ファイル構造検査による悪性 MS 文書ファイル検知手法の評価

大坪雄平 †　　三村守 ‡　　田中英彦 ‡

† 内閣官房情報セキュリティセンター
100-0014 東京都千代田区永田町 2-4-12
yuhei.otsubo@cas.go.jp

‡ 情報セキュリティ大学院大学
221-0835 神奈川県横浜市神奈川区鶴屋町 2-14-1

あらまし 実行ファイルを埋め込んだ MS 文書（Rich Text または Compound File Binary）ファイルを用いた標的型攻撃が発生している．これに対し，われわれは MS 文書ファイルのサイズや構造に関する情報を検査することで，悪性文書ファイルを検知する手法を提案した．しかしながら，CFB ファイルに対する評価や既存の手法との比較が不十分であった．本論文では，以前検証を行った検体に悪性一太郎ファイルも加えて提案手法の有効性を検証する実験を行った結果，98.5%の悪性 MS 文書ファイルを検知することができた．さらに Microsoft Office 2010 から導入されたファイル検証機能と検知率を比較する実験を行った結果，提案手法の有効性が確認できた．

Evaluation of Methods to Detect Malicious MS Document File using File Structure Inspection

Yuhei Otsubo†　　Mamoru Mimura‡　　Hidehiko Tanaka‡

†National Information Security Center
2-4-12 Nagata-cho, Chiyoda-ward, Tokyo 100-0014, JAPAN
yuhei.otsubo@cas.go.jp
‡Institute of Information Security
2-14-1 Tsuruya-cho, Kanagawa-ward, Yokohama 221-0835, JAPAN

Abstract Targeted attacks using MS document (Rich Text or Compound File Binary) files that contain executable files are popular. We proposed detection methods of malicious MS document files checking file size, file structure and so on. However, our previous paper has the insufficient evaluation to CFB files and other methods. In this paper, we added malicious Ichitaro files to previous specimen. Our methods could detect 98.5% of the malicious MS document files in the experiment. In addition, Microsoft added OFV (Office File Validation) to Microsoft Ofice 2010, we compared our methods's detection rate and OFV's one. The experimental result shows the effectiveness of our methods.

1　はじめに

近年では，特定の組織や個人を狙って情報窃取等を行う標的型攻撃が顕在化している．標的型攻撃の防御策として最新のパターンファイルを適用したウイルス対策ソフトを導入したとしても，標的型攻撃に用いられるマルウェアを検知できないことがほとんどである．この原因としては，標的型攻撃に用いるマルウェアが最新のウイルス対策ソフトで検知できないことを確認した後，攻撃者は特定の組織や個人にマルウェアを送付しているということが考えられる．さ

らに，攻撃を秘匿するため，マルウェアの本体である実行ファイルが埋め込まれた悪性文書ファイルが送付されることもある．

標的型攻撃に用いられる実行ファイルが埋め込まれた悪性文書ファイルの典型的な動作を以下に示す．悪性文書ファイルを開くと，閲覧ソフトの脆弱性を攻撃する exploit と呼ばれる部分が動作し，shellcode（侵入した端末を制御できるようにするためのコード）が実行される．shellcode は文書ファイルに埋め込まれた実行ファイルやダミー表示用の文書ファイルを取り出し，実行ファイルを実行したりダミー表示用の文書ファイルを表示する．これによって悪性文書ファイルを閲覧した端末はマルウェアに感染する．一方，文書ファイルに埋め込まれた実行ファイルやダミー表示用の文書ファイルはウイルス対策ソフト等の検知を回避するため様々な方式でエンコード（符号化）されているほか，ダミー表示用の文書ファイルの表示内容は通常の文書ファイルと変わらないため，一般に，悪性文書ファイルの受信者には実行ファイルが埋め込まれた悪性文書ファイルと通常の文書ファイルとを区別することは困難である．

