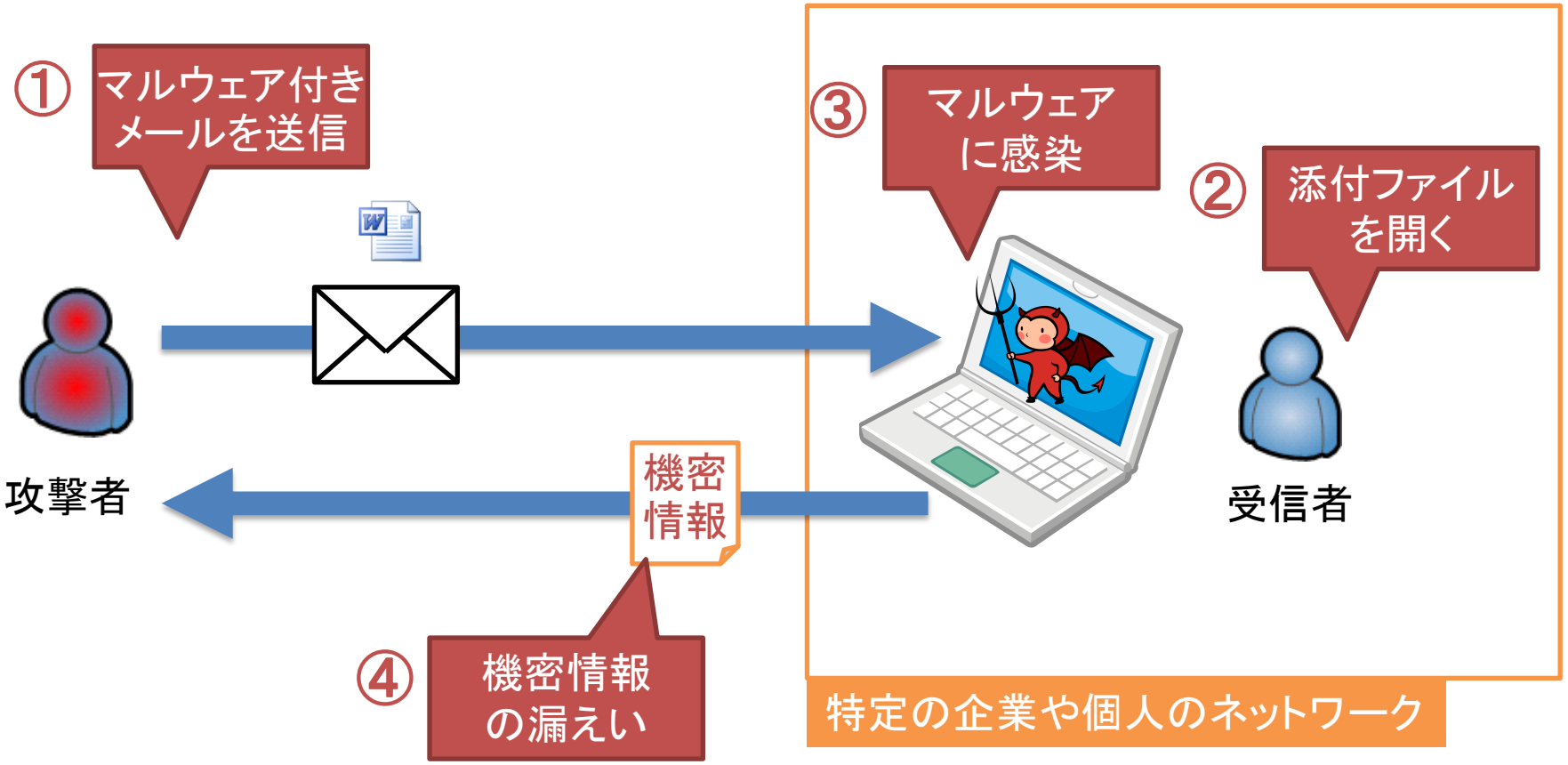
ファイル構造検査による悪性MS文書ファイルの検知手法の評価

大坪　雄平
三村　　守
田中　英彦

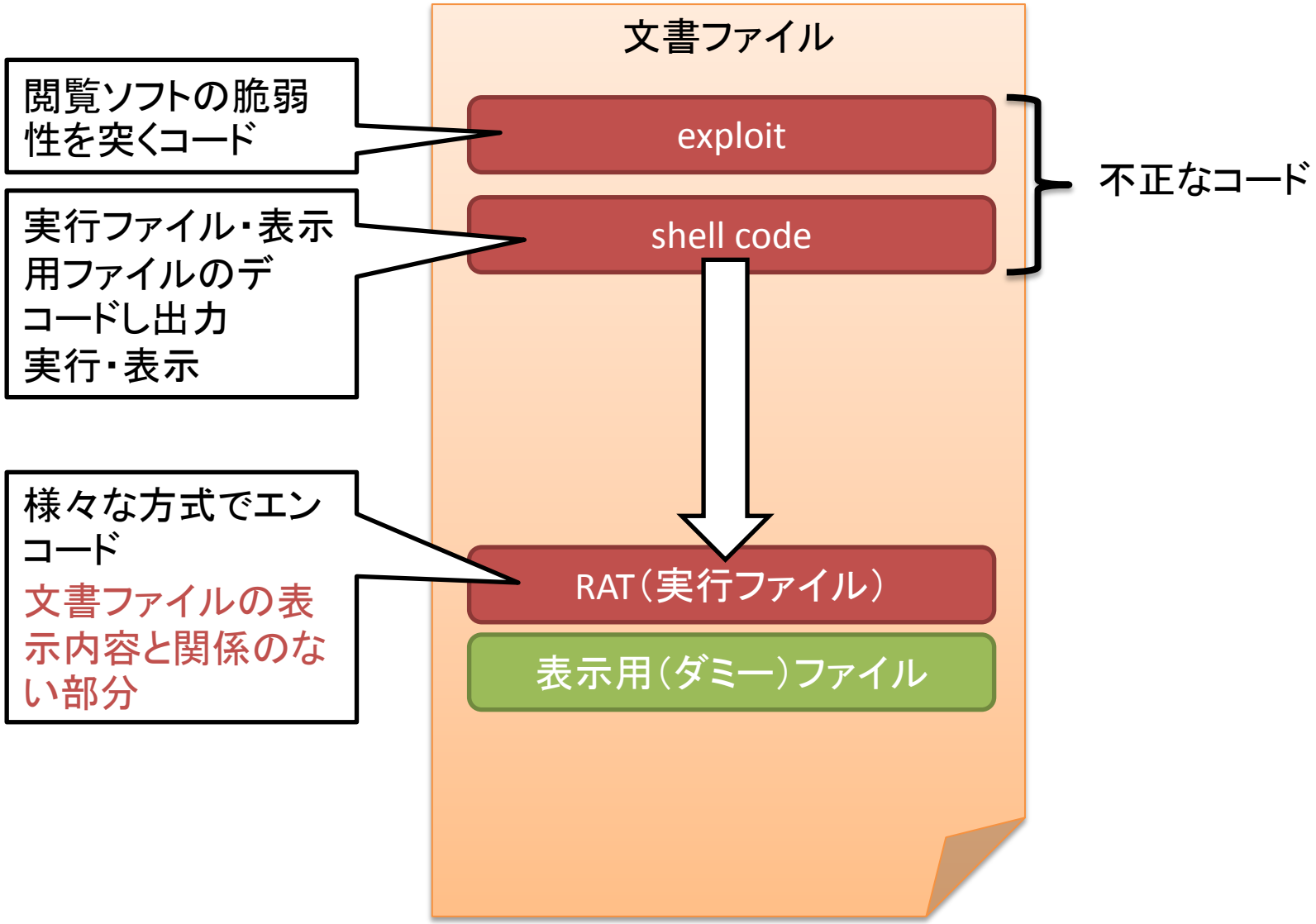
目次

標的型攻撃の脅威



実行ファイルが文書ファイルに埋め込まれた場合
受信者がマルウェアに気づくことは難しい

実行ファイルが埋め込まれた悪性文書ファイルの構造



関連研究

静的解析

特徴	課題
