Vol.2019-MPS-123 No.49
Vol.2019-BIO-58 No.49
2019/6/19

文字コードの分布を考慮した特定データ検知手法の検討

西山魁人[†1]　鈴木海斗[†1]　松田健[†1]　園田道夫[†2]

概要：コンピュータ上で操作することができるファイルは様々な種類のものが存在する.その中には,悪意のあるプログラムが含まれていたり,ユーザーの PC に何らかの害を与えるファイルが含まれている場合もある.本稿では,バイナリエディタを用いていくつかのファイル形式の文字コード,特にアスキーコードの分布の状態について調査することで,特定のデータやコードを含むファイルを自動検知する手法について考察を行う.

キーワード：USB，セキュリティ，検知

Examination of specific data detection method considering distribution of character codes

KAITO NISHIYAMA[†1]　KAITO SUZUKI[†1]　TAKESHI MATSUDA[†1]
MICHIO SONODA[†2]

Abstract: There are various types of files that can be manipulated on the computer. It may contain malicious programs, or it may contain files that do some harm to your PC. In this paper, we examine a method of automatically detecting a file containing specific data or code by examining the character code of some file formats, especially the distribution state of ASCII code, using a binary editor.

Keywords: USB, Security, Detection

1. はじめに

日本で使用されているソフトウェアの脆弱性に関する情報を取りまとめている Japan Vulnerability Notes (JVN)のウェブページ[1]では，直近で発表された脆弱性の情報を収集することができる．共有されている情報のほとんどは，古くから知られている攻撃手法に関するものであり，技術的な対策方法が知られていたとしても，一般ユーザーがそのような攻撃そのものを知らなかったり，対策や対応策を知らなかったりするため，一般ユーザーにとってわかりにくい脅威を，わかりやすくするための手法に関する研究[2]は重要であると言える．本研究では，一般ユーザーが日常的に使う機会の多い USB メモリに焦点を当て，USB メモリの中身にどのような種類のファイルが含まれているかを簡易的に調べるための手法について検討する.

2. バイナリデータ

コンピュータ上で扱うことができるファイルは,バイナリデータとして見ることができる.バイナリデータとは,コンピュータが扱うことができるデータのことである.コンピュータ上のファイルをバイナリエディタ[3]で開くとファイルの種類ごとに特徴を持っていることがある.これを利用することで,目で見てどの種類のファイルか判断することができる.

本研究では,「Binary Editor BZ」というバイナリエディタを利用した.このバイナリエディタは,様々なファイルを 16 進数で表示することができる.また,1 バイトを 1 ドットとみなし図 1 のようなビットマップを表示することもできる.ビットマップの色付けのルールは以下のようになっている.

・白色：0x00(NULL 文字)
・水色：0x01~0x1F(ASCII 制御文字)
・赤色：0x20~0x7F(ASCII 文字)
・黒色：0x80~0xFF(MSB)

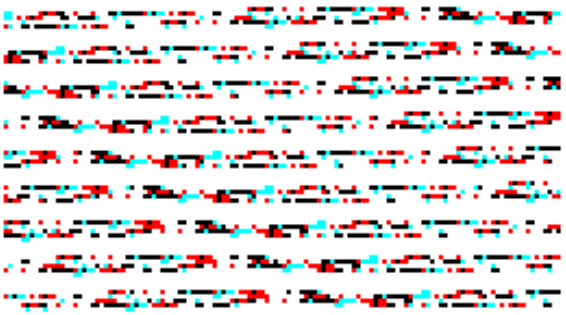

図 1 ビットマップの例
Figure 1 Example of bitmap

†1 長崎県立大学情報セキュリティ学科
†2 国立研究開発法人情報通信研究機構

Vol.2019-MPS-123 No.49
Vol.2019-BIO-58 No.49
2019/6/19

3. バイナリの分布と検知

本研究では,実行ファイルと USB メモリ接続時の通信データを用いて調査を行った.以下にバイナリエディタで出力したヘッダー部分のビットマップの図を示す.

バイナリエディタは,ビットマップを表示することはできるが ,サイズが小さなファイルであってもバイナリデータとして見ると情報量が多くなってしまう.そこでヘッダー部分周辺のビットマップの一部を切り抜くことで,ファイルの種類を特定できないか調査を図 4 の例のように行った.図 5,図 6 は,図 2,図 3 のビットマップからそれぞれ 30 ドット×30 ドットの正方形を切り抜き,900 バイト分のデータに対してバイナリエディタのビットマップの色付けルールを元に数値の出現回数を集計した度数分布表である.

