JOIV : Int. J. Inform. Visualization, 6(4) - December 2022 856-861

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Entropy Based Method for Malicious File Detection

Muhammad Edzuan Zainodin [a,*], Zalmiyah Zakaria [a], Rohayanti Hassan [a], Zubaile Abdullah [b]

[a] School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310, Skudai, Johor, Malaysia
[b] Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, Malaysia
Corresponding author: *muhammadedzuan@graduate.utm.my

Abstract— Ransomware is by no means a recent invention, having existed as far back as 1989, yet it still poses a real threat in the 21st century. Given the increasing number of computer users in recent years, this threat will only continue to grow, affecting more victims as well as increasing the losses incurred towards the people and organizations impacted in a successful attack. In most cases, the only remaining courses of action open to victims of such attacks were the following: either pay the ransom or lose their data. One commonly shared behavior by all crypto ransomware strains is that there will be attempts to encrypt the victims' files at a certain point during the ransomware execution. This paper demonstrates a technique that can identify when these encrypted files are being generated and is independent of the strain of the ransomware. Previous research has highlighted the difficulty in differentiating between compressed and encrypted files using Shannon entropy, as both file types exhibit similar values. Among the experiments described in this study, one showed a unique characteristic for the Shannon entropy of encrypted file header fragments, which was used to differentiate between encrypted files and other high entropy files such as archives. The Shannon entropy of encrypted file header fragments has a unique characteristic in one of the tests discussed in this study. This property was used to distinguish encrypted files from other files with high entropy, such as archives. To overcome this drawback, this study proposed an approach for test case generation by enhancing the entropy-based threat tree model, which would improve malicious file identification. The file identification was enhanced by combining three entropy algorithms, and the test case was generated based on the threat tree model. This approach was then evaluated using accuracy measurements: True Positive, True Negative, False Positive, False Negative. A promising result is expected. This method solves the challenge of leveraging file entropy to distinguish compressed and archived files from ransomware-encrypted files in a timely manner.

***Keywords*— Entropy; malicious; ransomware.**

Manuscript received 15 Jan. 2022; revised 16 Aug. 2022; accepted 7 Oct. 2022. Date of publication 31 Dec. 2022.

I. Introduction

Software testing is a type of research used to offer information to stakeholders regarding the quality of the software product being tested. Typically, the test procedures involve running a program or application to find software faults (errors or other problems) to ensure that the software product meets the desired requirements as well as determine its vulnerabilities to security issues. As a result, security testing has become an integral part of the software development process, manifesting itself in a variety of actions aimed at ensuring a certain level of security for the application in development.

Security, however, can be considered a typical non-functional requirement, and testing approaches can be customized to meet those needs. In this context, we look at a new way for creating test cases for application security testing, notably in the event of ransomware. Ransomware is a sort of malicious software that encrypts data and attacks a computer system's availability. Until the ransom is paid, the attacker encrypts the victim's data and holds it hostage.

The ransomware infection has grown in scope, cost, complexity, and impact since its discovery roughly 30 years ago. Security experts and ransomware producers are constantly engaged in an arms race, with the former acting as the defender aiming to protect their digital infrastructure from ransomware attacks and the latter being the attacker aiming to assault said infrastructure. As a result of recent changes in working practices as a result of COVID-19, which prompted a greater proportion of individuals to work from home, ransomware attacks have spread, and attack mechanisms have changed.

Given a set of mixed files, the challenge is to generate the test cases to identify the malicious threat of each file. The test cases should be generated from the testing threat model

representing the modified entropy analysis technique that focuses on reducing the false positive rate detection.

Several researchers have investigated the use of file entropy as a reliable approach for identifying encrypted files. Keep the following in mind while evaluating the proposed detection methods in the reviewed research: Distinguishing between compressed and encrypted files by using the file entropy value is problematic because the total entropy values for multiple file formats are often identical.

This research was implemented based on some research questions. The research questions are as follows:

- RQ1: What are the current entropy analysis techniques to identify malicious files?
- RQ2: How to improve the entropy analysis technique to reduce false positive rate detection?

