Seeing the Unseen: Entropy-Guided Insights into Binary Files with GView

Andrada-Livia Antoneac
aantoneac@bitdefender.com
Al. I. Cuza University
Iași, Romania
Bitdefender

Gheorghiță Mutu
gmutu@bitdefender.com
Al. I. Cuza University
Iași, Romania
Bitdefender

Dragoș-Teodor Gavriluț
dgavrilut@bitdefender.com
Al. I. Cuza University
Iași, Romania
Bitdefender

Abstract

The number of known malicious samples has increased exponentially in the last decade, making it more complicated for a security researcher to identify and cluster them quickly. While for most scenarios, most data viewers relate to the file type (either binary or textual) to extract meaningful data, the protective measurements of different malicious payloads may make this task difficult.

Our research introduces an approach that develops an enhanced visualization mode within the open-source framework GView to address this challenge. It harnesses established entropy-based analytical principles to facilitate the identification of anomalies and intrinsic properties in binary data, irrespective of specific file formats.

Keywords

GView, Cybersecurity, entropy, visualization mode, binary file format, malware analysis, stenography, static analysis, automatic reasoning, scientific visualization, image processing

ACM Reference Format:
Andrada-Livia Antoneac, Gheorghiță Mutu, and Dragoș-Teodor Gavriluț. 2025. Seeing the Unseen: Entropy-Guided Insights into Binary Files with GView. In *Proceedings of 12th ACM Celebration of Women in Computing: womENcourage™ (womENcourage).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Security vendors have multiple approaches to win the ever-lasting battle against malware writers. The real challenge lies in this industry's focus on forensic investigations and security research, specifically in the areas of static analysis and dynamic analysis.

The effectiveness of static analysis over files relies on the capability of one tool to extract and correlate multiple information specific to that file's format. Although all analysis methods have inherent limitations, static analysis, in particular, can prove to be both time-consuming and frustrating. Attackers may conceal malicious payloads with encryption and packing methods. In such cases, entropy becomes a powerful tool for detecting these obscured threats.

womENcourage, Brașov, Romania

https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 Related Work

The potential power of entropy and byte frequency, while used in static analysis, is brought up by Gregory Conti et al. [2], presenting the impact of file format, semantics, and structure analysis in contrast with only text-based one.
Later, a similar approach comes from Ming Xu et al. [17] analyzing the file's visual representation. The survey conducted by M. Wagner et al. [16] surfaces the utility of entropy in classification and detection. Another type of entropy representation is detailed in [5] utilizing entropy graphs and the similarity criterion.

Exploiting visual representations in static analysis is extended to opcodes by Hashemi and Hamzeh [6]. Notable papers on entropy used as straight classification or artificial intelligence models come from Hamad Naeem et al. [8], Otsubo et al.[10], Hui Guo et al.[4], Zainodin et al.[20]. Deep learning classification over malware visualization is depicted in Anson Pinhero's approach [11].

3 Problem Description

Many analysis approaches imply classifying a sample based on static [4] [20] or dynamically [15] [12] extracted features, while being continuously accompanied by adversarial attacks [7]. Even though solutions involving automatic detection exist, scenarios often arise where an analyst needs to provide a quick verdict.

Threat actors might rely on packers to avoid sample detection. These cases were thoroughly researched [9] [3] [13]. Although high overall entropy may prevent direct anomaly detection, this characteristic alone promptly guides the analysis toward unpacking within seconds.

An entropy-based view enables rapid identification of anomalous zones or patterns in a sample, regardless of its binary format. Without such visual cues, static analysis slows down—an especially critical drawback in zero-day scenarios.

4 Solution

We propose a visualization plugin that we are currently developing inside a specialized framework, GView [18] [19]. The framework is based, at its core, on an extensible Text-User-Interface (TUI) library called AppCUI[1] that ensures our tool works cross-platform (Windows, Linux, MacOS).

With its ability to provide guided analysis for various file types, automatic artifact recognition, extraction, coherent correlation and inference, and meaningful and intuitive views at multiple granularities, GView seeks to explore potential malicious objects.

