International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 06 - Issue 06 || June 2020 || PP. 11-15

An Analytical View of Interface on Security Logs Adapting Deep Learning Techniques

Prateek Bajaj, Sumaiya PK

Abstract: Amongst all aspects of Security in Applications and Products, Security Log Analytics has taken a priority lower than many other aspects for a plethora of reasons. However, the power it holds over a responsive, as well as a proactive approach towards securing applications, and in lieu products is unmatched. Security Log Analysis does result in evidence of attacks. A lack of such, thus, results in several security threats going unnoticed. The issue, though, during log analysis with only human labor and intervention is a longer response time and a lot of manual effort. Thus, automating log-analysis is one way through which the technology of the future is helping out in making the products of today secure.

Deep learning algorithms, as popular as they are, have for some time now helped in multiple applications for making intelligent decisions with the help of huge amounts of data. This paper provides a look into a comprehensive deep-learning algorithm that would help in security log analysis for the generic use-cases, that can be tweaked according to the particular product's log-data is elaborated.

Introduction

With advancements in the field of software and security in the recent years, the advent of attacks of products, software, and networks have drastically increased; digitization has although transformed organizations in ways that were unfathomed a few years back, there is still huge susceptibility towards cyber-attacks. (Chuvakin, Anton, et al.)

Such attacks do not just have a technological concern, but also an economic and a social impact to the organization. To avoid such concerns risking businesses, there came a concept of Secure Operations Centers (SOC's) that allow for monitoring security devices of organizations continuously. Such an organizational unit is needed to detect and/or prevent such attacks. Such SOC's bring into use the concept of SIEM tools (Security Incident & Event Management) software that provide a centralized place to collect and manage logs from different sources.

SIEM software, as they are commonly known, are used for collecting, processing, analyzing, and formulating huge log files from varied sources. (Mitchell, Tom, et al.)

However, a major concern with SIEM tools is the ginormous abundance of logs that are generated every second. This makes it almost impossible for human beings to keep track of errors, and detection of threats.

In this research paper, the major challenge is to overcome some of the challenges faced by SIEM tools for log analysis and to provide a study on how Machine Learning can be used to implement intelligent log analysis, with minimum efforts.

Issues with Current Siem Log Analysis

1. Difficulty in selecting specialists for SIEM: Due to the complexity of such tools, it is extremely difficult for organizations to find specialists just for the task of using such tools.
2. Cumbersome Processes: Due to detailed time consuming tasks that are present to understand, analyze and deduce outcomes from logs, it is a tedious task to generate and work on lengthy processes.
3. Huge amounts of logs: Since the motive here is to deal with huge amounts of log data, it is extremely difficult to handle and manage, let alone analyze logs. (Pellissier, Ren, et al.)

Due to such prevalent issues, there is a proposal for a detailed machine learning algorithm that caters to log analysis in an intelligent manner.

LOG ANALYSIS 101

Before diving into complex machine learning algorithms and techniques for analyzing logs, it is imperative for users to understand log analysis as a science.

Application faults often have an immediate business impact, such as unavailability of service or feature, and even data corruption. In some cases, application faults may be also related to security issues. Therefore, identifying these events can be a critical task.

As opposed to the security logs, which are structured and predictable, the application layer springs from a multitude of developers, including in-house, commercial teams, etc.

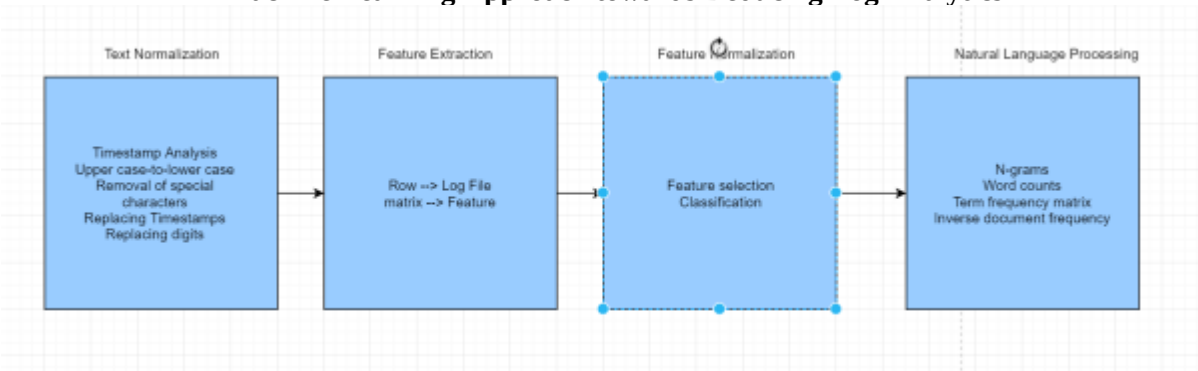
The following example illustrates a connectivity problem (the user was not able to connect to the app). *(Fawcett, Tom, et al.)*

In order to investigate the source of the problem, the user searches for the term "socket", which returns hundreds of results. However, using the Augmented Search semantic analysis the use can immediately identify the connection problems and authentication failure indications, although they are very different in their wording and ID.

