Proceedings of Student-Faculty Research Day, CSIS, Pace University, May 5th, 2017

Big Data False Alarms: Improving Data Leakage Detection Solutions

Eric Amlie, Peter Gelsomino, Aditya Goswamy Giri, Juan Rodriguez, and Pedro Vasseur
Seidenberg School of CSIS, Pace University, Pleasantville, New York

Abstract – Today, the security of sensitive and confidential information is a crucial concern for all organizations. A large amount of sensitive information is stored, transferred, and utilized digitally. This paper examines various methods of confirming and prioritizing alarms with the intent of collecting "true positives" and minimizing "false positives" as they are presented for examinations in the security logs of data leak detection applications. A true positive is an alarm that is captured in security logs that are, in fact, true. A false positive is also an alarm captured in a security log marked as true, but is realistically false. This can cause a large amount of extra work and confusion for security teams trying to decipher which alarm they need to focus their attention on. The material presented in further reading will examine specific security logs using big data analytic methods to determine what circumstances are likely to trigger false alarms. By identifying common triggers of false alarms and the most effective methods for re-classifying and prioritizing those alarms, this study strives to potentially reduce false positive rates in security logs.

Index Terms– big data, data analysis, data leak detection, data loss prevention, false positives, security logs.

I. INTRODUCTION

To large companies such as Target, Home Depot, BlueCross BlueShield, and many other organizations, the damages data breaches causes when they occur can be catastrophic. These breaches can cause companies to lose status and reputation, or encounter enormous financial losses. Data Loss Prevention and Data Leak Detection methods have become critical to maintaining information security, and preventing these data breaches. As these technologies are still being developed, honed, and perfected, they do not always provide an applicable level of accuracy. In many cases, they will return logs that contain "noisy" data, or a large number of false positives that have to be identified and thrown out. Because this has to be done manually, going through "noisy" data takes time away from legitimate threats, and slows down the response time to true alarms. In order to help improve the efficiency of Data Loss Prevention and Data Leak Detection technologies, it is important to find methods that reduce these false positives.

In looking to reduce the number of false positives in security logs that track data leaks and data loss, the false alarm team wants to examine the issue of false positives from two perspectives. The team will begin by analyzing the logs and identifying the specific circumstances, attributes, or actions that trigger false alarms in the hopes of creating rules that would be effective at reducing their number. Identifying commonalities in false alarms may lead to instructing an application detecting leaks to provide more accurate results. The team will also use an algorithm and data mining application to assist in the classification of true and false alarms. This combined effort will assist an organizations security department, hopefullly limiting the number of false positives that security analysts make a decision on.

It is equally important to consider the role that data status plays in the monitoring of data leaks. Sensitive data can be considered "at rest" – being stored and not in transit or in use – or "in motion", in transit across a network [6]. The methods employed for protecting data will differ depending on the state of the data, and is therefore likely to trigger false positives for different reasons. Data at rest is usually monitored by access controls and file permissions; false alarms triggered by attempts to access this data may warrant re-evaluating file permissions and user groups. Data in motion would be monitored by the Data Loss Prevention/Data Leak Detection (DLP/DLD) applications. The false alarms are triggered by data in traffic that may require a more careful examination of the rules that govern incidents, as well as modifications to make them more specific.

II. RELATED WORKS

In the article, *Strategies to Reduce False Positives and False Negatives in NIDS*, the author Timm describes the design of NIDS, Network-based Intrusion Detection

System, as one of three models and are the simplest and most common [13]. The NIDS systems are great at identifying known attacks, however, they are unable to detect unknown or even slightly modified attacks. They also have the potential to produce many false positives by picking up the attack signature in non-attack traffic. This occurs when a user references an attack or includes text that is part of a known attack signature. Another known system is Anomaly-Based systems that use weighting to predict the probability of an intrusion based on the frequency that the traffic occurs. This method is better than signature models at reducing the chances that data passes through as a false negative. Alternatively, Timm describes them as less flexible due to their mathematical focus.

In *Using Fuzzy Cognitive Maps to Reduce False Alerts in Som-based Intrusion Detection Sensors*, the author Mahmoud Jazzar builds on this idea of anomaly-based systems by including other factors to estimate the abnormality of an individual packet [7]. The weights include availability, similarity, occurrence, relevancy, independent and correlation factors with an effect value to more accurately estimate a total degree of abnormality. The weights are computed using a neural network to make fuzzy cognitive maps. This technique may offer a reduction in false positives but the black-box nature of the neural network make it harder to understand the inner workings of the system and why traffic is being labeled what it is.

In their efforts to develop standards to address insider threats to information security, Mark Guido and Marc Brooks in their paper *Insider Threat Programs Best Practices* identify the key components of a comprehensive insider threat mitigation program. They identify clear security policy, strong monitoring and auditing measures, and complementary preventative controls as important parts of a high-level program. They go on to establish best practices for the mitigation of insider threats, which include developing and issuing acceptable use policy to users, utilizing continuous monitoring, utilizing active prevention in tandem with monitoring, and identifying and examining user behavior that may precede a data leak [5].