The first factor that is being considered is the small difference in entropy values between encrypted and compressed files. Previous studies using Shahnon's entropy only focused on processing files at the header. Some works use Shannon's entropy and focus on processing the file as a whole [1]–[41]. Therefore, in this study, a particular file (or file type) can be characterized by a bit value representing its information content. For instance, text files containing written English have been identified as having a file entropy of 3.25 to 4.5 bits. Compressed files, such as ZIP archives, have a higher entropy level, typically just over six bits [20]. This led to using the entropy of a file being written as a means to determine the existence of malicious activities for some cases of malware detection.

However, the use of file extensions as an indicator is open to abuse since the extension of a file or its magic number can be changed by the attacker at any time, which in turn could prevent the identification of said files by the operating system and thus allow the attacker to evade detection. The major concern of using entropy for file type classification is that when considering entropy as a gauge, most compressed and encrypted data share similar characteristics, which means that more work is required to investigate the application of entropy to these file types. This study also used a threat tree to model the file type identification. Based on the threat tree model, test cases can be generated.

II. Materials and Methods

This section describes famous applications and techniques commonly used in big data stream processing. In addition to that, several issues related to these applications were discussed. Aside from that, the research methodology model that has been adopted in order to carry out this study is also explained. It is known as an entropy-based threat testing model for malicious file detection. It comes in five phases: feasibility study and collection of data, literature review, improved malicious file identification, the result analysis, and documentation, as shown in Figure 1.

Phase 1 of the research process consists of a feasibility study and data collection. The research would include a complete review of the research's domain of interest, as well as the benefits and drawbacks. This phase is divided into four sections: identifying problems in the domain of interest, defining the study's goal and objectives and deciding the scope of the requirement analysis. The data and other materials available for experimental purposes are the most important outcome of this review. The researcher would be able to define their study problem in a meaningful framework. The researcher would then rewrite the research problems in as much detail as feasible.

This research uses a public dataset, the Govdocs1 dataset, which is widely used in digital forensics. It was introduced by Garfinkel et al. (2009), and it can be downloaded from http://digitalcorp ora.org/corp/nps/files/govdocs1/. Table 1 tabulates the file types of samples in the dataset.

TABLE I
File types samples

Type of sample files	Number of sample files	File size range
PDF	10	4-91KB
DOC	10	26-467 KB
TXT	10	1-112KB
PPT	10	35-998KB
ZIP	10	2-41KB
GIF	10	3-114KB
7z	10	2-31KB
JPG	10	13-137KB
PS	10	32-726KB

