Summary: Detecting Malicious Domains via Graph Inference - Anomaly Detection Homework 6

Vishal A. Bhalla[1] and Nithish Raghunandanan[1]

***Abstract*— This report summarizes the approach and implementation within the enterprise security domain, for detecting malicious domains. The spread of malicious domains is the root cause of malware infections inside an enterprise and this approach scales well on a data collected at the global enterprise level. We have highlighted the methodology, results and the contribution of this paper to the set of malware detection techniques.**

I. INTUITION

Analysis of large data sets to extract actionable security information, and hence to improve enterprise security, however, is a relatively unexplored area [1]. Scaling is a challenge with existing machine learning techniques that require accurately labeled training sets along with a large number of features. Hence these techniques are resource intensive or computationally expensive which causes additional delay in detection.

Alternatively to achieve scalability, the authors model the malicious domain detection problem as a graph inference problem by adapting a faster approximate estimation algorithm i.e. Belief Propagation [2] because it scales well to large graphs, and also takes advantage of the structure of malware communication. The incorporated approach in this paper uses event logs and minimal data from existing blacklists and whitelists. For scalability and identifying new malicious domains not present in the blacklists, it is then applied over to event logs collected at a global enterprise over 7 months.

II. MODEL

A. ADOPTED METHODOLOGIES

1) Graph Inference Approach: They start by using an enterprises host-domain graph, adding a node for each host in the enterprise and each domain accessed by the hosts and an edge connecting the two. Each node has a state, e.g., malicious or benign. A domain can be malicious or benign, based on its presence in a domain black list or a domain white list respectively. Similarly, a malware infected host is known to be malicious otherwise it is benign. The nodes with a known state act as ground truth while the rest are unknown nodes.

2) Belief Propagation:

$$P(X = x) = \frac{1}{Z} \exp(\sum_j w_j f_j(x)) \tag{1}$$

[1]Master of Science students in the Department of Informatics, Technische Universitat Munchen, Germany. Email: vishal.bhalla@tum.de and nithish.raghunandanan@tum.de

B. Flow Diagram & Description

C. Implementation

1) Predicates and Formula:

2) Weight Learning:

3) Inference:

III. EVALUATION CRITERIA (DATASET)

A. Training Phase

B. Pseudo Code

The pseudo code depicts the sequence of events in the pipeline on each sentence extracted from a web link.

Algorithm 1 Belief Propagation

```
1:  procedure BELIEFPROPAGATION()
2:  Phase:
3:      if Phase = Training then
4:      Web Crawling:
5:          for each URL in URLList do
6:              text ← text + WebCrawlText(URL)
7:               // Web crawl text from the given link
8:      else
9:          if Phase = Test then
10:     Activity Label Mapping:
11:             for each activity in ActivityList do
12:                 map ← ActivityLabel(actId, lblName)
13:                  // Activity Label Mapping for each action performed
14:                 // by the subject
        // End of If clause to check Phase
```

C. Testing Phase

D. Results

A graphical depiction like a scatter plot highlights the difference between the probabilistic estimate for each tuple in both phases.

1) Interpretation of Results:

IV. RELATED WORK

V. CONCLUSION AND FUTURE WORK

REFERENCES

[1] A. Labrinidis and H. Jagadish, "Challenges and opportunities with big data," *Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 2032–2033, 2012.

[2] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring artificial intelligence in the new millennium*, vol. 8, pp. 236–239, 2003.