Hot Topics: Information Retrieval for Network Security[*]

Dustin Arendt[†]
United States Air Force Research Lab
Wright-Patterson AFB, Dayton, Ohio, USA
dustin.arendt.ctr@us.af.mil

ABSTRACT

Topic modeling is performed on 12 million firewall log entries to depict changes in the network over time.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: General; C.2 [**Computer-Communication Networks**]: Security and Protection (e.g., firewalls); H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical user interfaces (GUI)*

General Terms

Human Factors, Security

Keywords

firewall log, stacked graph, topic modeling

1. INTRODUCTION

Data from the network security domain presents many challenges to analysis and visualization and is a rapidly growing research area [6, 9]. Chief among these challenges is the fact that security log files generally contain huge amounts of structured but non-numerical data, preventing normal techniques for data analysis from being easily applied. Therefore, the author presents an analysis and visualization pipeline designed to tackle the non-numerical problem and reduce the amount of data to a manageable level. This is accomplished by approaching the problem from an information retrieval [4] standpoint, and applying techniques including bag of words abstraction and topic modeling. This is accomplished by assuming log entries are words and documents are sequences of log entries within corresponding time intervals. This approach is similar to work by Ribinson [8], where individual log entries are treated as documents and topic modeling was applied to discard (exfiltrate) large quantities of "normal" data.

2. METHODS & RESULTS

For this experiment, the author has used the firewall log from the 2011 VAST Challenge[1] (MiniChallenge 2). This log contains roughly 12 million entries spanning a period of 3 days and includes port scans and a denial of service attack. The topics over time are visualized as a stacked graph [5, 3] and provide a visual overview that shows emergent visual structures corresponding with the time intervals of some of attacks known in the ground truth of the dataset.

The overall analysis pipeline is as follows: tokenize each entry in the log file to create a "vocabulary"; create a document corpus by assigning documents to time intervals; determine the topics for the corpus and each document; and visualize the topics over time and vocabulary (an example of each is shown in Fig. 1).

Tokenization is the process in which words (e.g. a sequence of letters from an alphabet) are assigned a unique identifier. The firewall log is structured as a table with an arbitrary number of rows and a fixed number of fields. Fields can include information such as source/dest IP's, source/dest ports, sessions, protocol, etc. The log is tokenized by considering each row as a word, except for the time stamp, which is used later. The alphabet is the set of possible values any field in the firewall log may take, though not every letter from the alphabet is possible in each position of a word, and each word has the same length because each row has the same fields. Ideally we wish the alphabet to be small, so it is necessary to treat IP addresses and ports in a special way, since there are billions of possible IP's and tens of thousands of possible ports. This is done by applying the following mapping to each IP, port, and service (possibly containing redundant port information), which reduces the alphabet to just 113 characters.

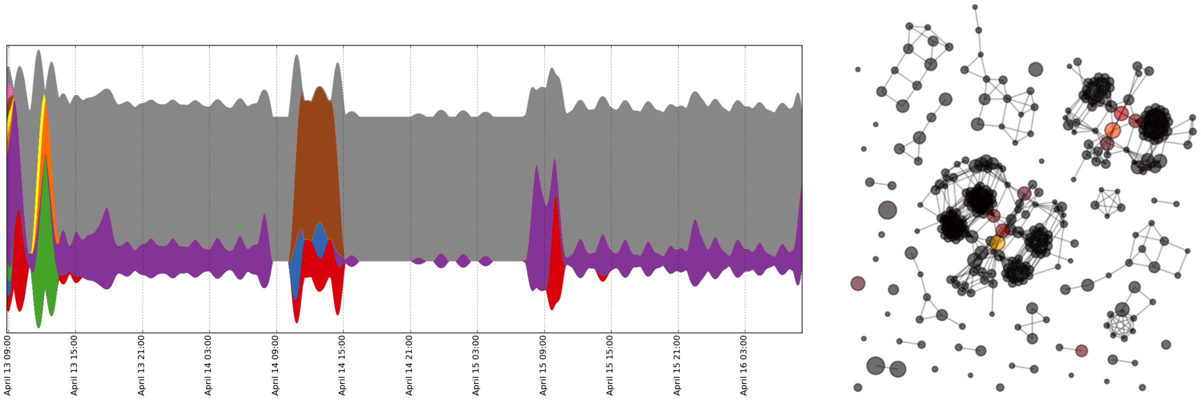
$$\begin{aligned} \text{IP address} &\mapsto \{\text{workstation, server, external, unknown}\} \\ \text{port} &\mapsto \lfloor \log_{10}(\text{port}) \rfloor \\ \text{service[:port]} &\mapsto \text{service} \end{aligned}$$

