A Review of Some Windows' File Metadata Which Could Highlight Indicators of Compromise

Peter Bentley, School of Business, Computing and Social Sciences, University of Gloucestershire, Cheltenham, UK

Abstract – Advanced Persistent Threats are known to obfuscate their malware through encryption, encoding, change of file extension. This paper reviews the files through analysis of the File Name, Index of Coincidence, alphabet size and File Extension Separator to highlight files which may be candidates for malware. It uses one bespoke program to calculate the Index of Coincidence and is mainly Living off the Land i.e. uses Windows software for other data manipulation.

Keywords –Microsoft Windows; Encrypt; Decrypt; Encode; Decode; Compression, Advanced Persistent Threat (APT); Malware; File Extension; Index of Coincidence; Indicator of Compromise; Base64, Alphabet Length; Living off the Land.

1. Introduction

This paper is the fourth in an occasionally issued series of post-doctoral Monographs. It is produced to demonstrate cost-effective ways to identify some types of malware. It should be read in conjunction with other metadata analysis (Bentley, 2023a).

The cyber security industry publishes the outcome of their analysis of Advanced Persistent Threats (APT) mainly in the form of white papers. Most of the documentation for this paper is based on two repositories of white papers, blogs and some academic papers: this author has compiled a repository of such papers, derived from web searches which were used as the basis for previous research and to which is continually added.

The Metadata analysed is: Filename, Filetype Extension, Index of Coincidence. File Alphabet Length, File length and number of dots/full stops in the filename extension.

One C program was written in support of this paper. All other software is taken from Microsoft i.e. this work for this paper is Living off the Land. The data under analysis is taken from all 9596 files in the C:\Windows\System32 directory Windows 10 Enterprise machine which is in regular use.

It is accepted that no one technique will find all malware. However, part of the point of defence is to increase attackers' costs (Bentley, 2021, pp 210 - 214).

This paper is agnostic towards the origin and intent of APTs.

2. Literature Review

It is known that Windows is a file-based system:

"NTFS is based round a relational database. This is the MFT or Master File Table. All "objects" stored on the volume are regarded as files, except the Partition Boot Record"

(Sammes and Jenkinson, 2007, p. 217)

Filenames have a specific format (Microsoft, 2024a). No academic list of filename extensions could be found. It is not good academic practice to reference wiki but only one extensive list was found (Wikipedia, 2025).

It has been documented (Bentley, 2023b) that APTs may obfuscate their attacks by:

- Encrypting or encoding file;
- changing the file extensions (.exe renamed .pdf) (Panda-Security, 2015, p. 4);
- using BASE64 and other base coding.

Each of these techniques changes the makeup of the contents of the file with respect to the file extension. For example, encryption and encoding will change the natural language content of Microsoft Word file to random text but if the file extension is unchanged then it will still be .docx. A file extension change will not change the internals of a .pdf to a .exe file.

One statistic that may highlight such changes is the Index of Coincidence.

The Index of Coincidence is a well-known statistic (Friedman, 1935). It is given by

$$\frac{c\ \sum_{i=1}^{c} f_i(f_i - 1)}{N(N-1)}$$

Where c is the alphabet length, f_i is the frequency count for the i-th character in the c-long alphabet and N is the file length or sum of the frequencies of the characters. The theoretical range of the Index of Coincidence [1,c]. This can be easily seen by having a file where all the

characters have equal frequency and another file where the file is populated by just one characters. In the first file, as N tends to infinity, the numerator tends to denominator divided by c and the Index of Coincidence equals 1. In the latter file there is only one character. The numerator equals the denominator which equals c. A demonstration of this for hexadecimal characters is given in Appendix A.

Bentley (Bentley, 2021, pp 254-255) has shown that a file may have as many as 12 file extensions. The work also discusses the use of the Index of Coincidence as a file categoriser. The work does not consider alphabet length and its use as a a possible Indicator of Compromise to highlight encrypted or encoded traffic.

It is known that attackers may perform DLL sideloading (Mitre, 2025) to achieve persistence and escalate privileges. This is performed by inserting the attacker's routine earlier in the search order for the program that is running. The attacker's routine has the same name as the legitimate routine but will be in a different directory.

