Analyzing and Detecting Malicious Content: DOCX Files.

Ayyad Mohammad Naser, Mohammad Hjouj Btoush,

Computer Science Dept.,
Al-Balqa Applied University,
Salt, Jordan
Eng.ayyad@hotmail.com, hujooj@yahoo.com

Ali Hussein Hadi

Computer Science Dept.,
Princess Sumaya University for
Technology, Amman, Jordan
ali@ashemery.com

***Abstract*-This paper presents the process of analyzing and detecting malicious content which is DOCX Files. Recently these files are secure, popular and reliable documents used by attackers as an instrument to attack users. The attackers have converted their ways from server-side to client-side attacks. The attackers focused on client-side attacks and also use DOCX as a tool to spread malicious code in document files to ease the attacking process. However, DOCX files suffer from many security issues, similar to previous Office formats; attackers are still embedding to hide malware thanks, macros and similar features to OLE objects. The aim of this paper is to create a system to analyze the structure of DOCX files and classify these files to suspicious or benign. One file was scanned randomly through a direct application and it was found to have suspected words which classify this file as a malicious. The suspected words were determined by extracting the malicious DOCX file. After that, the suspected words were confirmed by comparing this malicious file with a benign file. By depending on the presence of these suspected words or not, there is no any need to open the DOCX file that causes damage to the computer and this system will play a very important role to the users who deal with the threat.**

Keywords- ***OOXML, Client-Side Attack, Analysis DOCX, Suspicious DOCX, Malicious DOCX, Structure Scan.***

I. INTORDUCTION

Over the last 10 years; organizations, corporations, and individuals have become more dependent on computers to achieve specific purposes. Microsoft Office is a very popular kind of Office that runs on computers. Microsoft Office 2007 and 2010 using XML and ZIP technologies [1].DOCX file became very important because it depends on two pillars: ZIP archive and security files [2]. The DOCX is one of the most popular file formats to interchange documents between users. Attackers have newly converted their method from server-side to client-side attacks the international adoption of the DOCX format, has released DOCX the critical vector for malware distribution. The feature of DOCX files and the content possibly lead to some security problems that can be used to put malicious elements to hold malware and steal data from of these features possibly contain embedding in DOCX file. Hackers focus on attacking DOCX because it uses a unique safe merit, which is trusted by computer [3]. Unfortunately, the anti-virus systems used are not capable of detecting malicious content [4].

This research will focus on one of the most extensively used attack vectors of malicious document XML and focusing on a new technique of attack which known “client-side attack” that exploits vulnerabilities on the client used through the user that lack of security awareness that is considered as a point to start his attacks to control the network or system. This paper seeks to propose a new system based on analyzing and detecting malicious content and will test DOCX files that contain malicious and benign content. After an analysis process, the system will be able to classify benign or suspect files.

II. BACKGROUND

Starting in 2000, Microsoft announced about releasing XML and gives libraries, consumers, and government the permission privacy for using XML file format. In December 2006, PEGSCO reports that Microsoft Company has embraced a new technique “XML format”. In 15th of November 2005, there was co-submission of Office Open XML Formats to ECMA International [5].

MS Office 2007 and 2010 files use a new technique file format depending on OOXML format that is different from the old techniques, whereas, MS Office 1997 and 2003 files are saved as the binary format [6]. An Open Office XML file depends on the following [2][7]:

- Parts: files in ZIP archive
- Relationship: the relationships between the parts and packages or between parts.
- Package: ZIP archive.

For all package, a ZIP item containing information around the relevance between packages and parts. Similarly, there is a relationship ZIP archive items that include information about relationships between the parts and the package or among parts of the document. As shown in Figure 1, the structured information is first encoded into XML and compressed, so the new documents will take up less space than previous format.it also shows the structure of OOXML format. The OOXML has new extensions distinguished from the previous type of documents as tabulated in Table 1 [1].

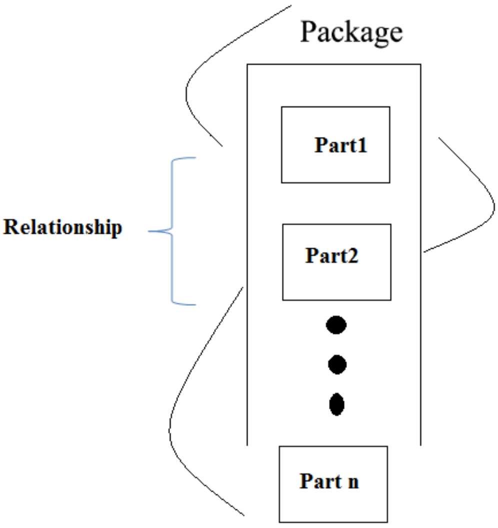
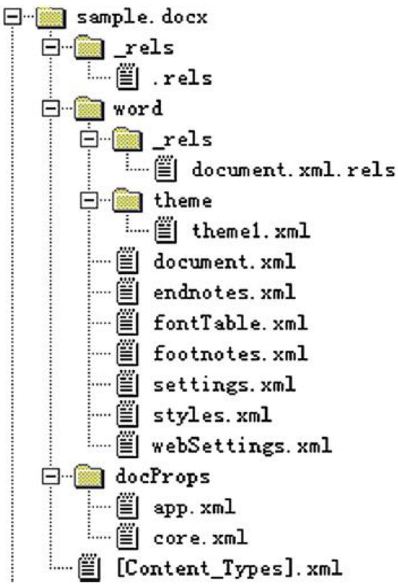


Fig 1: structure of Office Open XML format.