われわれが Microsoft 社の開発した Rich Text[1]（rtf 拡張子）や CFB（Compound File Binary）[2]（doc，xls，ppt，jtd，jtdc 拡張子）の MS 文書ファイルに実行ファイルが埋め込まれた悪性 MS 文書ファイルを分析したところ，多くの悪性 MS 文書ファイルで通常の MS 文書ファイルとファイル構造に違いがあることが明らかになった．そこでわれわれは，MS 文書ファイルのファイル構造を検査し，悪性 MS 文書ファイルを検知する手法を提案した [3]．本論文では，提案手法の有効性を確認することを目的とし，検体に悪性一太郎ファイルを加えて検知率の評価を行うほか，Microsoft 社のファイル検証機能 [4] との比較も行う．

2　関連研究

われわれの提案手法は，MS 文書ファイルを対象に悪性 MS 文書ファイルであるか否かを検査する．以下，悪性 MS 文書ファイルの検知に関連する先行研究について述べる．

文献 [5] では，様々な形式の悪性文書ファイルに埋め込まれた実行ファイルを自動的に抽出するツールが提案されている．この手法では，実行ファイルを埋め込む際に使用される様々なエンコード方式を自動的に解読し，実行ファイルに頻出する文字列を検索することで実行ファイルを抽出することができる．MS 文書ファイル専用の解析ツールである OfficeMalScanner[6] は，MS 文書ファイルから不正なコードによく利用されるコードを検索したり，文書ファイルに埋め込まれた実行ファイルや別の文書ファイルをヘッダに使われる文字列を検索することにより抽出することができる．

文献 [7] では，MS 文書ファイルの構造を検査することにより，MS 文書ファイルに埋め込まれた，表示内容と関係のないデータを解析するツールが提案されている．このツールは MS 文書ファイル内でデータが秘匿される可能性がある 4 種類の場所を表示する．

Microsoft 社が Microsoft Office 2010 から導入したファイル検証機能は，Office 文書ファイル（xls，doc，ppt，pub 拡張子）を開こうとした場合に，正常なファイル構造の Office 文書ファイルか否かを検証し，正常なファイル構造でない場合に閲覧するか否か確認するポップアップウインドウが表示される機能である．ファイル検証機能がどのような仕組みでファイル構造を検証しているかは不明であるものの，ファイル構造を検証するという，われわれの提案と同じ目的の機能であるので，ファイル検証機能における検知率との比較を行う．

3　悪性 MS 文書ファイルのファイル構造

exploit の多くは閲覧ソフトの脆弱性を利用しており，閲覧ソフトが通常読み込む部分に埋め込まれている．一方，実行ファイルやダミー表示用の文書ファイルを閲覧ソフトが通常読み込む部分に埋め込むと，閲覧ソフトが誤動作したり，表示される内容が文字化け状態になってしまう．したがって，実行ファイルやダミー表示

用の文書ファイルは閲覧ソフトが通常読み込まない部分に埋め込まれることが多い．その結果，ファイル構造に通常の文書ファイルとは異なる特徴が表れる．

われわれが 2009 年から 2012 年の間に複数の組織において採取した実行ファイルが埋め込まれた悪性 MS 文書ファイルのファイル構造を分析し，判明した悪性 MS 文書ファイルの特徴を以下に示す．詳細については文献 [3] に示すとおりである．

3.1 Rich Text の場合

3.1.1 基本構造

Rich Text は Microsoft 社により開発された文書ファイルフォーマットの 1 つである．Rich Text のデータは通常，7bit の ASCII 文字列で記述されており，プレーンテキストに装飾やレイアウトのための制御用の文字列を付加した形式となっている．Rich Text ファイルの最初の文字は，“{” である．Rich Text は入れ子構造となっており，ファイルの最後の文字は，ファイルの最初の “{” に対応する “}”（EOF）となっている．閲覧ソフトは “}”（EOF）より後のデータについて通常は何もしない．

3.1.2 特徴 1：EOF 違反

一般的な Rich Text では “}”（EOF）がファイルの末尾となっていたが，実行ファイルが埋め込まれた Rich Text では EOF の後にデータが追加されているものがほとんどであった．

3.2 CFB の場合

3.2.1 基本構造

CFB は Microsoft 社により開発された複合ファイルフォーマットの 1 つである．Microsoft Word，Microsoft Excel や Microsoft PowerPoint 等で保存されるときに使用される doc，xls，ppt という拡張子のファイルは CFB を利用しており，文書ファイルに利用される様々なデータを

