MLDED: Multi-Layer Data Exfiltration Detection System

Mohammad Ahmad Abu Allawi, Ali Hadi, Arafat Awajan
Computer Science Dept.
Princess Sumaya University for Technology (PSUT)
Amman, Jordan
mohammad.allawi@gmail.com, {a.hadi, awajan}@psut.edu.jo

***Abstract*— Due to the growing advancement of crime ware services, the computer and network security becomes a crucial issue. Detecting sensitive data exfiltration is a principal component of each information protection strategy. In this research, a Multi-Level Data Exfiltration Detection (MLDED) system that can handle different types of insider data leakage threats with staircase difficulty levels and their implications for the organization environment has been proposed, implemented and tested. The proposed system detects exfiltration of data outside an organization information system, where the main goal is to use the detection results of a MLDED system for digital forensic purposes. MLDED system consists of three major levels Hashing, Keywords Extraction and Labeling. However, it is considered only for certain type of documents such as plain ASCII text and PDF files. In response to the challenging issue of identifying insider threats, a forensic readiness data exfiltration system is designed that is capable of detecting and identifying sensitive information leaks. The results show that the proposed system has an overall detection accuracy of 98.93%.**

Keywords—Data Leakage; Data Exfiltration; Data Breach, Data Hiding, Data Theft, Data Loss

I. INTRODUCTION

The increasing growth of the network systems information exchange along with the high advances of information technologies in everyday life, and the increasing in the complexity of attacks regardless of the enhanced security measures make the need for network security of crucial essence [1].

Data Exfiltration can be defined as the "Unauthorized extraction of information or data from a host"[2]. The essential importance of data as enterprise assets is leading to an increasing concern and a serious growing need for systems that detect, prevent, and mitigate such violations. Since a boost in insider threats has been recorded specifically in incidents of data exfiltration, data exfiltration becomes a principal part of each network security system.

Data Exfiltration happened in most cases as a sequence of an intrusion on a victim machine. The intrusion can range from a complex snippet of code written by the attacker or by an organized team of intruders to guiltless unauthorized login by a close friend [3]

Cybercrime nature has been transformed to be more robust, automated and designed with high difficulty levels. A variety of security models and algorithms are proposed, developed and implemented, but they do not reach to a level they are fit to detect this complicated level of attacks. Furthermore, these crime ware services represented by organization sensitive data leakage and exfiltration are recently sold as a principal part of a growing underground economy. This underground economy has provided an organized market where more complicated crime wares raised and created [4].

Therefore, in this research a Multi-Layer Data Exfiltration Detection (MLDED) system has been proposed based on multi-level detection framework that can handle the security breaches from a plain level to the sophisticated one. Each level deals with a specific complexity degree of data exfiltration attack. MLDED is designed to be for forensic purposes in addition to its high capability in data leakage detection.

II. LITERATURE REVIEW.

Data-exfiltration systems are designed in many techniques and using different algorithms depending on the scenario and application under study.

Shu et al. suggested Data Leakage Detection DLD Algorithm called fuzzy fingerprint, which can be used to detect the unauthorized data leakage or extraction due to haphazard human errors or application imperfections with emphasis on the Privacy-Preserving feature of sensitive data.

Kemerlis, et.al. 2010 built a system called iLeak that composed of three major components: Uaudites, Inspectors, and Trail Gateway. Where Uaudites is an agent installed on client's computer that monitors the activities, these components are OS tools that could be a limitation of the system.

Liu, et al. (2009) developed a systematic multi-level algorithm called Sensitive Information Dissemination Detection (SIDD) system. SIDD consists of three major building blocks: Identification of network level application, Content signature generation and detection, and Covert communication detection. SIDD system depended on the idea of con-tent signature matching. This technique is fruit yielding just in cases when the whole monitored content is identical to the media sensitive data

Al-Bataineh and White (2012) proposed a novel technique to analyze and detect the malicious data exfiltration in web traffic. The pro-posed technique analyze the data leakage exfiltration behaviors of one of the most well-known data stealing botnets; Zeus, where the attempts to malicious data stealing is identified by a classification algorithm.