III. APPROACHES TO DATA LOSS PREVENTION

A. DLP: A Summary

Data Loss Prevention, often shortened to DLP, is a strategy for making sure that end users do not send sensitive or critical information outside the corporate network. DLP needs to be implemented and enforced at a strategic level rather than just providing DLP tools to a network. To solve this challenge, the team needs to base the solution on the context in which data is accessed. For example, an employee goes to work in an office uses his own iPhone to check corporate email and then downloads a PDF to look at for later. Seemingly this sounds like an innocent action, but this scenario poses a few threats:

- How did the user connect his phone to the corporate network, Internet or LAN?
- How does the organization ensure that only trusted devices can connect?
- Was the user ever authenticated and was it logged for audit purposes?
- Was the e-mail attachment a corporate document? If it were, would it be subject to a data classification scheme where DLP is administered?
- If this classified document was marked for Internal Use Only, how can we be sure that it is secure from being copied by a third party?
- What happens if the device is stolen or lost? What options does the company have in relation to remote wipe, recovery of data or device encryption?

Using the above example, it is easy to see why it is necessary to have a plan in place to guard against data loss. These are the steps required to complete a successful DLP implementation:

- Identify the data that needs to be protected
- Classify the data according to business information levels.
- Appoint data owners.
- Set a policy for data handling and implement DLP controls to make them available to the data owners.
- Use DLP reporting tools to identify violations.
- Act on DLP violations by adjusting DLP controls, HR improvement, or both.

Requirements for the DLP fall into two categories, one for data in motion and one for data at rest. Data in motion, or network DLP, deals with data moved over to the corporate network. It can include data going and coming from the Internet or other networks and applications. Data at rest deals with data hosted on servers or in storage. This includes data on file shares, database servers or content management systems. A comprehensive DLP solution will secure both types of data but it can be complex to make. The team will be focusing this paper on the data at rest. DLP challenges created by noisy security logs where false alarms are predominant.

When designing a DLP solution, the first step is understanding how the solution will integrate with other network components and security protocols already deployed. DLP would integrate with any firewall and content inspection solutions already deployed. A typical network DLP deployment's integration should have its Internet firewall forward outbound traffic to the content inspection solution. The inspection would submit any traffic containing matching data to the DLP solution for inspection. The DLP solution would then make a firewall to block said traffic. DLP Applications, Methods, and Best Practices

Now that we have identified the need for DLP measures, we can begin to look at the various ways in which DLP solutions achieve their intended goal. There are different methods by which these applications will aim to detect sensitive data as it travels the network, and determine if the policies regarding how sensitive information is handled have been breached. The exact methods employed by an organization will ultimately depend on the type of sensitive data the organization is looking to track, and the their overall security goals. Some organizations may have large databases to store customer information. This may include personal information consisting of social security numbers, or credit card information. Others may be looking to safeguard sensitive text documents containing confidential organizational information and trade secrets.

Depending on the specific data an organization looking to protect, DLP applications can employ different methods to detect breaches. For example, there are methods that rely on detecting leaks within the content of the data and comparing the content of network traffic against the words and patterns found in sensitive documents. Pattern matching is employed to detect instances of certain numerical patterns, for example xxx-xx-xxxx for a social security number, or xxxx xxxx xxxx xxxx for a credit card number. Keyword matching is similar, working instead towards detecting matching words rather than numbers.

The problem with the methods above is that in many cases are flawed, or not comprehensive enough to provide acceptable results. For example, pattern matching, while theoretically useful for detecting things like social security numbers in traffic, can be easily fooled if the action is not accidental. This could be performed by an attacker who intentionally modified the format of the numbers in order to bypass detection. Similarly, keyword detection can be bypassed by modification. Furthermore, if not given the ability to examine keywords in the context of the document, it is likely that the application will generate a large number of false alarms, making it more difficult to respond to the legitimate ones, and ultimately complicating the problem we are seeking to solve. [11]

There are certain additional steps that can be taken to improve the efficacy of a DLP application. For example, when looking to detect credit card numbers, the application of the Luhn algorithm helps to differentiate between arbitrary strings of sixteen digits, and potentially valid credit card numbers [11]. Because this algorithm is able to give a fairly accurate determination of whether the arrangements of digits are a valid credit card number, you can reduce the instances of false positives for this type of detection by employing it. Unfortunately, a similar algorithm has not yet been determined for applying the same logic to social security numbers, so there is not necessarily an across the board fix for the weaknesses inherent in these methods.

Data or document fingerprinting is an alternative method to keyword or pattern matching. Instead of looking for a keyword match within the content of the document, the document itself becomes the match. A sensitive document or pieces of a sensitive document are fingerprinted, or assigned a cryptographic hash value. This hash value is then compared against the hash values created by fingerprinting traffic in the same way. Though more effective than the aforementioned methods, this method can also be bypassed by making modifications to the documents being transmitted, as modifications to the contents of the document will result in differences in the hash values, which may prevent detection. [9] While this may help in cases of inadvertent or accidental leaks, it still does not help to stop a data leak in cases where the intent was malicious.

