Preliminary investigation method for segregating malware-encrypted files from the regular traffic

Aarzoo Gosain[1]*, B.E. (I.T.), Commander Rakesh Kumar Gosain, Ret.[2], B.Tech. (Electrical), M.Sc. (Physics)

[1] Netaji Subhas Institute of Technology (University of Delhi), New Delhi, India
[2] Marine Electrical Engineer Ex-Indian Navy; Scientist F, Ex-PMO, India

* Corresponding Author. Email: aarzoogosain@gmail.com

Synopsis

Background: In this digital era, where a massive number of files are being transferred every second, it is impossible to open and check every file for its effect on the computer system. For some files containing malware, the attack is launched once that file is opened. Furthermore, opening each file is very time-consuming. Therefore, the requirement of 'file checking without opening' arises. This paper introduces a standalone file segregator tool which conducts a preliminary investigation of incoming file traffic of a computer system. It can detect suspected malicious or hidden code snippets in those files without opening their contents. This file segregator tool calculates Shannon's entropy value of every file in the system's incoming file traffic before passing them onto the central system. The entropy of a file lies within the range of 0 to 8 bits per byte of information. A lower value indicates character uniformity, while a higher entropy value indicates randomness. Files with a higher entropy value are flagged as 'suspected' and are separated from the regular traffic. Such files are likely to be encrypted with some malware.

Results: Incoming traffic of files is routed through this file segregator tool, and some files are flagged by segregation. It is seen that the flagged files were either encoded or encrypted and had an entropy value of 7 bits or more.

Conclusion: It is found that the value for Shannon's entropy for a standard text file is generally less than 4 bits. Audio and video files are generally encoded and have a higher entropy value. The files being encrypted using sophisticated encryption techniques like Advanced Encryption Standard (A.E.S.) tend to have an even higher value of Shannon's entropy, i.e., greater than 7 bits. Such files can be potential malware carriers; hence, they are flagged by this file segregator tool.

Relevance in the Naval Industry: By removing the suspected files from the network traffic and allowing the transfer of the safe files, secured communication is established on an integrated command and control network. Jamming of a system can be prevented when a Denial of Service (DoS) or Distributed Denial of Service (DDoS) attack is perpetrated against it. As this file segregator tool can prevent malware attacks, national security agencies can use it to strengthen their threat prevention mechanisms.

Keywords: Shannon entropy, malware-encrypted, network traffic, randomness, file segregation

1. Introduction: Measuring randomness of a file

In Information theory, measuring a file's entropy is an integral concept. The randomness in file characters can convey its encryption status. Encryption is a process of modifying the data of a given file so that it becomes unintelligible to users who are not its intended audience (Mahajan & Sachdeva 2013). By carrying out the encryption operation on a particular file, it ensures that any patterns in the natural language cease to exist, and randomness is induced into the characters of the said file. A file's entropy is measured to quantify this induced randomness by using Shannon's entropy equation.

Shannon's entropy calculation is a method in information theory that measures information content in a file. It measures the probability of occurrence of any specific character in the file based on its preceding characters (Shannon 1948). In his research paper titled 'A Mathematical Theory of Communication', Sir Claude E. Shannon first propounded this method for calculating entropy in 1948.

Authors' Biographies

Aarzoo Gosain is an alumnus (batch of 2021) of Netaji Subhas Institute of Technology (University of Delhi). She holds a Bachelor of Engineering in Information Technology and has published a conference paper in the IET journal regarding AI-based real-time noise suppression. She has led several projects in the fields of Artificial Intelligence and Cybersecurity during her course of undergraduate study.

Cdr. Rakesh Kumar Gosain, Ret. is an Indian Navy veteran and has served as a Scientist 'F' in PMO, India. He is a marine electrical officer having versatile professional experience ranging from Naval electrical and weapon systems to project and contract management of multi-billion dollar G to G projects of strategic interest to his credit. His area of expertise is missiles and associated Close-in Weapon Systems (CIWS) and Sensors.

The calculated value of Shannon's entropy can lie between 0 to 8 bits per byte of information. A lower entropy value indicates character uniformity, which means predicting the succeeding character based on the previous characters is possible. In comparison, a higher entropy value indicates randomness in the file. It has been observed that the files that are either encoded or encrypted often have a higher entropy value (Bat-Erdene et al. 2017). The difference between an encoded and encrypted file is that the former can be converted to its original form without using a key, and the latter requires a key for its decryption. In other words, compressing the contents of a file is essentially a form of encoding, whereas encryption refers to the process of transferring information secretly to its intended receiver.

