Ransomware: Analysis of Encrypted Files

Houria MADANI[1], Noura OUERDI[2], Abdelmalek Azizi[3]
Faculty of sciences, Mohammed First University, Oujda, Morocco[1, 2, 3]

***Abstract*—Ransomware is a type of malware that damage the system by encrypting all the files existing in the computer. To get access, the victim has to pay a ransom to get a key to decrypt his data. When the virus is running in machine, the user cannot stop it on the first try, so he may lose his entire files. One of the goals of this work is to detect ransomware based on encrypted files in real time and to minimize the cost of losing files. We will try to do an analysis of a received file (without opening it and seeing its contents). This scanning action can prevent a ransomware from spreading in the system. Most Ransomware files are sent in ".exe" format, but in this work, we will try to use other file formats that can accept malware, for example, .doc or .docx, .xls or .xlsx, .ppt or .pptx, .jpg, etc. In fact, an attacker can focus only on the files that contain useful data. In this paper, we are going to identify the types of files if they are suspicious or normal (without opening them) from their headers. For that first, we are going to analyze each extension separately (.docx, .exe, .pptx, .xlsx, .jpg, etc.) by identifying their headers and signatures. Then we will take several files with different extensions to analyze them by doing a program who detect if a file is benign or suspicious.**

***Keywords*—Ransomware; encrypted files; signature; file format; static analysis**

I. Introduction

In recent years, ransomware attacks continue to explode exponentially around the world; the cost keeps falling and exploit different sectors.

Researchers and cybersecurity specialists are still looking for a solution to detect this attack and even to slow down its growth in order to find an effective and reliable solution. We see many solutions, but not 100% sure, because hackers are always attentive and updated with the new technologies, they use more sophisticated techniques to follow the evolution and bypassing the protection techniques.

This study focuses on the examination of the behavior and method in which ransomware encrypts files. Ransomware can infiltrate a device in various formats like .exe, .docx, .ppt, etc. A user may open a .docx file without realizing it is an unsafe file that contains metadata that can damage their computer. Therefore, we aim to analyze the files (without opening them) before and after ransomware encryption, in order to distinguish between a typical file and a suspicious one.

In this paper, we will make a study on files to differentiate between a normal file and a suspicious one. For that in Section II, we will approach some "state of the art" concerning the study of files to give you an idea of the current research on this subject. In Section III, we will see our objectives and working methodology to identify and detect a normal file from another suspect one. We will discuss the results that we have had in Section IV. At the end, we sum up with a conclusion and some perspectives.

*Corresponding Author.

II. State of the Art

As you know, attackers are very inventive when they want to target a victim and we find, often, that emails are the trickiest (more than 90%) way [1] for them to create a link between the attacker and the target. Fig. 1 explains how ransomware attacks your machine:

Ransomware detection techniques [2]–[5] are becoming more and more competitive, and each researcher has his own method and technique. If we take the detection of ransomware or malware in general, using file headers, several researchers work focus on a single file extension like PE (Portable Executable) files [6]–[8], but there is not enough research on the detection of ransomware using the headers of different extensions.

The authors in [9] proposed a new classification model based on machine learning techniques to detect and classify malicious and benign PE files based on their headers information. The experimental results proved that the Random Forest algorithm yields a higher accuracy (99.68%) compared to other algorithms. The tests were performed on 211,067 malware samples obtained from the VirusShare database [10]. Manavi and Hamzeh [11] presented a method for detecting ransomware using the PE header. They used a Convolutional neural network (CNN) to identify ransomware by converting the header bytes into 32*32 pixel images. The use of a header is advantageous, but transforming it into an image would necessitate the use of a network with additional layers in order to extract its features.

To detect ransomware, the authors [12] used a static method. They proposed a method that is based on the bytes extracted from the header of the executable file using LSTM network to build the detection model.

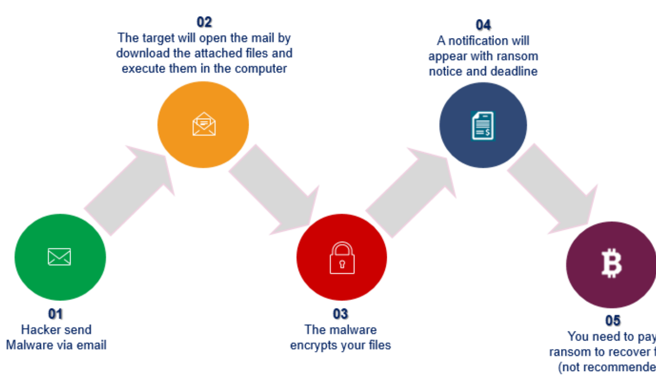

Fig. 1. Ransomware attack phases.

