社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－DPS－126 (47)
2006－CSEC－ 32 (47)
2006／3／17

自己ファイルREADの検出による 未知ワーム・変異型ワームの検知方式の提案

鈴木 功一* 松本 隆明** 高見 知寛* 馬場 達也** 前田 秀介** 西垣 正勝*

*静岡大学情報学部, ** NTTデータ技術開発本部,

あらまし．ワームの感染は，ワーム自身を他のPCにネットワーク経由でコピーすることに他ならない．よってワームの感染行動は，OSのファイルシステム上では，自分自身のファイルをREAD（コピー）し，これを通信APIにWRITE（ペースト）するという動作として現れる．本稿では，この「ワームの自己ファイルREAD」を検出することにより，ワームを検知する方式を提案する．原理的にはワームは必ず自己ファイルREADを行うため，本方式によれば未知ワームや変異型ワームも検知可能であると考えられる．また本方式は，エンドユーザのPCにおける各プロセスのファイルアクセスを常時監視することにより実装可能であるため，ワームのリアルタイム検知も実現できる．本稿では本方式のコンセプトを示した上で，ファイルアクセスを監視するモニタツールを用いて擬似的に本方式の未知ワーム検知能力を検証する．

An unknown-worm and mutated-worm detection scheme based on capturing self-initiated READ behavior

Koichi Suzuki* Takaaki Matsumoto** Tomohiro Takami*
Tatsuya Baba** Shusuke Maeda** Masakatsu Nishigaki*

*Faculty of Informatics, Shizuoka University, ** R&D Headquarters, NTT Data corp

Abstract. Worm infection is just to copy the worm onto other PC by way of a network connection. Therefore, it is observed as the following behaviors; (1) COPY: read their own executable file, and (2) PASTE: write the file onto the stream communication API. This paper proposes to use this type of worm's "self-initiated READ behavior" for unknown-worm detection. It is expected that the worm detection scheme based on capturing self-READ behavior could be applicable to a variety of worms including mutated-worm since this behavior is basically found in most of them. Moreover, this scheme could achieve real-time worm detection because the self-READ behavior can be done just by capturing the file access of every process. In this paper, the conceptual design of the proposed scheme is described and its feasibility is investigated by using a tool kit to capture the file access in the OS.

1．はじめに

これまでに様々な未知ワーム検知手法が提案されてきており，その代表的なものが，プログラムの振る舞いにおける「ワームらしさ」を検出するビヘイビアブロッキング法[1]である．ワームらしい振る舞いとしては，一般的に，レジストリの改ざん，システムファイルの書き換え，外部への感染活動などが規定される[1]．ビヘイビアブロッキング法は，プロセスが発行するシステムコールなどを検査することによりコンピュータ上で動作しているプログラムの動きを監視し，ワームらしい振る舞いをした場合に，それをワームとして検知する．

ビヘイビアブロッキング法はワームらしいプログラムをすべて検知することができるため，基本的には，亜種ワームやポリモーフィック／メタモ

ーフィック型ワームを含むすべての未知ワームを検知することが可能である．しかし，レジストリの改ざんやシステムファイルの書き換えは，OSのアップデートや正常のアプリケーションプログラムのインストール時にも発生する振る舞いであり，また，外部への感染活動と正常の通信を確実に見極めるには限界がある．このため，ビヘイビアブロッキング法には，ワームと類似した動きをする正常なプログラムを誤検知してしまうという大きな問題がある．すなわち，ビヘイビアブロッキング法の効果を高めるためには，真に「ワームらしい振る舞い」を見極めることが重要となる．

そこで本稿では，「ワームは，感染行動の中で自分の複製を作成するにあたって，自分自身のファイルをREADする」というワーム特有の振る舞いに着目し，プロセスのファイルアクセスをリアルタイムで監視することによりワームを検知する方法を提案する．原理的にはワームは感染行動の際に必ず自己ファイルREADを行うため，本方式によれば未知ワームや変異型ワームも検知可能であると考えられる．