The formula to calculate Shannon entropy is given below.

$$H = -\sum_{i=1}^{n}(p(i) * \log_2 p(i)) \tag{1}$$

Where H is Shannon's entropy value, $p(i)$ is the probability that the variable in question will take the state 'i', and 'n' is the total number of states.

The logarithm of that probability defines the storage space required by that variable state (Vajapeyam 2014). Due to the information representation being in bits, the base of the logarithm is taken as '2'. Since the probability always lies between 0 and 1, it causes the logarithm function in (1) to take on a negative value. A minus sign precedes Shannon's entropy equation to balance this anomaly and make the entropy value positive.

2. Process of file investigation

The process of a preliminary investigation for segregating the suspected files from the regular traffic is depicted in Figure 1.

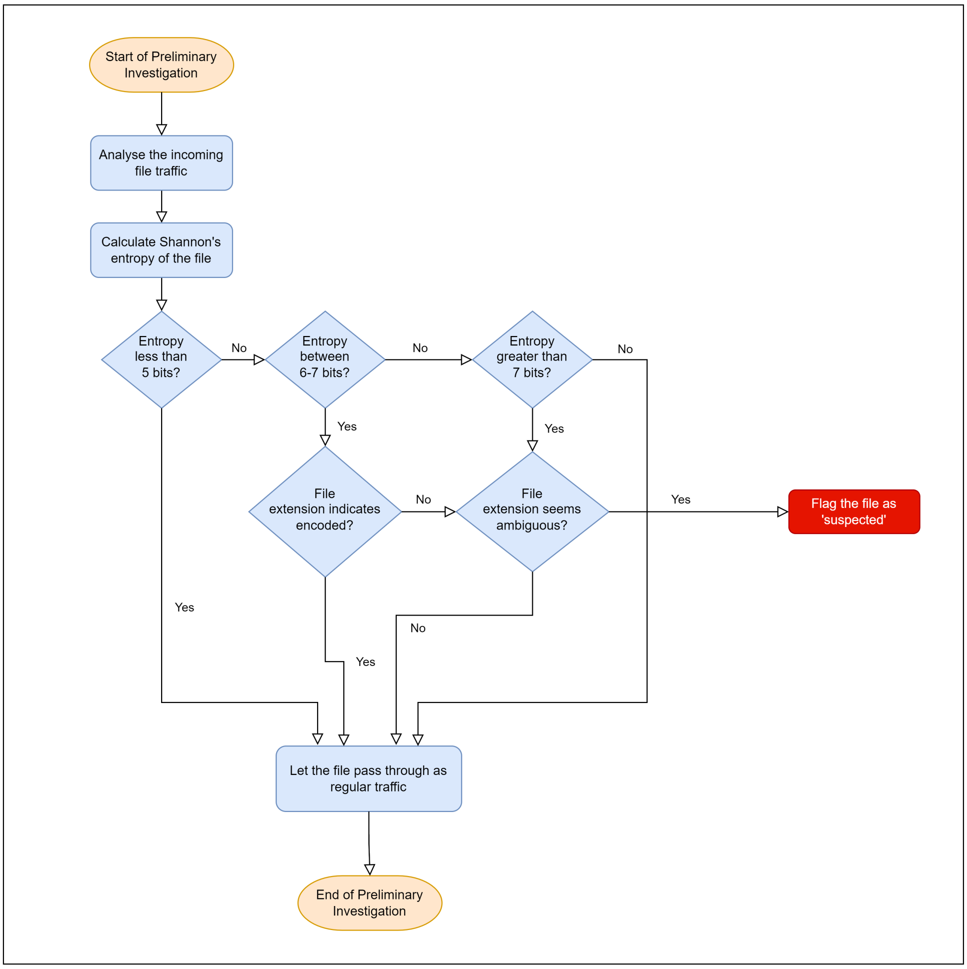
Start of Preliminary Investigation
Analyse the incoming file traffic
Calculate Shannon's entropy of the file
Entropy less than 5 bits?
No
Entropy between 6-7 bits?
No
Entropy greater than 7 bits?
No
Yes
Yes
File extension indicates encoded?
No
File extension seems ambiguous?
Yes
Flag the file as 'suspected'
Yes
No
Yes
Let the file pass through as regular traffic
End of Preliminary Investigation

Figure 1: Process of the preliminary investigation for flagging suspected files

The incoming traffic of the files is first passed through this investigation filter. It analyses and calculates Shannon's entropy of every file. If the file's entropy comes out to be less than 5 bits, then the file is passed through the filter. Otherwise, if the file's entropy value lies between 6 to 7 bits, then its file extension is analysed. If it is an encoded file, it is passed, as usual; otherwise, it is flagged as 'suspected'.