The modification of the file header changes its structure. Therefore, they did the extraction of the executable file headers, then they processed the byte sequence that builds the file header with LSTM network, and they separated the ransomware samples, from the benign samples to form the template. With this technique, they managed to detect ransomware with 93.25% accuracy without running the program.

Subedi et al. [13] employed data mining techniques to recognize and detect ransomware families using both static and dynamic analysis at three different levels: assembly, function calls and library. They also created an analytical tool that uses reverse engineering to create signatures for identifying ransomware families. Arabo et al. [14] proposed a dynamic analysis approach to gather ransomware API properties, which is then utilized to test 9 Machine Learning classifiers and a neural network. The goal of this research is to understand the link between a process's behavior and its nature, to detect if it is a ransomware or not. With a detection rate of 75.01%, Random Forest surpasses other classifiers. The benefit of this technique is that it does not require a signature database, but rather a collection of ransomware and non-ransomware data. The detection rate of the classifiers may be better by improving the dataset.

Before encrypted files were moved to a backup disk, Lee et al. [15] utilized machine learning techniques to detect and classify infected files. The training step was implemented at the backup system according to their recommendation. It identified files from various users and file types, as well as determining file entropy thresholds. These thresholds were transmitted to client hosts in order to decide whether a new version of the file was encrypted or not. The authors in [16] suggest a two-stage mixed ransomware detection approach using Markov model with the Random Forest technique to detect ransomware. Random Forest has the best detection rate of 97.3%.

The paper [17] emphasizes the capabilities of behavior-based detection mechanisms to identify crypto ransomware, demonstrating the limitations of signature-based detection approaches. In [18], Nieuwenhuizen proposed a ransomware detection scheme using behavior analysis and machine learning. Although the specific features were not revealed, their created feature set included properties such as payload persistence, anti-system restoration, stealth methods, environment mapping, network traffic, and privilege elevation that were extracted from the behavior of a malicious set up. Author employed the support vector machine (SVM) method as the classification technique in addition to the behavioral features related with data transformation behavior, such as huge file encryption.

The effect of certain ransomware families on the Windows platform is demonstrated and analyzed by Mohammad [19]. He deduces that most families of ransomware behave in a similar way when it comes to affect file system and registry entities. Furthermore, all types of ransomware generate files in the Windows system files and rename other files. To do the experiments, the author used Windows 7, Oracle VirtualBox VM, Cuckoo sandbox, and Virtual windows 10. The author concludes that monitoring system file and registry activities can protect against ransomware. He also mentions that Windows 10 is more effective than Windows 7 regarding malware. The best method to follow as a recommendation is to regularly back up company or individual data.

III. Methodology

As mentioned at the beginning, our goal is to detect whether a file is suspicious or not (regardless of its content), from its header which will be identified from its extension. This leads us to detect ransomware from encrypted files in real time.

It is well known that each extension has a fixed header according to the standards. If the header differs from the standard state, we deduce that it is suspicious.

To achieve our goal, we took several files with different extensions, if we take an extension, for example ".docx", and we open some files with the same extension using the Hexadecimal editor (Hex Editor Neo [20]), we found that they have the same signature, also called "Magic number".

Namely, each file extension (.docx, .pptx, .xls, .exe, .dll, .jpg, etc.) has a fixed and specific "Magic number".

Table I shows some files with different extensions and their signatures or Magic number, for clear and normal files.

According to a deep study on Microsoft Office files, we notice that their signature is different, the "x" added at the end made many differences. If we take the extensions .doc and .docx (the same thing for .xls, .ppt/.xlsx, .pptx), the differences are seen in Table II.