This paper extends Bentley's work to produce a low-cost malware identifier by creating a shall bespoke program and using Microsoft software. Just as APTs are Living Off the Land (Wüest and Anand, 2017, pp. 7, 15-16)) then so is this the work documented in this paper.

3. The Bespoke Program, extns.exe

The program, extns.exe, is used in conjunction with the Microsoft program forfiles (Microsoft, 2023):

forfiles /P C:\Windows\System32 /S /M * /C "cmd /c D:\<Redacted>\extns.exe @path @ext @fsize"

The output of forfiles (full file path, extension and file size is used as input to extns.exe which outputs the input, followed by the file size used (the maximum is 10,000) alphabet length of the file and calculated Index of Coincidence of the file. This data is imported into a spreadsheet for manipulation. Within the spreadsheet, for each file, the number of dots (full stops, periods) in the filename is calculated.

4. Analysis

It is noted that not all file extensions in the appendices are displayed due to page size restrictions. The full graph is viewable in the spreadsheet**.**

4.1. File Names

Possibly the strongest metadata characteristic is the filename and this may be used to look for DLL sideloading.

With the spreadsheet it is possible to produce a pivot table containing a frequency count of filenames. Any filename with a count greater than one is a candidate for DLL sideloading.

Analysis revealed, for example:

- Two identically named .bud files - a backup or data file created by a specific program;
- Two identically named .gpd files – printer description file;
- more than one akshasp.sys file – associated with Sentinel/Aladdin HASP software.

Other identically named files were observed. While possibly benign it does highlight the legitimacy of this anti-malware technique.

4.2. File Extensions

The file extension count for all 9596 files is given in Table 1:

Number of dots in filename	0	1	2	3	4	5	6
Count	9	6824	904	9144	1692	22	1

Table 1: Count of file extensions

The file with six file extensions has a final file extension of .cat (Microsoft, 2024b) and has an Index of Coincidence of 2.75742 on a file length of 9448 consisting of a 256-long alphabet. A low Index of Coincidence and maximum alphabet length makes this file a good candidate for having been encrypted. The file is in a sub-directory pathname contains a Globally Unique Identifiers (GUIDs).

4.3. Index of Coincidence

Appendix B plots the Index of Coincidence against the file types (Note: The work is on all files but not all can be displayed). Recall that one measure of encryption is the closeness the Index of Coincidence is to 1.

The lowest Index of Coincidence for a .exe file is for C:\Windows\system32\curl.exe. It is also noted that alphabet length of the file 243 characters, of the 665135 the program extns.exe defaulted to the first 10,000. A low Index of Coincidence and high alphabet length make it a good candidate for it being encrypted. The file may be viewed in Notepad and confirmed this is an unencrypted executable.

The Index of Coincidence and alphabet lengths for .dat files varies. The Index of Coincidence ranges from 2.20297 to 190.06498, and the alphabet length ranges from 41 to 256 (the maximum). .dat files may be any format and are used by specific programs (Fileformat, 2025).

4.4. Alphabet Length

Appendix C plots the alphabet length against file type and from this it is easy to see files that use the full 256 characters and hence may be random/encrypted.

4.5. File Length

Appendix D plots the file length against Index of Coincidence and from this it is easy to see files that are statistically flat hence may be random/encrypted.

4.6. Base 64 and Other Encoding

Bentley (Bentley, 2023a) highlights the use of Base 64 and other Base encoding. The spreadsheet may be sorted by the alphabet length field and 11 different file extensions are noted. Again, this is a simple way to highlight files of a certain structure.

The idea may be extended to other Bases.

4.7. Looking for Non-standard File Extensions

This paper hypothesises that there may be file extensions used by APTs that are not in any known set of file extensions. No academic reference could be found for such a list. The only only that could be found (Wikipedia, 2025) should not be referenced in n paper such as this but an exception is being made.