Files whose entropy values are more than 7 bits and their extensions seem ambiguous are also flagged as 'suspected'. All other files are passed through the filter as regular traffic. This stage marks the end of the preliminary investigation process.

Encryption increases the entropy of files because it tampers with the patterns and induces haphazard behaviours into the file's contents.

It is seen that most malicious software like viruses, worms, and Trojan horses are embedded into files using encryption which are then transferred over the Internet to targetted recipient systems (Or-Meir et al. 2019). The changes in a file's entropy value can be observed to detect the presence of any encryption, as shown in the example below.

Suppose a file named 'example.txt' contains the following text:

'Hi!
What a nice day it is today! Shall we go for a walk?'

Table 1 below represents the changes in Shannon's entropy value, for example.txt, after it undergoes encoding and encryption operations. Base64 encoding is performed on the given file. It is a form of encoding which converts binary or textual data into printable American Standard Code for Information Interchange (ASCII) format. The Advanced Encryption Standard (A.E.S.) algorithm is used for the encryption operation with a key size of 128 bits. It follows a robust encryption scheme which converts a block of input text into a block of ciphertext. Both of these blocks are the same size as the key used. It is the most widely used algorithm for transferring confidential data over a network.

Table 1: Entropy trends of example.txt

S. No.	**Operation Performed**	**Shannon's entropy (bits)**
1	None	4.13
2	Base64 encoding	4.95
3	A.E.S. 128-bit key encryption (Mode: E.C.B.)	5.37

It is evident that the file's entropy increases after the encryption and encoding operations, but the increment in entropy after encryption is more than that after encoding. Hence, this is a clue in the preliminary investigation process to segregate maliciously encrypted files from the regular traffic.

3. Inbuilt implementations of Shannon's entropy function

Below are some of the inbuilt implementations for Shannon's entropy function. These implementations can be executed on the Windows and Linux platforms. Some of these utilities are proprietary by their originator companies, and others are open-sourced.

3.1. Sandfly-filescan utility

This utility is helpful for scanning files and reporting their entropies. It works on Linux or UNIX-type executable files (The Sandfly Security Team 2022). When a file is packed with malware in Linux, it gives a very high entropy value. This tool helps to filter out malicious executable files. By running the following command, it can be checked whether a file is suspicious or not based on its generated entropy value.

```
sandfly-filescan -file /dev/shm/suspicious_file
```

3.2. Sigcheck utility

This utility can be executed using the Windows command prompt. Sigcheck produces the version, timestamp and other signature details about a particular file as its result. Users can scan a file for malicious software using the antivirus engines embedded in this utility. All the computations of this method use the system's internal sigcheck procedures.

The following command is used to initiate the Sigcheck utility function.

#$cmd = "sigcheck -a `"" + $FNAME + "`""

3.3. Entropy utility

The entropy utility generates the maximum entropy value of the file supplied as input. This command can also be executed on the Windows command prompt like the Sigcheck utility function. The command to launch the entropy utility function is given below.

./entropy path/to/file.bin

4. Experimental results

4.1. Preliminary filtering results

Table 2 indicates the results of the preliminary filtering of the incoming file traffic (Netresec 2022). The reported categories are plaintext, compressed/encoded, and encrypted files. In cryptography, plaintext files refer to human-readable files that do not need a decryption key for their usage. Binary files, document files, and regular text files fall into this category. Compressed/encoded files require a specific encoding and decoding scheme to make them comprehensible to the users. The same is true of encrypted files, which are unintelligible to the users unless they have the key and the particular algorithm for decryption.