文書ファイルに不正なコードや実行ファイルが埋め込まれているか否かを判定	不正なコードや実行ファイルは様々な方式でエンコードされている

動的解析

特徴	課題
文書ファイルを実際に閲覧して挙動を解析。	悪性文書ファイルが動作する環境を整備する必要がある

我々の提案

文書ファイルの仕様はエンコードに依存しないことに着目し悪性文書ファイルの特徴を検知する手法を提案

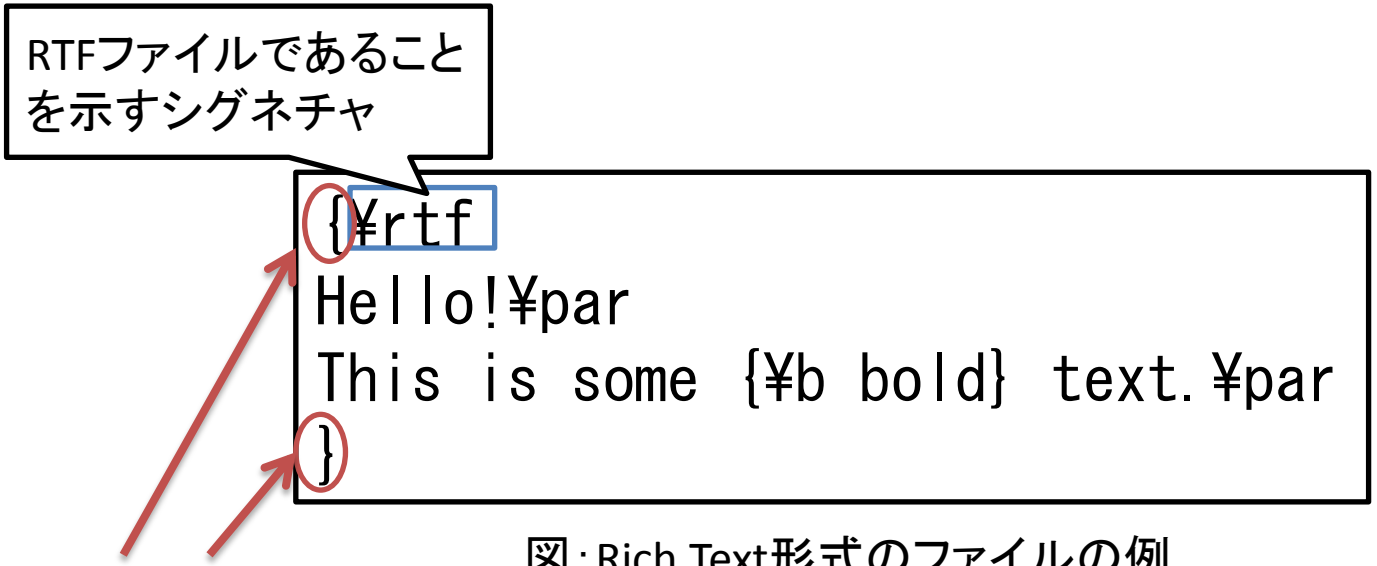
IOT22で発表

本研究

提案した手法の評価

Rich Text形式のファイルの基本構造