We now have two files: a list of filename extensions under analysis from the C Drive (A) and a list of known filename extensions in a control set (C). We wish to find all extension in A that are not in C:

We combined both files; Boolean terms this file has data A&C as well as A|C. Putting this file into a spreadsheet and putting the data through a pivot table counting frequencies of filename extension: all extensions that have a count of 2 are from A & C; all extensions that have a count of 1 are from A | C. We copy the later and combine with a copy of our C Drive set (A) and put this set of data through a pivot table. Those filename extensions which appear twice are, therefore, unique to A i.e.:

$$A \,\&\, \overline{AC} = A \,\&\, (\bar{A} \mid \bar{C})$$

$$= A \,\&\, (\bar{A} \mid \bar{C})$$

$$= A\,\bar{A} \mid A\bar{C}$$

$$= A\bar{C}$$

In practical terms this means that our final dataset which contains file extensions unique to A should have a count of two. For the data under analysis there were no file extensions found which were not in the control set.

5. A Critique of this Work

The bespoke program extns.exe calculates the Index of Coincidence for a 256-character alphabet and does not take a Bayesian view of calculating the Index of Coincidence for the presented file extension.

The data for this paper was taken from C:\Windows\System32. Malware may not be restricted to this folder and the range could be extended.

6. Suggested Lines of Further Work

There may be white papers and hence, encryption algorithms not analysed in this paper. Researchers may wish to perform deeper analysis on some of the algorithms presented in this paper as well as the more complex assembler routines.

7. Concluding Remarks

This paper has demonstrated a cost-effective method of highlighting possible malware on a Microsoft Windows system and, by demonstrating such, may have increased the business costs of attackers.

REFERENCES

Bentley, P. (2023a) The Use of File Size Parity of Windows. exes and. dlls as a Malware Indicator of Compromise. Available at: https://eprints.glos.ac.uk/12778/ (Accessed: 4th August 2025).

Bentley, P. (2023b) A taxonomy of encryption and encoding algorithms used by advanced persistent threats with emphasis on bespoke encryption algorithms. Available at: https://eprints.glos.ac.uk/12874/ (Accessed: 4th August 2025).

Bentley, P. (2021) *The Treatment of Advanced Persistent Threats on Windows Based Systems* (Doctoral dissertation, University of Gloucestershire). Available at: https://eprints.glos.ac.uk/10332/ (Accessed: 4th August 2025).

Fileformat (2025) What is a DAT file? Available at: https://docs.fileformat.com/database/dat/ (Accessed: 4th August 2025).

Friedman, W. F. (1935) The Index of Coincidence and Its Application in Cryptanalysis. Washington: United States Government Printing Office Available at: https://archive.org/details/41761039080018 (Accessed: 4th August 2025).

Microsoft (2024a) Naming Files, Paths, and Namespaces Available at: https://learn.microsoft.com/en-us/windows/win32/fileio/naming-a-file (Accessed: 4th August 2025).

Microsoft (2024b) Catalog files and digital signatures Available at: https://learn.microsoft.com/en-us/windows-hardware/drivers/install/catalog-files (Accessed: 7th August 2025).

Microsoft (2023) forfiles Available at: https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/forfiles (Accessed: 4th August 2025).

Mitre (2023) Hijack Execution Flow: DLL Available at: https://attack.mitre.org/techniques/T1574/001/ (Accessed: 4th August 2025).

Panda-Security (2015) Operation “Oil Tanker” the Phantom Menace. Panda Security Available at: http://www.pandasecurity.com/mediacenter/src/uploads/2015/05/oil-tankeren.pdf (Accessed: 4th August 2025).

Sammes, T. and Jenkinson, B. (2007) Forensic Computing a Practioner's Guide. London 2010: Springer-Verlag.

Wikipedia (2025) List of filename extensions Available at: https://en.wikipedia.org/wiki/List_of_filename_extensions (Accessed: 4th August 2025).