There are several other DLP methods that can detect files containing specific information. One way is to determine frequencies for the threshold and test documents that are needed to prove the method works. In one test that uses semantic similarity detection for DLP measures those frequencies of the test documents and if they are higher than the threshold, then the document did not go through. According to Euzenat, semantics "provides the rules for interpreting the syntax which do not provide the meaning directly but constrains the possible interpretations of what is declared" [2]. Compared to conventional DLP approaches, which use syntactic features, the singular value method identifies files based on semantics. The singular value method discovers the semantic features contained in the training set, which has the documents being tested. Unlike regular expression methods, this particular approach extracts a small number of critical semantic features and requires a small training set. Existing tools concentrate mostly on data format where most industry applications would be better served by monitoring the semantics of information in the enterprise.

No matter which method or combination of methods is being employed, it is important that it be periodically assessed for efficacy, and the rate of both false positives and false negatives examined to determine if they fall within acceptable levels. Currently, it is the consensus of experts, that the majority of detected alarms are false in nature.[3]

C. Preventative Controls

There are additional measures that can be taken to complement Data Loss Prevention applications in achieving their goal. Developing acceptable use policy involves creating a list of rules and accepted user behaviors, as well as restricted user behaviors. Examples of could include rules prohibiting the addition of email attachments to external email addresses, rules outlining the process for printing sensitive information or requesting a hard copy of a sensitive document, and rules prohibiting risky web browsing behavior. Outlining these behaviors will both serve to identify what sort of restricted behaviors should trigger alarms and indicate to users what actions should be avoided. By making users aware of what actions are considered acceptable use and which actions are restricted, an organization can cut down on any false alarms triggered by a user inadvertently performing a restricted action because they were not aware that it was restricted. While some of these rules may be implemented using the honor system, it is also possible, and usually advisable, to implement preventative controls to ensure that policies are being appropriately followed, and reduce the number of incidents logged by security applications.

Preventative controls are best used in tandem with monitoring and auditing. Preventative controls can include a variety of access control measures taken to restrict user access to sensitive information, as well as restricting certain user behaviors. Rather than simply providing users with an acceptable use policy and encouraging them to follow it, you can take steps to ensure that behaviors that increase the risk of a data leak, and that have no benefit to the business processes of the organization, can be eliminated entirely. For example, you might elect to block external email services to ensure all email traffic is conducted through an email server being monitored for data leaks. If there is no business need for USB ports to be active, disabling USB ports to prevent sensitive data from being stored on removable media devices can be a beneficial policy. This also eliminates the risk of users compromising your system by using personal removable media devices infected with malicious software, either inadvertently or intentionally. Another example of preventative controls would be implementing secure printing procedures to hold employees accountable for hard copies of sensitive information.

These examples are by no means comprehensive. The specific preventative controls enacted by an organization will depend on the environment, and the business processes being performed within that environment. It is important that security measures do not place an unnecessary burden on users, or impact business processes in a negative way. That being said, utilizing preventative controls not only strengthens your security measures to prevent data leaks, it can also help reduce the number of false positives generated by security logs by helping to reduce the total number of incidents documented by the logs. If removable media ports are disabled, there is no need for the logs to attempt to determine whether secure information is being transferred to removable media by an unauthorized user. By eliminating the potential for an incident, you can reduce the overall number of incident reports. These measures will neither protect against all data leaks, nor remove all instances of false positives, however, concluding that additional steps must be taken to address these.

Monitoring and auditing on a continuous basis is necessary for the effective prevention of data leaks; this monitoring will generally be done by the Data Loss Prevention/Data Leak Detection application that has been implemented by the organization, but if the data collection performed by the monitoring application generates too much data or the data that is generated is too noisy for security analysts to respond to in a reasonable amount of time, it is not fulfilling its purpose. With that in mind, the team will analyze the provided security logs to identify traits that trigger false alarms, and determine the best methods for confirming and prioritizing alarms with the intent of underscoring true positives and downgrading false positives as they are presented for examinations in the security logs of data leak detection applications.

IV. PREVIOUS WORK

Previous study of this topic at the Seidenberg School of Computer Science and Information Systems, gained an enormous amount of ground researching, testing, and analyzing techniques to assist in the output of a DLP application. The data they studied was an example of data in motion and data they examined generated an alarm while in transit over the network. The alerts occurred over the span of about four hours during normal business hours, and recorded 352 incidents. Of the 352 incidents, 40 were later determined to have been false alarms. [2]

The first data analysis technique used was data clustering and identifying the attributes recorded in the logs that resulted in a false positive. Due to the shortness of time and lack of a computer assistant, the team decided to that the 352 recorded incidents in this sampling included alarms triggered by twenty-three different users. However, they were able to identify that of these twenty-three different users, only three of them were generating false positives; the rest were generating only legitimate alarms.

This information gave the team an extreme advantage, knowing which data was false and which was not. One of their biggest challenges was, the three users that generated the false alarms, also generated legitimate true alarms. What they found to be useful was the target UID path of the uploads which triggered the false positives. All of the false positives shared the same target UID path, which itself was unique from the target UID path of any other user's alarms, legitimate or otherwise. By flagging this target UID path as being a false alarm, the rate of false positives in the given data dropped from 11% to 3.9%.