2．自己ファイル READ の検出によるワーム検知

ワームの感染は，ワーム自身を他のPCにネットワーク経由でコピーすることに他ならない．ファイルのネットワーク経由のコピーを，コピー元のPCにおけるOSの観点で見ると，その処理は1）コピー元のファイルを読み出し（READ），2）これを通信APIであるWINSOCKに引き渡す（WRITE），という動作により実行される．よってワームが感染活動を行うにあたっては，自分自身のファイル（ワーム本体のファイル）をREADするという動作が発生する．正規のプログラムの中にも自分自身の本体のファイルをREADするプログラムが存在するが，著者らが確認した限りでは，ワームのように自分自身のファイルからそのデータのほぼ「すべて」をREADする正規のプログラムは現在のところ発見されていない．以降，本稿では上記のようなワームが自分自身のファイルの大部分をREADするという動作を「自己ファイルREAD」と呼ぶことにする．

本稿では，この「ワームの自己ファイル READ」を，ビヘイビアブロッキング法におけるワームらしい振る舞いとして利用する．具体的には，OSのファイルシステムをフックすることにより，エンドユーザのPCにおける全プロセスのファイルアクセスを常時監視し，自己ファイルREADを行ったプロセスをリアルタイム検知する．よって，ワームが感染活動を開始した瞬間にこれを発見することが可能である．

本方式の特長として，本方式は従来の方法では検知が困難であった変異型ワームに対しても有効に検出が可能である点が挙げられる．変異型ワームは感染の度に自分自身を暗号化（ポリモーフィック型）または難読化（メタモーフィック型）により自己改変するが，それらの処理は1）自分自身のコードをREADしてメモリに展開し，2）コードの自己改変を行う，という手順で行われる．すなわち，原理的には変異型ワームも必ず自己ファイルREADを行う．よって本方式は，変異型ワームを含むすべての未知ワームを確実に検知することが可能であると期待される．

ただし本方式は，ワーム本体のファイルがOSのファイルシステム内に格納されないタイプのワームは検知対象外となる．例えば，PCのメモリ上でのみ動作し，ファイルシステムの中にワーム本体のデータを書き込むことがないCodeRed[2]のようなワームは，自己ファイルREADのイベントが発生しないため検知不可能である．しかし，このようなメモリ常駐型のワームは，コンピュータがリブートされた際に自分自身を再起動させることができないため，基本的には電源が切られることのな

いサーバマシンを狙ったものがほとんどである．このため本方式は，エンドユーザのPCに対しては実用的に使用可能なのではないかと考えられる．

3．検知能力に関する検証

3．1　模擬実装

本方式では，ワームの自己ファイルREADをリアルタイムで捉えるために，OSのファイルシステムを監視する必要がある．これは，Windowsのファイルシステムをフックすることにより，エンドユーザのPCにおけるファイルアクセスを常時モニタリングすることで達成される．具体的には，PC内で実行中の全プロセスのファイルアクセスを監視し，自分自身の本体のファイルの大部分をREADしたプロセスが発見された時点でアラートを上げる．

ファイルシステムのフックを行うにあたっては，フィルタドライバに手を加えることで比較的容易に実装が可能である．また，ファイルシステムのフックは，動的ヒューリスティック法[3]のようにPC上で常駐的に仮想環境を用いるより低負荷であるため，リアルタイム検知が可能であると考えられる．(ただし本稿では，次に述べるように，まだ実際のシステムを構築していないため，オーバヘッドの実測については今後の課題である．)

本稿では本方式のコンセプトの提案に主眼を置いているため，現時点では実際のファイルシステムフィルタドライバについては未実装である．そこで，OSのファイルアクセスをリアルタイムに監視するモニタツールである可能なFileMon[4]を用いて，本方式の可能性を探ることにする．

FileMonによりPC内で発生したすべてのファイルアクセスを記録し，プロセスが自分自身の本体のファイル（の大部分）をREADしていないかどうかをチェックする．すなわち，パス名Aのファイルを実行することによって生起されたプロセスが，パス名Aのファイル（の大部分）をREADした場合に，ワームの疑いありと判断する．