Although most of proposed data exfiltration techniques de-pend on the idea of whole content matching or fuzzy hashing, they exhibits a shortage in detection of the partial sensitive data exfiltration or the data leakage attacks that designed to steal data from a sensitive database of considerable sizes

III. PROPOSED SYSTEM

MLDED System consists of two main phases: Pre-processing phase and Leakage detection phase where each phase works in cascading manner, namely, one cannot proceed to the next phase unless the duty of the previous one is performed. Each phase consists of building blocks that may work in both modes: parallel or sequentially depending on the involved phase.

Leakage detection phase composed of three major detection levels; Hashing, Keyword Extraction and labeling levels. Each detection level is designed to handle a specific difficulty level of data leakage attack. Fig. 1. depicts the MLDED system prototype.

978-1-4673-8499-5/15 $31.00 © 2015 IEEE
DOI 10.1109/CyberSec.2015.29

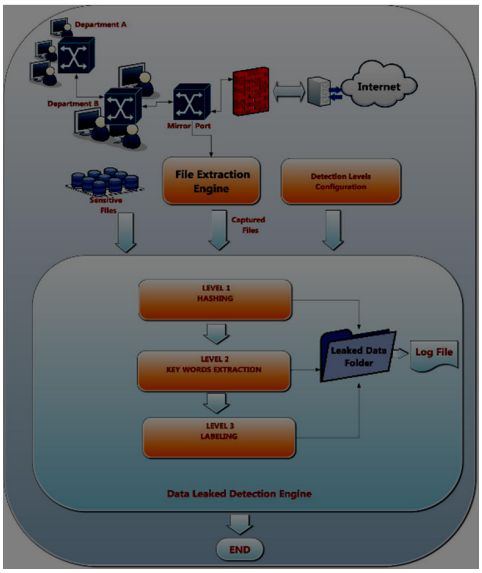

Fig. 1. MLDED System Block Diagram

A. *Pre-Processing Phase*

The data exfiltration pre-processing phase of MLDED system proceeds in two major stages: 1) File Extraction Engine 2) Detection Engine Levels Configuration stage. In this research Bro-IDS was used to extract pdf, HTML and txt files from the network traffic, a custom Bro-IDS script was written to do this task. Every level of MLDED must be configured before executing MLDED system. Thus, MLDED system requires three inputs: Captured Folder, Sensitive folder, and levels configuration

B. *Data Leakage Detection Phase*

The MLDED system consists of three basic detection levels. These levels detect Positive Leaked files and Negative leaked files.

Positive Leaked files are the files that detected by the detection level and considered leaked ones, whereas the negative leaked files are the files that considered normal (not leaked) with respect to the detection level. The negative leaked files of one level are considered the input to the next level. Thus, the levels are divided depending on the difficulty level of the designed attack.

1) *Level (1): Hashing*

This level is considered as the first detection line of the MLDED system against sensitive data leakage. The main building blocks of the first detection level of MLDED system are the hash function, hash value-comparing module.

A hash function usually means a function that does compressing process, which means the output of this function is shorter than the input [5]. Moreover, hash function can be considered as a deterministic function that always gives the same value for an object. The hash code does not change depending on where the object is stored.

In this research, the object is the files in both captured and sensitive folders. Therefore, each file in the extracted folder and the sensitive folder has a unique hash value. This hash value used to compare the files that captured to that exist in the sensitive folder, if the hash values are the same, then that sensitive file is leaked by an insider in the target organization. Otherwise, the file will be stored in the negative leaked files of the level (1) to be used as input for the next detection level, namely, level (2). In this research SHA1 was used due to its simplicity

2) *Level (2): Keyword Extraction*

This level used more complicated filter in order to enhance the capability of this level to detect the sensitive data exfiltration attacks that designed in a higher difficulty level. The negative leaked files that does not exceed the data leakage threshold of level (1) is transferred as input to this level in association with the sensitive folder of the target organization.A process of re-filtration according to different leakage threshold is conducted. MLDED system consists of three major building blocks: Common English words removal, Regular Expressions (regex) functions, and Regex Comparing module

Common English words removal module removes the common English words that have the highest frequency in documents that written in English language and this is conducted for both sensitive folder and for the negative leaked files coming from level (1).