Last semesters team discovered a pattern where any file uploads that took more than one second were accurately classified as a false alarm and any file that took one second or less was then sorted based on the number of changes that were made to file ownership. With this in mind, the team decided to use a decision tree. The majority of files had only one change in ownership and were split between actual and false alarms. Each file that had more than one change in ownership was a true alarm. The tree seemed to be effective however still misidentified some cases.

Analysis through clustering proved to be a useful tool for identifying and weeding out problematic users, and may provide a suitable solution for reducing the number of false positives generated among users in a single network. However, in order to use it to successfully predict future false alarms, a larger data set would need to be analyzed to determine the best way to cluster false positive traffic.

The decision tree algorithm proved to be more applicable to multiple environments, as it was able to take into consideration various attributes of the data. It allowed the consideration of more attributes at a single time and how the value of one attribute might affect another. Unfortunately due to project limitations, there were also various aspects of the data the team was unable to incorporate into their analysis, leaving many options for additional things to consider in future research.

This semester's team leveraged as much as possible on the previously done work, by applying to the data-at-rest condition, especially to Big Data security logs.

The team's client had created slides to relay the current status and desired states of false alarms handling. The teams objective is to further reduce the amount of false alarms by creating an algorithm that is both predictive and proactive to help limit the amount of logs the security analyst sees before the make a decision whether to act on, or discard an alarm. *Figure 1* below, represents the current state with a voluminous alarms log report, while *Figure 2* represents the team's final objective.

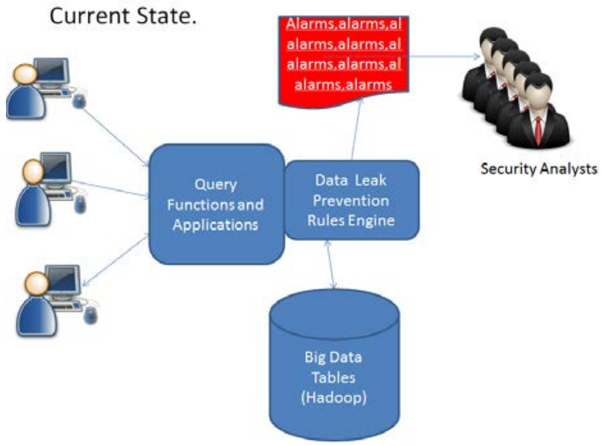

Current State.
Alarms,alarms,al alarms,alarms,al alarms,alarms,al alarms,alarms
Security Analysts
Query Functions and Applications
Data Leak Prevention Rules Engine
Big Data Tables (Hadoop)

Figure 1: Big Data Teams Current State

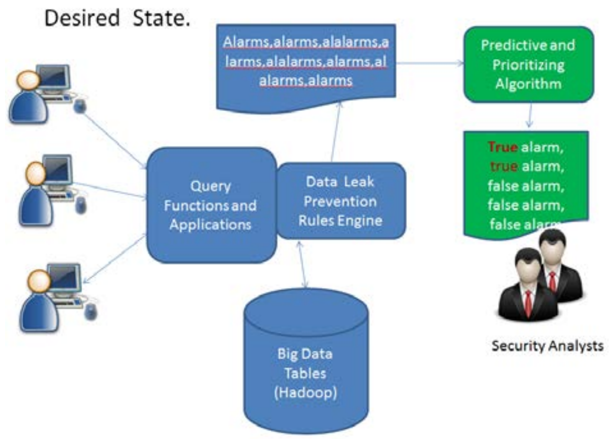

Desired State.
Alarms,alarms,alalarms,a larms,alalarms,alarms,al alarms,alarms
Predictive and Prioritizing Algorithm
True alarm, true alarm, false alarm, false alarm, false ala
Query Functions and Applications
Data Leak Prevention Rules Engine
Big Data Tables (Hadoop)
Security Analysts

Figure 2: Big Data Teams Final Objective

V. RAPIDMINER & DATA SETS

Due to security concerns, the team failed to obtain real life security logs. To continue project work, the false alarm team has been given a test security log in excel format to mimic the columns and values of DLP security logs to perform analysis by the client. The false alarm team used the decision tree ID3 algorithm used by the previous semester team.

The team used a data mining tool called RapidMiner to assist in analyzing the security data. RapidMiner was chosen because it was a well known software program that is free and open source, as well as a strong recommendation by the client due to his real life experience using it in the field. RapidMiner was also chosen because of its capabilities of implementing decision tree analysis and using the ID3 algorithm. RapidMiner will output a decision tree based on the imported the security log excel formatted data.

The training data set is a security log report that consists of seven columns. The first six columns contain variables that are candidates for predicting the value of the seventh variable, whether it is a true or false alarm. The variable column titles from A-G respectively are: timestamp, requestor, role, component accessed, request type, violation type, and alarm. To the team's advantage, the value of which row is associated with a true or false alarm is already given in the seventh column. The challenge is that no consistency exists in the data and each false and true alarm reveals no obvious pattern. The RapidMiner program must choose which columns create precedence over others and which combination will eventually result in the prediction of false alarms when new instances are given to the model's algorithm.