TABLE I. Signature of Files with Different Extensions

File extension	Hex signature	Size	ASCII Signature
.doc , .ppt , .xls	**D0 CF 11 E0 A1 B1 1A E1**	8 Bytes	ĐÏ.à¡±.á
.docx , .pptx , .xlsx	**50 4B 03 04 14 00 06 00**	8 Bytes	PK......
.pdf	**25 50 44 46**	4 Bytes	%PDF-
.png	**89 50 4E 47 0D 0A 1A 0A**	8 Bytes	.PNG….
.jpg	**FF D8 FF E0**	4 Bytes	ÿØÿà
.dll	**4D 5A 90 00**	4 Bytes	MZ
.exe	**4D 5A**	2 Bytes	

TABLE II. Difference Between .Doc and .Docx

	DOC	DOCX
Version	Came in when the Microsoft Word was 1st delivered and was utilized until 2003 variant of "Word".	Came in with Word 2007 and has been the default extension from then for all new Word versions.
Storage	A DOC is saved in a binary file that contains all the related formatting and relevant informations.	A DOCX file is actually a zip file with all XML files associated with the document.
File size	The DOC format has a ***greater*** size than the DOCX format.	The DOCX format has a ***smaller*** size than the DOC format.

To analyze files, we have taken some files (.docx, .doc, .pdf, .exe, .png) as an example, and see their structures (this study of files is very difficult to make because there is a lack of information on the structure of the files, for example, for the header "how many bytes occupied", for the contents "where does it begin?", etc.):

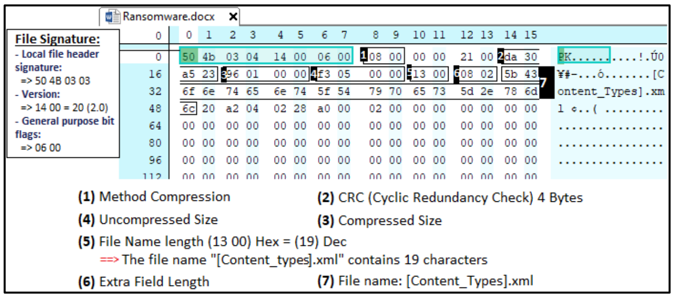

Fig. 2. ".Docx" File signature.

From the article [21], the header is always at the beginning of the file and is exactly 512 bytes in length. Fig. 2 and Fig. 3 show you some information about the header of the **".docx"** and **".doc"** file, respectively.

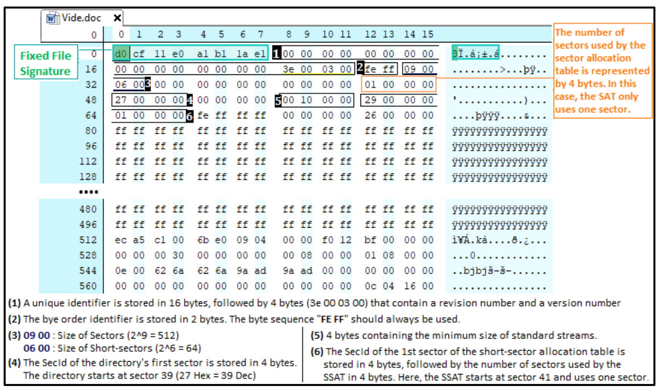

Fig. 3. ".doc" File signature (empty file).

Fig. 4 and 5 show you the structure and header of the ".pdf" and ".exe" files, respectively.

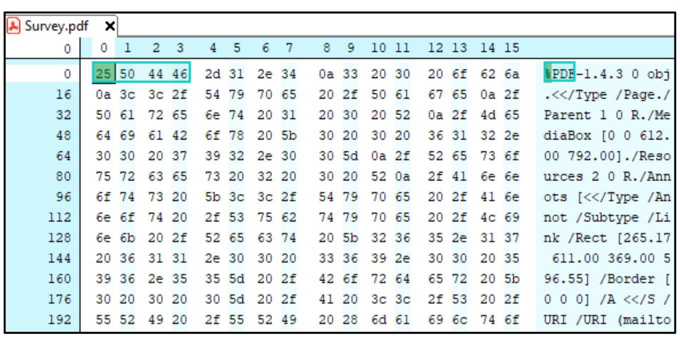

Fig. 4. ".PDF" File Signature.

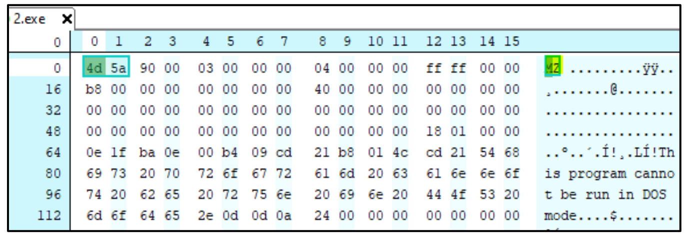

Fig. 5. ".exe" file signature.