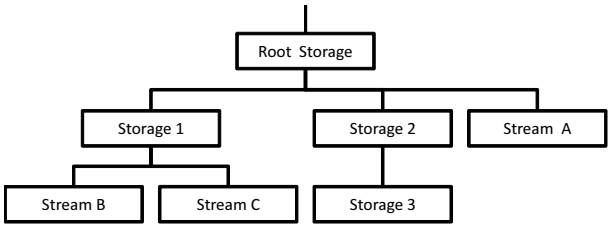

図 1: CFB の階層構造

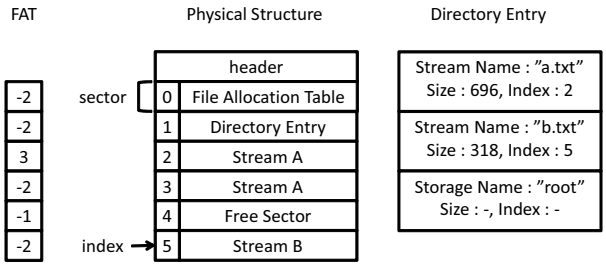

図 2: CFB のファイル構造

1 つのファイルに集約して保存している．また，JustSystems 社が開発した日本語ワープロソフトである一太郎で使用される jtd および jtdc という拡張子のファイルも CFB を利用している．CFB の階層構造を図 1 に示す．CFB の階層構造はファイルシステムによく似た構造となっており，ファイルに相当する Stream とディレクトリに相当する Storage の集合体となっている．CFB のファイル構造を図 2 に示す．CFB のファイル構造はヘッダと sector と呼ばれる一連の index 番号が振られた小さなブロックの集合で構成され，Stream は sector に分割して格納されている．各 sector の管理情報は FAT（File Allocation Table）という領域で管理されている．各 Stream，Storage の名称，サイズ，親子関係などの情報は DE（Directory Entry）という領域で管理されている．

3.2.2 特徴 2：ファイルサイズ違反

一般的な CFB ファイルのファイルサイズはヘッダサイズを除くと sector サイズの倍数であるが，実行ファイルが埋め込まれた CFB ファイルの中には，ヘッダサイズを除いたファイルサイズが sector サイズの倍数でないものがあった．

3.2.3　特徴 3：FAT 参照不可能領域

FAT において参照することのできる領域の大きさは，FAT に割り当てられた領域の大きさに依存している．一般的な CFB ファイルはヘッダを除いたすべての領域を FAT において参照することができていたが，実行ファイル埋め込まれた CFB ファイルの中には，FAT で参照可能な領域の上限を超えたファイルサイズのものがあった．

3.2.4　特徴 4：Free Sector 位置違反

一般的な CFB ファイルについて，ファイル末尾に該当する sector が Free Sector（未使用の sector）であることはなかったが，実行ファイルが埋め込まれた CFB ファイルの中には，ファイル末尾に該当する sector で Free Sector であるものがみられた．

3.2.5　特徴 5：使途不明の sector

CFB では，sector は，DIFAT（Double-Indirect FAT），FAT，miniFAT，DE，Stream または Free Sector の 6 種類に分類される．ここでいう DIFAT は，FAT に使用されている sector を管理するための領域であり，miniFAT はある一定サイズ未満の Stream をまとめて管理するための領域である．

一方，実行ファイルが埋め込まれた CFB ファイルの中には，上記 6 種類に分類できない sector を持つものがあった．

4　実験

4.1　試験プログラムの概要

これまでに示した 5 つのファイル構造上の特徴を検知するプログラムを，オープンソースのプログラミング言語である Python を用いて実装した．実装したプログラムの概要を図 3 に示す．試験プログラムは文書ファイルを引数として受け取り，悪性 MS 文書ファイルの特徴を検知するコマンドラインプログラムである．実装の詳細については文献 [3] に示すとおりである．