The team was given a test and validation data set by the client and mimic real world security logs. The data sets were the same format as the training data set mentioned in the earlier paragraph. The difference between the validation and the test data sets is the test data set has an empty column 7 that specifies whether the alarm record is true or false, while the validation data set confirms which columns are indeed true or false. To clarify, the training data set has completely different data logs. It is used only as a “training” mechanism for RapidMiner to help the program learn how to react.

VI. EXPERIMENTATION & RESULTS

A. Test 1

With both the training data set and RapidMiner installed, the team ran an analysis in RapidMiner. Initial analysis of the data concluded that separate users created a certain percentage of alarms. *Chart 1* below shows a representation of that data:

User	Alert Total	% of Total (104)
Analyst	74	69.24%
Business User	28	26.93%
Administrator	4	3.4847%

Chart 1: Breakdown of user activity in the security training log.

Without adjusting column weight and establishing that column 7 (the true or false column) as the label, the team let RapidMiner determine the results of the dataset. The test data was imported into RapidMiner and applied against itself in order to allow the program to learn through the use of the ID3 algorithm. Once imported, two columns were chosen for the example set – Role and Alarm – columns C and G The data was run through the ID3 algorithm and set as the data model by which the same data was run against. The second set of data was set as “unl” or unlabeled which, according to RapidMiner [4], is not explicitly used when training the model, but only used in predicting “the value of the attribute with (a) label role” [5].

Using two sets of test data (Data model for Test sample training v1.xlsx (DMTSTv1) and Test sample test validation v2.xlsx (TSTVv2)) a design process was set up in RapidMiner in order to have the new data, TSTVv2, learn from the older data. This would allow the output of the process to show which Component Accessed caused a true or false alarm based on criteria such as Violation Type and Requestor. In this scenario, both User A and B attempted to access components Element 1 and Element 2 which caused a true positive alarm. The goal is to understand why this combination caused the true positive alarm.

In setting up RapidMiner, the usage of the ID3 algorithm was again employed. This time making changes within the process to reveal more precise results. Parameters were set to achieve correctness with the data in the spreadsheet. *Figure 3,* shows the process as set up for this example.

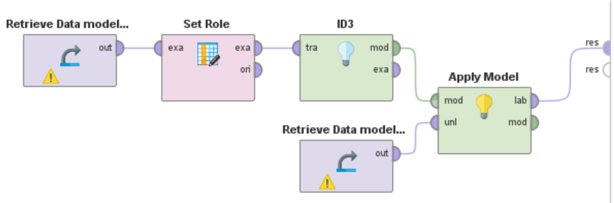
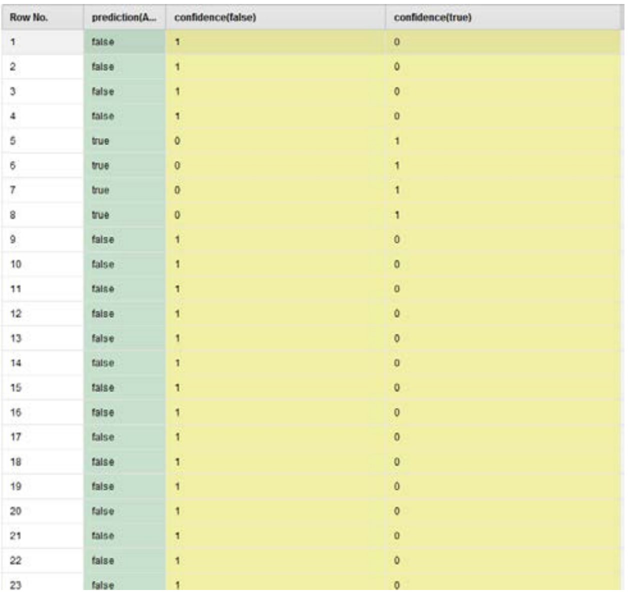


Figure 3: RapidMiner ID3 Algorithm Diagram with Test and Production Data Sets

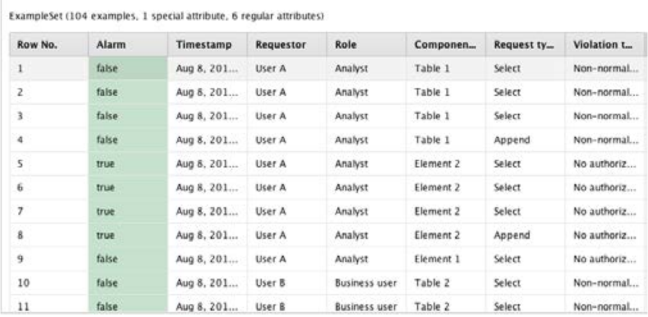
The results provided a view in terms of *confidence* of the data. This implies that the program was able to identify which alarms were nearly actual as being false or true in terms of 1 being positive and 0 being negative, shown in *Figure 4 and 5* below.



Row No.	prediction(A...	confidence(false)	confidence(true)
1	false	1	0
2	false	1	0
3	false	1	0
4	false	1	0
5	true	0	1
6	true	0	1
7	true	0	1
8	true	0	1
9	false	1	0
10	false	1	0
11	false	1	0
12	false	1	0
13	false	1	0
14	false	1	0
15	false	1	0
16	false	1	0
17	false	1	0
18	false	1	0
19	false	1	0
20	false	1	0
21	false	1	0
22	false	1	0
23	false	1	0

Figure 4: RapidMiner Results Confidence of Data

After this was executed, RapidMiner has the capabilities of understanding the column fields data set, and have the ability to apply it to a new production data set that does not include column G of the true and false alarm value. RapidMiner will determine its best guess of what rows will be true or false alarms.