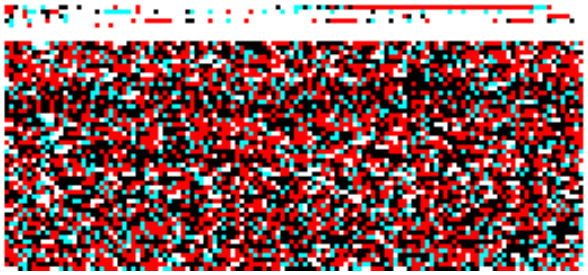

図 2　Lhaca076.exe のビットマップのヘッダー部分

Figure 2 Header part of Lhaca076.exe bitmap.

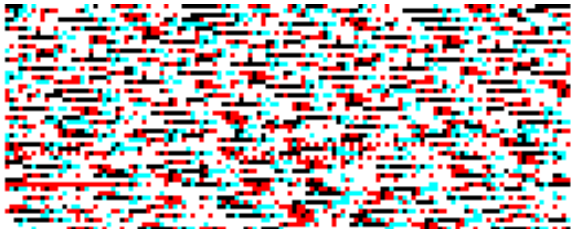

図 3 USB メモリの通信データのビットマップのヘッダー部分

Figure 3 Header part of USB memory communication data bitmap.

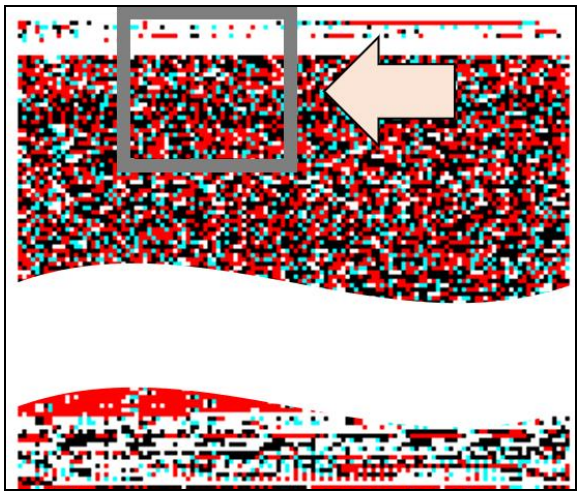

図 4 バイナリデータの切り抜き方の例

Figure 4 Example of how to crop binary data

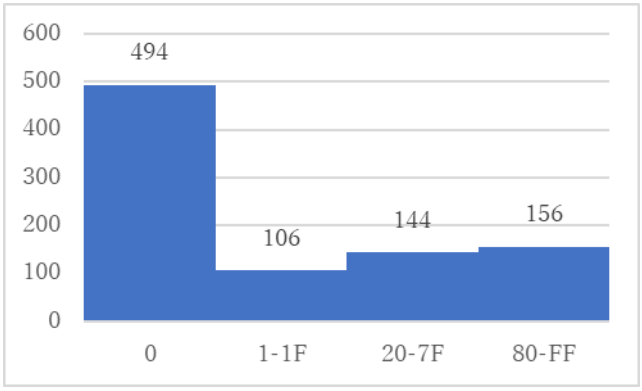

図 5 USB メモリの通信データの一部の度数分布表

Figure 5 Partial Frequency table of communication date of USB memory

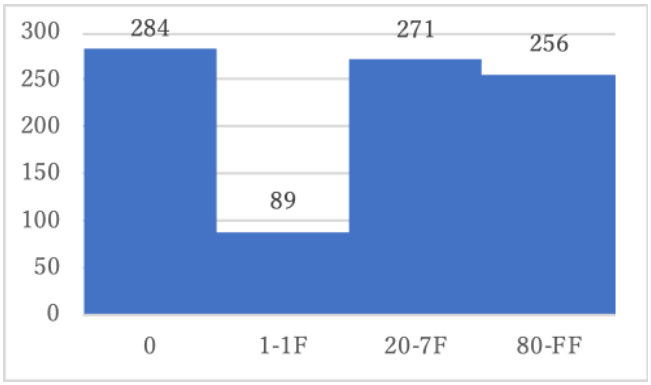

図 6 Lhaca076.exe のビットマップの一部の度数分布表

Figure 6 Partial frequency table of Lhaca076.exe bitmap

4. 調査結果

図 5 図 6 より, ビットマップの見え方だけでなく, 各種数値の出現回数も USB メモリの通信データと実行ファイルでは異なることが確認できた. 特に 20～7F(アスキー文字)の出現回数が 2 倍近く異なることが確認できた.

5. 考察と今後の課題

本研究では, 実行ファイルと USB メモリの通信データのバイナリデータで比較を行った. 今回調査したファイル形式以外のファイルも同様に調査し, それぞれどのような特徴があるのかを調査することが今後の課題である.

参考文献

[1] Japan Vulnerability Notes , https://jvn.jp
[2] 大谷康介, “ファイル情報の可視化による分類法の検討“. 情報処理学会第 77 回全国大会, 2015 (参照 2019-05-15).
[3] Binary Editor BZ , http://devil-tamachan.github.io/BZDoc/