ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 6, Issue 11 , November 2019

Data Leakage Detection and Prevention Solutions

ObidjonBozorov, DadakhonSharofov

National University of Uzbekistan named after MirzoUlugbek,City Tashkent, Uzbekistan
National University of Uzbekistan named after MirzoUlugbek,City Tashkent, Uzbekistan

ABSTRACT: This paper will examine various methods of reducing false positives in the security logs of data leak detection/data loss prevention applications to determine the best method for weeding out false positives, and prevent them from creating noisy security logs. This paper will also examine specific security logs using big data analytics methods to determine what circumstances are likely to trigger false alarms, and help determine which attributes may help to reduce their number. By identifying common triggers of false alarms as well as the most effective methods for reducing them, we hope to come up with recommendations for creating more effective security logs that can be more closely analyzed to detect leaks.

KEY WORDS:confidentiality, data analysis, big data, data leak detection, false positives, security logs,data loss prevention.

I.INTRODUCTION

The security of sensitive and confidential information is a serious concern for most organizations operating in today's world, where so much of this sensitive information is stored, transferred, and utilized digitally. Data breaches have become a fairly frequent occurrence, and the damage that they cause when they occur can be severe, from loss of status and reputation to violation of regulatory standards to enormous financial losses. Data Loss Prevention and Data Leak Detection methods have therefore become crucial to maintaining information security, and preventing these data breaches. As these technologies are still being developed, honed, and perfected, they do not always provide the necessary 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 efficacy 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, we want to examine the issue of false positives from two perspectives. We 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. If we are able to identify commonalities in false alarms, we may be able to instruct the application detecting leaks in how to do so more effectively. We also want to consider the best practices for configuring these applications, as well as the policies and procedures being employed in the environment generating the incidents that are being logged, in an effort to determine if preventative controls would be helpful at reducing the overall number of incidents logged, thereby also reducing the number of false alarms.

It is also 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. 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; false alarms triggered by data in traffic may require a more careful examination of the rules that govern incidents, and modification to make them more specific.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 6, Issue 11 , November 2019

II. RELATED WORKS

In his article *Strategies to Reduce False Positives and False Negatives in NIDS*, Timm describes the design of NIDS systems as one of three models. Signature-Based models are the simplest and most common [9]. These 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 happens when a user references an attack or includes text that is part of a known attack signature. Anomaly-Based systems 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 something passes through as a false negative but 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*, Mahmoud builds on this idea of anomaly-based systems by including other factors to estimate the abnormality of an individual packet [4]. 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 [3]. We have singled these out because of the way in which these practices could positively impact the number of false positives received by DLP/DLD applications by preventing incidents that might trigger those false alarms.

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, we need to base the solution on the context in which data is accessed. For example, someone who goes to work in an office uses his own iPhone to check corporate email and then downloads a PDF to look at for later. Sounds like a simple process, 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?

With this in mind, we can see why it is necessary to have a plan in place to guard against data loss, and why there are so many specialized applications that have been designed to address this need. 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.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 6, Issue 11 , November 2019

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.

When designing a DLP solution we need to understand how it's going to 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, then 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.

B.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 overall security goals of the organization. Some organizations may have large databases to store customer information, which might include personal information like 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 you're looking to protect, DLP applications can employ different methods to detect it. For example, there are methods that rely on detecting leaks within the content of the data, 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 xxxxxxxxxxxxxxxx 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, they 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, but rather being 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. [9]

There are certain additional steps that can be taken to improve the efficacy of your 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. [7] 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. [6] 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].

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 6, Issue 11 , November 2019

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.

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 rules regarding user behaviors that might be found in an acceptable use policy 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, you 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, so 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, we will analyze the provided security logs to identify traits that trigger false alarms, and determine the best methods for reducing the number of false alarms present in the logs.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 6, Issue 11 , November 2019

IV.DATA ANALYSIS

A. False Positives

The goal of our research is to determine a way in which the number of false positives found in security logs can be reduced to provide analysts with more effective data, and allow for the more timely analysis of potential threats. The term "false positive" describes an instance where the monitoring system in place – a Data Leak Detection application or Intrusion Detection System, for example – triggers an alert that malicious activity is present, but the traffic or behavior in question is not actually malicious. These false positives are logged alongside genuine threats, creating noisy data that takes longer to analyze, increasing the amount of time that it takes to detect and mitigate genuine malicious activity.