IV. Results and Discussion

We took a corpus[22] that contains a large number of files with different extensions, and I encrypted them with a python program, adding to its files an '.enc' extension to make the difference between a clear file and an encrypted file. As an example, I took four files for each different extension (.doc, .docx, .ppt, .pdf); we got the following result:

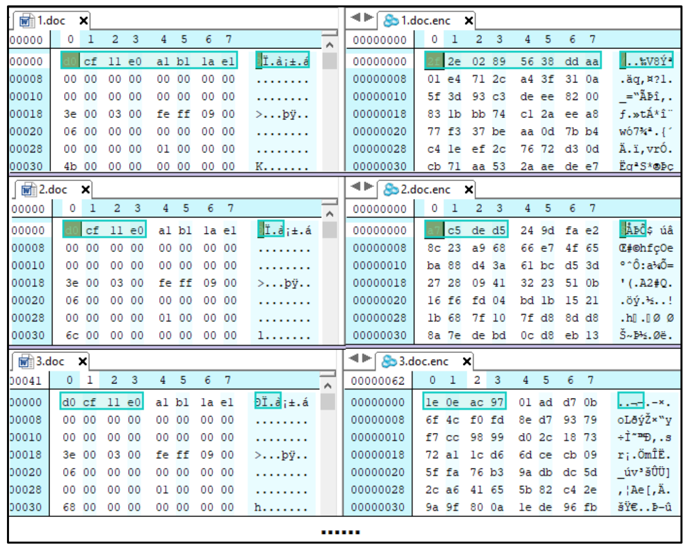

Fig. 6. The difference between encrypted and clear file headers [.doc].

For the files " *.doc " (in Fig. 6), those on the left are clear files, their header should be normal [D0 CF 11 E0 A1 B1 1A E1]. While on the right, you see that there is an extension added at the end " *.doc.enc ", this means that they are encrypted files (the encrypted file of each clear file, e.g. "1.doc.enc" is the encrypted file of "1.doc"), and even their header is different. What is relevant is that each encrypted file has a different header from the other file, we have [2F 2E 02 89 56 38 DD AA], [A7 C5 DE D5 24 9D FA E2], etc.

The same thing for the files " *.docx " (in Fig. 7), those on the left are clear files, their header should is [50 4B 03 04 14 00 06 00], while on the right you see the encrypted files and even their header is different. We have [A1 D1 05 C2 6C 8B 1D 19], [EF 34 21 9F FE 78 65 FC], etc.

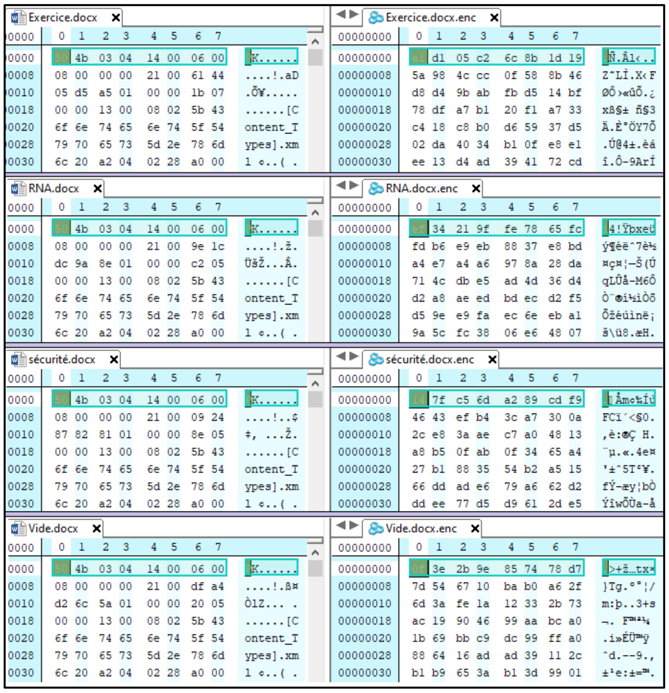

Fig. 7. The difference between encrypted and clear file headers [.DOCX].