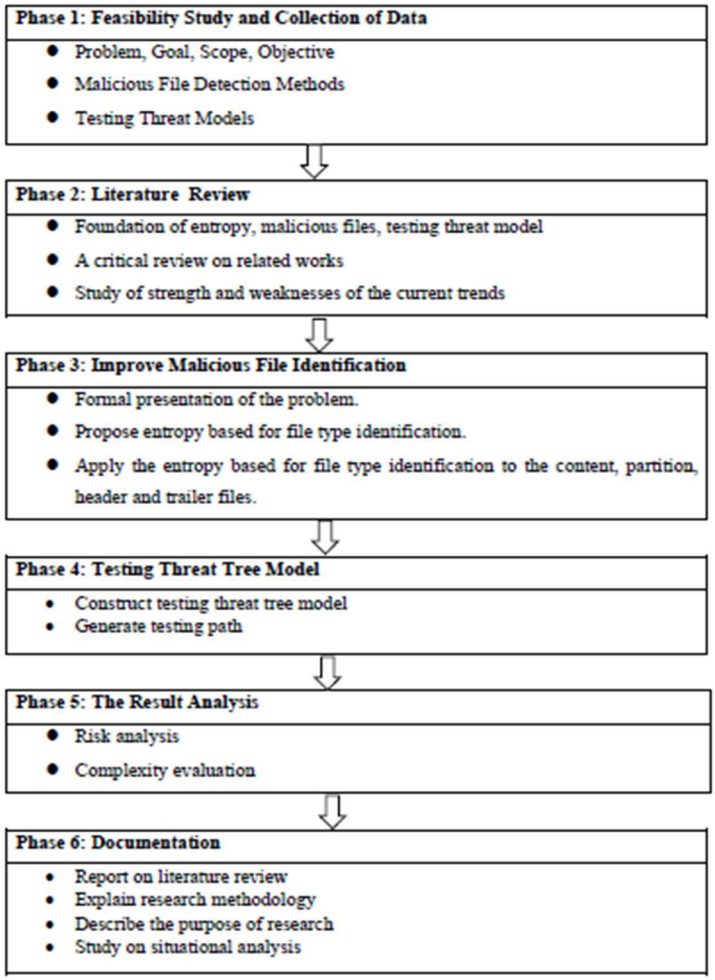

Fig. 1 Overview of the research process

Fig. 2 Steps for entropy-based file identification

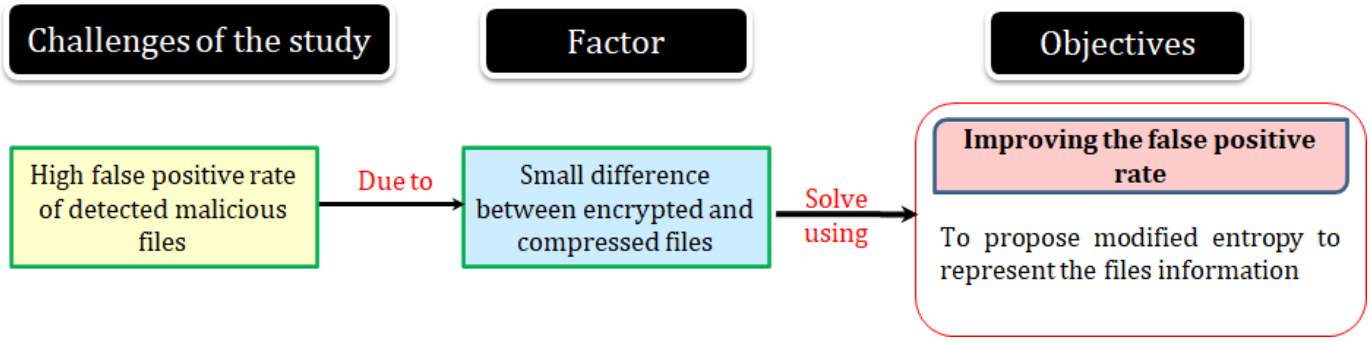

Fig. 3 Challenge, factor, and objective of the study

This study's second part focuses on the literature review (Phase 2). Before beginning the investigation, this step includes defining the foundations of knowledge relevant to the issues in malicious file detection and testing threat modeling. The researcher conducts an exhaustive literature search relating to the research problem at this phase. As a first stage, various publicly available publications such as academic journals, conference proceedings, reports, and books are examined and reviewed. It starts by identifying the present labor that goes into compiling the entropy and threat model data. This study's associated works include a review of informal domain modelling, goal modeling techniques, metrics for measuring risk and complexity, as well as the study's trends and directions. This phase also identifies the strengths and flaws of related works so that improvements can be made and implemented in this study.

An enhanced technique can be utilized to quickly identify the generation of these encrypted files during this phase (Phase 3: Improve Malicious File Identification). Only the first few bytes of the file being written are tested, and this file sample is analyzed to determine whether the file being written is encrypted or not. This study will implement Shannon, Renyi, and Tsallis' entropy. In theory, the disclosed technique may be used to warn the user of questionable behavior, prevent files from being written, or initiate a live forensic investigation. Although this strategy will not eliminate data exfiltration before encryption, it is beneficial in stopping the encryption of the user's data, thereby neutralizing a substantial amount of the attack. Figure 2 shows the steps taken to identify the malicious file.