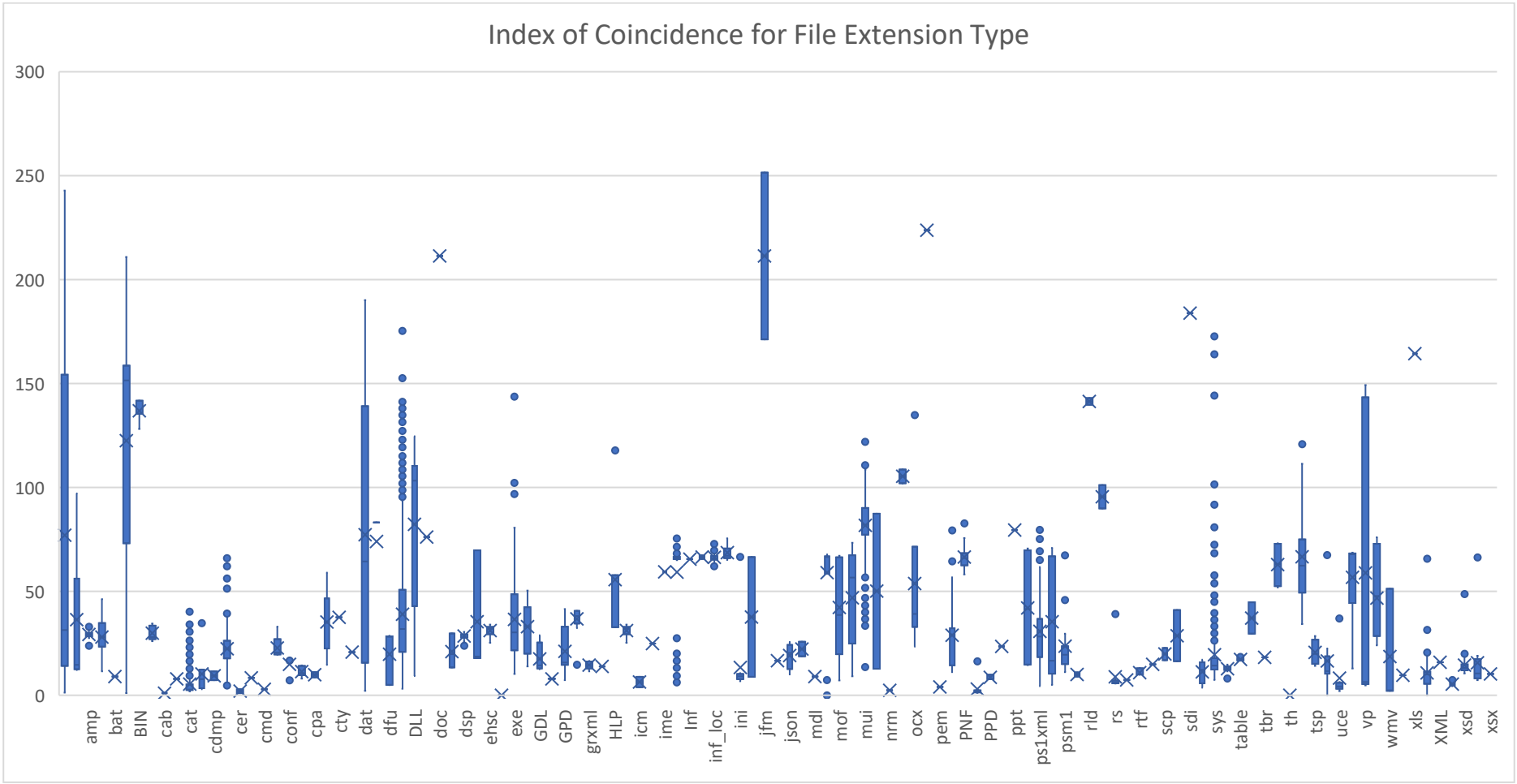
Wüest, C. and Anand, H. (2017) Living Off the Land and Fileless Attack Techniques. Symantec Available at: https://docs.broadcom.com/doc/istr-living-off-the-land-and-fileless-attack-techniques-en (Accessed: 4th August 2025).

Appendix A: Theoretical Index of Coincidences for Extremes of Flat and Rough File Frequency Counts