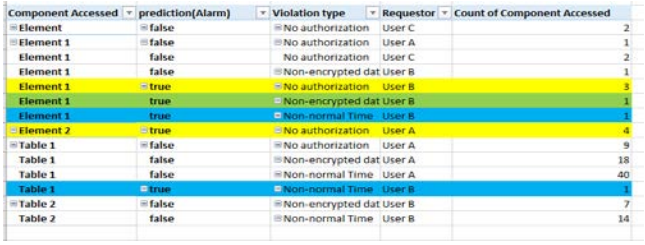


ExampleSet (104 examples, 1 special attribute, 6 regular attributes)

Row No.	Alarm	Timestamp	Requestor	Role	Componen...	Request ty...	Violation t...
1	false	Aug 8, 201...	User A	Analyst	Table 1	Select	Non-normal...
2	false	Aug 8, 201...	User A	Analyst	Table 1	Select	Non-normal...
3	false	Aug 8, 201...	User A	Analyst	Table 1	Select	Non-normal...
4	false	Aug 8, 201...	User A	Analyst	Table 1	Append	Non-normal...
5	true	Aug 8, 201...	User A	Analyst	Element 2	Select	No authoriz...
6	true	Aug 8, 201...	User A	Analyst	Element 2	Select	No authoriz...
7	true	Aug 8, 201...	User A	Analyst	Element 2	Select	No authoriz...
8	true	Aug 8, 201...	User A	Analyst	Element 2	Append	No authoriz...
9	false	Aug 8, 201...	User A	Analyst	Element 1	Select	No authoriz...
10	false	Aug 8, 201...	User B	Business user	Table 2	Select	Non-normal...
11	false	Aug 8, 201...	User B	Business user	Table 2	Select	Non-normal...

Figure 5: RapidMiner Output Results of both Data Set

Figure 6, highlights the relationships between Component Accessed, Violation Type, and Requestor in regards to the alarm output. The colors used separates the violation types. Further review of the data shows that the component most accessed was Element 1 which generated a No Authorization violation 3 times with User B – the Business User, 1 Non-encrypted data violation and 1 No authorization violation also for User B.



Component Accessed	prediction(Alarm)	Violation type	Requestor	Count of Component Accessed
⊟ Element	⊟ false	⊟ No authorization	User C	2
⊟ Element 1	⊟ false	⊟ No authorization	User A	1
Element 1	false	No authorization	User C	2
Element 1	false	⊟ Non-encrypted dat	User B	1
Element 1	⊟ true	⊟ No authorization	User B	3
Element 1	true	⊟ Non-encrypted dat	User B	1
Element 1	true	⊟ Non-normal Time	User B	1
⊟ Element 2	⊟ true	⊟ No authorization	User A	4
⊟ Table 1	⊟ false	⊟ No authorization	User A	9
Table 1	false	⊟ Non-encrypted dat	User A	18
Table 1	false	⊟ Non-normal Time	User A	40
Table 1	⊟ true	⊟ Non-normal Time	User B	1
⊟ Table 2	⊟ false	⊟ Non-encrypted dat	User B	7
Table 2	false	⊟ Non-normal Time	User B	14

Figure 6: Relationships between Component Accessed, Violation Type, and Requestor in regards to the alarm output.

There was a total of 104 alarms, 10 of which, 9.62%, were labelled as true positive. User B, which is in the role of the Business User, generated the most amount of true positive alarms, 6 in total – Non-normal Time, Non-encrypted data, and No Authorization. User A, our Analyst role, generated 1 for No Authorization.

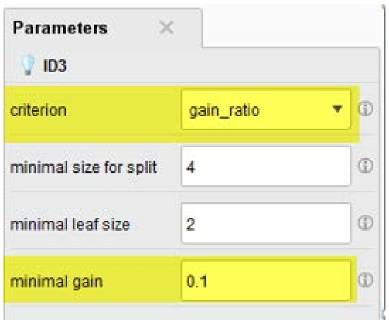
Noting these aspects, there is the question of whether or not RapidMiner is taking the "Time Stamp" column into account regarding true and false positives and if this aids in generating such alarms. While false positives are viewed before and after 7PM, all of the true positives have occurred after such time.

B. Test Two

For the second test, the team was given a second data set from the client. This data set was different form the training data set given in the last test. The team used the original training set as the basis for entering into the ID3 algorithm. Then the team plugged in the second set, similar to *Figure 2* shown in the previous paragraphs.

After running the model, the team got unexpected results. The model was incredibly inaccurate and ended up returning only 43% accuracy with a total of 43 false alarms and 77 true alarms.