Meanwhile, in Figure 3, the study's challenge, factor and objective are shown to align the steps involved. In Phase 4, evaluation and analysis of this study are conducted. This study considers whether the requirement is feasible and adequate to be implemented based on the confidence factor evaluation that has been carried out. The documentation phase is the last step in the research process. This section explains the documentation needs, such as the objectives and literature review. The final conclusions are based on the study's overall findings. The documentation stage is the penultimate in the final step before the research product can be given.

The algorithms involved in this study are shown below:

1) *Shannon entropy:*

$$H_\partial(x) = \sum_{i=1}^{n} p(x_i) \log_\partial \frac{1}{p(x_i)} \quad (1)$$

2) *Renyi entropy:*

$$H_\partial(x) = \frac{1}{1-\partial} \log_\partial \left(\sum_{i=1}^{n} p(x_i)^\partial\right) \quad (2)$$

3) *Tsallis entropy:*

$$H_\partial(x) = \frac{1}{1-\partial} \left(\sum_{i=1}^{n} p(x_i)^\partial - 1\right) \quad (3)$$

Where H=entropy
N= number of bytes in the sample
$p(x_i)$ = probability of byte i appearing in the stream of bytes
∂=parameterized value

The two public datasets mentioned earlier, consisting of the binary and textual file types from "Govdocs1 dataset", are used for the evaluation of the proposed feature selection techniques. The performances of the feature selection technique are measured based on average accuracy, using the following formulas;

$$\text{Overall Accuracy} = \frac{\sum Number of true positives for all types}{Total number of file types} \quad (4)$$

$$\text{Accuracy} = \frac{Number\ of\ true\ positives}{Number\ of\ true\ positives + Number\ of\ false\ positives} \quad (5)$$

III. Results and Discussion

In this chapter, the initial results of the study are presented. The dataset used in the experiments is downloaded from govdocs1 website. 10 files from each file type were selected randomly for the experiments. The sample file was randomly selected and compressed into zipping and 7z files for the compressed file. For encrypted ransomware files, the sample files were exposed to the ransomware Wannacry and Cerber strain in an isolated testing environment. The encrypted ransomware files were then downloaded to the testing environment for analysis. Each sample file was analyzed 33 times, starting with the first 8 bytes of the file content up to the first 256 bytes in the 8 bytes increment for each run. At the final run of the experiment, the entire file content was analyzed. At each run, the entropy value of each file was calculated using Shannon entropy, Renyi entropy, and Tsallis

entropy. For Renyi and Tsallis entropy, alpha value of 2, 3, 4 & 5 was run. The average of each entropy profile was presented in Tables 2 and 3.

After the entropy values are produced, the file types with the entropy of both Shannon and Renyi entropy with values above 7 are analyzed further for encrypted file identification. From table 2, WannaCry, ZIP, GIF, and 7z are selected for further analysis. Files encrypted with Cerber strain of ransomware is excluded due to the files having a low entropy value for Renyi entropy. This is due to the nature of Cerber ransomware which encrypts the second half of the file content rather than the entire content. File types having entropy values less than seven are also excluded from further analysis.

The file type having an entropy value of more than seven are then analyzed by calculating the average entropy of each file of the sample file. The entire content is calculated by Shannon and Renyi entropy of the first 256 bytes. Then for each averaged entropy calculated, a base value is selected: the 256 bytes having a base value of 7 and the entire file content having a base value of 8. After that, a difference between the base value and the average entropy value is calculated.

IV. CONCLUSION

In this paper, we concluded that modeling the threat file identification in security testing is carried out not only to define users' needs, objectives and functions, but synthesized solutions should also be conducted iteratively to optimize performance requirements. Two challenges have been discovered in this research, namely cases where regular files with a high header entropy were identified as ransomware encrypted (False Positive), and cases where files encrypted by ransomware with a low header entropy were classified as normal files, resulting in classification errors (False Negative). As a result, there is a problem with the high false positive rate of dangerous files discovered. In future work, we plan to build a threat testing model based on test cases generated from this method.