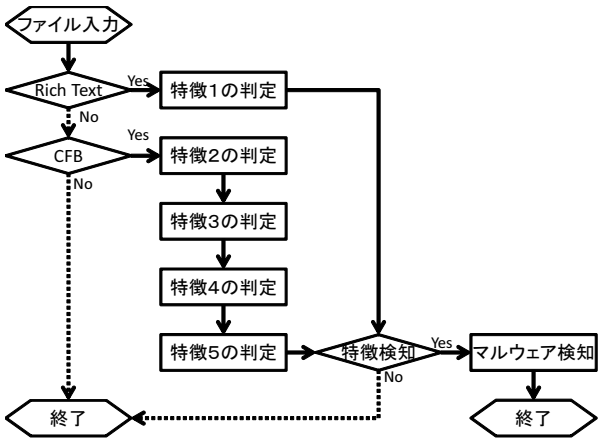

図 3: 試験プログラムの動作の概要

4.2　実験内容

試験プログラムの性能を評価するため，悪性 MS 文書ファイルを入力して結果を分析する．実験の対象となる MS 文書ファイルの概要を表 1 に示す．rtf，doc および xls 拡張子以外の悪性 MS 文書ファイルにも試験プログラムが有効であることを確かめるため，表 1 の検体は，文献 [3] で行った実験に使用した検体に一太郎で使用される jtd および jtdc という拡張子のファイルを加えたものである．これらの検体は，2009 年から 2012 年までに複数の組織において採取した固有のハッシュ値を持つ MS 文書ファイルで，分析により実行ファイルが埋め込まれていることをあらかじめ確認しているものである．ただし，拡張子は doc であるものの実際の中身は Rich Text であるものは rtf としている．これらの検体を試験プログラムに入力し，検知の成功率および平均実行時間を求める．また，試験プログラムの検知率と，採取した当時の最新パターンファイルを適用した大手ベンダのウイルス対策ソフトの検知率，OfficeMalScanner の検知率およびファイル検証機能の検知率を比較する．

実験を実施する環境は表 2 に示すとおりであり，実験はすべて仮想マシン上で行った．

表 1: 検体の概要

拡張子	検体数	平均容量 (KB)
rtf	98	266.5
doc	36	252.2
xls	49	180.4
jtd/jtdc	17	268.5
合計	200	243.0

表 2: 実験環境

CPU	Core i5-3450 3.1GHz
Memory	8.0GB
OS	Windows 7 SP1
Memory(VM)	2.0GB
OS(VM)	Windows XP SP3
Interpreter(VM)	Python 2.7.3

表 3: 試験プログラムの検知率

拡張子	検知数	検知率	平均実行時間
rtf	97 / 98	99.0%	0.021s
doc	35 / 36	97.2%	0.062s
xls	48 / 49	98.0%	0.051s
jtd/jtdc	17 / 17	100.0%	0.201s
合計	197 / 200	98.5%	0.051s

表 4: 検体の特徴ごとの検知状況

	検知数	検知率
特徴 1	97 / 98	99.0%
特徴 2	79 / 102	77.5%
特徴 3	92 / 102	90.2%
特徴 4	99 / 102	97.1%
特徴 5	98 / 102	96.1%

4.3 実験結果

検体の拡張子ごとの検知率を表 3 に示す．この表における検知数の欄は，検知数/検体数 を表している．検知の成功率は全体で 98.5%であった．また，平均実行時間は約 0.051s であり，最も実行時間が長いもので 0.391s であった．

検知に成功した 197 体の検体の特徴ごとの検知状況は表 4 に示すとおりである．表中の特徴 1 は rtf 拡張子のファイルの検知数であり，特徴 2 から特徴 5 までは doc 拡張子のファイルの検知数，xls 拡張子のファイルの検知数および jtd/jtdc 拡張子のファイルの検知数を合算した値である．

次に，試験プログラムの検知率と，大手ベンダのウイルス対策ソフトの検知率，OfficeMalScanner の検知率およびファイル検証機能の検知率との比較結果を表 5 に示す．実験に用いた検体に対しては，採取した当時の最新のパターンファイルを適用した大手ベンダのウイルス対策ソフトでも 20.0%から 21.0%の低い確率でしかマルウェアを検知することができなかった．しかも，ウイルス対策ソフトで検知できるマルウェアの種類には重複があったため，3 種類のウイルス対策ソフトを組み合わせた場合（T，S，M 社 AV）でも，検知率は 43.0%であった．