The encoded files comprise Z.I.P. files and R.A.R. format files. Base64 encoding is used on some files, whereas, for encryption, the A.E.S. algorithm with a key size of 128 bits is used. The mode used for encryption is Electronic Code Book (E.C.B.). This mode is the simplest and the fastest among all other modes (Almuhammadi & Al-Hejri 2017), chosen not for showcasing the strength of the encryption but for depicting the properties of an encrypted file.

The three file types shown in the 3rd column of the given table are plaintext (P), encoded/compressed (C), and encrypted (E).

Table 2: Preliminary investigation results

File number	**File name**	**File type (P/C/E)**	**Shannon's entropy (bits)**	**Investigation result (Pass/Suspected)**
1.	Letter.rar	C	4.98	Pass
2.	Desert.mp3	C	5.27	Pass
3.	Entropy_mater.zip	C	5.59	Pass
4.	Textpdf.pdf	C	5.59	Pass
5.	Testcompress.7z	C	5.84	Pass
6.	Flowers.jpg	C	5.98	Pass
7.	Image_compress.jpeg	C	6.31	Pass
8.	Base64_text.docx	C	6.52	Pass
9.	Tongo_movies.exe	C	6.70	Borderline suspect
10.	Tongo_movies.7z	C	6.73	Borderline suspect
11.	Flowers_min.jpg	C	6.83	Borderline suspect
12.	Textdoc.vbs	E	5.04	Pass

Table 2 (continued)

13.	Flowers.encrypt	E	5.39	Pass
14.	Settings.encrypt	E	5.69	Pass
15.	Text.encrypt	E	5.75	Pass
16.	Abridged_mode.encrypt	E	7.43	Suspected
17.	ABC.encrypt	E	7.66	Suspected
18.	Letter.vbs	E	7.73	Suspected
19.	Textpdf.vbs	E	7.87	Suspected
20.	Testimage.vbs	E	7.97	Suspected
21.	Binary_data.dat	P	3.45	Pass
22.	Letter.docx	P	3.86	Pass
23.	Import.html	P	4.42	Pass
24.	Text.txt	P	4.44	Pass
25.	Project_detailts.xml	P	4.75	Pass
26.	ABC.rtf	P	5.06	Pass
27.	Abridged.xlsx	P	5.81	Pass

4.2. Analysis

The table contains 27 file entries whose entropy ranges from around 3 to 8 bits. The names of the files are given for an easy understanding of the contents and type of these files. The files are grouped according to their types. Then, the files are arranged in each group according to their entropies.

The plaintext files have the lowest entropy amongst the two groups because they are the standard text documents, spreadsheets, and binary files. Naturally, they will have a low entropy because of their contents' uniformity and coherence (Canfora, Mercaldo & Visaggio 2016). Likewise goes for the encoded/compressed files.

Encoded or compressed files have a pattern in their contents because any data encoding method has a pre-defined procedure for being carried out, and any output generated will conform to those procedural rules. The entropy of the encoded files is more than the plaintext files because they contain a lot more variety of information compressed in them as compared to the plaintext files.

The encrypted files have a haphazard behaviour. Some files have entropy as low as 5 bits, and some have a high entropy of 7.6 to 7.9 bits. The files having an entropy higher than 7 bits may contain obfuscated data or suspicious code snippets in them. Therefore, such files are flagged as 'suspected'.

4.3. Graphical representation of the investigation results

Figure 2 represents the contents of Table 2 graphically. The figure represents file numbers on X-axis and entropies on Y-axis. Entropy lies between 0 to 8 bits only, but for representation purposes, the Y-axis also contains a marker of 9 bits.

This graph depicts the entropy patterns in the three file groups. The plaintext file group is represented in 'green'; they have the lowest entropy values.

The entropy values for the encoded/compressed files are shown as 'blue'. They have an entropy ranging from 5 to 6 bits and are consistent in their behaviour.

A pattern can be seen in the entropy values of both plaintext and encoded file types. The threshold for flagging the suspected files is set at 7 bits. This line for the threshold value is shown in 'red'. Only some of the encrypted files have crossed this threshold value, so these files are flagged. Some encoded files are also very close to the threshold, marking them as 'borderline suspects'.

The malware is generally hidden as a secret code in the image and video files. This technique is known as steganography. This hidden malware code can also be encrypted to avoid its detection, and as soon as the user opens that particular file in their system, the malware attack is executed.