TABLE II
AVERAGE ENTROPY VALUE FOR ENTIRE FILE CONTENT

File type	Entropy						
	Shannon	**Renyi**	**Tsallis**	**Avg (S+R+T)**	**Avg(S+R)**	**Avg(S+T)**	**Avg(R+T)**
WannaCry	**7.97867**	**7.93006**	133.3323225	49.7470175	**7.954365**	70.65549625	70.63119125
Cerber	**7.60988**	6.17962	133.33092	49.04014	6.89475	70.4704	69.75527
ZIP	**7.91613**	**7.4883875**	133.332205	49.5789075	**7.70225875**	70.6241675	70.41029625
GIF	**7.92861**	**7.449405**	133.332125	49.57004667	**7.6890075**	70.6303675	70.390765
7z	**7.90206**	**7.2914375**	133.332155	49.50855083	**7.59674875**	70.6171075	70.31179625
PDF	**7.35673**	5.335295	133.32926	48.67376167	6.3460125	70.342995	69.3322775
DOC	3.59757	1.2974275	128.20016	44.3650525	2.44749875	65.898865	64.74879375
PPT	6.24189	3.506915	132.6192325	47.4560125	4.8744025	69.43056125	68.06307375
TXT	4.7267	3.307095	39.7203075	15.91803417	4.0168975	22.22350375	21.51370125
PS	4.77113	3.660555	47.999725	18.81047	4.2158425	26.3854275	25.83014

TABLE III
AVERAGE ENTROPY VALUE FOR THE FIRST 256 BYTES OF FILE CONTENT

File type	Entropy						
	Shannon	**Renyi**	**Tsallis**	**Avg(S+R+T)**	**Avg(S+R)**	**Avg(S+T)**	**Avg(R+T)**
WannaCry	**7.16567**	6.759155	84.0605225	32.6617825	6.9624125	45.61309625	45.40983875
Cerber	4.56806	3.375885	21.960515	9.968153333	3.9719725	13.2642875	12.6682
ZIP	**7.00185**	5.6756825	80.8826225	31.18671833	6.33876625	43.94223625	43.2791525
GIF	5.96619	4.7886025	60.65012	23.8016375	5.37739625	33.308155	32.71936125
7z	**7.00185**	5.6756825	80.8826225	31.18671833	6.33876625	43.94223625	43.2791525
PDF	3.97682	2.452395	23.9697025	10.1329725	3.2146075	13.97326125	13.21104875
DOC	1.50822	0.7793525	10.9956225	4.427731667	1.14378625	6.25192125	5.8874875
PPT	1.81374	0.9779375	13.0091025	5.266926667	1.39583875	7.41142125	6.99352
TXT	4.61647	3.6404575	21.44138	9.899435833	4.12846375	13.028925	12.54091875
PS	5.20101	4.4181	28.3759575	12.6650225	4.809555	16.78848375	16.39702875

TABLE IV
AVERAGE ENTROPY VALUE FOR WANNACRY FILE TYPE

	Avg(S+R) 256 bytes	**Difference 256 bytes**	**Avg(S+R) Entire content**	**Difference Entire content**	**Total Difference (256+Entire content)**
File 1	6.85236	0.14764	**7.94946**	0.05054	0.19818
File 2	6.76036	0.23964	**7.9185**	0.0815	0.32114
File 3	6.94774	0.05226	**7.9842**	0.0158	0.06806
File 4	**7.01266**	-0.01266	**7.91544**	0.08456	0.0719
File 5	6.85536	0.14464	**7.98324**	0.01676	0.1614
File 6	6.65042	0.34958	**7.88644**	0.11356	0.46314
File 7	6.84872	0.15128	**7.9342**	0.0658	0.21708
File 8	6.74204	0.25796	**7.92858**	0.07142	0.32938
File 9	6.83116	0.16884	**7.9973**	0.0027	0.17154
File 10	6.90376	0.09624	**7.90046**	0.09954	0.19578