Table 1: Extensions of MS office file

File Type	Extensions
- Microsoft word - Macro-Enabled	- .docx - .docm
- Microsoft Excel - Macro-Enabled	- .xlsx - .xlsm
- Microsoft Powerpoint - Macro-Enabled	- .pptx - .pptm

Figure 2 displays the directory structure of a typical MS Word 2007 and 2010 document. In the root, we find a file called “[Content_Types].xml” of the ZIP archive. Functionality, this file saves a dictionary with content types for each the other parts inside the package [8][9].



```
sample.docx
  _rels
    .rels
  word
    _rels
      document.xml.rels
    theme
      theme1.xml
    document.xml
    endnotes.xml
    fontTable.xml
    footnotes.xml
    settings.xml
    styles.xml
    webSettings.xml
  docProps
    app.xml
    core.xml
  [Content_Types].xml
```

Fig 2: The directory structure of Word

As shown Figure 3, the OOXML document is divided into three parts to define the printable text: paragraph, run, and text elements [6]. The relationships between the three elements are also explained. in Figure 4 illustrates OOXML format and each <w:p> indicate a paragraph within the <w:body>. <w:r> represents a run, and the paragraph can be divided into multiple runs. Finally, <w:t> is the text element. There can be group of texts <w:t> elements within a <w: r>.

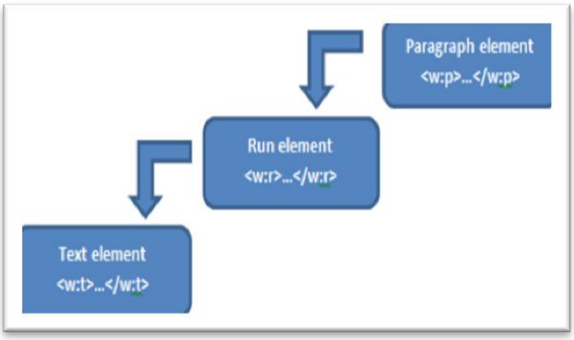
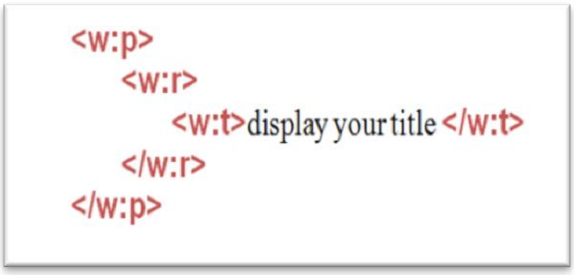


Fig 3: The relationships among three elements.



```
<w:p>
    <w:r>
        <w:t>display your title </w:t>
    </w:r>
</w:p>
```

Fig 4: Basic text written through the XML.

III. LITERATURE REVIEW

There is a forensic way which depends on the unique value of the revision identifier that specifies the source of suspicious electronic documents [10]. This way applies to electronic documents as the use of the format Office Open XML (OOXML) such as: MS Office 2007 and MS Office 2010. According to the distinction revision identifier that is extracted from documents, the investigators can detect if the suspicious document and other document are from the same source. Forensic investigators can specify if the original document and the copy are from the same source by detecting its revision identifier value. Forensic investigator can specify that the two documents by using the same way and then showing the way for OOXML format files to detect the creator information and the real time information.

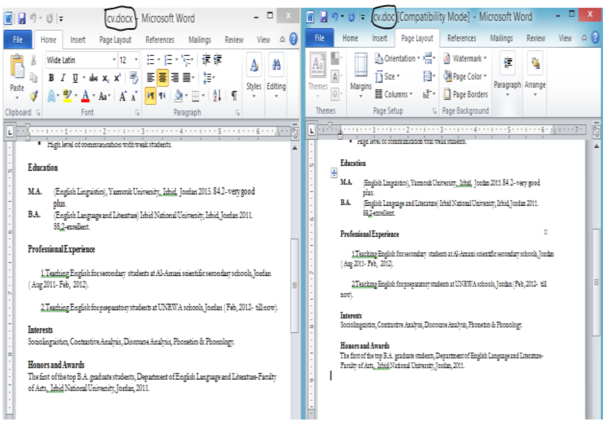
The author refers to the possibility of concealing information that used MS Office file, and that researcher is aware of these documents which contain a very huge amount of space unused and metadata. That is used to conceal information [11]. The author mentioned that there are a lot of electronic documents that are located on individual systems and on wide networks. Electronic document includes important information like secret trade and private data. As electronic documents may use digital signify in forensic checking, they used forensic examiners to considerate specific applications on content which help forensic examiner to detect the relevance between different electronic documents. The author proposed a new technique for investigating Microsoft power point files which include very important information about writing process [2].