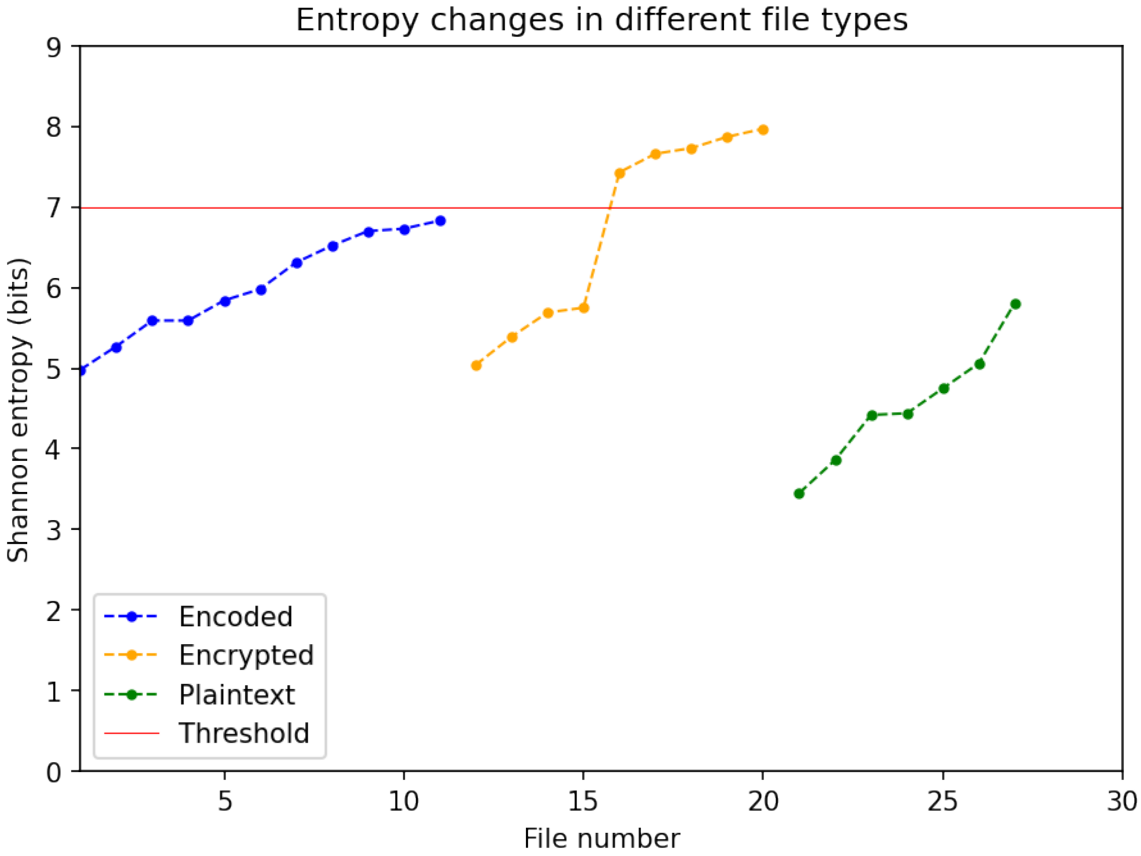

Figure 2: Entropy trend of different file types

5. Applications in the Naval Industry

The technologies in the world are evolving and enhancing every second. In today's modern era, cyber warfare has become a harsh reality. A file segregator tool can be built on the concept of filtering malicious files from the regular network traffic using Shannon's entropy. The entropy of the files will be calculated using equation (1). Such a tool will be beneficial to equip the Naval Industry better to face the upcoming challenges. It will route the harmful files into a sandbox environment where the embedded malware could be further inspected, thereby preventing these files from entering the central system. Mentioned below are some applications of this tool, precisely laid out in the context of the Navy.

5.1. Establishing secured communication

The usage of unmanned ships and submarines is on the rise in the maritime industry due to the advances in automation and reducing the cost of human factors (Danilin et al. 2021). Therefore, ensuring the information security aspect of these unmanned vehicles has become essential.

Malware attacks can be launched anywhere, and they can cause the behaviour of these vehicles to go rogue. Such attacks can also hamper the communications between the Naval ships or submarines with the base and vice versa.

When installed on the Naval base, this file segregator contributes to a smoother flow of information. Any incoming threat in the form of a malicious file can be encountered and stopped before the attack can even be launched.