The average sentence length in plain English language is (15-20) words in average [7]. Therefore, the clean negative leaked files that will be filtered depending on the count of words that contained in each file in such a way, each clean file contained less than seven words is excluded from the rest of Level (2) processing steps which leads to low false positive rates. As an instance of this case, if the target organization is a hospital, then, as a natural result, a high percentage of sensitive files of the hospital will contain "Patient Record Number ". Let, for example, one of the clean files contains only "Patient Record Number: 125236 ", then this file will be detected as leaked file. This example demonstrates one of the frequent cases that increase the False Positive Rate (FPR) of the system, which leads to overall system performance degradation.

The second major module of Level (2) is the Regular Expressions or regex (sometimes-called regexp) which is a sequence of characters that forms a search pattern, mainly for use in pattern matching with strings [8]. Regular expressions module of level (2) consists of a set of functions that extract pre-specified patterns (keywords) that found in the negative leaked files of level (1).

The regular expressions that applied at this level can be classified into two main categories: Static Regex and Dynamic Regex. Static regular expressions are established using the sensitive keywords that common for all organizations while the dynamic regular expressions are established based on the keywords that are specific for a target organization depends on the nature of target enterprise or business, thus it different from one organization to another.

Level (2) Mechanism

Firstly, the negative leaked data that come out of the level (1) that is considered in addition to sensitive folder as the inputs of level (2) are processed to remove the common words that repeated frequently in the English language. This will result in clean versions of both sensitive folder and the negative leaked data folder, which leads to overall enhancement in the process of leakage detection at the level (2) and less computation time. Then, these clean folders will be as inputs to regular expressions functions.

These functions search for a pattern, and then, extract the string contents of files from both folders individually based on pattern matching with file strings. This yields two types of strings depending on the source file: sensitive strings and negative leaked file strings.

Then, string comparing module will compare the sensitive strings and the negative leaked strings to find the similar strings and compute the similarity percentage between them, which will determine the positive leaked data, if exceeded the threshold. Otherwise will be specified as negative leaked data that will pass to level (3). The similarity percentage can be computed as illustrated in the following equations:

Let C_{Si} : $Content\ of\ i^{th}\ Sensitive\ file$,

C_{Ej} : $Content\ of\ j^{th}\ Extracted(Captured\ file)$,

h : $hash\ value\ of\ a\ file$

H: $Hash\ Function$

$$R_{si} = REG(C_{si}), R_{Ej} = REG(C_{Ej})$$

$where\ , REG(C_{si}), and\ REG(C_{Ej})\ are\ the\ Regex\ functions\ and$

R_{si} *and* R_{Ej} *are the string outputs of Regex functions of the contents* C_{Si} *and* C_{Si} *respectively*.

$$l_{Si} = |R_{si}|; \textit{length of } R_{si}$$

$$l_{Ei} = |R_{Ej}|; \textit{length of } R_{Ej}$$

$$I_i = R_{si} \cap R_{Ej}$$

$$\lambda_i = \frac{|I_i|}{l_{Ei}}$$

λ_i : *represent the similarity percentage.*

3) Level (3): Labeling

A higher order filtration can be achieved at this level, such a way meets the high difficulty level of the designed data leakage attack that passed form level (2) to level (3) as negative leaked files. Level (3) built on the basis that every document created in the organization contains specific keywords that consists the backbone of the document, namely, it carries the most important information that best described the document. Therefore, a sensitive keywords file is established as pre-processing step for this phase, which consists an input of this level. Employees of a target organization particularly who deal and /or handle sensitive data in the organization are responsible for extracting specific keywords out of each file they created and then store it in a specific file called the sensitive keywords file. This file is filled with sensitive keywords and continue evolving along with the development of organization business.

This level is called "labeling" due to the fact that labels (sensitive keywords) will be used in the process of comparing which is considered a higher level of data-leakage sensing.

Level (3) Mechanism

In this level, each employee of the target organization who responsible for dealing, editing, and modifying sensitive files, will process these sensitive files in such a way that sifting the sensitive files to select the most sensitive keywords that signify it.