DOCX files are new documents, Microsoft office use XML format for office 2007 and 2010 depends on the same purpose which depend on Zip archive with an open schema contrasting to good-old format (Ms-word, Excel, power point,…etc). The author talked about the security system and its technical issues containing XML, and Zip obfuscation processes which are followed to create a security system to extract malicious content from document [12].

The researchers show many different techniques for malware such as hex editing, reverse engineering and repacking to avoid host-based Anti-Virus (AV) system. These techniques circumvent SSL/HTTP and SMTP. Lastly, it is important to detect and find new and anonymous malware; new malware detection system based on honeynet systems is surveyed [13].

The researchers talked about the dangerous effect of malware through the Internet. They also mention some of these malware's like (viruses, Trojan, and worm). These are considered as a threat to a computer security system. By using a particular approach for analyzing malicious behavior through an effective way including extracting and fixing some part of a document [14].

Open Office XML involves the compressed ZIP file which is known as the package [15]. ZIP has been chosen as a packaged pattern for the office XML formats, the (Figure 5 and 6) DOCX file illustrate the ZIP compressions property that decrease the size of the document nearly to three-quarters percent (75%) [16].



- High level of communication with weak students.

Education

M.A. (English Linguistics), Yarmouk University, Irbid, Jordan 2015. 84.2- very good plus.
B.A. (English Language and Literature) Irbid National University, Irbid, Jordan 2011. 88,2-excellent.

Professional Experience

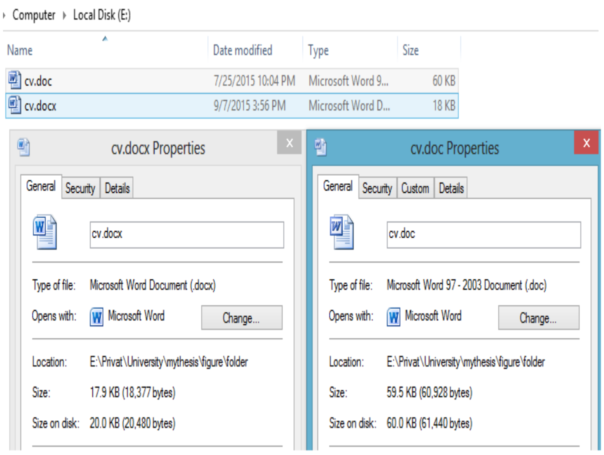
1.Teaching English for secondary students at Al-Amari scientific secondary schools, Jordan (Aug 2011- Feb, 2012).

2.Teaching English for preparatory students at UNRWA schools, Jordan (Feb, 2012- till now).

Interests
Sociolinguistics, Contrastive Analysis, Discourse Analysis, Phonetics & Phonology.

Honors and Awards
The first of the top B.A. graduate students, Department of English Language and Literature-Faculty of Arts, Irbid National University, Jordan, 2011.

Fig 5: content with DOCX and DOC



Name	Date modified	Type	Size
cv.doc	7/25/2015 10:04 PM	Microsoft Word 9...	60 KB
cv.docx	9/7/2015 3:56 PM	Microsoft Word D...	18 KB

cv.docx Properties

cv.docx
Type of file: Microsoft Word Document (.docx)
Opens with: Microsoft Word
Location: E:\Privat\University\mythesis\figure\folder
Size: 17.9 KB (18,377 bytes)
Size on disk: 20.0 KB (20,480 bytes)

cv.doc Properties

cv.doc
Type of file: Microsoft Word 97 - 2003 Document (.doc)
Opens with: Microsoft Word
Location: E:\Privat\University\mythesis\figure\folder
Size: 59.5 KB (60,928 bytes)
Size on disk: 60.0 KB (61,440 bytes)

Fig 6: Show using document

IV. ANALYZING AND DETECTING DOCX FILE

The experiment was tested on a host with the Windows 8 operating system. The specifications and identifiers which are used in the experiment are recorded in Table 2. A virtual machine was used to test the DOCX file in order to detect the malicious files and to keep the host operating system safe.

Table 2: Specification of Testing Machine

Hardware/software	Specification
Operating System	Windows 8
OS-VM	Windows 7
RAM	4 GB
RAM-VM	1.50 GB
CPU	2.50 GHz
CPU-VM	Core^{TM} i5
Language	Python 2.7.3

Table 3 consists of MD5 hash value and shows the size of the dataset used in the experiment; it consists of 25 DOCX files as dataset and 57 DOCX files as a sample. DOCX files were downloaded from a variety of sites and from Google which was used to experiment test.