RTFのデータはテキスト形式を用いており、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※



図：Rich Text形式のファイルの例

最初の‘{‘に対応する’}’がファイルの最後（EOF）

※：wikipediaより引用

特徴1:EOF違反

EOFの後にデータが存在

表示に影響を与えないファイル末尾に実行ファイルを挿入

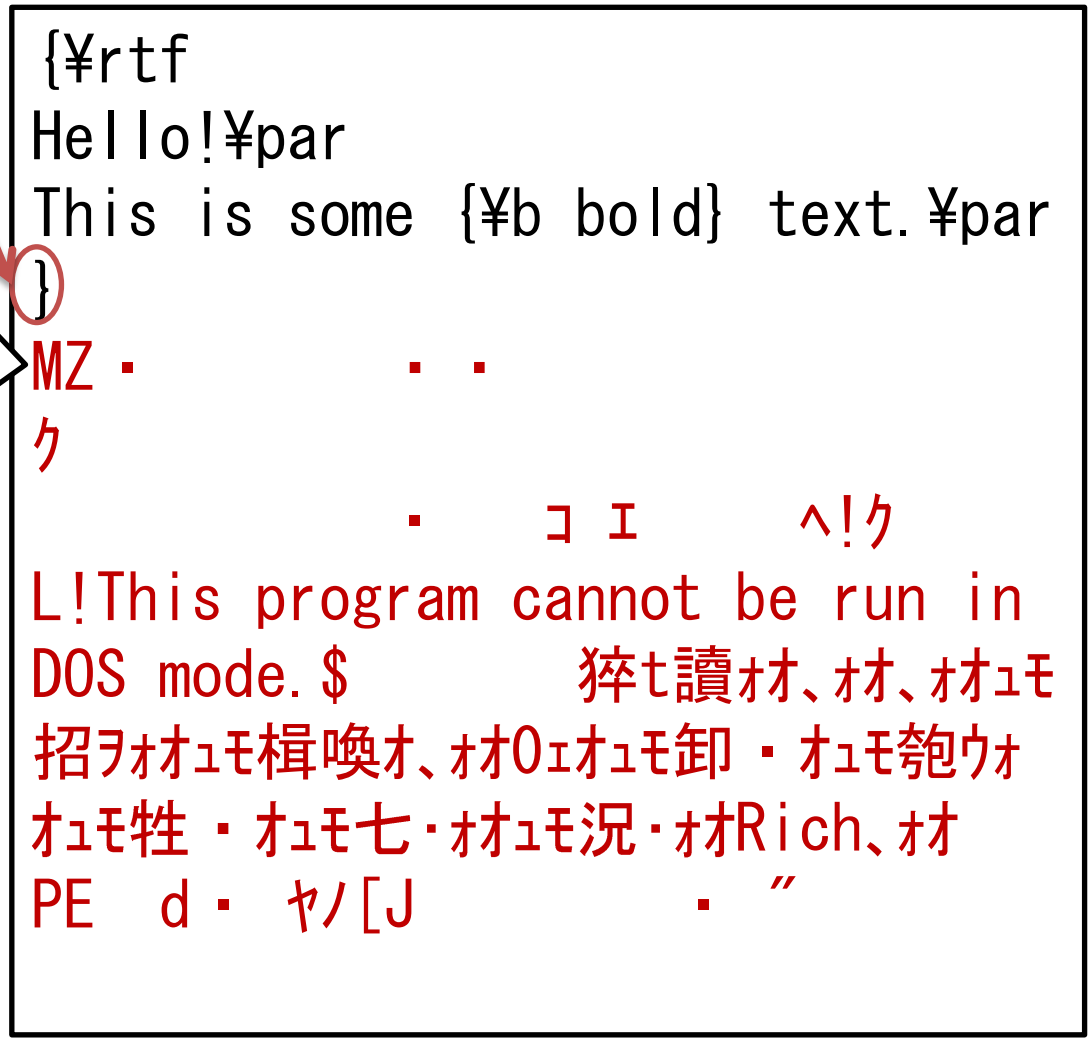
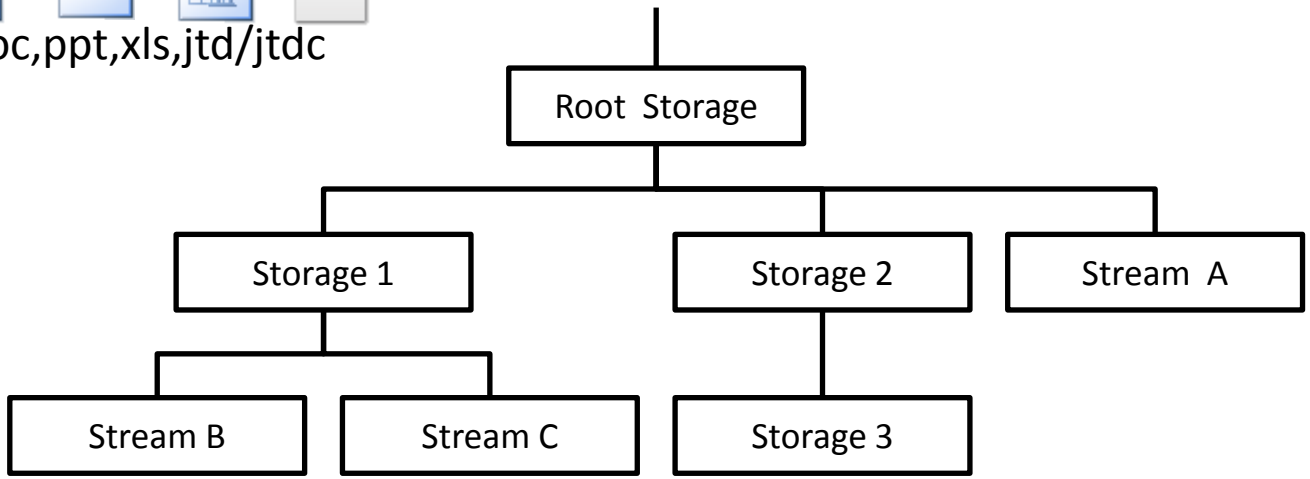