In order to get more accurate results, the team looked into the parameters section of the ID3 algorithm. First of the parameters is the criterion. There are four choices to choose from. One is the information_gain. This parameter minimizes the entropy, a gradual decline of unpredictability, of the data. Next is the gain_ratio that minimizes the distance of the attribute values. Third is the gini_index, that measures the pollutantance of each attribute in the data set. Forth is accuracy, that gives the highest chance of accuracy for the created tree [4]. The other parameter the team looked into was the minimal_gain. This parameter is in charge of splitting the nodes (data columns) into a tree. If the minimal gain is higher, it will cause the tree to have a smaller number of splits and will result in a smaller tree. Lower the gain, bigger the decision tree, that results in more broken down decision making [4]. The team experimented on these values to see which combination will cause the most accurate results. *Figure 7* below shows the default parameter settings for the ID3 decision tree.



ID3

criterion: gain_ratio

minimal size for split: 4

minimal leaf size: 2

minimal gain: 0.1

Figure 7: ID3 default parameters.

The team changed the criterion to "accuracy" and the minimal gain to 0.9. The team then ran the model with the new changes and got vastly different and more improved results. The model now returned 13 true alarms and 107 false alarms. See *Chart 2* below comparing the two models.

Test	Criterion	Minimal gain	False Alarms	True Alarms
1	Gain_ratio	0.1	43	77
2	accuracy	0.9	107	13

Chart 2: Comparison between model when run with different parameters

The team then looked at the decision tree available for the model. Since the ID3 algorithm is based on a predictive decision tree model, RapidMiner will show how it makes its decisions based on the training data set entered to learn from. RapidMiner first makes its decision based on which component is accessed. And then goes down the tree into the other columns as shown in *Figure 8* below.

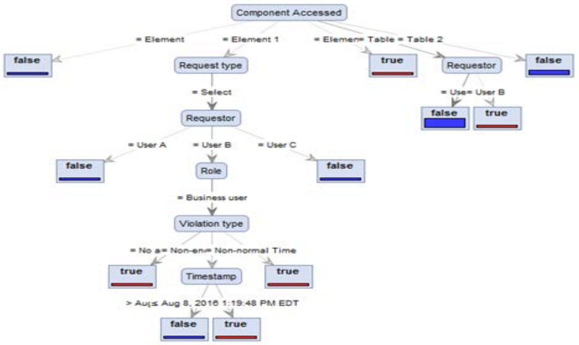
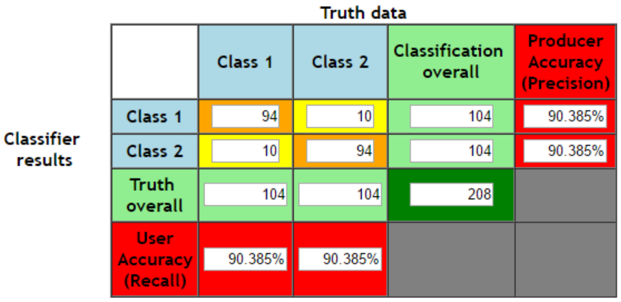


Figure 8: RapidMiner decision tree based on Test Two

In regards to the data provided, RapidMiner revealed that the true and false positive predictions had an accuracy of 90.39%. Through the use of a confusion matrix shown in *Chart 3* below, 94 false positive results and the 10 true positive results were able to provide the generated output.



		Truth data			
		Class 1	Class 2	Classification overall	Producer Accuracy (Precision)
Classifier results	Class 1	94	10	104	90.385%
	Class 2	10	94	104	90.385%
	Truth overall	104	104	208	
	User Accuracy (Recall)	90.385%	90.385%		

Chart 3: Confusion Matrix of Classifier Results vs. Truth Data [17]

C. Test 3

While RapidMiners new results were much more accurate based on the new parameters, it is still not 100% accurate based on our validation data set. Since the team had increased the gain ratio to its max number, the next step was to adjust the models decisions in order to get a more accurate result. We can change RapidMiner's root node to look at a different column first before the program determines decisions. For example, RapidMiner can look at the requestor first instead of the component accessed. The team decided to remove certain columns for the test data set, to see if RapidMiners results could become more accurate. The thought process here is, if the team removes the Time Stamp column because it is not leading any significance into determining true or false alarms, maybe Rapid Mienr will give the team a higher accuracy than 90.35% obtained in test two. Throughout testing, obtaining this result derived unsuccessful and the team came to the conclusion that including all data columns would provide the most accurate results.

VII. PROJECT LIMITATIONS

One major limitation is the data. It is extremely difficult to find a company or organization that is comfortable supplying an academic institution with real world data for students to study. Security logs usually contain sensitive data to its organization and customers and are not willing to take the risk of confidentiality with students. This will hinge the authenticity of the teams research and analysis, but with the right knowledge of similar real world data, the team hopes be able to get as close to actual results as possible.

VIII. CONCLUSION

In conclusion, the Big Data False Alarms team was unsuccessful in 100% accuracy for determine true and false alarms based on security logs. However, the team was able to prove that through analysis, data mining programs such as RapidMiner, and an ID3 decision algorithm, security analysts can greatly reduce the number of false positives coming through the pipelines. Throughout testing, the team was able to produce 90.39% accuracy when given a data set of over 120 alarms. This project should be continued next semester with the goal of further increasing the capabilities of using RapidMiner and the use of the ID3 decision tree. With more exposure and testing through real life security data sets, the teams can gain more experience in judging how to manipulate the program into lowering the amount of false positives.