実験に使用した OfficeMalScanner のバージョンは v0.58 であり，jtd/jtdc を含む CFB ファイルについては，一般的な shellcode のパターンを検索する “SCAN” オプションおよび総当たりで実行ファイル等を検索する “BRUTE” オプションを使用して実行した．また，Rich Text ファイルについては，OfficeMalScanner に同封されている RTFScan を，“SCAN” オプションを使用して実行した．表中の OfficeMalScanner の検知数は OfficeMalScanner，RTFScan いずれかで検知した数を示す．OfficeMalScanner の検知率は 91.0%であった．

ファイル検証機能の実験には，Microsoft Office 2007 にファイル検証機能を有効にするために必要な更新プログラム（KB2464583（MS11-021），KB2464605 および KB2509488（MS11-023））を適用し，Office ファイル検証機能のアドイン（KB2501584）をインストールしたものを使用した．doc 拡張子および xls 拡張子の悪性文書ファイルを閲覧ソフトで読み込んだときに，ファイル検証機能が正常なファイル構造でないと判定し，ポップアップウインドウが表示され閲覧処理が開始されなかった場合にファイル検証機能による検知とした．実験に用いた検体に対するファイル検証機能の検知率は 38.8%であった．また，ファイル検証機能で検知できて，試験プログラムで検知できなかったものはなかった．

表 5: ウイルス対策ソフト等との検知率の比較

	検知数	検知率
試験プログラム	197 / 200	98.5%
T 社 AV	42 / 200	21.0%
S 社 AV	40 / 200	20.0%
M 社 AV	42 / 200	21.0%
T，S，M 社 AV	86 / 200	43.0%
OfficeMalScanner	182 / 200	91.0%
ファイル検証機能	33 / 85	38.8%

5 考察

5.1 検知に失敗した原因

試験プログラムが検知に失敗した検体を分析した結果，失敗の原因は，exploit および shellcode に連結する形で実行ファイルが埋め込まれているためであった．exploit および shellcode は閲覧ソフトが通常読み込む部分に埋め込まれることが多いことから，exploit および shellcode が埋め込まれた部分には本論文で論じたような特徴は現れないことが多い．exploit および shellcode と実行ファイルやダミー表示用の文書ファイルが別々の場所に埋め込まれている場合はわれわれの提案手法で検知することができるが，exploit および shellcode と実行ファイルやダミー表示用の文書ファイルが連結している場合はわれわれの提案手法で検知することはできない．一方，検知に失敗した 3 個について，OfficeMalScanner では 2 個検知することができ，Handy Scissors[5] では 1 個検知することができたが，いずれのツールでも検知することができなかったものは 1 個であった．

5.2 ファイル検証機能との差異

ファイル構造を検証するという，われわれの提案と同じ目的の機能を持つファイル検証機能の検知率は 38.8%であり，試験プログラムの検知率の 98.5%と比較して低い結果となった．検体の特徴ごとの検知状況とファイル検証機能の検知状況を比較したが，今回の実験で使用した検体では関連性を見出すことができなかった．加えて，各特徴の検知率のうち最も検知率の低い特徴 2 の検知率はファイル検証機能の検知率の値の 2 倍以上となっており，ファイル検証機能の機能に特徴 2 から特徴 5 までの判定が包含されてとは考えにくい．したがって，われわれの提案手法は，ファイル検証機能とは異なる手法である可能性が高いと考えられる．

5.3 試験プログラムの効果

試験プログラムは，検査処理に要する時間の平均値はわずか 0.051s で，98.5%という高い確率で悪性 MS 文書ファイルを検知することに成功した．試験プログラムは高速に検査することが可能であることから，試験プログラムを組織内のメールサーバ等で自動実行させれば，組織内に到達するメールの簡易チェックを実施することが可能である．添付ファイルがパスワードで暗号化された zip ファイルであった場合，ウイルス対策ソフトでは通常中身を検査することができない．一方，パスワードで暗号化された zip ファイルであっても，格納されているファイルの名称とサイズは復号しなくても判明する．これは，パスワードで暗号化された圧縮ファイルに格納された悪性 MS 文書ファイルにも特徴 2 の判定が適用可能であることを示している．