TABLE V
AVERAGE ENTROPY VALUE FOR ZIP FILE TYPE

	Avg(S+R) 256 bytes	Difference 256 bytes	Avg(S+R) Entire content	Difference Entire content	Total Difference (256+Entire content)
File 1	6.28058	0.71942	**7.51252**	0.48748	1.2069
File 2	6.34272	0.65728	**7.52944**	0.47056	1.12784
File 3	6.13634	0.86366	**7.92576**	0.07424	0.9379
File 4	6.34004	0.65996	**7.36352**	0.63648	1.29644
File 5	6.15814	0.84186	**7.69188**	0.30812	1.14998
File 6	6.23334	0.76666	**7.22358**	0.77642	1.54308
File 7	6.17274	0.82726	**7.60302**	0.39698	1.22424
File 8	6.32036	0.67964	**7.55924**	0.44076	1.1204
File 9	6.29052	0.70948	**7.98018**	0.01982	0.7293
File 10	6.36994	0.63006	**7.35022**	0.64978	1.27984

TABLE VI
AVERAGE ENTROPY VALUE FOR GIF FILE TYPE

	Avg(S+R) 256 bytes	Difference 256 bytes	Avg(S+R) Entire content	Difference Entire content	Total Difference (256+Entire content)
File 1	1.59032	5.40968	**7.44516**	0.55484	5.96452
File 2	6.39662	0.60338	**7.87034**	0.12966	0.73304
File 3	6.40012	0.59988	**7.86686**	0.13314	0.73302
File 4	5.97978	1.02022	**7.8137**	0.1863	1.20652
File 5	5.54048	1.45952	**7.9382**	0.0618	1.52132
File 6	5.6066	1.3934	**7.93588**	0.06412	1.45752
File 7	6.09748	0.90252	**7.96512**	0.03488	0.9374
File 8	5.47818	1.52182	**7.90562**	0.09438	1.6162
File 9	1.51078	5.48922	6.72638	1.27362	6.76284
File 10	5.64084	1.35916	5.9852	2.0148	3.37396

TABLE VII
AVERAGE ENTROPY VALUE FOR 7Z FILE TYPE

	Avg(S+R) 256 bytes	Difference 256 bytes	Avg(S+R) Entire content	Difference Entire content	Total Difference (256+Entire content)
File 1	5.8296	1.1704	**7.4322**	0.5678	1.7382
File 2	5.9888	1.0112	**7.37184**	0.62816	1.63936
File 3	5.81372	1.18628	**7.88656**	0.11344	1.29972
File 4	5.94398	1.05602	**7.12868**	0.87132	1.92734
File 5	5.94098	1.05902	**7.4776**	0.5224	1.58142
File 6	6.01536	0.98464	6.96046	1.03954	2.02418
File 7	5.9053	1.0947	**7.29602**	0.70398	1.79868
File 8	6.04482	0.95518	**7.46848**	0.53152	1.4867
File 9	6.02254	0.97746	**7.98084**	0.01916	0.99662
File 10	5.90406	1.09594	**7.13294**	0.86706	1.963

ACKNOWLEDGMENTS

This research was supported by Universiti Tun Hussein Onn Malaysia (UTHM) Tier 1 (Vot Q027). Universiti Teknologi Malaysia also supported this study under UTMER FASA 1/2020 (VOT NO 19J59).

REFERENCES

[1] Frank Swiderski, W. S. (2004). Threat Modeling. Microsoft PressDiv. of Microsoft Corp. One Microsoft Way Redmond, WA United States.

[2] Amirani, M. C., Toorani, M., & Shirazi, A. A. B. (2008). A new approach to content-based file type detection. Proceedings - IEEE Symposium on Computers and Communications. https://doi.org/10.1109/ISCC.2008.4625611

[3] Ammann, P., & Offutt, J. (2016). Introduction to Software Testing. In Introduction to Software Testing. https://doi.org/10.1017/9781316771273

[4] Bajpai, P., & Enbody, R. (2020). An Empirical Study of Key Generation in Cryptographic Ransomware. International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2020. https://doi.org/10.1109/CyberSecurity49315.2020.9138878

[5] Bat-Erdene, M., Kim, T., Park, H., & Lee, H. (2017). Packer detection for multi-layer executables using entropy analysis. Entropy, 19(3). https://doi.org/10.3390/e19030125