Table 3: DOCX files was collected for the experiment

Category	Number of files	Size of files
DOCX files-Data set	25	4.60 MB
DOCX files-sample	57	7.61 MB
MD5	206	697 MB

V. FILTER DOCX FILE METHOD

There are many researches that have been used to hide the data, but there is a little use of the structure DOCX files which depends on XML file format. Although the complexity of the file format DOCX, it faces some of threats after supporting the company has a zip archive. After parsing the DOCX files structure, and how the hacker can threaten the system by embedding exploits to circumvent security detection. Filter DOCX files was designed to parse and classify DOCX file which is based on keywords. The system will parse a given DOCX file, and extract the suspicious elements contained inside the file. Finally, we will get a report to describe this file as a suspicious or benign.

The filter DOCX files focus on two Phases:

1) Structure Scan phase

The highlight of the designing method is shown in Figure 7. Structure scan depend on keywords by scanning that are exist in within keywords file identified by the user. After that the DOCX file and the keywords file are read to calculate hash value for them.

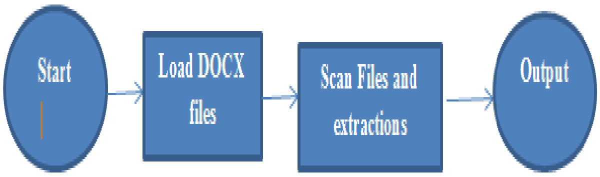


Fig 7: System Scan architecture

The DOCX file is read as a sequence of byte in order to analyze it easily. This study uses the DOCX file to analyze the embedding file. There are two filters used in order to check the file, which includes the keywords:

- OLE and o:lock
- Vanish

The main objective of the structure scans (4.1) is to extract various elements located within DOCX, which are used in regular expressions to parse the DOCX and search for those known the keywords. The system will be starting the search for keywords that are not directly connected to the files. At the end of the system will give a caution if the file is a malicious, or a benign, or a suspicious.

2) Filter DOCX phase

The highlight of the designing method is shown in Figure 8. The system checks the hash value of the DOCX file if it is exist in hash value in database. If no the hash value of the DOCX file does not exist, the system adds to the hash value database. If the hash value DOCX file found in the database. This indicates that the DOCX file was scanned before.

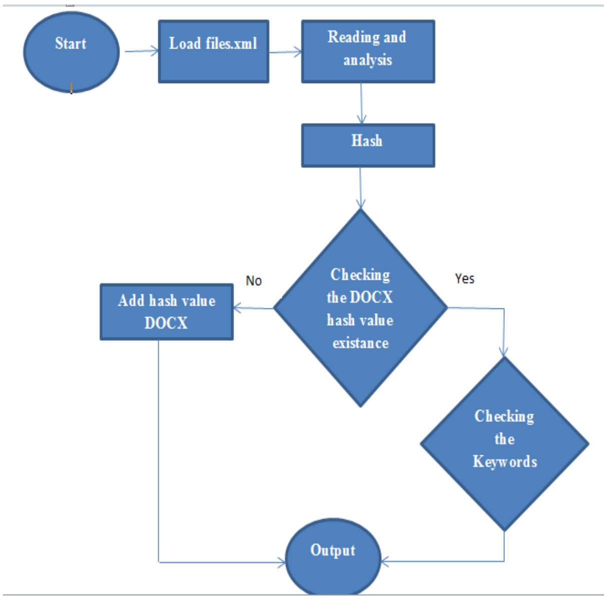


Fig 8: Filter DOCX architecture

A. Reading and Analysis

Reading the DOCX files, and the scanning files, i.e. the keyword files are open XML and calculating the hash value.

B. *Calculate hash value*

The first manner the system will calculate the hash value for each DOCX files in addition to the keyword was extracted from DOCX file, which will be used to identify and classify the malware samples as well as the elements inside DOCX files. It can be used when inquiring for documents that have been already analyzed which can save the user's time if the document has been analyzed before. After scanning DOCX files, the final step enables the user to choose how to display the output format of the scan through the console or copy the output to text file named by the name of the scanned DOCX Files.

VI. RESULTS AND ANALYSIS

When downloading samples from web site it is likely to be some malicious. The presence of malicious files in benign samples or benign files in malicious samples will produce negative results on the studied experiment. The dataset consist of 25 files of malicious and benign DOCX files were used to evaluate the proposed method in order to detect suspicious DOCX files. Structure scan method was used for scanning DOCX files to search the pre-selected keys that have been obtained from the malicious DOCX files. Three files were obtained by entering the path which contains DOCX files, as shown in Figure 9.