図:実行ファイルを埋め込まれたRich Text形式のファイルの例

CFB（Compound File Binary）形式のファイル（doc、xls、ppt）

複数のデータを階層構造で1つのファイルに格納できる
Microsoft社が作成したアーカイブ形式
Micrsoft Word等で使用されている

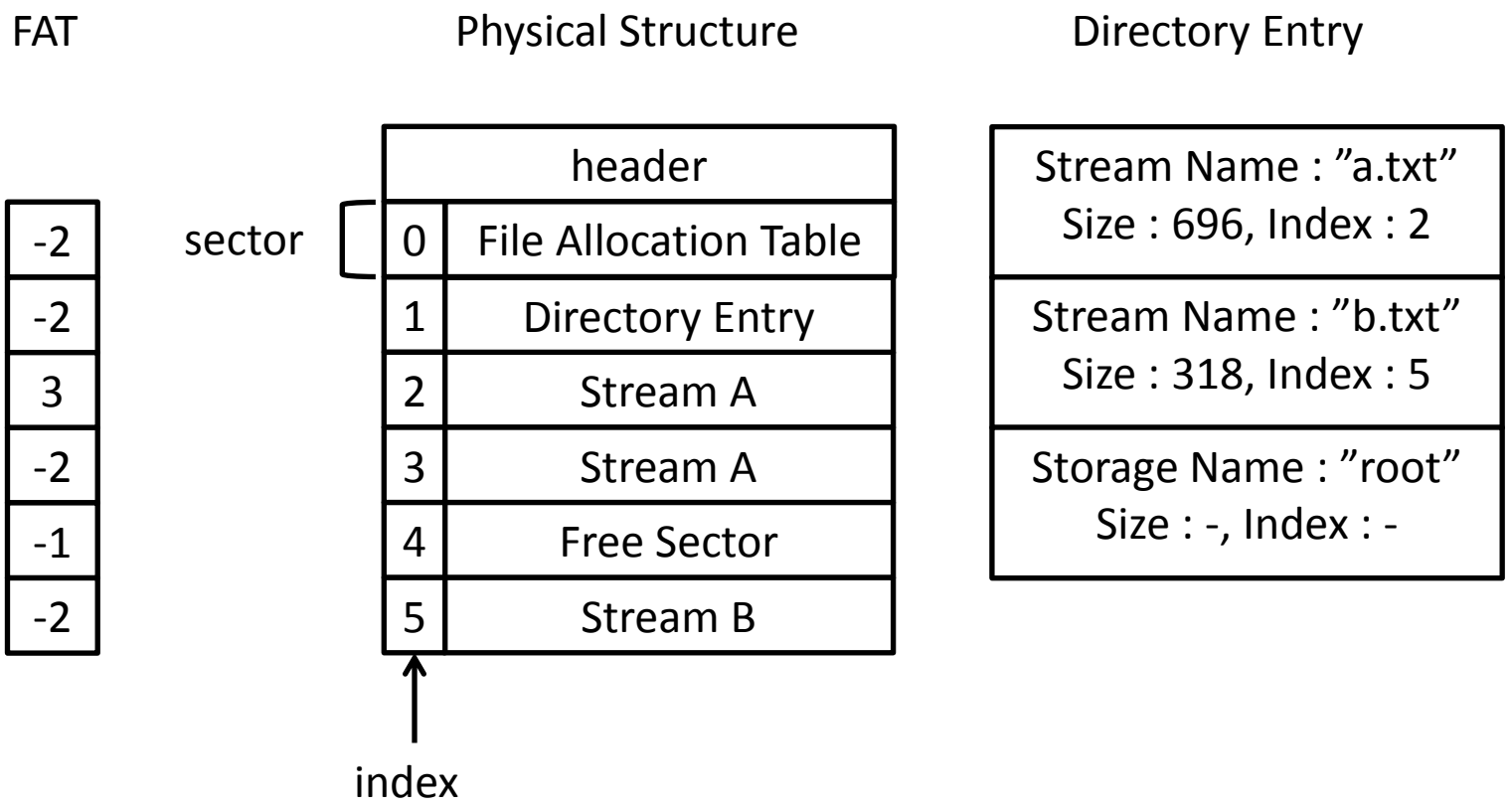