Therefore, for each sensitive file created, there is corresponding sensitive keywords that represent it. The process of files categorization into sensitive or non-sensitive ones depends on the file owner. In addition, selecting the most sensitive keywords of a specific file is highly dependable on the owner/creator of the file.

On the other hand, sensitive files for an organization may be not sensitive to other one. At the same time, the sensitivity of files may range from extremely sensitive to open for each employee in an organization. In our case, we restrict the sensitivity of file to be one of two states: sensitive or non-sensitive, which enhance the simplicity of the proposed model in terms of proof of concept.

The preliminary step in Level (3) is to prepare the sensitive keywords of each sensitive file, which is conducted by the owner of the file.

Once the sensitive keywords for each sensitive file is available, a searching for these sensitive keywords in the positive leaked files of level (2) is conducted. The similarity ratio of number of sensitive keywords that matched to the number of sensitive keywords that exist in the sensitive file will determine whether the file is leaked or it is normal. The similarity ratio is evaluated depending on the following equations:

Let L_{Si} : *Labels of* i^{th} *Sensitive file*,

C_{Ej} : *Content of* j^{th} *Extracted*(*Captured file*),

$$l_{Si} = |R_{si}|; \textit{length of } R_{si}$$

$$l_{Ei} = |R_{Ej}|; \textit{length of } R_{Ej}$$

$$\omega_{ij} = L_{si} \cap C_{Ej}$$

$$\zeta_{ij} = \frac{|\omega_{ij}|}{|L_{si}|}$$

ζ_{ij} : *represent the similarity percentage.*

Sensitive Keywords Feedback Mechanism

Since the process of sensitive keywords sifting is human dependable, adding a mechanism that controls this process is essential. Thus, a sensitive keywords feedback mechanism is suggested as a future work to raise the automation of MLDED system and to enhance its detection performance. The proposed feedback mechanism based on the idea of using the learning system techniques (algorithms) in the process of keywords sifting where the learning system will sift the sensitive keywords for each sensitive file and assigning weight value. This will result in a set of weighted sensitive keywords for each sensitive file. Then, a comparison process between the file-owner created sensitive keywords and the learning system created sensitive keywords is conducted in order to remove the repeated sensitive keywords. Net weighted sensitive keywords are ordered in ascending manner in sake of sifting the top (n) sensitive keywords that will be appended to the keywords originally saved in a special labels file which leads to dynamic updating of this file. Fig. 2. depicts the general framework of the proposed sensitive keywords feedback mechanism.

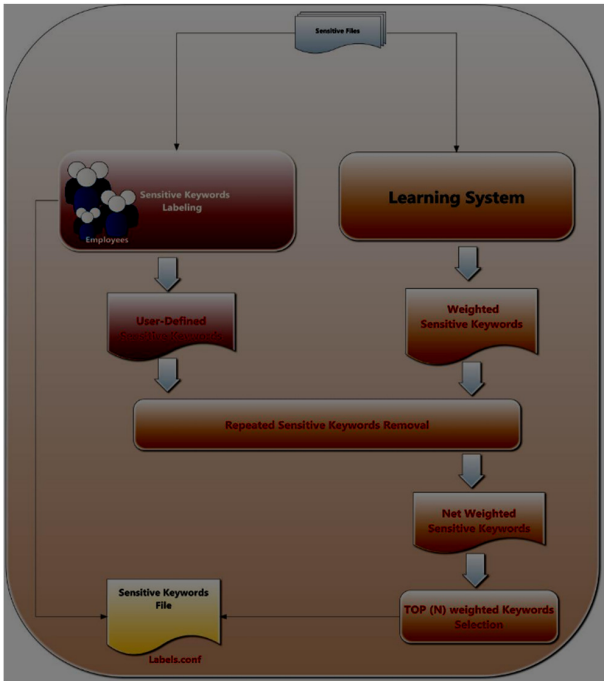

Fig. 2. Sensitive Keyword Feedback

IV. TESTING AND EXPERIMENTAL RESULTS

MLDED system is implemented using Python 2.7 as a programming language in PyCharm Community Edition 3.4 as a powerful integrated development environment and Linux Ubuntu 12.04 platform with Intel Core i5 2Due CPU 2.3 GHZ with RAM 10.0GB. MLDED has been officially approved and tested in a retail company located in Jordan called Izzat Marji Group.