At the same time, it is also important to consider the risks inherent in attempting to remove false positives from the data. We do not want the reduction of false positives to increase the instances of false negatives, or legitimate alarms that were not detected by the application. These represent risks to the organization that can be much more serious than the inconvenience of noisy data, or the time it takes to investigate a false alarm.

Organizing and dealing with recorded logs and alerts generated by security sensors like the DLD applications, IDS, firewalls, and servers are not a simple job. Since these sensors are independent, alerts are sent to an analyst party to analyze these alerts in order to determine whether a data leak occurred, and what data may have been lost. It is important to understand the various actions or behaviors that trigger these alerts in order to use the necessary tools, methods, and techniques that will allow us to reduce the false alerts rate and increase the detection rate. It is important that in looking to reduce false positives, we do not increase the risk of a legitimate threat bypassing the detection method.

There are two different ways we can study false alerts reduction. One way is through studying and reducing it at the sensor level, that is, to prevent the application from detecting it as an alarm at all or alternatively, removing it after it's been detected on the logs. In the paper by O. A. Abouabdalla, one person proposed a solution to the problem by using fuzzy cognitive maps (FCM), which has computing modeling techniques generated from the compensation of fuzzy logic and neural network. He measures availability, similarity, occurrence, relevancy, independent, and correlation factors, which he then assigns an effect value for each one of the factors to estimate the total degree of maliciousness per packet. We measure the effect/influence value between 0 and 1, where 0 means normal relation and 1 means high relation.

There are three types of agents for the data mining techniques (clustering, association rules, and sequential association rules). The clustering based agents extract properties from traffic in terms of frames and tries to make the normal traffic in the training stage. If the unknown traffic is far from the normal cluster it is considered an attack. The association rule-based agent captures the rule of selected features and in the detection phase, the agents count the rules of each connection to be matched and if the frequency is less than the threshold, it is classified as an attack. The sequential association rule-based agents capture the sequential patterns in network traffic to assist the association mining process. In the detection phase the agents tests the abnormal connections with the packet/time frame. If it is larger than the threshold, it will be declared as an attack. In the decision making stage, the alert is evaluated to see if it is generated from both clustering based and rule based, else it is considered a false alert from one side, which ends up eliminating the other side.

Another way to reduce the false alerts rate is by classifying the alerts into two classes, continuous and discontinuous where the continuous patterns represent a real threat and the discontinuous patterns show the sequences mixed with noisy data. Reducing the false alerts here will be after denoting all alerts in one sequence Xi by the length m, Xi then will be expanded to a number of sequential patterns that are generated by extracting all possible combinations. Other approaches of reducing false alerts are based on data mining methods that provide automatic intrusion detection capabilities by mining knowledge from audit data to sort out acceptable behavior from malicious behavior.

B. Data Analysis Techniques

The first data analysis technique we plan to apply to the security logs will be clustering the traffic,both normal and anomalous. Cluster analysis is a type of data analysis that looks at the similarities in a given set of data and groups the data based on those similarities. We plan to use numerous clustering techniques to try to separate normal and abnormal traffic, as well as legitimate attacks and false positives.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 6, Issue 11 , November 2019

If all normal traffic can be grouped in a reasonable cluster, anything outside of that group can be identified as an intrusion. If this method is still resulting in a large number of false positives, we can also see if the false positives can be grouped into a reasonable cluster. If there are noticeable traits or triggers that the false positives all have in common that is not present in normal traffic, and is not present in legitimate alerts, it may be possible to cluster the false positives, and indicate that items in that cluster may be safely disregarded. If the false positives do not have enough attributes that distinguish them from legitimate alerts, this method may not be effective at weeding them out.

The alternative clustering approach will be to group all known instances of actual attacks and label everything outside of the cluster as normal traffic. This will likely produce much fewer false positives at the cost of more false negatives. Therefore, this is likely a less effective approach, as we do not want to increase the number of incidents that evade detection in our effort to reduce the false positives in the logs.