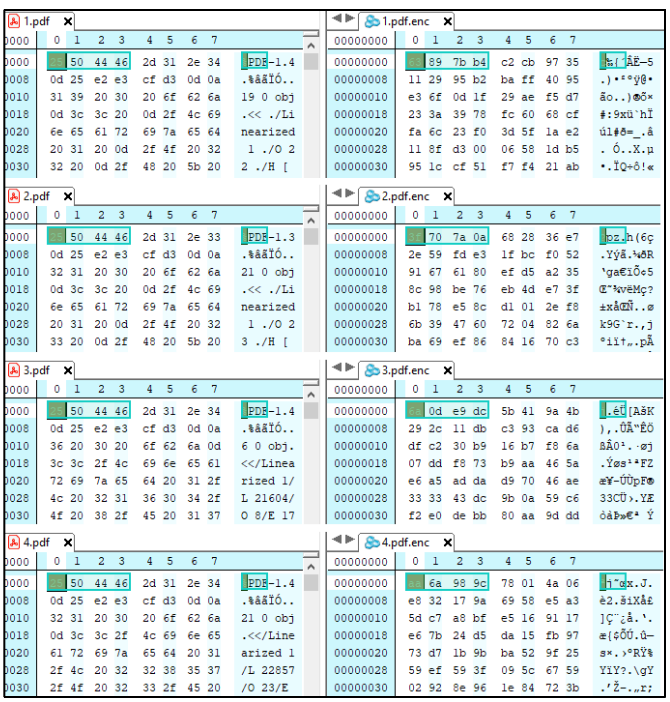

Fig. 8. The difference between encrypted and clear file headers [.PDF].

As you can see in Fig. 6, Fig. 7 and Fig. 8, the signature of a clear file and its encrypted is not the same; the case is the same for the other extensions. We also notice that the signature is fixed for any clear file (with different extension), but for encrypted files, it is not fixed and differs between each file.

The program is done by Python language to make this study and detect if a file is suspicious or normal from its signature.

Each extension has a "Magic Byte". We instantiated our dataset by creating a dictionary with the file extension as a key and its "Magic byte" as value, and then we analyze the file. If the file does not contain the corresponding signature, i.e. it has a different header than the one presented in our dataset; we deduce that it is a suspect file. We have also dealt with the case of a file without extension, if we give it to our program, it analyzes the header and if it does not find the corresponding signature, it sends us back that it's a suspicious file, otherwise, if everything is normal the result is: "This is a benign file, its extension is: … ". Fig. 9 shows the result of a file without extension that is benign.

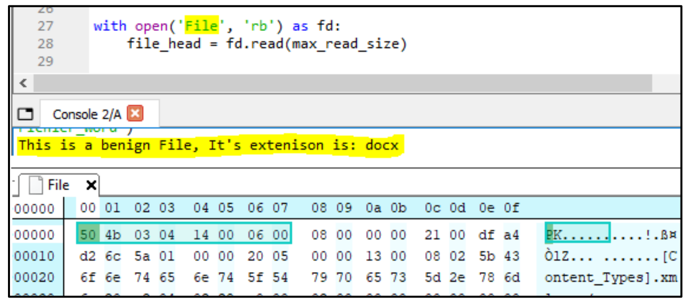

Fig. 9. Analysis of a file without extension.

Fig. 10 shows you that we have performed an analysis for several files with different extensions.

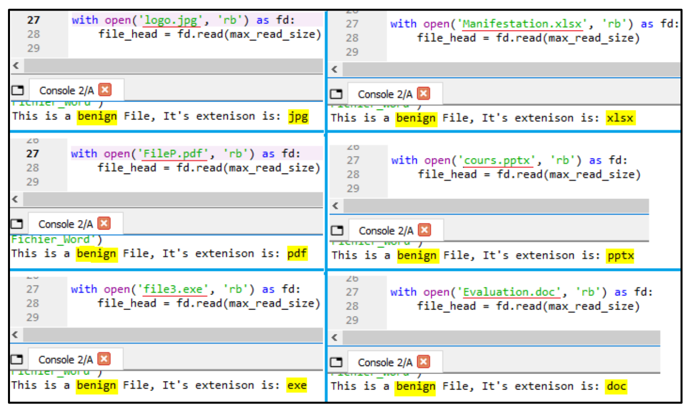

Fig. 10. Analysis of several normal files with different extensions.