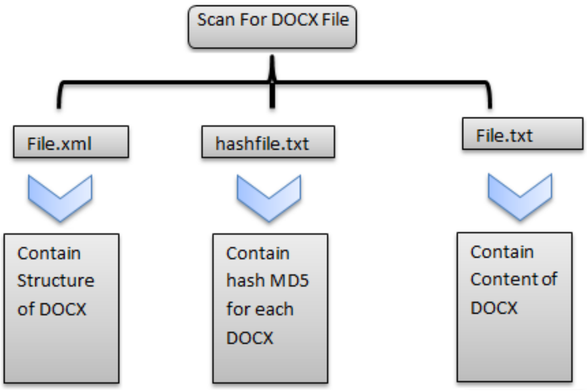
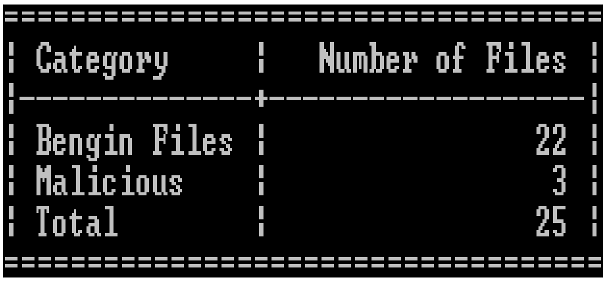


Fig 9: Output Scan DOCX

The 25 files were classified as benign and malicious:

Table 4: The DOCX collected for the experiments



```
====================================
¦ Category      ¦   Number of Files ¦
¦---------------+-------------------¦
¦ Bengin Files  ¦                22 ¦
¦ Malicious     ¦                 3 ¦
¦ Total         ¦                25 ¦
====================================
```

B. *ExperimentS*

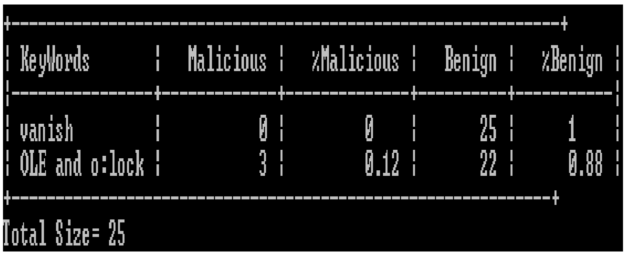
This paper consists of two experiments:

- The first experiment focuses on a new system to analyze and detection malicious DOCX files and explains the results achieved by Structure Scan.
- The second experiment use a sample DOCX files were downloaded a variety of site from google to test experiment and used for experiment test.

1) *Experiment 1*

In this experiment, the Structure Scan was tested using Python regular expressions are used to search for the suspicious features and calculate their frequencies in both the benign and malicious dataset. After running the scan on the dataset, the results were tabulated, as shown the Table 5. The percentages was calculated by dividing the number of files which represent the malicious on the total number of the sample which consist of 25 DOCX files according to the below equation 4.1.

Table 5: Structure Scan result.



```
+---------------------------------------------------------+
¦ KeyWords      ¦ Malicious ¦ %Malicious ¦ Benign ¦ %Benign ¦
¦---------------+-----------+------------+--------+---------¦
¦ vanish        ¦         0 ¦      0     ¦     25 ¦    1    ¦
¦ OLE and o:lock ¦        3 ¦      0.12  ¦     22 ¦    0.88 ¦
+--------------------------------------------------------+
Total Size= 25
```

$$\text{Percentage} = \frac{\sum \text{Malicious dataset with a frequency feature}}{\sum \textit{Total Dataset}} \qquad (1)$$

It was found that the presence of these two keywords: Vanish, and (OLE and o: lock) are indicators for suspicious file. These characteristics could be used to differentiate between benign and suspicious files. By calculating the number of files from the hypothesis proposed above, we must be aware of how these characteristics are presented in the DOCX files. Each DOCX files may contain both features.

a) *Detection Accuracy*

Before explaining the detection rates achieved below, we must be aware about the four terms used for evaluation [17]:

- False Positive (FP): The number of files classified as malicious from benign samples.
- False Negative (FN): The number of files classified as benign from malicious samples.
- True Positive (TP): The number of files classified as malicious from malicious samples.
- True Negative (TN): The number of files classified as benign from benign samples.

In this experiment, the Performance of the method is evaluated by using the terms below [17]:

- True positive rate (TPR) based on the classification of positive correctly ,
- False positive rate (FPR) based on the classification of negative, and
- Accuracy(ACC) based on measuring the number of absolutely correctly classified instances:

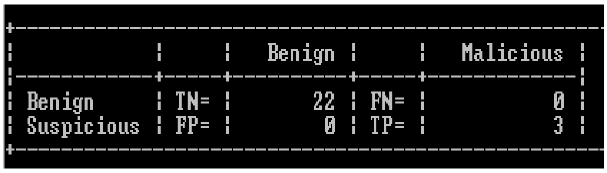
$$\text{True Positive Rate (TPR)} = \frac{TP}{TP+FN} * 100\% \quad (2)$$

$$\text{False Positive Rate (FPR)} = \frac{\text{FP}}{\text{FP+TN}} * 100\% \quad (3)$$

$$\text{Accuracy (ACC)} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} * 100\% \quad (4)$$

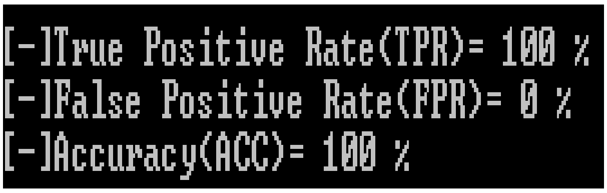
By using the equations which explained previously, the structure of XML files can be evaluated in terms of the presence of Vanish and (o:OLE and o:lock). These characteristics can determine the efficiency of the system and the accuracy of detection, as shown the Table 6.