doc,ppt,xls,jtd/jtdc



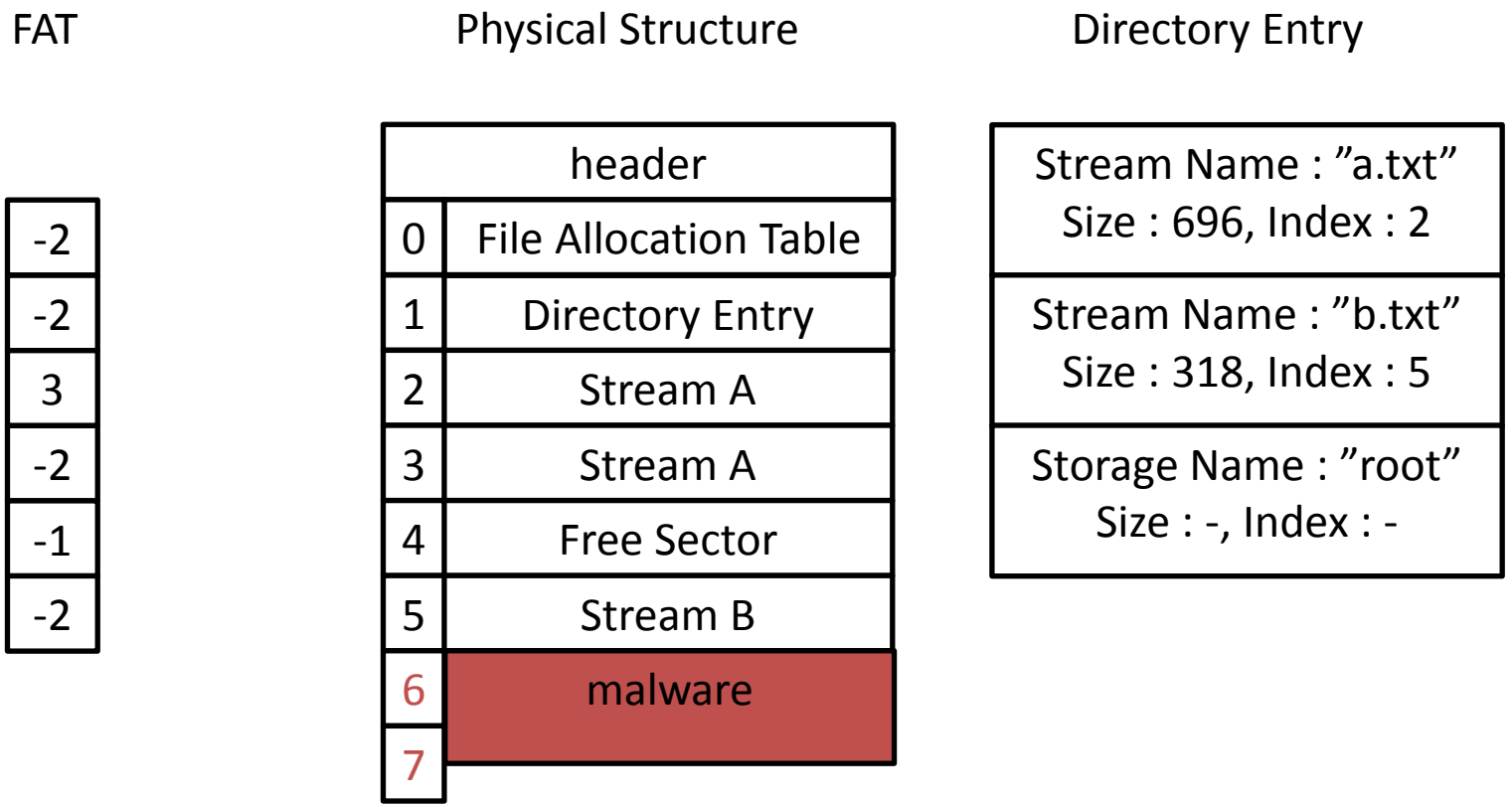
ファイルシステムでいうと
Stream → ファイル
Storage → フォルダ

引用：[MS-CFB] – v20130118 Compound File Binary Format （http://msdn.microsoft.com/en-us/library/dd942138.aspx）