[6] Beebe, N. L., Maddox, L. A., Liu, L., & Sun, M. (2013). Sceadan: Using concatenated N-gram vectors for improved file and data type classification. IEEE Transactions on Information Forensics and Security, 8(9). https://doi.org/10.1109/TIFS.2013.2274728

[7] Bereziński, P., Jasiul, B., & Szpyrka, M. (2015). An entropy-based network anomaly detection method. Entropy, 17(4). https://doi.org/10.3390/e17042367

[8] Chew, C. J. W., & Kumar, V. (2019). Behaviour based ransomware detection. Proceedings of 34th International Conference on Computers and Their Applications, CATA 2019. https://doi.org/10.29007/t5q7

[9] Conti, G., Bratus, S., Shubina, A., Sangster, B., Ragsdale, R., Supan, M., Lichtenberg, A., & Perez-Alemany, R. (2010). Automated mapping of large binary objects using primitive fragment type classification. Digital Investigation, 7(SUPPL.). https://doi.org/10.1016/j.diin.2010.05.002

[10] Damashek, M. (1995). Gauging similarity with n-grams: Language-independent categorization of text. Science, 267(5199). https://doi.org/10.1126/science.267.5199.843

[11] Davies, S. R., Macfarlane, R., & Buchanan, W. J. (2021). Differential Area Analysis for Ransomware Attack Detection within Mixed File Datasets. https://doi.org/10.1016/j.cose.2021.102377

[12] Divakaran, D. M., Liau, Y. S., & Thing, V. L. L. (2016). Accurate in-network file-type classification. Cryptology and Information Security Series, 14. https://doi.org/10.3233/978-1-61499-617-0-139

[13] Ezhilarasan, M., Thambidurai, P., Praveena, K., Srinivasan, S., & Sumathi, N. (2008). A new entropy encoding technique for multimedia data compression. Proceedings - International Conference on Computational Intelligence and Multimedia Applications, ICCIMA 2007, 4. https://doi.org/10.1109/ICCIMA.2007.22

[14] Felderer, M., Büchler, M., Johns, M., Brucker, A. D., Breu, R., & Pretschner, A. (2016). Chapter One – Security Testing: A Survey. Advances in Computers, 101.

[15] Fitzgerald, S., Mathews, G., Morris, C., & Zhulyn, O. (2012). Using NLP techniques for file fragment classification. Proceedings of the Digital Forensic Research Conference, DFRWS 2012 USA. https://doi.org/10.1016/j.diin.2012.05.008

[16] Garg, A., Curtis, J., & Halper, H. (2003). Quantifying the financial impact of IT security breaches. Information Management and Computer Security, 11(2–3). https://doi.org/10.1108/09685220310468646

[17] Guo, J., He, J., & Huang, N. (2016). Research of Multiple-type Files Carving Method Based on Entropy.

[18] Hall, G. A. (2006). Sliding Window Measurement for File Type Identification. Proceedings of the 1997 ACM Symposium on Applied Computing.

[19] Iwamoto, K., & Wasaki, K. (2016). A Method for Shellcode Extractionfrom Malicious Document Files Using Entropy and Emulation. International Journal of Engineering and Technology, 8(2). https://doi.org/10.7763/ijet.2016.v8.866

[20] Karampidis, K., & Papadourakis, G. (2017). File Type Identification - Computational Intelligence for Digital Forensics. The Journal of Digital Forensics, Security and Law. https://doi.org/10.15394/jdfsl.2017.1472

[21] Karampidis, K., Papadourakis, G., & Deligiannis, I. (2015). File Type Identification - A Literature Review. 9th International Conference on New Horizons in Industry Business and Education, NHIBE 2015.