If we take the example in Fig. 11, you can see that the result is "this is a suspicious file", even though the file has the extension ".doc". In effect, sometimes attackers send files that look normal with a legal extension, while the file is infected by the ransomware, so as you can see, our program perfectly analyzes the header of the given file identifying its signature, and it found that its signature does not match to the normal signatures.

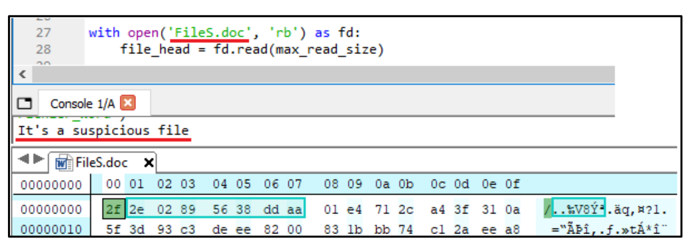

Fig. 11. Analysis of a suspicious file.

We have tested our program on files encrypted by Ransomware with ".lmas", as you can see in Fig. 12, we have taken as an example the file “formation.xlsx.lmas”, the “.lmas” extension is added after ".xlsx" extension. We got the followed result:

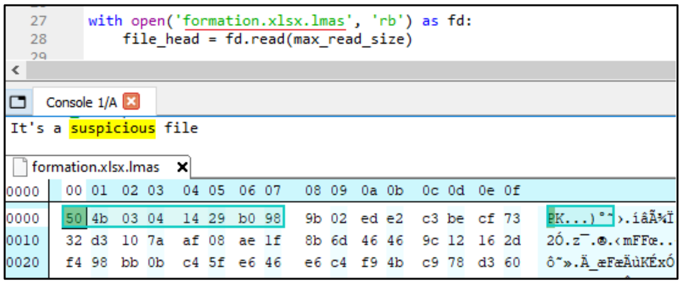

Fig. 12. Analysis of a file encrypted by Ransomware.

As you can see, Ransomware infects the file «formation.xlsx», it is encrypted and the attacker has added the extension ".lmas" to the file.

We know that "xlsx" has a fixed signature (see Table I); in the Fig. 12, we can see that the first 4 bytes are similar to the first 4 bytes of the normal xlsx file (50 4B 03 04), but the difference is in the next 4 bytes. Therefore, our program was able to detect that this file is encrypted by ransomware so it is a suspicious file without opening it.

V. Conclusion

In this work, we have made a Python program that allows to detect a suspicious file from another normal one, we started by studying the header of files of different extensions separately, later we extracted the header of each file and compared the headers of a normal file with another encrypted one. With this study, we could deduce that a normal file has a fixed and unchangeable extension, once it is changed the file is suspicious.

In the upcoming work, we will conduct a dynamic analysis by executing ransomware files in a simulated environment. This will allow us to extract ransomware encrypted files and analyze them in order to develop and implement our own neural network. This network will be trained to identify ransomware files by first learning the characteristics extracted from the ransomware encrypted files, and then using that knowledge to detect ransomware when a "vulnerable" file is downloaded onto a victim's device.

References

[1] « Cyberattacks 2021: Statistics From the Last Year », Spanning, 18 janvier 2022. https://spanning.com/blog/cyberattacks-2021-phishing-ransomware-data-breach-statistics/ (consulté le 7 mars 2022).

[2] S. Kok, A. Abdullah, N. Jhanjhi, et M. Supramaniam, « Ransomware, threat and detection techniques: A review », Int. J. Comput. Sci. Netw. Secur, vol. 19, no 2, p. 136, 2019.

[3] C. V. Bijitha, R. Sukumaran, et H. V. Nath, « A Survey on Ransomware Detection Techniques », in Secure Knowledge Management In Artificial Intelligence Era, vol. 1186, S. K. Sahay, N. Goel, V. Patil, et M. Jadliwala, Éd. Singapore: Springer Singapore, 2020, p. 55-68. doi: 10.1007/978-981-15-3817-9_4.

[4] C. Beaman, A. Barkworth, T. D. Akande, S. Hakak, et M. K. Khan, « Ransomware: Recent advances, analysis, challenges and future research directions », Computers & Security, vol. 111, p. 102490, déc. 2021, doi: 10.1016/j.cose.2021.102490.