CFB（doc、xls、ppt）のファイル構造

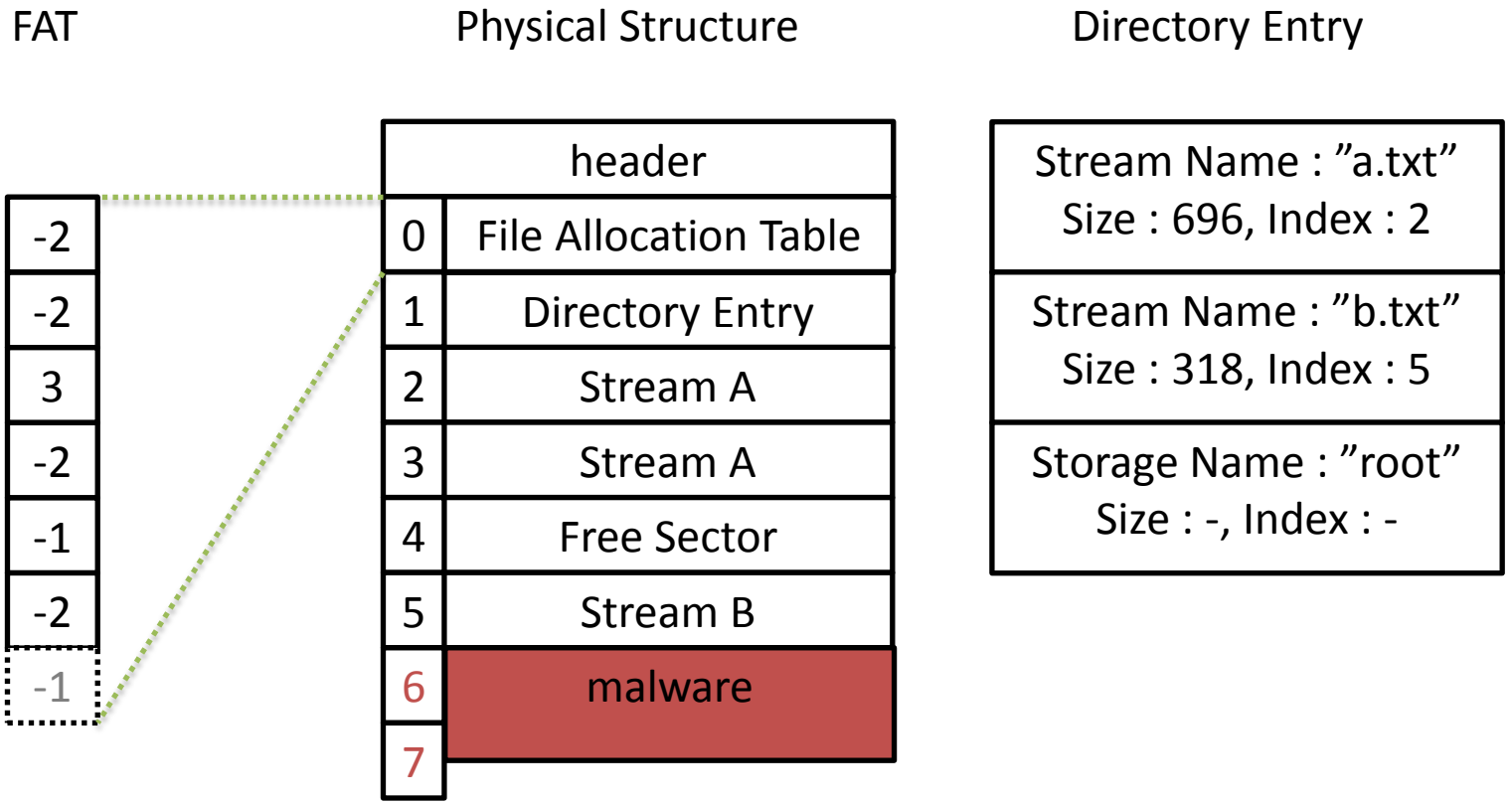


特徴2:ファイルサイズ違反



ヘッダを除いたファイルサイズが
sectorサイズ単位でない

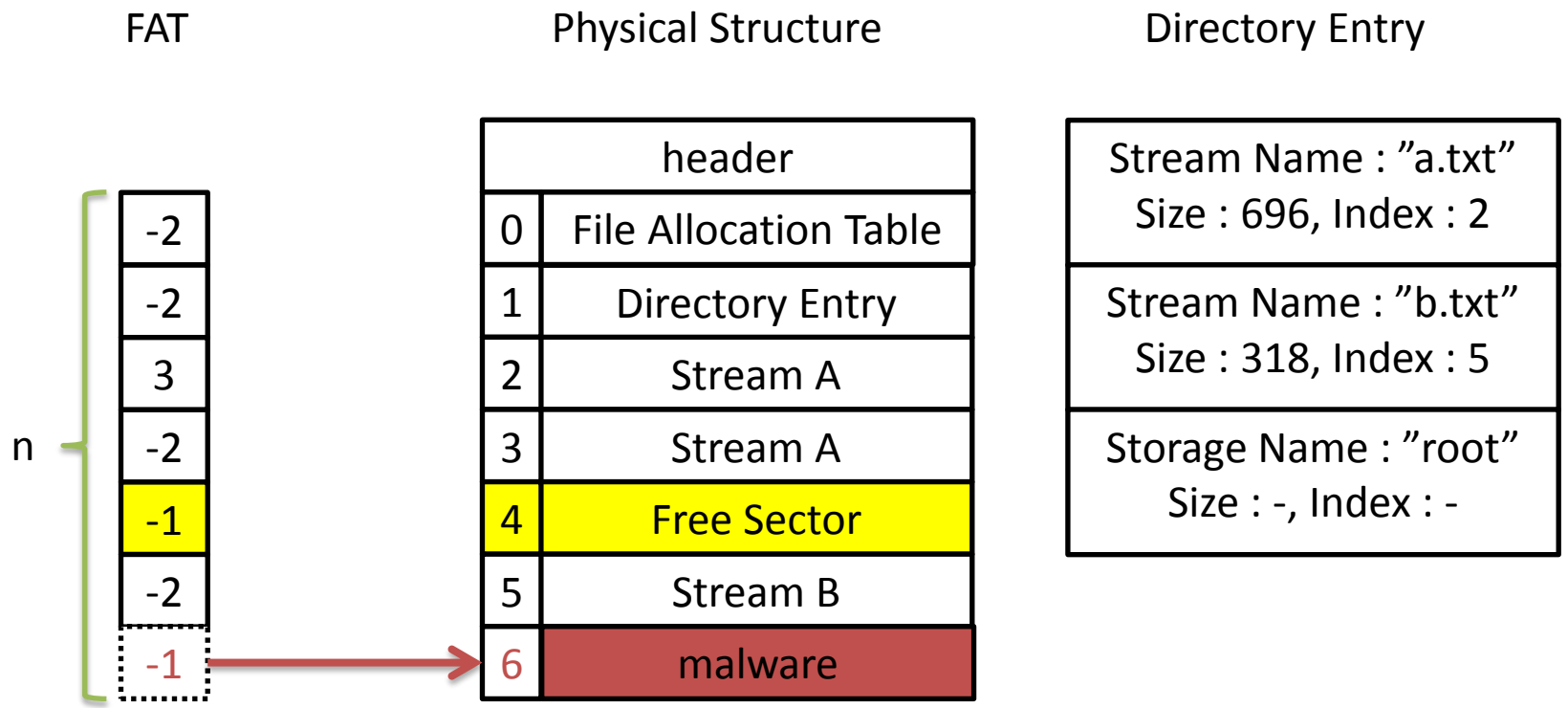
特徴3：FAT参照不可領域



FATで参照可能な領域
通常は
FATセクタ数×128×512 (Byte)