The implemented MLDED system is tuned where each level was run as a separate unit to obtain the best level configuration, and then the overall of MLDED system has been tested. The collection process of the dataset is

conducted in Izzat Marji Group by using 160 sensitive files of the target organization. Then, capturing the network traffic for a whole day, and creating "fake" sensitive leaked files where four of the employees shared in the process of MLDED system simulation. Data traffic on Monday is the largest in Izzat Marji Group among four working days (Sunday, Monday, Tuesday, and Wednesday), thus it was selected to be the test sample for MLDED system. Monday has the highest network traffic, where on that day 4299 files were extracted from the network. Therefore, those 4299 files extracted on Monday will be used as the dataset of the MLDED system-tuning phase. In addition, this dataset will be used to test the operation of MLDED system.

Evaluation Criteria

The performance of MLDED system is evaluated using true positive (TP), true negative (TN), false negative (FN), false positive (FP), Accuracy, False Positive Rate (FPR), Detection Rate (DR), False Negative Rate (FNR), and Precision.

True positive indicates the number of leaked sensitive files that are correctly classified by the data-exfiltration system as leaked sensitive files and it is considered a sign of proper detection of data leakage whereas the True negative indicates the number of legal files that are correctly passed as legal ones by the data-exfiltration system.

False positive indicates the files that were incorrectly classified as leaked sensitive files, whereas they are non-leaked insensitive files and it represents the accuracy of the detection system. False negative indicates files that were incorrectly detected as legal files, whereas they are leaked sensitive files (intrusion activities). A false negative is a direct sign of the inability of the data-exfiltration system to detect the intrusion.

Experimental Results of Level (1) Tuning Phase

This section present and discuss the experimental results obtained in case of running the level (1) of the MLDED system as a separate unit to ensure its performance. Ten sensitive files have intentionally been leaked in sake of evaluating the performance of this level of the MLDED system where the fake leaked files was designed to suit the complexity of the level (1). Table 1 and Table 2 shows the experimental results achieved at this level.

Table 1 The Standard performance measures of Level (1)

Standard performance Measure	No. of files
TP	10
TN	4289
FP	0
FN	0

Table 2 The performance evaluation metrics Level (1) /Tuning Phase

MEASURE	TNR	DR	FPR	FNR	ACCURACY	PRECISION
VALUE	100 %	100%	0	0	100%	100%

Since this level checks for file hashes and uses the hash values to compare the files, it achieved high detection rate of 100% and with high accuracy and sensitivity of 100%. These results for this level were expected, since no file can bypass if it is completely leaked, because the leaked and sensitive file share the exact hash value. The high precision of this level will enhance the performance of the other subsequent levels (level (2) and level (3)) by decreasing the number of leaked data samples to be checked.

Experimental Results of Level (2) / Tuning Phase

This section present and discusses the experimental results achieved when running level (2) of the MLDED system as a separate unit to ensure its performance. Level (2) has three leakage thresholds that measure the percentage of the captured file that contained in the corresponding sensitive file. After doing different experiments, three leakage percentages have been chosen for the tests as shown in Table 3:

Table 3 Leakage Threshold Percentage Meaning of Level (2)

Leakage Threshold	Meaning
30%	30% of the captured file contained in the corresponding sensitive file
60%	60% of the captured file contained in the corresponding sensitive file
75%	75% of the captured file contained in the corresponding sensitive file
90%	90% of the captured file contained in the corresponding sensitive file

The implemented MLDED system level (2) was run on these thresholds separately where each threshold shows different experimental results and different performance in terms of both standard measures (TN, TP, FP, and FN) and its derivatives.

In this experiment, nine sensitive files have intentionally been leaked in sake of evaluating the performance of this level of MLDED system where the intrusion files (leaked files) were designed to suit the complexity of the level (2). In case of the 30% threshold, level (2) successfully captured 12 leaked files out of 12 fake leaked files. Table 4 and Table 5 list the performance of level (3) at threshold of 30%.