[5] S. I. Bae, G. B. Lee, et E. G. Im, « Ransomware detection using machine learning algorithms », Concurrency Computat Pract Exper, vol. 32, no 18, sept. 2020, doi: 10.1002/cpe.5422.

[6] S. Poudyal, K. D. Gupta, et S. Sen, « PEFile Analysis: A Static Approach To Ransomware Analysis », The International Journal of Forensic Computer Science, vol. 14, p. 34-39, oct. 2019, doi: 10.5769/J201901004.

[7] T. Rezaei et A. Hamze, « An Efficient Approach For Malware Detection Using PE Header Specifications », in 2020 6th International Conference on Web Research (ICWR), Tehran, Iran, avr. 2020, p. 234-239. doi: 10.1109/ICWR49608.2020.9122312.

[8] V. Verma, S. K. Muttoo, et V. B. Singh, « Detection of Malign and Benign PE Files Using Texture Analysis », in Information Systems Security, Cham, 2020, p. 253-266. doi: 10.1007/978-3-030-65610-2_16.

[9] I. Abdessadki et S. Lazaar, « A New Classification Based Model for Malicious PE Files Detection », International Journal of Computer Network and Information Security, vol. 11, p. 1-9, juin 2019, doi: 10.5815/ijcnis.2019.06.01.

[10] « VirusShare.com ». https://virusshare.com/ (consulté le 25 février 2022).

[11] F. Manavi et A. Hamzeh, « A New Method for Ransomware Detection Based on PE Header Using Convolutional Neural Networks », in 2020 17th International ISC Conference on Information Security and Cryptology (ISCISC), Tehran, Iran, sept. 2020, p. 82-87. doi: 10.1109/ISCISC51277.2020.9261903.

[12] F. Manavi et A. Hamzeh, « Static Detection of Ransomware Using LSTM Network and PE Header », in 2021 26th International Computer Conference, Computer Society of Iran (CSICC), Tehran, Iran, mars 2021, p. 1-5. doi: 10.1109/CSICC52343.2021.9420580.

[13] K. P. Subedi, D. R. Budhathoki, et D. Dasgupta, « Forensic Analysis of Ransomware Families Using Static and Dynamic Analysis », in 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, mai 2018, p. 180-185. doi: 10.1109/SPW.2018.00033.

[14] A. Arabo, R. Dijoux, T. Poulain, et G. Chevalier, « Detecting Ransomware Using Process Behavior Analysis », Procedia Computer Science, vol. 168, p. 289-296, 2020, doi: 10.1016/j.procs.2020.02.249.

[15] K. Lee, S.-Y. Lee, et K. Yim, « Machine Learning Based File Entropy Analysis for Ransomware Detection in Backup Systems », IEEE Access, vol. 7, p. 110205-110215, 2019, doi: 10.1109/ACCESS.2019.2931136.

[16] J. Hwang, J. Kim, S. Lee, et K. Kim, « Two-Stage Ransomware Detection Using Dynamic Analysis and Machine Learning Techniques », Wireless Pers Commun, vol. 112, no 4, p. 2597-2609, juin 2020, doi: 10.1007/s11277-020-07166-9.

[17] P. S. Goyal, A. Kakkar, G. Vinod, et G. Joseph, « Crypto-ransomware detection using behavioural analysis », in Reliability, Safety and Hazard Assessment for Risk-Based Technologies, Springer, 2020, p. 239-251.

[18] D. Nieuwenhuizen, « A behavioural-based approach to ransomware detection », Whitepaper. MWR Labs Whitepaper (2017), p. 20.

[19] A. Mohammad, Analysis of Ransomware on Windows platform, International Journal of Computer Science and Network Security 20.6 (2020): 21-27. 2020. doi: 10.13140/RG.2.2.11150.59202.

[20] « Free Hex Editor: Fastest Binary File Editing Software. Freeware. Windows ». https://www.hhdsoftware.com/free-hex-editor (consulté le 25 février 2022).

[21] D. Rentz, « The Microsoft Compound Document File Format », [Internet]. Available: http://www.openoffice.org.zaxyproxy.com, p. 25, août 2007.

[22] « Digital Corpora Downloads: corpora/files/govdocs1/threads/ ». https://downloads.digitalcorpora.org/corpora/files/govdocs1/threads/ (consulté le 7 mars 2022).