Table 6: Detection results for Structur Scan



```
+---------------------------------------------------------
¦            ¦     ¦  Benign ¦     ¦  Malicious ¦
¦------------+-----+---------+-----+------------¦
¦ Benign     ¦ TN= ¦      22 ¦ FN= ¦          0 ¦
¦ Suspicious ¦ FP= ¦       0 ¦ TP= ¦          3 ¦
+---------------------------------------------------------
```

Depending on Table 5 and equations 2, 3 and 4, the results could be seen in Figure 10:



```
[-]True Positive Rate(TPR)= 100 %
[-]False Positive Rate(FPR)= 0 %
[-]Accuracy(ACC)= 100 %
```

Fig 10: Extraction results

Depending on the results from Table 3, it was found that three files were detected as suspicious from three malicious files with 100%. Remarkably, it was found also that the falsely detected files were zero with zero percent as suspicious files from 22 benign files.

b) Performance Evaluation

To evaluate the Filter DOCX files, and how to detect suspicious DOCX files, a comparison was done with a related work entitled by OpenDocument and Open XML security (Lagadec P., 2008), this paper have no results, the researcher was depending on the presence of oleObject.bin as a keyword in this previous method[1], as shown the Figure 11. This method had been used to scan the DOCX files, then locate the keyword in the DOCX files and describe the way that this keyword was embedded and stored in these documents. This method depends on two values; if yes, it means that the file is suspect but if No, it means that the file is benign. A structure scan was designed to detect oleObject.bin keyword in the DOCX files, as shown Figure 12**.** It was found that there are two suspicious DOCX files**,** in the sample that consist of 25 DOCX files and have a similar dataset.

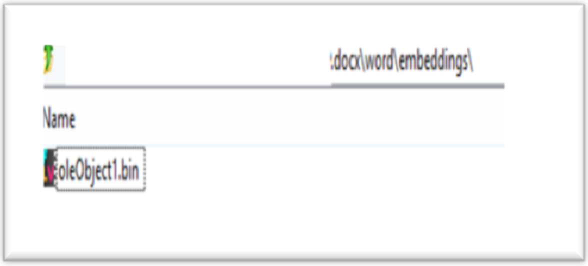
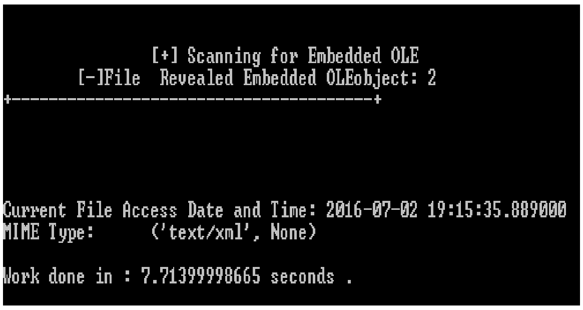


Fig 11: Stored OLE object



```
                [+] Scanning for Embedded OLE
         [-]File  Revealed Embedded OLEobject: 2
+--------------------------------------+



Current File Access Date and Time: 2016-07-02 19:15:35.889000
MIME Type:       ('text/xml', None)

Work done in : 7.71399998665 seconds .
```

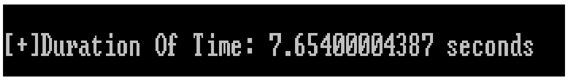
Fig 12: Structure scan OLE object

The researcher depended on the presence of oleObjec1.bin as a keyword in this search. In this study, it was found that three DOCX files as malicious out of 25 were detected as suspicious files, as shown Table 4. Unlike, to the previous method which detects two DOCX files out of 25 as suspicious files.

c) Performance Evaluation of Time

The time represents the main factor in measuring the speed of performance. The python program was used to import time in order to calculate the execution time by start time, end time, and the difference between the end and start time represent the duration of the implementation as show Figure 13 to show the time was achieved.

[1] OpenDocument and Open XML security (Lagadec P.)(2008).