Table 4 The Standard performance measures of Level (2) at 30%

Standard performance Measure	No. of files
TP	13
TN	3632
FP	654
FN	0

Table 5 The performance evaluation metrics of Level (2) at 30%

MEASURE	TNR	DR	FPR	FNR	ACCURACY	PRECISION
VALUE	84.741 %	100%	15.25%	0	84.78%	1.949%

As shown in Table 4 and Table 5, Level (2) of the implemented MLDED system has achieved a high detection rate (DR) of 100 % if the leakage threshold set to 30%. However, the precision decreased severely to less than 2% due to higher FP in comparison with the attained TP values.

On the other hand, true negative rate (TNR) of this level is lower than that achieved by level (1) due to the high false positive rate of this level, which affect the overall performance of MLDED system. In case of the 60% threshold, level (2) successfully captured 13 leaked files out of 13 fake leaked files. Table 6 and Table 7 list the performance of the level (2) if the threshold set to 60%.

Table 6 The Standard performance measures of Level (2) at 60%

Standard performance Measure	No. of files
TP	13
TN	4078
FP	208
FN	0

Table 7 The performance evaluation metrics of Level (2) at 60%

MEASURE	TNR	DR	FPR	FNR	ACCURACY	PRECISION
VALUE	95.14%	100%	4.85%	0	95.1617%	5.88%

As shown in Table 6 and Table 7, Level (2) of the implemented MLDED system has achieved the same high detection rate of 100 % as achieved in case of the 30% threshold since the FN is zero.

However, the accuracy that achieved at 60% threshold is more than that achieved in case of 30% due to decreasing in the FP value. To increase the performance of the level (2) and in an attempt to achieve a better detection

rate, the leakage threshold has been increased to 75%. Tables 8 and 9 show the performance evaluation measures of this level at a threshold of 75%.

Table 8 The Standard performance measures of Level (2) at 75%

Standard performance Measure	No. of files
TP	13
TN	4042
FP	23
FN	0

Table 9 The performance evaluation metrics of Level (2) at 75%

MEASURE	TNR	DR	FPR	FNR	ACCURACY	PRECISION
VALUE	99.4%	100%	0.0056	0	99.44%	36.111%

The final test was increasing the threshold to 90%, where the level (2) successfully captured 12 leaked files out of 13 fake leaked files, which means, one sensitive file is leaked and it passed through MLDED system.

The FP value that resulted in this threshold is the best among the rest of thresholds that used in this level; however, it shows degradation in accuracy and false negative rate. Table 10 and Table 11 list the performance of the level (3) at a threshold of 90%.

Table 10 The Standard performance measures of Level (2) at 90%

Standard performance Measure	No. of files
TP	12
TN	4273
FP	14
FN	1

Table 11 The performance evaluation metrics of Level (2) at 90

MEASURE	TNR	DR	FPR	FNR	ACCURACY	PRECISION
VALUE	99.6%	92.3%	0.0032	0.076	99.6%	46.153%

A comparison has been done in sake of comparing between the performances of level (2) leakage thresholds that was used during this level's tests. Fig. 3. shows the detection rate comparison of the leakage thresholds of level (2).

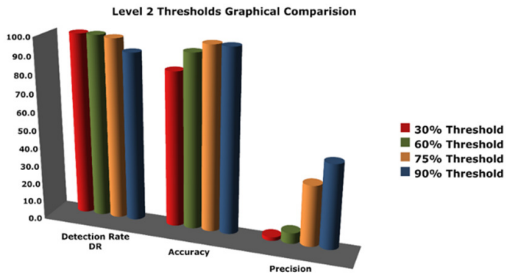

Fig. 3. Graphical comparison of Level (2).

As shown in Fig. 3. 75% leakage threshold has the highest detection rate associate with highest accuracy. Thus, it will be chosen as the best threshold in the phase of testing the overall MLDED system.

On the other hand, a leakage threshold of 90% has the highest precision with the lowest detection rate, and 30% threshold has poor precision and accuracy values, 60% threshold has a comparable performance to 75% but with low precision. Therefore, as a result, thresholds of (30%, 60% and 90%) will not be selected as the best thresholds to be used in the testing phase of the whole performance of MLDED system.