5.2. Preventing system jamming by outside threats

A Denial of Service (DoS) attack can render a system unavailable at the time when its critical operations are needed the most. So many superfluous requests are sent into the targeted system that it becomes jammed, and no helpful information can be drawn from it (Murlidharan & Osadciw 2006). The attack decreases the efficiency and reliability of the system. Likewise is the case of a Distributed Denial of Service (DDoS) attack, which requires even greater mechanisms for its mitigation.

This file segregator tool can prevent such jamming of the system in case of DoS and DDoS attacks. The segregation presented here is done manually, so it is presently beneficial on smaller systems only. However, an Artificial Intelligence-based classifier can be built in the future using the foundation of this present manual segregator.

The AI-based classifier will be able to prevent a whole DDoS attack by judging the incoming file threat and sending the malicious traffic to a sandbox, thus ensuring that the file traffic which reaches the central system is harmless.

5.3. Protection of the critical information infrastructures

Critical information infrastructures comprise the most crucial information databases to a particular nation. For example, the Aadhaar database of India, the Social Security number database of the United States of America, and the driver's license database of any particular country, come under the category of critical information infrastructures. Every country has built agencies to protect its critical information infrastructures.

Many malware attacks are launched on these critical databases to steal their information. The classified information should not be divulged to the common public. However, the hackers break into these databases and steal, modify, or encrypt the information in exchange for money or national secrets (Nickolov 2008).

One such malware attack occurred in India in February 2009, where over 600 Ministry of External Affairs (M.E.A.) computers were hacked into by spyware (The Times of India 2009). This file segregator tool can be installed in the agencies responsible for protecting these critical databases.

5.4. Avoiding outside tampering in the military operations

There are some covert military operations which need utmost confidentiality and secrecy. Many missile launches are tampered with using computer-launched malware attacks. The data relating to a covert military operation can be breached using inside sources or an outsider system. Either way, the operation is hindered, and the covert agenda of that nation is revealed.

National security agencies worldwide use some threat prevention and detection mechanisms when working in covert military activities. Some of these activities use satellites for coordination between the various teams involved. The soldiers' lives are at risk if the operation becomes compromised.

Spyware and ransomware attacks can be prevented by adequate file segregation. National security agencies can use this file segregator tool to strengthen their threat prevention mechanisms. This way, only the essential and the intended information will be communicated. This file segregator will flag all other superfluous, harmful or anonymous file traffic.

6. Conclusions

Often there is a massive amount of incoming file traffic in a given computer system at any time. Opening every file and analysing its contents is impossible because it places heavy demands on the system regarding the time and space required. Some of these files may even contain malware that launches the attack once that file is opened. Hence, using this file segregator tool as a preliminary investigative device for a system can be beneficial.

Most of the time, it is seen that the malware code is encrypted within the standard text or audio-video files. This file segregator tool, developed on the concept of Shannon's entropy, can differentiate and filter such encrypted files from the regular traffic, maintaining the system's security.

The entropy of a file lies between 0 to 8 bits per byte of information. Plaintext and encoded files have a lower entropy value than encrypted files. The files having entropy greater than 7 bits are flagged as 'suspected'. All other files are passed through as regular traffic. Some encoded files have a higher entropy value, almost touching the threshold value of 7 bits. Such files are marked as 'borderline suspects'.

Sometimes, this file segregator tool gives false negatives and false positives also. It may work as a heuristic but should still be considered a preliminary investigation tool only.

7. Future scope: AI-based classifier

The algorithm for segregating malicious files presented in this paper can be extended to suit real-world applications more efficiently. One such method of advancing the functionalities of this file segregator tool is to implement it using Artificial Intelligence (A.I.).

An AI-based classifier will be able to judge the incoming file traffic and classify the files into two categories: ' Normal' and 'Suspected'. The classifier will be trained on a dataset containing at least a thousand file entries.

One suitable dataset is the 'Benign & Malicious PE files' dataset for detecting malware-ridden files (Kaggle 2018). This dataset is on the Kaggle platform, a repository of datasets for carrying out AI-based projects. The algorithm for this AI-based classifier will be as follows:

- A code calculates Shannon's entropy of an incoming data file.
- The classifier is trained on the dataset mentioned above. The training algorithm and entropy calculation code will function simultaneously so that the classifier learns the characteristics of both types of files, namely, 'Benign' and 'Malicious'.
- Files with ambiguous properties and a higher entropy value are flagged as 'Malicious' and are filtered out from the usual traffic. All other files are marked as 'Benign'.
- Now, as the classifier recognises both types of files, it is tested in real-time by feeding it a file and recording its response.
- The accuracy of the classifier is calculated based on the accuracy of its responses, i.e., whether it can segregate malware-ridden files from the regular traffic or not.