Each word (i.e. row in the firewall log) is associated with a specific time. Given this pairing, a document corpus can be induced from the log file by assuming one document covers a particular time interval. For this experiment documents correspond to five minute time intervals. For efficiency, documents are represented using the "bag of words" abstraction, a fundamental concept in information retrieval [4]. This implies that only the word frequencies, and not the particular order they occur in the document, are stored. Given the alphabet, and the restrictions the structure of the firewall log places on how log entries can appear, there can be up to 3.88×10^8 words in the vocabulary. Despite this, there are only 305 unique words observed in the firewall log's vocabulary after tokenization.

[*] Distribution A: Approved for public release; distribution unlimited. 88ABW Cleared 8/26/2013; 88ABW-2013-3828.

[†] Postdoctoral Research Associate of the National Academy of Sciences of the United States of America

[1] `http://hcil.cs.umd.edu/localphp/hcil/vast1`



(a)

(b)

Figure 1: (a) Hot topics–the topics of the firewall log over time. (b) The firewall "vocabulary" with nodes highlighted according to a particular topic.

The next step in the analysis pipeline is to use topic modeling as a form of dimension reduction. The goal of topic modeling is to summarize each document as belonging to one or more topics. The number of topics should be smaller than the vocabulary; as a rule of thumb, only a few hundred topics are used for very large document corpora. The author has decided on Latent Dirichlet Allocation (LDA) [1] to perform the topic modeling. Each LDA topic is a probability distribution over words, and each document has a distribution over topics. Tokenization and LDA are performed using Gensim [7], an open source Python package.

At this point the log's original length (number of entries) and dimension (vocabulary size) is greatly reduced and can be easily visualized. Since each document corresponds to a particular time interval (having a logical ordering) and has a distribution over topics, a stacked area graph [3] is a good choice for presenting the topics over time. The topics over time for the firewall log are shown in Fig. 1(a). The time series is smoothed by convolution with a Hanning window to improve readability. This visualization is much like the concept for ThemeRiver [5], which visualizes themes in a document corpus over time. However, because topics here are not recognizable words, but are distributions over unique log file entries, it is useful to also see each word rendered in terms of its relationship to the other words in the vocabulary and their overall frequency. This is shown in Fig. 1(b) a network is formed by connecting words whose hamming distance is exactly 1.

3. FUTURE WORK

An interactive visualization is in development using D^3 [2] where selecting a topic in the stacked graph highlights the corresponding nodes in the network and displays the relevant firewall log entries from the vocabulary. Such interaction is necessary to effectively explore the dataset due to use of the bag of words and LDA abstractions. Other datasets should be explored to begin evaluating the effectiveness of this approach. Additionally, the bag of words abstraction makes it simple to merge different data sources (i.e. IDS, PCAP, etc.) into a single document, potentially improving the efficacy of the analysis. Finally, this approach also leaves open the possibility to "train" the LDA topics on a set of previously known set of attacks or scenarios that can be used to classify network intrusions in realtime.

4. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of Machine Learning Research*, 3:993–1022, 2003.

[2] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[3] L. Byron and M. Wattenberg. Stacked graphs–geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252, 2008.

[4] D. A. Grossman and O. Frieder. *Information Retrieval: Algorithms and Heuristics*. Springer, 2004.

[5] S. Havre, B. Hetzler, and L. Nowell. ThemeRiver: Visualizing theme changes over time. In *IEEE Symposium on Information Visualization*, pages 115–123, 2000.

[6] R. Marty. *Applied Security Visualization*. Addison-Wesley, 2009.

[7] R. Řehůřek and P. Sojka. *Gensim–Statistical Semantics in Python*. EuroScipy, 2011.

[8] D. G. Ribinson. Statistical language analysis for automatic exfiltration event detection. Technical report, Sandia National Laboratories, 2010.

[9] H. Shiravi, A. Shiravi, and A. A. Ghorbani. A survey of visualization systems for network security. *IEEE Transactions on Visualization and Computer Graphics*, 18(8):1313–1329, 2012.