IX. FUTURE WORKS

If this project was continued, there are numerous directions the team could take. One route is to explore an extensions algorithm of ID3 called C4.5. The C.45 algorithm can handle both steady and various attributes. [12]. This could result in more accurate results based on the sample data sets. Another step that can be taken is to acquire real life security logs. If a company or institution would be willing to give the team a sample data set, it will allow much more accurate results from experimentation and will greatly assist in the quality of work that the team is doing.

Another feature that could be explored, if committed to the ID3 algorithm, is developing an automated interface to the Detection Engine to provide the findings of the ID3 model. The Detection Engine is a component of the DLP process and raises an alarm when it recognizes an access role being violated. If a team can fine tune this engine to receive automatic feedback information based on confirmed true and false alarms, this could help reduce noisy volume for security analysts.

X. REFERENCES

[1] Alneyadi, S., Sithirasenan, E., & Muthukkumarasamy, V. (2015, 20-22 Aug. 2015). Detecting Data Semantic: A Data Leakage Prevention Approach. Paper presented at the Trustcom/BigDataSE/ISPA, 2015 IEEE.

[2] Buckridge, Jessica, Finklestein Ezra, Hasanramaj, Marlon, Vasseur, Pedro (2016). Improving Data Leakage Detection and Prevention Solutions by Reducing False Positives in Security Logs, 2016, Seidenberg School of CSIS, Pace University, Pleasantville, New York

[3] Euzenat, Jerome. Ontology Matching. Springer-Verlag Berlin Heidelberg, 2007, p. 36

[4] GmbH, RapidMiner. "ID3 (RapidMiner Studio Core)." ID3 - RapidMiner Documentation. RapidMiner, n.d. Web. 04 Dec. 2016.

[5] Guido, M. D., & Brooks, M. W. (2013, 7-10 Jan. 2013). Insider Threat Program Best Practices. Paper presented at the System Sciences (HICSS), 2013 46th Hawaii International Conference.

[6] IndustryEnergy, By. "Data Protection: Data In Transit vs. Data At Rest." Digital Guardian. Digital Guardian, 13 Oct. 2016. Web. 08 Nov. 2016.

[7] Jazzar, M., A.B. Jantan, Using fuzzy cognitive maps to reduce false alerts in some-based intrusion detection sensors, in: Proceeding of the Second Asia International Conference on Modelling & Simulation, 2008.

[8] Peng, W., J. Chen, & H. Zhou, An Implementation of ID3 Decision Tree Learning Algorithm, University of New South Wales, School of Computer Science and Engineering, Sydney, Australia, 20p.

[9] Petkovic, M., Popovic, M., Basicevic, I., & Saric, D. (2012, 11-13 April 2012). A Host Based Method for Data Leak Protection by Tracking Sensitive Data Flow. Paper presented at the Engineering of Computer Based Systems (ECBS), 2012 IEEE 19th International Conference and Workshops.

[10] Protection of sensitive data from malicious e-mail, by C. Alexander and C. Nachenberg. (2009, Nov 10). US 7617532 B1 [Online]. Available: https://www.google.com/patents/US7617532

[11] RapidMiner, "ID3 - RapidMiner Documentation," 2016. [Online]. Available: http://docs.rapidminer.com/studio/operators/modeling/predictive/trees/id3.html. [Accessed 10 November 2016].

[12] Singh, Sonia, and Priyanka Gupta. "Comparative Study ID3, CART, C4.5 Decision Tree Algorithm: A Survey." International Journal of Advanced Information Science and Technology (IJAIST) 27.27 (2014): n. pag. University of Delhi, Department of Computer Science, July 2014. Web. 04 Dec. 2016. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.685.4929&rep=rep1&type=pdf>.

[13] Shabtai, A., Y. Elovici, and L. Rokach, A survey of data leakage detection and prevention solutions. Springer Briefs in Computer Science, Springer, 2012.

[14] Shapira, Y., B. Shapira, & A. Shabtai, Content-based data leakage detection using extended fingerprinting, Ben-Gurion University of the Negev, Israel, 2013, 12p.

[15] Timm, K., Strategies to reduce false positives and false negatives in NIDS, Security Focus Article, available online at: http:// www.securityfocus.com/infocus/1463, 2009.

[16] Tosun, A., & Bener, A. (2009, 15-16 Oct. 2009). Reducing false alarms in software defect prediction by decision threshold optimization. Paper presented at the

[17] Vanetti, Marco. "Confusion Matrix Online Calculator." Confusion Matrix Online Calculator. N.p., 2007. Web. 01 Dec. 2016. <http://www.marcovanetti.com/pages/cfmatrix/>.

[18] Xiaokui, S., Danfeng, Y., & Bertino, E. (2015). Privacy Preserving Detection of Sensitive Data Exposure. IEEE Transactions on Information Forensics and Security, 10(5), 1092-1103. doi:10.1109/TIFS.2015.23983633rd International Symposium on Empirical Software Engineering and Measurement, 2009. ESEM 2009.