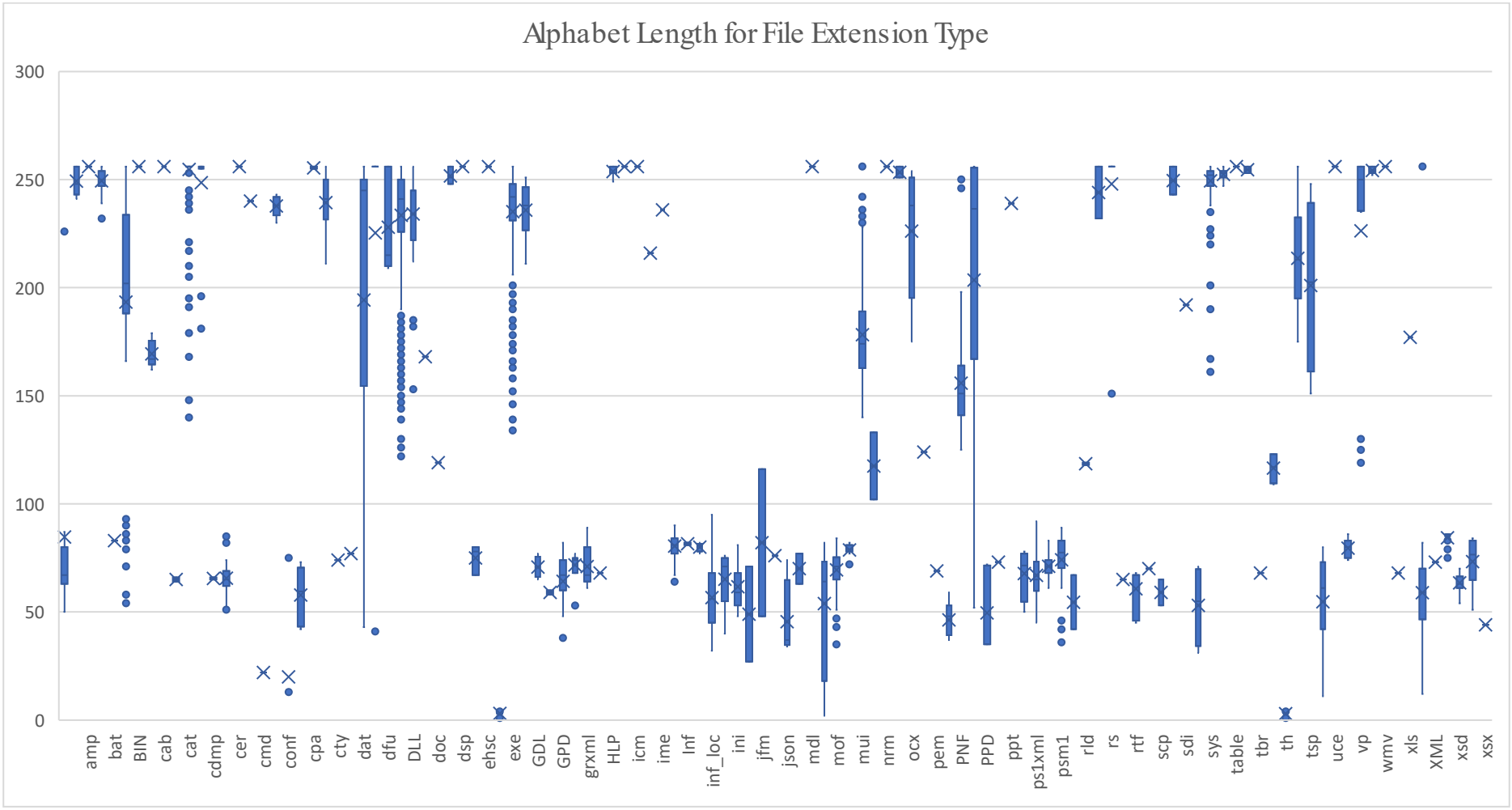
Charac-ter		Frequency Count (f)			f(f-1)		Charac-ter		Frequency Count (f)			f(f-1)
0		100			9900		0		1600			2558400
1		100			9900		1		0			0
2		100			9900		2		0			0
3		100			9900		3		0			0
4		100			9900		4		0			0
5		100			9900		5		0			0
6		100			9900		6		0			0
7		100			9900		7		0			0
8		100			9900		8		0			0
9		100			9900		9		0			0
A		100			9900		A		0			0
B		100			9900		B		0			0
C		100			9900		C		0			0
D		100			9900		D		0			0
E		100			9900		E		0			0
F		100			9900		F		0			0
	Sum* (Sum-1)	2558400		Sum	158400			Sum* (Sum-1)	2558400		Sum	2558400

Index of Coincidence	0.99062	Index of Coincidence	16

Appendix B: Index of Coincidence for File Extension Type



Appendix C: Alphabet Length for File Extension Type



Appendix D: Index of Coincidence for File Length