Experimental Results of Level (3) / Tuning Phase

This section presents and discusses the experimental results achieved when running the level (3) of the MLDED system as a separate unit to ensure its performance. As explained in chapter four, level (3) has three leakage percentages (thresholds) that measure the percentage of the sensitive keywords leakage of a file. After performing experiments, three leakage percentages have been proposed as shown in Table 12:

Table 12 Leakage Percentage Meaning of Level (3)

Leakage Threshold	Meaning
30%	30% of keywords the captured file contained in a sensitive file
60%	30% of keywords the captured file contained in a sensitive file
90%	30% of keywords the captured file contained in a sensitive file

The implemented MLDED system level (3) was run on these thresholds separately where each threshold shows different experimental results and different performance in terms of both standard measures (TN, TP, FP, and FN) and its derivatives.

In the first experiment, nine sensitive files had intentionally been leaked for the sake of evaluating the performance, where each employee (who shared in the MLDED system simulation) was asked to select an arbitrary piece of data out of the sensitive files that had intentionally been leaked.

In case of the 30% threshold, level (3) captured 23 files. These captured files include the files that correctly leaked (TP) and that were misclassified as leaked ones (FP). Table 13 and Table 14 list the performance of the level (3) at 30% leakage threshold.

Table 13 The Standard performance measures of Level (3) at 30%

Standard performance Measure	No. of files
TP	8
TN	4176
FP	115
FN	1

Table 14 The performance evaluation metrics of Level (3) at 30%

MEASURE	TNR	DR	FPR	FNR	ACCURACY	PRECISION
VALUE	99.67%	92.3%	0.0032	0.076	99.6%	46.153%

As shown in Tables 13 and 14 above, MLDED system has achieved a high detection rate of 100% if the leakage threshold set to 30%. However, the precision decreased dramatically due to the high FP in comparison with the TP values. The high value of FP is a natural result of using a low leakage threshold where the contents compared on a low threshold that will select any file contains 30% of a sensitive file, which may be in most cases a legal file. With the aim of seeking a better accuracy, two other thresholds have been used and tested. Tests with a threshold of 60% and 90% have also been tested. Table 15 and Table 16 lists the experimental results achieved when running level (3) of the MLDED system with a threshold of 60%.

Table 15 The Standard performance measures of Level (3) at 60%

Standard performance Measure	No. of files
TP	8
TN	4176
FP	115
FN	1

Table 16 The performance evaluation metrics of Level (3) at 60%

MEASURE	TNR	DR	FPR	FNR	ACCURACY	PRECISION
VALUE	99.8 %	88.8%	0.001	0.11	99.86%	61.53%

The following Tables 17 and 18 lists the experimental results achieved when running level (3) of the MLDED system with a threshold of 90%.

Table 17 The Standard performance measures of Level (3) at 90%

Standard performance Measure	No. of files
TP	3
TN	4290
FP	0
FN	6

Table 18 The performance evaluation metrics of Level (3) at 90%

MEASURE	TNR	DR	FPR	FNR	ACCURACY	PRECISION
VALUE	100 %	33.3%	0	66.6%	99.86%	100%

Referring to Fig. 4. a threshold of 60% has the highest detection rate of 88.89% with the highest precision of 66.6%, thus this threshold value will be used for this phase of MLDED system testing

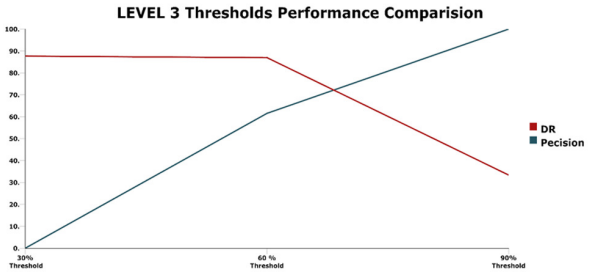

Fig. 4. Graphical comparison of level (3) thresholds

MLDED System: Overall Testing

This section presents and discusses the experimental results achieved when running the MLDED system as a whole depending on the best configuration parameters that obtained in the phase of tuning which shown in Table 19. The testing dataset is the same that used in tuning phase and it consists of 4299 captured files, the sensitive folder consists of 160 files. To test the overall MLDED system, 32 fake leaked files have been transmitted.