Inside GView, we name this plugin Entropy Visualizer. It is a generic plugin, meaning it is suitable for any input file or buffer

[1] https://github.com/gdt050579/AppCUI

object. As the framework supports opening multiple objects, a researcher can visualize the buffer exposed in multiple visualization modes.

Shannon entropy [14](the mathematical foundation of this solution) of a discrete random variable X is calculated using the formula:

$$H(X) = -\sum_{i} p(x_i) \log_2 p(x_i)$$

Where $H(X)$ is the entropy of the random variable X, $p(x_i)$ is the probability of outcome x_i, $\log_2$ is the base-2 logarithm. Its reasoning and context are briefly described in this article [1]. In our case - applying this formula over byte blocks, a byte has values between 0 and 255. Based on this, Shannon entropy values range from 0 to 8.

We can split this range into three intervals: 0 to 6 - the area is likely to contain plain text/data; 6 to (8-ϵ) - the area is likely to contain binary data; (8-ϵ) to 8 - the area will likely contain encrypted data. ϵ, is an estimated difference between absolute chaos (8) and actual entropy, differentiating binary from encrypted data. The file will be represented by blocks of different colors corresponding either to a certain entropy value or one of the defined intervals, resulting in two visualizer methods.
The Rényi entropy [1] and the Tsallis (information) entropy [1] are also included in the visualizer.

4.1 Results

From our tests, we identified several cases where computing the entropy on raw or augmented data or correlating both resulted in the following outcomes:

(1) identifying a malicious MZPE file embedded into a PNG file: On VirusTotal[2], the PNG is detected only by 12/58 vendors. At the same time, the embedded one has 44/71 detections, showing significant difficulty in dealing with this kind of attack.
(2) identifying a malicious PHP script embedded into a JFIF file: "Special Strings" are automatically recognized in the visualizer. These are anomalies in a JFIF file, which should contain only metadata and content according to its purpose.
(3) identifying a tampered executable sample (malicious payload added): for multiple files like [3] [4] we have seen a change in the entropy (very high) values at the end of the files.
(4) identifying samples from the same family (Sality malware - file infectors): files in the same malware family have similar patterns in terms of entropt values.

5 Conclusion

We implemented several types of entropies and techniques that a security researcher can use to filter or pinpoint an anomaly in a sample quickly. They can evaluate a file in seconds based on patterns.

We plan to apply a preprocessing step that unpacks the file before using any algorithm. After any preprocessing steps, we can develop a flag extraction or hash computation method. The first one can be used as input for convolutional neural networks, which aim to use both methods for malware detection.

[2]https://www.virustotal.com
[3]VirusTotal report for sample #4
[4]VirusTotal report for sample #5