[22] Karen, S., & Angela, O. (2008). NIST - Technical Guide to Information Security Testing and Assessment Recommendations. Nist Special Publication, 800. https://doi.org/10.6028/NIST.SP.800-115

[23] Karresand, M., & Shahmehri, N. (2006). File type identification of data fragments by their binary structure. Proceedings of the 2006 IEEE Workshop on Information Assurance, 2006. https://doi.org/10.1109/iaw.2006.1652088

[24] Lee, K., Lee, S. Y., & Yim, K. (2019). Machine Learning Based File Entropy Analysis for Ransomware Detection in Backup Systems. IEEE Access, 7. https://doi.org/10.1109/ACCESS.2019.2931136

[25] Li, W. J., Wang, K., Stolfo, S. J., & Herzog, B. (2005). Fileprints: Identifying file types by n-gram analysis. Proceedings from the 6th Annual IEEE System, Man and Cybernetics Information Assurance Workshop, SMC 2005, 2005. https://doi.org/10.1109/IAW.2005.1495935

[26] Lyda, R., & Hamrock, J. (2007). Using entropy analysis to find encrypted and packed malware. In IEEE Security and Privacy (Vol. 5, Issue 2). https://doi.org/10.1109/MSP.2007.48

[27] McDaniel, M., & Heydari, M. H. (2003). Content based file type detection algorithms. Proceedings of the 36th Annual Hawaii International Conference on System Sciences, HICSS 2003. https://doi.org/10.1109/HICSS.2003.1174905

[28] McGraw, G. (2006). Software Security: Building Security in. Proceedings - International Symposium on Software Reliability Engineering, ISSRE. https://doi.org/10.1109/ISSRE.2006.43

[29] Pareek, H., & Hyderabad, C.-D. (2014). Entropy and n-gram analysis of malicious PDF documents. https://www.researchgate.net/publication/235974671

[30] Paul, C. B., Co-Advisor, R. C., & Fargues, M. P. (2017). NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA ENTROPY-BASED FILE TYPE IDENTIFICATION AND PARTITIONING.

[31] Potter, B., & McGraw, G. (2004). Software security testing. In IEEE Security and Privacy (Vol. 2, Issue 5). https://doi.org/10.1109/MSP.2004.84

[32] Rényi, A. (1955). On a new axiomatic theory of probability. Acta Mathematica Academiae Scientiarum Hungaricae, 6(3–4). https://doi.org/10.1007/BF02024393

[33] Revo, R., Made, G., Sasmita, A., Agus, I. P., & Pratama, E. (2020). Testing for Information Gathering Using OWASP Testing Guide v4 (Case Study : Udayana University SIMAK-NG Application). Jurnal Ilmiah Teknologi Dan Komputer, 1(1).

[34] Shannon, C. E. (1948). A Mathematical Theory of Communication. Bell System Technical Journal, 27(3). https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

[35] Shannon, C. E. (1997). The Mathematical Theory of Communication. M.D. Computing, 14(4). https://doi.org/10.2307/410457

[36] Shannon, M. M. (2004). Forensic Relative Strength Scoring: ASCII and Entropy Scoring. In International Journal of Digital Evidence Spring (Vol. 2, Issue 4). www.ijde.org

[37] Tian-yang, G., Yin-sheng, S., & You-yuan, F. (2010). Research on Software security testing. World Academy of Science, Engineering and Technology, 70. https://doi.org/10.5281/zenodo.1081389

[38] Tsallis, C., Mendes, R. S., & Plastino, A. R. (1998). The role of constraints within generalized nonextensive statistics. Physica A: Statistical Mechanics and Its Applications, 261(3–4). https://doi.org/10.1016/S0378-4371(98)00437-3

[39] Xie, H., Abdullah, A., & Sulaiman, R. (2013). Byte Frequency Analysis Descriptor With Spatial Information For File Fragment Classification. 25–26.

[40] Young, A. L., & Yung, M. (2017). Cryptovirology: The birth, neglect, and explosion of ransomware: Recent attacks exploiting a known vulnerability continue a downward spiral of ransomware-related incidents. In Communications of the ACM (Vol. 60, Issue 7). https://doi.org/10.1145/3097347

[41] Young, A., & Yung, M. (1996). Cryptovirology: extortion-based security threats and countermeasures. Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy. https://doi.org/10.1109/secpri.1996.502676.