Table 19 Best levels configurations

Level	Best Configuration
Level(1)	True or False (No thresholds)
Level(2)	75%
Level(3)	60%

The experimental results of the overall MLDED system testing are shown in Tables 20 and 21.

Table 20 The Standard performance measures of MLDED system

Standard performance Measure	No. of files
TP	32
TN	4221
FP	46
FN	0

Table 21 The performance evaluation metrics of MLDED system

MEASURE	TNR	DR	FPR	FNR	ACCURACY	PRECISION
VALUE	98.92%	100%	1.1%	0%	98.93%	41.02%

MLDED System: Throughput

Although MLDED system is designed for digital forensics purposes and the time efficiency is out of scope of this research, MLDED took about six hours to process 4299 captured files, which means MLDED can process twelve files per minute.

V. CONCLUSION AND FUTURE WORK

In this research, we can conclude with the following points:

1. A multi-layer data-exfiltration detection system has been implemented and tested.
2. The MLDED system achieved an overall detection accuracy of 98.93% and FPR of 1.1% which is a promising result.
3. The experiments and evaluations of MLDED have been performed in a real-world environment, with real data.

Many other areas and work could be added to improve the work done such as:

1. Apply machine learning techniques to make initial classification of captured files.
2. Add the support of other document file formats such as: DOC, DOCX, XML, XLS, and XLSX.

REFERENCES

[1] Chatterjee Tanusree, and Bhattacharya Abhishek (2014, Feb), VHDL Modeling of Intrusion Detection and Prevention System (IDPS) A Neural Network Approach. International Journal of Computer Trends and Technology (IJCTT) – volume 8 number one.

[2] Sharma, P. (2013). A multilayer framework to catch data exfiltration (Doctoral dissertation, UNIVERSITY OF MARYLAND, BALTIMORE COUNTY).

[3] Stamati-Koromina, V., Ilioudis, C., Overill, R., Georgiadis, C. K., & Stamatis, D. (2012, September). Insider threats in corporate environments: a case study for data leakage prevention. In Proceedings of the Fifth Balkan Conference in Informatics (pp. 271-274). ACM.

[4] Udo-Akang, D. (2014) Cyber Perspectives: Internet Exploitation and Business Survivability. International Journal of Business, Humanities and Technology.

[5] Shu, X., Zhang, J., Yao, D. D., & Feng, W. C. (2015). Rapid and Parallel Content Screening for Detecting Transformed Data Exposure. Retrieved February 20, 2015 from http://people.cs.vt.edu/danfeng/papers/BigSecurity-Yao-2015.pdf

[6] Kemerlis, V. P., Pappas, V., Portokalidis, G., & Keromytis, A. D. (2010, October). iLeak: A lightweight system for detecting inadvertent information leaks. In Proceedings of the sixth European Conference on Computer Network Defense (EC2ND) (pp. 21-28).

[7] Liu, Y., Corbett, C., Chiang, K., Archibald, R., Mukherjee, B., & Ghosal, D. (2009, January). SIDD: A framework for detecting sensitive data exfiltration by an insider attack. In System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference, IEEE, (pp. 1-10).

[8] Al-Bataineh, A., & White, G. (2012, October). Analysis and detection of malicious data exfiltration in web traffic. In Malicious and Unwanted Software (MALWARE), 2012 7th International Conference, IEEE, (pp. 26-31).

[9] Coron, J. S., Dodis, Y., Malinaud, C., & Puniya, P. (2005, January). Merkle-Damgård revisited: How to construct a hash function. In Advances in Cryptology–CRYPTO 2005, Springer, Berlin Heidelberg, (pp. 430-448).

[10] Cutts, M. (2013). Oxford guide to plain English (4th ed.). UK: Oxford university Press

[11] Goyvaerts, J. (2006). Regular Expressions: The Complete Tutorial. USA, Princeton University: Lulu Press