After identifying each cluster, we will adjust the input signals to see if a better fit can be found and therefore reduce the number of false positives by more accurately identifying the groups. The more effectively we are able to group traffic, the better chance we have of identifying and removing false positives without allowing attacks.

Another technique we plan to use is the ID3 decision tree as a way to tell which attributes are most telling when trying to separate legitimate and illegitimate traffic. ID3 is an algorithm that is used to create a decision tree, which is a model made up of a root node, branch nodes, and leaf nodes meant to indicate decisions. Similar to the clustering, decision trees will be trained on past logs and then tested on a separate portion to see their effectiveness in reducing false positives. Utilizing the ID3 algorithm will allow us to identify which traits or attributes of the data present in the security logs are most useful in grouping and identifying different types of traffic, including the false alarms. In this way, we hope to find better ways to identify false alarms and remove them from the data in order to generate less noisy data.

V.TESTING DATA ANALYSIS METHODS

The data that we will be analyzing for our research is the output of a Data Loss Prevention application. The output is in the form of an activity log which displays information security alerts for file uploads to a restricted access server. This data is therefore an example of data in motion, as the data that we are examining generated an alarm while in transit over the network.

These alerts occur over the span of about four hours during normal business hours, and recorded 352 incidents during that time period. Of these 352 incidents, 40 were later determined to have been false alarms. Our goal in analyzing this data is to reduce the number of false alarms from 40, or 11% to less than 5%, which we believe would be a more reasonable rate of false positives.

A. Data Analysis Using Clustering

Our first data analysis method that we will be applying to the security logs will be clustering. In order to cluster the data, we needed to first look at the various attributes recorded in the logs. For each log, there is a record of the date and time, the username of the user uploading the file, the file that they were uploading, the read/write permissions of the file, the recipient of the file, and the full target UID path of the file.

Because the date and time of the instances of false positives were distributed throughout the four hour window, and because all of these instances occurred during normal business hours, the date and time information for the incidents did not appear to be an immediately useful attribute for separating traffic for clustering purposes.

The 352 recorded incidents in this sampling included alarms triggered by twenty-three different users. However, we 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. With this in mind, we separated the users into two groups, those generating false positives, and those generating only legitimate alarms in order to analyze them further to determine whether the users could be flagged and their alarms disregarded, or whether we would need to take additional attributes into account in order to provide the most accurate results.

The three users who generated false alarms also generated legitimate alarms, so clustering by user alone and flagging incidents by those users as false positives would mean the application would not be able to detect legitimate incidents triggered by these users. Therefore, it was necessary to examine further attributes of the alarms triggered by these three users.

Of the three users, User 1 was responsible for twenty six instances of false positives, more than half of the total false positives contained in the data; being able to weed out this user's false alarms would have a very notable impact on the overall percentage of false alarms. User 1 also generated one legitimate alarm. While the legitimate alarm did differ in

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 6, Issue 11 , November 2019

file type from the false alarms, that did not necessarily appear to be a reliable attribute to use in order to cluster this data, as the file type was found in both legitimate and false alarms of other users.

What we did find to be unique and likely a more reliable attribute 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 was 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 drops from 11% to 3.9%.

User 2 presented a more challenging data set. Unlike User 1, the alarms generated by User 2 were all of the same file type, so we were unable to separate the data based on that attribute. Additionally, the file names and locations of the uploads were all remarkably similar, and did not offer a logical cluster. Finally, the target UID path for this user was shared by both the false alarms, and the legitimate alarms, making it impossible for us to use it to differentiate between the two. Clustering did not offer us a logical way to separate the 12 false alarms triggered by this user, and flagging the user would mean allowing the 4 legitimate alarms generated to go undetected.

User 3 presented a similar situation to User 1; though the user generated both false and legitimate alarms, the false positives had a unique target UID path that we were able to use to cluster those instances. With that in mind, we were able to add that target UID path to our cluster of false positive data, further reducing the number of false positives by 2, dropping it down to 3.4%.