References

[1] PA Bromiley, NA Thacker, and E Bouhova-Thacker. [n. d.]. Shannon entropy, Renyi entropy, and information. ([n. d.]).
[2] Gregory Conti, Erik Dean, Matthew Sinda, and Benjamin Sangster. 2008. Visual Reverse Engineering of Binary and Data Files. In *Visualization for Computer Security*, John R. Goodall, Gregory Conti, and Kwan-Liu Ma (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–17.
[3] Dhruwajita Devi and Sukumar Nandi. 2012. Detection of packed malware. In *Proceedings of the First International Conference on Security of Internet of Things* (Kollam, India) *(SecurIT '12)*. Association for Computing Machinery, New York, NY, USA, 22–26. doi:10.1145/2490428.2490431
[4] Hui Guo, Shuguang Huang, Cheng Huang, Fan Shi, Min Zhang, and Zulie Pan. 2020. Binary File's Visualization and Entropy Features Analysis Combined with Multiple Deep Learning Networks for Malware Classification. *Security and Communication Networks* 2020 (12 2020), 1–19. doi:10.1155/2020/8881760
[5] Kyoung-Soo Han, Jae Hyun Lim, Boojoong Kang, and Eul Gyu Im. 2015. Malware analysis using visualized images and entropy graphs. *International Journal of Information Security* 14 (2015), 1–14. https://api.semanticscholar.org/CorpusID:8498534
[6] Hashem Hashemi and Ali Hamzeh. 2019. Visual malware detection using local malicious pattern. *Journal of Computer Virology and Hacking Techniques* 15 (03 2019). doi:10.1007/s11416-018-0314-1
[7] Bojan Kolosnjaji, Ambra Demontis, Battista Biggio, Davide Maiorca, Giorgio Giacinto, Claudia Eckert, and Fabio Roli. 2018. Adversarial Malware Binaries: Evading Deep Learning for Malware Detection in Executables. In *2018 26th European Signal Processing Conference (EUSIPCO)*. 533–537. doi:10.23919/EUSIPCO.2018.8553214
[8] Hamad Naeem, Bing Guo, Muhammad Rashid Naeem, and Danish Vasan. 2019. Visual malware classification using local and global malicious pattern. *Journal of Computers* 6 (2019), 73–83.
[9] Lakshmanan Nataraja, Grégoire Jacobb, and B. S. Manjunatha. 2010. DETECTING PACKED EXECUTABLES BASED ON RAW BINARY DATA. https://api.semanticscholar.org/CorpusID:5876296
[10] Yuhei Otsubo, Akira Otsuka, Mamoru Mimura, and Takeshi Sakaki. 2020. o-glasses: Visualizing X86 Code From Binary Using a 1D-CNN. *IEEE Access* 8 (2020), 31753–31763. doi:10.1109/ACCESS.2020.2972358
[11] Anson Pinhero, Anupama M L, Vinod P, C.A. Visaggio, Aneesh N, Abhijith S, and AnanthaKrishnan S. 2021. Malware detection employed by visualization and deep neural network. *Computers Security* 105 (2021), 102247. doi:10.1016/j.cose.2021.102247
[12] Edward Raff, Jon Barker, Jared Sylvester, Robert Brandon, Bryan Catanzaro, and Charles K. Nicholas. 2017. Malware Detection by Eating a Whole EXE. In *AAAI Workshops*. https://api.semanticscholar.org/CorpusID:33641567
[13] M Shafiq, S Tabish, and Muddassar Farooq. 2009. PE-Probe: Leveraging Packer Detection and Structural Information to Detect Malicious Portable Executables. (07 2009).
[14] C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal* 27, 3 (1948), 379–423. doi:10.1002/j.1538-7305.1948.tb01338.x
[15] Philipp Trinius, Thorsten Holz, Jan Göbel, and Felix C. Freiling. 2009. Visual analysis of malware behavior using treemaps and thread graphs. *2009 6th International Workshop on Visualization for Cyber Security* (2009), 33–38. https://api.semanticscholar.org/CorpusID:13866201
[16] Markus Wagner, Fabian Fischer, Robert Luh, Andrea Haberson, Alexander Rind, Daniel A. Keim, and Wolfgang Aigner. 2015. A Survey of Visualization Systems for Malware Analysis. In *Eurographics Conference on Visualization (EuroVis) ; STARs - State of The Art Reports*, Rita Borgo (Ed.). The Eurographics Association, 105–125. doi:10.2312/eurovisstar.20151114
[17] Tantan Xu, Ming Xu, Yizhi Ren, Jian Xu, Haiping Zhang, and Ning Zheng. 2014. A File Fragment Classification Method Based on Grayscale Image. *Journal of Computers* 9 (08 2014). doi:10.4304/jcp.9.8.1863-1870
[18] Raul Zaharia, Dragos Gavrilut, Gheorghita Mutu, and Dorel Lucanu. 2024. Interactive Assistance in Malware Dissemination Detection and Analysis. In *Proceedings of the 1st Workshop on Security-Centric Strategies for Combating Information Disorder* (Singapore, Singapore) *(SCID '24)*. Association for Computing Machinery, New York, NY, USA, Article 7, 6 pages. doi:10.1145/3660512.3665526
[19] Raul Zaharia, Dragoş Gavriluţ, Gheorghiţă Mutu, and Dorel Lucanu. 2024. GView: A Versatile Assistant for Security Researchers. arXiv:2404.09058 [cs.CR]
[20] Muhammad Zainodin, Zalmiyah Zakaria, Rohayanti Hassan, and Zubaile Abdullah. 2022. Entropy Based Method for Malicious File Detection. *JOIV : International Journal on Informatics Visualization* 6 (12 2022), 856. doi:10.30630/joiv.6.4.1265