ウイルス対策ソフトはマルウェアに対応するパターンファイルを作成して検知するが，マルウェアは日々新たなものが出現している．OfficeMalScanner は不正なコードによく利用されるコードを検知するが，エンコードされたり未知の不正なコードは検知できない．Handy Scissors[5] はエンコード方式を解析し埋め込まれた実行ファイルを検知するが，未知のエンコード方式を利用したものは検知することができない．実験に用いた検体は，少なくとも採取した時点では大手ベンダの最新のパターンファイルを適用したウイルス対策ソフトでも，ほとんど検知することができない未知のマルウェアであった．それにもかかわらず，試験プログラムはパターンファイルを用いずに高い確率で悪性 MS 文書ファイルを検知することに成功した．さらに，試験プログラムは悪性 MS 文書ファイルに埋め込まれていた exploit や実行ファイルのエンコード

方式を解析することなく高い確率で悪性 MS 文書ファイルを検知することに成功した．また，われわれの提案手法と同様にファイル構造を検証するファイル検証機能と検知率を比較した結果，ファイル検証機能の検知率 38.8%に対し試験プログラムの検知率は 98.5%であり，試験プログラムが検知できないものでファイル検証機能で検知できるものは確認できなかったことから，われわれの提案手法の有効性が確認できた．

文書ファイルの仕様は，攻撃者の意志で変更することが困難であり，その結果，悪性 MS 文書ファイルのファイル構造はマルウェアやエンコード方式と比較して時間に対する変化が少なくなる．試験プログラムはその悪性 MS 文書ファイルのファイル構造を検査対象としており，今後プログラムを更新しなくても高い検知率を維持することが可能であると考えられる．また，いわゆるゼロデイ攻撃にも本論文の提案手法は有効である．例えば，実験に用いた jtd/jtdc 拡張子の検体 17 個中 14 個は検体入手時点で対策の取られていない脆弱性を利用したものであったが，試験プログラムはすべて検知することができた．doc 拡張子，rtf 拡張子などのファイルでも同様で，未知の脆弱性を利用した検体であっても，実行ファイルを埋め込む場所が変わらなければ検知することができると考えられる．

一方，われわれの提案手法は，原理的に exploit および shellcode はほぼ検知することはできない．したがって，exploit および shellcode に連結する形で実行ファイルが埋め込まれているものや exploit および shellcode のみが埋め込まれてるもの，例えば実行ファイルを外部のサーバ等からダウンロードするようなものは検知することができない．これらについては，われわれの提案手法以外の方法で検知する必要がある．

6　おわりに

本論文では，ファイル構造検査により悪性 MS 文書ファイルを検知するという，われわれの提案手法の有効性を検証する実験を行った結果，平均実行時間 0.051s で 98.5%の悪性文書ファイルを検知することができた．ファイル構造は攻撃者の意志で仕様を変更することが不可能であることから，提案手法は長期にわたり有効である．

参考文献

[1] Microsoft：Rich Text Format (RTF) Specification, version 1.9.1(online), `http://www.microsoft.com/en-us/download/details.aspx?id=10725` (2013-05-22).

[2] Microsoft：[MS-CFB]: Compound File Binary File Format(online), `http://msdn.microsoft.com/en-us/library/dd942138.aspx` (2013-05-22).

[3] 大坪雄平，三村守，田中英彦：ファイル構造検査による悪性 MS 文書ファイルの検知，情報処理学会研究報告, Vol.2013-IOT-22, No.16 (2013).

[4] Microsoft：Microsoft Office 向けの Microsoft Office ファイル検証機能の公開 (online), `http://technet.microsoft.com/ja-jp/security/advisory/2501584` (2013-06-18).

[5] 三村守，田中英彦：Handy Scissors：悪性文書ファイルに埋め込まれた実行ファイルの自動抽出ツール，情報処理学会論文誌, Vol.54, No.3, pp.1211-1219 (2013).

[6] Boldewin, F.：Analyzing MSOffice malware with OfficeMalScanner(online), `http://www.reconstructer.org/papers/Analyzing%20MSOffice%20malware%20with%20OfficeMalScanner.zip` (2013-05-08).

[7] Hyukdon, K. Yeog, K. Sangjin, L. and Jongin, L.：A Tool for the Detection of Hidden Data in Microsoft Compound Document File Format, *ICISS '08 Proceedings of the 2008 International Conference on Information Science and Security*, pp.141-146 (2008).