B. Data Analysis Using Decision Tree

The second data analysis method that we will be applying to the security logs will be the ID3 decision tree algorithm. As described above, the attributes present in the logs include the date and time of the file upload, the user uploading the file as well as the recipient, the file name, permissions, and target path. We will identify which of these attributes are the most important for the purposes of separating legitimate alarms from false alarms, and then implement the decision tree accordingly to determine how those attributes will help us reduce the number of false positives.

The attributes that we chose to test were the user, elapsed time from start to finish of file upload, and the number of changes to ownership/group permission. We found that when all attributes were included, the decision tree had one node and sorted the logs as true or false alarms simply based on the user ID. If the entry was created by either User 1 or 2, the alarms were classified as false and all others were classified as true. (Fig. 1)

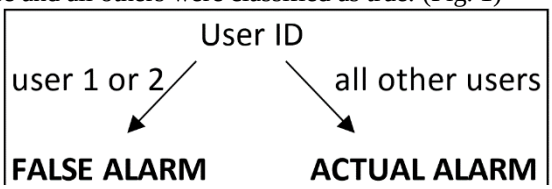

Fig. 1. Initial decision tree analysis utilizing all three chosen attributes: UserID, time elapsed during file upload, and number of changes to permissions, which generates a tree with a single node.

To gain further insight into the data we chose to eliminate the user id and run the decision tree based on the other two attributes. This also allowed us to look at how the decision tree might work on additional data sets where there are not a handful of problematic users, but rather false alarms are spread out among users with legitimate alarms as well.

A pattern emerged where any file upload that took more than one second was 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. 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 somewhat effective however still misidentified some cases. (Fig. 2)

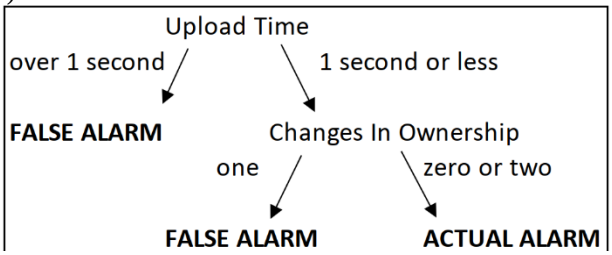

Fig. 2. Subsequent decision tree analysis wherein User ID was removed as an attribute, and instead analysis efforts focused on time elapsed during file upload, and number of changes to permissions, which generated a tree with more than one node.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 6, Issue 11 , November 2019

VI.EVALUATION OF EFFECTIVENESS OF DATA ANALYSIS METHODS

A. Effectiveness of Clustering Method

After analyzing the security logs through the clustering method, we have found that it was fairly effective at reducing the number of false positives in the logs. The initial rate of false positives in the sample data that we were working with was 11%. By clustering problematic users, and then zeroing in on the unique target UID paths of the false positives, we were able to reduce that rate to 3.4%, a much more reasonable false positive rate.

With that being said, this method cannot be used to predict false positives in other environments, and would not necessarily be sufficient to reduce false positives comprehensively in the environment in which it was being studied. The data we analyzed represented a portion of the logs generated by a single application; in order to provide more comprehensive reduction in false alarms for the organization which employed this application, it would be necessary to analyze more of their traffic to determine if there are other problematic users, and if they fit the same pattern and would be able to be clustered with the problematic users and target paths that we were able to identify in our sample.

If were able to identify all problematic users, or all problematic target paths which are generating these false alarms, it would be possible to whitelist those users, allowing the alarms they generate to be safely disregarded, and ultimately offering a sizeable reduction in the number of false positives being returned in the logs, allowing for more effective analysis of the remaining data, and more timely mitigation of legitimate threats.

While the specific parameters of the clustering would not be able to be applied to a different network or environment, the underlying method of clustering problematic users after they have been identified may be useful as a method for reducing false positives in additional situations.

B. Effectiveness of Decision Tree Method

After analyzing the security logs through the decision tree method, we have found that our initial decision tree, depicted in figure 1, had an effectiveness of 92% and only misclassified 2 entries on our training set. When using the user only decision tree to classify our test data, we were able to correctly identify 100% of false alarms because all false alarms were created by one of the same two users from our training data. It reinforced our findings from clustering that user was the most effective way to separate true and false alarms when the users are known. Once again, the issue remains of transferability to other networks and data sets outside of this local network.