FATで参照可能な領域にファイルが収まっていない

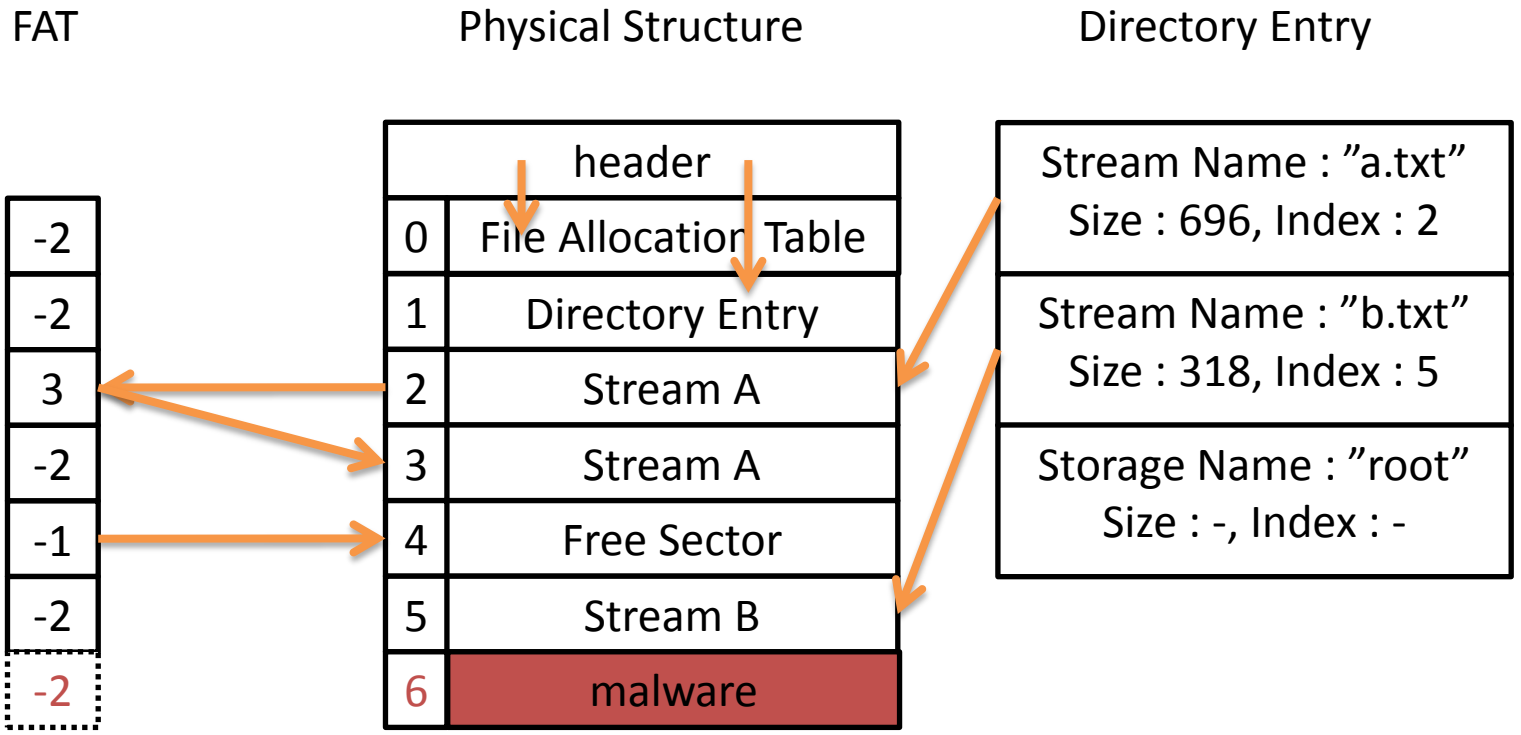
特徴4：Free Sector位置違反



sectorサイズが512バイトの場合
n = (ファイルサイズ-512)/512

ファイルの最後に対応するsector（n番目のsector）がFree Sector

特徴5：使途不明のsector



FAT（DI-FAT、miniFATを含む）、DE、Stream、FreeSectorに分類できないsectorがある

ファイル構造の検査による悪性MS文書ファイルの検知

検知名	検知内容	対象ファイル
手法1	特徴1：EOF違反	Rich Text(rtf)
手法2	特徴2：ファイルサイズ違反	CFB （doc,xls,ppt,jtd/jtdc)
手法3	特徴3：FAT参照不可領域	
手法4	特徴4：Free Sector位置違反	
手法5	特徴5：使途不明のsector	

実行ファイルが埋め込まれた場所に現れる特徴
exploitやshellcodeの場所には現れない

検体の概要

2009年から2012年までに複数の組織において受信した不審なメールに添付されたもの
実行ファイルが埋め込まれていることが確認されている文書ファイル

	悪性MS文書ファイル	
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

※docに拡張子が偽装されたRich Textはrtfでカウント

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

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

試験プログラムの検知率（特徴別）

	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%

ウイルス対策ソフト等との検知率の比較

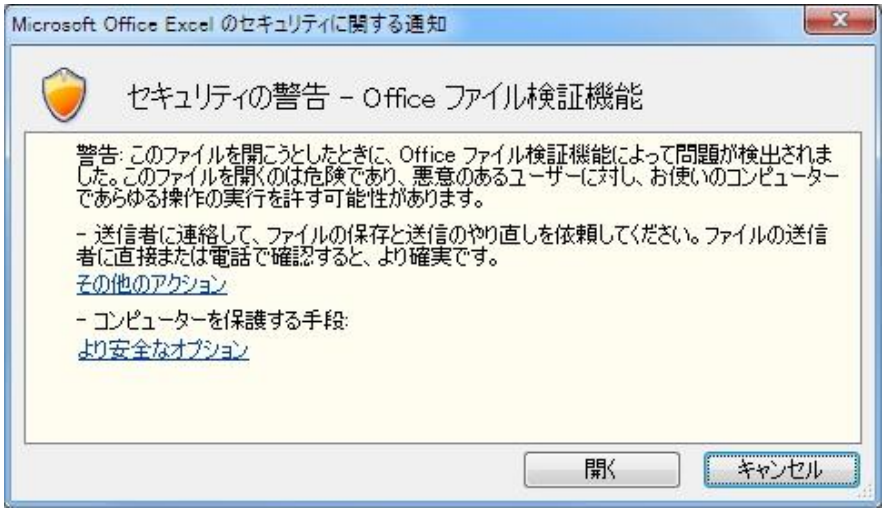
	検知数	検知率
試験プログラム	197 / 200	98.5%