To understand how log analysis works from textual perspective, some common text normalization techniques that could prove useful for log analysis would include:

1. Removing punctuation, e.g., . ; : ? ! = - [ ] — ¡ ¿ +, including the content between parentheses.
2. Removing definite and indefinite articles, e.g., a , an , the.
3. Removing weak words and prepositions: e.g., be, is are, of, at, such, after, from, being, do, done.
4. Replacing all numbers by the word NUMBER.
5. Replacing domain specific identifiers by corresponding words such as REGISTER or DIRECTORY.
6. Replacing all dates by DATE.
7. Replacing the upper-case letters by the corresponding lower-case letters.
8. Replacing present participles, past of verbs e.g. FAILING, FAILED by the main verb e.g. FAIL.
9. Replacing the plural noun e.g. ERRORS by the singular form e.g. ERROR.
10. Deleting the name of the day, for example Tuesday, or TUE. It is enough 20/Feb/2005, in order to reduce the length of the message.

A Machine Learning Approach towards Deducing Log Analytics



With the help of Machine Learning, in this case, the use of Decision Tree Algorithm, there is a classifier that helps in creating instances by drawing different attributes down to the root, without the need for explicit manual intervention. (K. Ryosuke. K. Kiyoshi. et al.)

A Machine Learning approach, to start with, can be divided into a 4-step extensive procedure:

1. Text Normalization: Normalization of text is the first step that is required for log analysis. It requires timestamp segregation as a separate attribute, that allows for anomaly detection at the most grassroot level.
Next comes bringing the text case as the same consistent one through the log file, making it easier to analyze.
Other text normalization techniques include removing special characters, replacing timestamps with a separate column, as discussed, and managing whitespaces.

2. Feature Extraction: The second important step towards building a machine understandable log file is to extract the relevant features required for analysis.
A popular way of achieving that is to convert each log row as a log file, and each matrix inside of that as a feature (date, error type, etc.)
This accounts for clear distinction between various features being extracted from the log files.

3. Natural Language Processing: With the help of the popular NLP algorithms and formats, handling sentences in log files would be easier than ever. With forms such as splitting the data using N-grams, Word Counts, Term Frequency Matrices, and/or Inverse Document Frequency allows for a clarity on understanding text from logs better using automation.

4. Decision Making & Anomaly Detection: This is where the preprocessing ends and the actual analysis begins.

International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 06 - Issue 06 || June 2020 || PP. 11-15

A Deeper Understanding towards the Algorithm

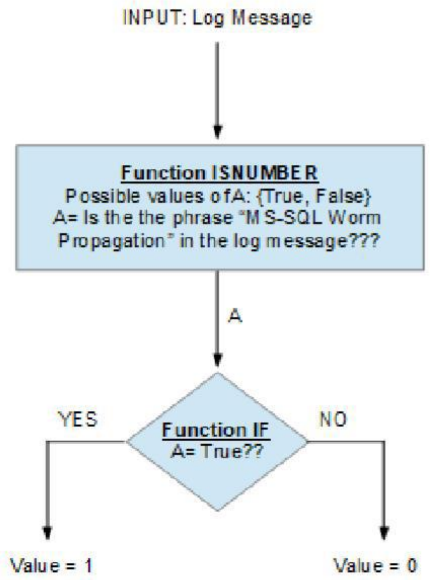


Figure 3.7: Function ISNUMBER and IF

The most important deductions that can be brought about with the help of machine learning is grouping of attribute values and anomaly detections. For instance, as shown in the image, the algorithm suggests that if phrases such as "Worm propagation" exist, the value of that attribute should be set to 1.

Segregating attribute outputs based on numerical values helps in generating graphical representations of outcomes without any manual efforts.

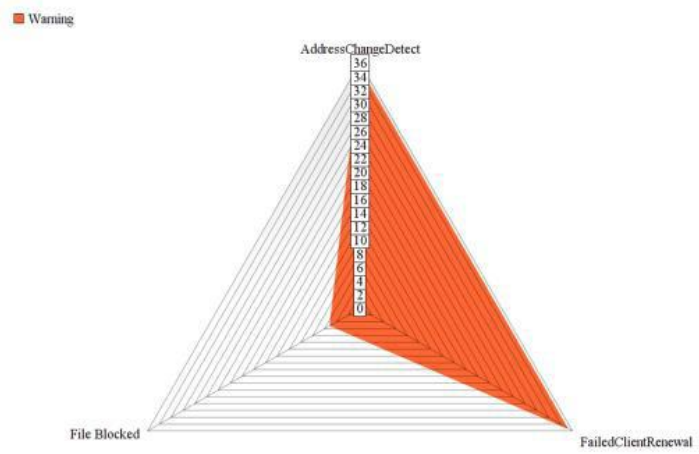
For the algorithm, the following 4 data types have been considered:

1. Numeric: can be real or integer numbers
2. Nominal: it is specified the possible values, i.e. value1, value2, value3
3. String: declaration of text values
4. Date: specifies the date and time of the data set. The format follows the ISO8601 yyy-MM-dd0 HH:mm:ss.