The second decision tree, depicted in figure 2, which had a root node based on elapsed upload time was able to accurately identify 100% of long uploads as false alarms and reduce the number of false positives by nearly 20%. This seems to be a result of small sample size because when tested on a new dataset, the rule incorrectly identified two actual alarms as false. Overall, the second decision tree had a success rate of 62% at classifying alarms as true or false when tested on a portion of our training set. When used to classify our test data set, the tree correctly labeled only 53% of the instances. This is too low to say that anything useful can be gained by using the tested attributes when labeling false alarms.

VII.PROJECT LIMITATIONS

There are a variety of factors that presented limitations for what we were able to analyze with this project. The majority of these limitations stem from the fact that the data sets which were made available to us had been anonymized, which presented a number of obstacles.

The first obstacle presented by the data sets is that we were not able to discern what application had generated the logs that we were working with, or what method the application was using in order to detect potential data leaks. As we had discussed earlier in this paper, there are variety of methods for detecting the loss of sensitive data, through pattern and keyword matching, document printing, and other methods. We also highlighted the strengths and weaknesses of these various methods, as well as algorithms that it may be possible to apply in order to help maximize the efficiency of the DLP application. Because we did not have information regarding which application we were using and what methods it employed, it is impossible to say whether it may be possible to eliminate some instances of false alarms by improving the efficacy of the detection method.

The second obstacle presented by the anonymization of the data set is the fact that we were not privy to what specific content in these files triggered the alarm, or if the alarm was triggered by content at all, or instead by a user behavior. If the alarm was triggered by specific content in the file, it would be possible to examine whether there were any

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 6, Issue 11 , November 2019

commonalities between the content in the false alarms – common keywords or phrases for example – and whether those commonalities were shared with any of the true alarms. This would have allowed us to determine whether the content shared any commonalities that could be whitelisted in order to reduce the number of false alarms being triggered. If instead the alarm was being triggered by a certain user behavior, it would be worth examining the permissions and access rights of that user to determine if there was a way to restrict that behavior in the future, in order to prevent them from triggering additional alarms.

With this in mind, there are still many opportunities for further research to be done on the topic to provide additional options for reducing false positives in DLP application generated security logs, and increasing the overall efficacy of these applications.

VIII.CONCLUSION

After identifying the various methods employed by Data Leak Detection and Data Loss Prevention applications, and performing data analysis on samples of the logs generated by the applications, we seemed to find moderate success on a the localized data by being able to group the problematic users. This was accomplished by both the decision tree and the clustering technique.

While clustering could provide useful information for monitoring a single network or analyzing a limited number of attributes it requires knowing what trends you are looking for before performing the clustering by choosing the attributes and number of possible clusters. Analysis through clustering was 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, and other networks may not necessarily have easily clustered data.

The decision tree algorithm, on the other hand, is likely more applicable to multiple environments, as it was able to take into consideration various attributes of the data besides the usernames which may be present in other data sets as well, whereas the usernames utilized in the clustering analysis are unique to this specific dataset and would therefore not apply to other environments. It allows the consideration of more attributes at a single time and how the value of one attribute might affect another.

Because of the limitations previously discussed, there are also various aspects of the data what we were not able to incorporate into our analysis, leaving many options for additional things to consider in future research.

REFERENCES

[1] Radwan R. Tahboub, Yousef Saleh. Data Leakage/Loss Prevention Systems (DLP) NNGT Journal: International Journal of Information Systems. Volume 1, 2014. –P. 13-19.

[2] Juraev G., Bozorov O. Problems and solutions to protect confidential information in the Republic of Uzbekistan in the "Electronic Government" // Tashkent, April 12, 2019. 108-111 p.

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

[4] 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

[5] International Conference.

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

[7] 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.

[8] 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.

[9] 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

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

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

[12] 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.

[13] *Xiaokui, S., Danfeng, Y., & Bertino, E. (2015). PrivacyPreserving Detection of Sensitive Data Exposure.IEEE Transactions on Information Forensics and Security, 10(5), 1092-1103. doi:10.1109/TIFS.2015.2398363*