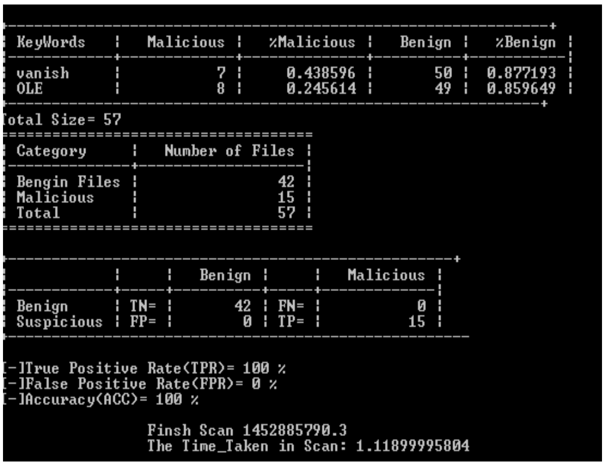
```
[+]Duration Of Time: 7.65400004387 seconds
```

Fig 13: Time taken in implementation

2) Experiment 2

A sample consists of 57 DOCX files were downloaded from a variety of sites from Google and used for experiment test:

- By using the structure scan in this study; 57 DOCX files were analyzed. The result detected that 0.263% DOCX files were suspicious files; 0.140% of them found to contain embedding and the duration time of implementation was 1.11 seconds, as shown in Figure 14.
- By using the OLE objects in previous method the result was two DOCX files out of 57 were suspicious files; both of them were found to contain embedding and the duration time of implementation was 18.86 seconds, as shown in Figure 15.



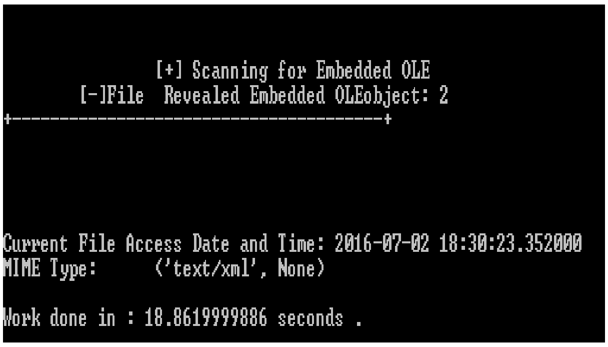
```
+--------------------------------------------------------------------+
| KeyWords   |  Malicious |  %Malicious |  Benign |  %Benign |
|------------+------------+-------------+---------+----------|
| vanish     |          7 |    0.438596 |      50 | 0.877193 |
| OLE        |          8 |    0.245614 |      49 | 0.859649 |
+------------------------------------------------------------------+
otal Size= 57
====================================
| Category     |  Number of Files |
|--------------+------------------|
| Bengin Files |               42 |
| Malicious    |               15 |
| Total        |               57 |
====================================

+-----------------------------------------------------+
|            |     |  Benign |     |  Malicious |
|------------+-----+---------+-----+------------|
| Benign     | TN= |      42 | FN= |          0 |
| Suspicious | FP= |       0 | TP= |         15 |
+------------------------------------------------------

[-]True Positive Rate(TPR)= 100 %
[-]False Positive Rate(FPR)= 0 %
[-]Accuracy(ACC)= 100 %

                Finsh Scan 1452885790.3
                The Time_Taken in Scan: 1.11899995804
```

Fig 14: Analysis of the test file using Structure Scan Method



```
           [+] Scanning for Embedded OLE
      [-]File  Revealed Embedded OLEobject: 2
+--------------------------------------+



Current File Access Date and Time: 2016-07-02 18:30:23.352000
MIME Type:      ('text/xml', None)

Work done in : 18.8619999886 seconds .
```

Fig 15: Analysis of the test file using OLE objects in previous Method

VII. CONCLUSION

Client's security and privacy occupies the first priority in this research. DOCX files are used by many users as document exchange format. Nowadays, DOCX files are used by hackers to run harmful codes on computers.

In this research, the structure of the DOCX files has been studied and analyzed in order to detect the keywords which refer to the suspicious files. After that, DOCX files were classified as suspicious or benign files. Eventually, the Structure Scan Method was designed for this purpose.

In conclusion, using Structure Scan Method is useful to detect suspicious DOCX files, reduce the duration time of implementation, and protect the DOCX files code on the computer from harmful by hackers. In addition, Structure Scan Method is more accurate and useful than the previous method.

Finally, Structure Scan Method is the best program may use by client-side to maintain and improve the system security, the dataset consists of 25 non-detected DOCX files which were collected for a malicious detection in this study. By using structure scan it was found that 0.88% files were classified as benign files and 0.12% as malicious ones. According to the presence of these suspected words and the time duration in the implementation 7.65 seconds in order to evaluate the Filter DOCX files, and how to detect suspicious DOCX files, a comparison was done with a related work entitled by previous method. This program was designed to calculate a percentage of suspicious files. Besides that, the researcher depends on the presence of oleObjec1.bin as a keyword, which detects 0.08% DOCX files as suspicious files and the time duration in the implementation 7.86 seconds. A sample consists of 57 DOCX files were downloaded from Google and they are used for experiment test. It was found that 0.74% files were classified as benign and 0.26% as suspicious ones and the time duration of implementation was 1.11 seconds. By using the previous method the result was detected two DOCX files were suspicious files and the time duration of implementation was 18.86 seconds. In addition, a program was designed to compare between Structure Scan and previous method and that the results above prove the Structure Scan more useful.

VIII. FUTURE WORK

- Design a method to analyze the embedding OLE object in the suspicious DOCX files and classify the files as malicious or benign.