This file segregation will become more efficient and reliable using an AI-based classifier. It will be able to segregate a million files in a much shorter duration of time. Errors in segregation can prove to be very detrimental in some cases, like that of covert military operations. The accuracy of this file segregator will be higher than its manual counterpart because there will be fewer chances of a human error occurring. Apart from its speed and efficiency, this file segregator tool's cost will also be less than the other hardware-based segregation tools.

Acknowledgements

The author would like to thank their co-author and mentor, Commander Rakesh Kumar Gosain, for providing immense support and valuable guidance in completing this conference paper. His thirty years of versatile Naval experience has been a huge stepping stone towards the successful accomplishment of the project, which otherwise would not have been possible. The author would also like to express gratitude towards their family and friends for providing immeasurable encouragement and motivation. Last but not least, a token of appreciation to the IMarEST team for giving this opportunity to showcase this research.

References

Almuhammadi, S. and Al-Hejri, I., 2017, April. A comparative analysis of A.E.S. common modes of operation. In *2017 IEEE 30th Canadian conference on electrical and computer engineering (C.C.E.C.E.)* (pp. 1-4). IEEE.

Bat-Erdene, M., Park, H., Li, H., Lee, H. and Choi, M.S., 2017. Entropy analysis to classify unknown packing algorithms for malware detection. *International Journal of Information Security*, *16*(3), pp.227-248.

Canfora, G., Mercaldo, F. and Visaggio, C.A., 2016. An hmm and structural entropy based detector for android malware: An empirical study. *Computers & security*, *61*, pp.1-18.

Danilin, G., Sokolov, S., Knysh, T. and Singh, V., 2021, May. Information Security Incidents in the Last 5 Years and Vulnerabilities of Automated Information Systems in the Fleet. In *International Scientific Siberian Transport Forum* (pp. 1541-1550). Springer, Cham.

Kaggle 2018, *Benign & Malicious PE Files'*, Kaggle, viewed 10 February 2020, <https://www.kaggle.com/datasets/amauricio/pe-files-malwares>.

Mahajan, P. and Sachdeva, A., 2013. A study of encryption algorithms A.E.S., D.E.S. and R.S.A. for security. *Global Journal of Computer Science and Technology*. Buckingham J.: "Hybrid Drives For Naval Auxiliary Vessels", Proceedings of the Pacific 2013 Conference, Sydney, Australia, October 7-9 2013.

Muraleedharan, R. and Osadciw, L.A., 2006, May. Jamming attack detection and countermeasures in wireless sensor network using ant system. In *Wireless Sensing and Processing* (Vol. 6248, p. 62480G). International Society for Optics and Photonics.

Netresec (2022) *Network Forensics and Network Security Monitoring*. Available at: https://www.netresec.com/?page=Resources (Accessed: 20 July 2022).

Nickolov, E.U.G.E.N.E., 2008, October. Modern trends in the cyber attacks against the critical information infrastructure. In *Regional Cybersecurity Forum* (Vol. 10).

Or-Meir, O., Nissim, N., Elovici, Y. and Rokach, L., 2019. Dynamic malware analysis in the modern era—A state of the art survey. *A.C.M. Computing Surveys (C.S.U.R.)*, *52*(5), pp.1-48.

Shannon, C.E., 1948. A mathematical theory of communication. *The Bell system technical journal*, *27*(3), pp.379-423.

The Sandfly Security Team, 2022, *Security Monitoring for Threats on Embedded Linux*, blog, viewed 29 May 2022, <https://www.sandflysecurity.com/blog/hunting-for-embedded-linux-systems-threats>.

The Times of India 2009, 'M.E.A. computers hacked again', *The Times of India*, 15 February, viewed 29 May 2022, < https://timesofindia.indiatimes.com/articleshow/4131304.cms >

Vajapeyam, S., 2014. Understanding Shannon's entropy metric for information. *arXiv preprint arXiv:1405.2061*.