalies in different attributes of Netflow data (source/destination IP & port, protocol). Each histogram detector has m bins.
- Rely on Kullback-Leibler distance for anomaly detection (p, q are reference and given distribution respectively):

\[ D(p||q) = \sum_{i=0}^{m} p_i \log(p_i / q_i) \]
Histogram Anomaly Detectors - Cont.

- Instead of training and recalibrating distributions for normal behavior the authors compare consecutive windows with each other.
- Based on observation they generate an alarm if the distance is greater or equal to three standard deviations.
- To identify bins that were responsible for the anomaly they iteratively eliminate bins based on their degree of deviation until KL distance falls below threshold.
Anomalous Bin Detection Convergence
Histogram Cloning

- To reduce the likelihood of normal events being flagged as anomalous, histogram cloning is employed.
- For each feature $n$ we have $k$ clones that use an independent hash function.
- A feature is selected if at least $l$ out of $k$ clones agree on that feature.
## Parameter Space

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n)</td>
<td>Number of detectors</td>
<td>5</td>
</tr>
<tr>
<td>(w)</td>
<td>Interval length</td>
<td>[5,10,15] min</td>
</tr>
<tr>
<td>(m)</td>
<td>Hash function length</td>
<td>[512,1024,2048]</td>
</tr>
<tr>
<td>(k)</td>
<td>Number of clones</td>
<td>1-50</td>
</tr>
<tr>
<td>(l)</td>
<td>Voting parameter</td>
<td>1-(k)</td>
</tr>
<tr>
<td>(s)</td>
<td>Minimum support</td>
<td>1% - 10%</td>
</tr>
</tbody>
</table>
Parameter Space - Discussion

- **n**: have 5 detectors in total since we rely on Netflow data (src/dst IP & port, protocol).
- **w**: tradeoff between detecting short disruptions and number of false alarms.
- **m**: tradeoff between detection sensitivity and memory space requirements.
- **s**: low values of s result in higher detection rate and more false positives, while large values would not detect most events.
Dataset

- Netflow traces from the SWITCH backbone connecting Swiss universities and research labs.
- 2.2 million IP addresses within SWITCH network.
- On average 92 million and 220 million packets per hour.
- Two continuous weeks starting on December of 2007.
- 31 anomalous events identified manually as ground truth.
Clone & Vote Count Analysis

- For a given interval that contains anomalies each histogram selects $b$ bins that are responsible for raising the anomaly flag.
- To study the effect of clone and vote count $(k, l)$ the authors rely on simulations.
- The probability of detecting an anomaly is shown by $P_a$.
- Probability of selecting normal flows through anomaly detector $P_n$. 
Clone & Vote Count - False Negative
Clone & Vote Count - False Positive
Accuracy of FIM Algorithm

- Based on the findings of the previous section the following values were selected for the histogram detectors:
  - $k = 3$
  - $l = 3$
  - $m = 1024$
- This translates to a true positive probability of $P_a = 0.51$ and a false positive probability of $P_n = 10^{-4}$ for $b = 25$.
- Given the output of these detectors how many false positive itemsets would be generated by FIM algorithm?
Accuracy of FIM Algorithm - Cont.

- All of the 31 anomalous intervals were detected (100% accuracy).
- 21 intervals didn’t generate a false positive (FP) itemset.
- For the remaining 10 intervals the number of FP itemsets is dependent on the minimum support threshold.
- Majority of FP itemsets are attributed to common traffic patterns such as web.
Accuracy of FIM Algorithm - Cont.
Conclusion

- Presented a new method for detecting network anomalies based on a combination of histogram detectors and FIM algorithms.
- Explored the scope of involved parameters through simulation.
- Histogram detectors could be employed to decrease the number of generated itemsets and decrease the computational overhead.
- Accuracy of 100% with an average between 2 and 8.5 FP itemsets.