- Improve the method to analyze the DOCX files which contain "Vanish" for hiding data in DOCX Files.

- Design a system to analyze any files which belong to OOXML to prevent hackers from using those files to hack clients.

IX. LIMETATIONS

There is many malicious DOCX files use the suspicious keywords referred in the experiments, also there are many benign DOCX files use them, which make the way unable to distinguish between benign and malicious DOCX files. The authors' way cannot detect any malicious DOCX files that do not use these keywords as the attack vector.

REFERENCES

[1] Rice F. (2006). Introducing the office (2007) open XML file formats.Microsoft Developer Network

[2] Park, J., & Lee, S. (2009). Forensic investigation of Microsoft PowerPoint files. Digital Investigation, 6(1), (pp. 16-24)

[3] Kittilsen, J. (2011). Detecting malicious PDF documents.

[4] Smutz C., & Stavrou A. (2014) Document Content Layout Based Exploit Protections.

[5] Jones B. (2007). History of office XML formats (1998-2006). MSDN blogs.

[6] Castiglione A., D'Alessio B., De Santis A., & Palmieri, F. (2011). New steganographic techniques for the OOXML file format (pp. 344-358). Springer Berlin Heidelberg.

[7] Fu Z., Sun X., Liu Y., & Li B. (2011). Forensic investigation of OOXML format documents. Digital Investigation, 8(1), (pp.48-55).

[8] Tom N. (2006). Office Open XML Overview. ECMA International, 14.

[9] Ehrli, E. (2006). Walkthrough: Word 2007 XML Format. Microsoft Corporation, June.

[10] Garfinkel S. L., & Migletz J. (2009). New XML-based files: implications for forensics. NAVAL POSTGRADUATE SCHOOL MONTEREY CA.

[11] Cantrell G., & Dampier D. (2004). Experiments in hiding data inside the file structure of common office documents: a stegonography application. In Proceedings of the 2004 international Symposium on information and Communication Technologies (pp. 146-151). Trinity College Dublin

[12] Lagadec P. (2008). OpenDocument and Open XML security (OpenOffice. org and MS Office 2007). Journal in Computer Virology, 4(2), (pp.115-125).

[13] Daryabar F., Dehghantanha A., & Udzir N., (2011). Investigation of bypassing malware defences and malware detections. InInformation Assurance and Security (IAS), 2011 7th International Conference on (pp. 173-178). IEEE.

[14] Grégio A. , Filho D., Afonso V. , Santos R. , Jino M., & de Geus P. (2011). Behavioral analysis of malicious code through network traffic and system call monitoring. In SPIE Defense, Security, and Sensing (pp. 80590O-80590O). International Society for Optics and Photonics.

[15] Mahoney M. (2012). Data compression explained. mattmahoney. net, updated May, VOL 7.

[16] http://www.maheshsubramaniya.com/article/microsoft-openxml-file-format-docx-pptx-xlsx-a-new-family-of-file-formats.html, [Accessed 14/June/2015].

[17] Moskovitch R., Nissim N., & Elovici Y. (2009). Malicious code detection using active learning. In Privacy, Security, and Trust in KDD (pp. 74-91). Springer Berlin Heidelberg.

Ayyad Mohammad Khamis Naser received the B.S degree in Business Networking and Systems Management from Philadelphia University, Jordan, in 2013, and the M.Sc. Computer Science from Al-Balqa Applied University, at Jordan. His area of interest includes DOCX file analyzing and detecting malicious content and network security.

Mohammad Hjouj Btoush
I have joined Al-Balqa Applied University- Jordan as a lecturer of computer science in 1992. During that time I co-authored a book: Computer Skills. In 2006, I joined the Informatics Research Group at Sheffield Hallam University-UK. My PhD thesis investigates the perception of users and providers of e-services in Jordan/Artificial Intelligence & Software Engineering. My current research interests include: Information security, Information systems evaluation, specifically, within the context of electronic Government, operating system, and software engineering.

- ✓ **Vice Dean for Graduate studies,** Faculty of Prince Abdullah Bin Ghazi of Information Technology, Al-Balqa' Applied University, Sep 2015 – 2016
- ✓ **Vice Dean,** Faculty of Prince Abdullah Bin Ghazi of Information Technology, Al-Balqa' Applied University, Dec 2010 – Sep. 2013.
- ✓ **Director of Computer Center,** Al-Balqa' Applied University, July 2016 – now.

Dr. Ali Hadi received the B.S. degree in computer science from Philadelphia University, Jordan, in 2002 and the M.Sc. and Ph.D. degree in computer information system from University of Banking and Financial Sciences, College of Information Technology, Jordan, in 2004 and 2010, respectively. He's a Senior Level Information Security Officer with 14+ years of professional experience working for different high-reputed companies. Since 2011 he's been teaching different computer security, digital forensics, and networking courses. He's also an author, speaker, and freelance instructor. His research interests include digital forensics, operating systems internals, malware analysis, and network security.