T社AV	42 / 200	21.0%
S社AV	40 / 200	20.0%
M社AV	42 /200	21.0%
T,S,M社AV	86 / 200	43.0%

各社AV　　:検体入手時点での最新のパターンファイルを適用したもの

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%
OfficeMalScanner（RTFScan）	182 / 200	91.0%
Officeファイル検証機能	33 / 85	38.8%

OfficeMalScanner :v0.58、MS文書ファイル専用の解析ツール
Officeファイル検証機能 :Office 2010から導入された正常なファイルか検証する機能



考察

検知に失敗した原因

原因はexploitやshellcodeに連結する形で実行ファイルが埋め込まれていた

doc　　:flashの中に実行ファイルが埋め込まれているもの
xls　　:VBAの中に実行ファイルが埋め込まれているもの
rtf　　:rtfのexploit部分と一体化（CVE-2010-3333）

今回の検知方式では検知不可能

試験プログラムの効果

- 実用性

検知率　98.5%
平均実行時間　0.051s

- 検知プログラムの更新頻度が低い

検査対象	更新頻度	
マルウェア	高	1日あたり20万個の新種
エンコード方式	中	マルウェア埋め込みツールの更新頻度に依存
ファイル構造	低	文書ファイル形式の更新頻度に依存

攻撃者の意志でコントロール可能

攻撃者の意志ではコントロールできない

- 暗号化された圧縮ファイルへの適用

拡張子とファイルサイズの情報のみで77.5%の検知率
パスワード付きzipは復号しなくても書庫のディレクトリ構造は復元可能

まとめ

- 実行ファイルが埋め込まれた文書ファイルのファイル構造上の特徴を調査し検知法を考察
- 上記検知法を実装した試験プログラムを作成し、検知率を評価
- 試験プログラムの効果について考察

今後の課題

- より詳細に構造検査する検知手法の考案、実装
- 他の文書ファイル形式への対応