An example of using the algorithm for segregation of attribute values based on types of logs includes:

WARN	[LeaseRene	org.apache Address ChangeDetect	chang	detected.	Old:	msra-sa-41	New:	msra-sa-41
WARN	[LeaseRene	org.apache Failed Client Renewal	to	renew	lease	for	[DFSClient_	for
WARN	[LeaseRene	org.apache Address ChangeDetect	chang	detected.	Old:	msra-sa-41	New:	msra-sa-41
WARN	[LeaseRene	org.apache Failed Client Renewal	to	renew	lease	for	[DFSClient_	for
WARN	[LeaseRene	org.apache Address ChangeDetect	chang	detected.	Old:	msra-sa-41	New:	msra-sa-41
WARN	[LeaseRene	org.apache Failed Client Renewal	to	renew	lease	for	[DFSClient_	for
WARN	[LeaseRene	org.apache Address ChangeDetect	chang	detected.	Old:	msra-sa-41	New:	msra-sa-41
WARN	[LeaseRene	org.apache Failed Client Renewal	to	renew	lease	for	[DFSClient_	for
WARN	[LeaseRene	org.apache Address ChangeDetect	chang	detected.	Old:	msra-sa-41	New:	msra-sa-41
WARN	[LeaseRene	org.apache Failed Client Renewal	to	renew	lease	for	[DFSClient_	for
WARN	[LeaseRene	org.apache Address ChangeDetect	chang	detected.	Old:	msra-sa-41	New:	msra-sa-41
WARN	[LeaseRene	org.apache Failed Client Renewal	to	renew	lease	for	[DFSClient_	for
WARN	[LeaseRene	org.apache Address ChangeDetect	chang	detected.	Old:	msra-sa-41	New:	msra-sa-41

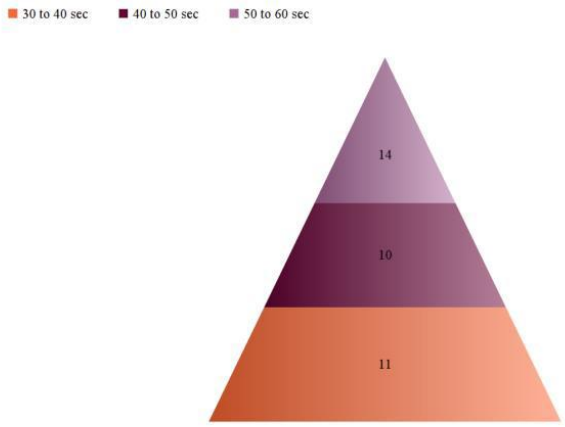
With the kind of data mentioned in the image above, the automated self-generated interpretations for the data include:



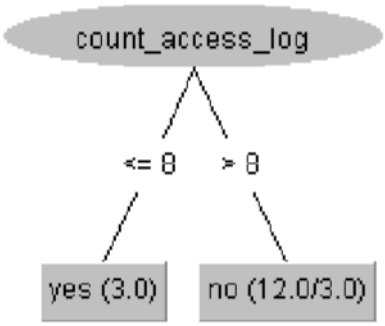
The concept for the algorithm revolves around segregating attributes, assigning numerical values to some (in case of categorical attributes), and assigning to the algorithm those attributes for detection of outcomes that the user considers necessary for the analysis.

Timestamps can be analyzed for understanding time intervals between security breaches (if any), along with duration and frequency.

Here's an example of analyzing timestamp values based on the algorithm:



Another such example of automating manual detection of trivial log tasks such as access to the system can be seen here:



Setting up pre-defined rules in the algorithm to suggest breaches based on number of access logs can be handled.

Conclusion

With the world moving towards automation and the idea of reducing manual intervention in the field of data and log analysis is a topic that could help change how analytics are considered and handled for huge amounts of data.

The algorithm proposed in the research allows for creation of graphical information that is easily interpretable, just with the deciding on attributes to be considered for the deduction of either anomalies or groups.

References

[1]. Anton Chuvakin. Scan34, http://old.honeynet.org/scans/scan34/. Last Accessed: 2018-11-03.
[2]. Tom M. Mitchell. Machine Learning. McGraw Hill, 1997.
[3]. Ren Pellissier. Business Research Made Easy. Juta Academic (September 5, 2008), ISBN-13: 978-0702177033.
[4]. K. Ryosuke O. Satoru and K. Kiyoshi. Minimizing false positives of a decision tree classifier for intrusion detection on the internet. Journal of Network and Systems Management, 16(4):399–419, 12 2008.
[5]. Tom Fawcett. An introduction to roc analysis. PatternRecognitionLetters,27(8):861 – 874, 2006. ROC Analysis in Pattern Recognition.