3．2　正検知実験

本方式によって実際に未知ワーム検知が可能であるかの実験を行った．ただし，未知ワームの入手が困難であるため，ここでは代表的な既知ワームのファイルアクセスを見ることで仮想的に未知ワーム検知が可能であるかを確認することとした．表1に本実験で用いたワームの種類を示す．

表1．検査対象ワームおよび検知結果

名前	型	シーケンシャルREAD	ブロックREAD
Sasser. C	脆弱性悪用型	○	×
Blaster. C	脆弱性悪用型	○	×
Beagle. X	メール送信型	○	×
Netsky. B	メール送信型	×	○
Netsky. D	メール送信型	○	×
Netsky. Z	メール送信型	×	○

本実験は，本学のLANから物理的に隔離されたネットワーク上で行った．隔離されたネットワーク上のPCでワームを実行し，FileMonでワームのファイルアクセスの観測を実施する．なお，実験に使用したPCのOSはセキュリティパッチの当たっていないWindows2000である．表1には，実験の結果も併記してある．

ワームのファイルアクセスを観測した結果を解析したところ，2種類の方法で自己ファイルREADを行っていることが確認された．以下，それぞれについて説明する．

・シーケンシャルREAD

Beagle.X，NetSky.D，Blaster.C，Sasser.Cにおいては，自分自身のファイル（ワーム本体）のファイルサイズなどを調べた上で，ファイルの中身をシーケンシャルにコピーするという動作が捉えられた．「Open Sequential Access」オプション付

OPEN	C:¥avserve2. exe	SUCCESS	Options: Open Sequential Access: All
QUERY INFORMATION	C:¥avserve2. exe	SUCCESS	FileAttributeTagInformation
QUERY INFORMATION	C:¥avserve2. exe	SUCCESS	Length: 15872
QUERY INFORMATION	C:¥avserve2. exe	SUCCESS	Attributes: RA
QUERY INFORMATION	C:¥avserve2. exe	SUCCESS	FileStreamInformation
QUERY INFORMATION	C:¥avserve2. exe	SUCCESS	Attributes: RA
QUERY INFORMATION	C:¥avserve2. exe	SUCCESS	FileEaInformation
CREATE	C:¥WINNT¥avserve2. exe	SUCCESS	Options: OverwriteIf Sequential Access: All
SET INFORMATION	C:¥WINNT¥avserve2. exe	SUCCESS	Length: 15872
QUERY INFORMATION	C:¥avserve2. exe	SUCCESS	Length: 15872
WRITE	C:¥WINNT¥avserve2. exe	SUCCESS	Offset: 0 Length: 15872
SET INFORMATION	C:¥WINNT¥avserve2. exe	SUCCESS	FileBasicInformation
CLOSE	C:¥avserve2. exe	SUCCESS	

図１．シーケンシャル READ を行うワーム（Sasser. C）の観測結果の一部

READ	C:¥Informations. Txt(大量のスペース)*. Exe	SUCCESS	Offset: 0 Length: 1024
READ	C:¥Informations. Txt(大量のスペース)*. Exe	SUCCESS	Offset: 1024 Length: 1024
READ	C:¥Informations. Txt(大量のスペース)*. Exe	SUCCESS	Offset: 2048 Length: 1024
READ	C:¥Informations. Txt(大量のスペース)*. Exe	SUCCESS	Offset: 3072 Length: 1024
:	:	:	:
READ	C:¥Informations. Txt(大量のスペース)*. Exe	SUCCESS	Offset: 20480 Length: 1024
READ	C:¥Informations. Txt(大量のスペース)*. Exe	SUCCESS	Offset: 21504 Length: 1024
READ	C:¥Informations. Txt(大量のスペース)*. Exe	END OF FILE	Offset: 22016 Length: 1024

図２．ブロック READ を行うワーム（NetSky.Z）の観測結果の一部

*NetSky. Z のプログラム名に大量のスペースが含まれるが，図では（大量のスペース）と表した．

きのファイル OPEN，ファイル CREATE，ファイル WRITE の処理の中で自分自身がシーケンシャルに READ されていると思われる．観測結果（の一部）を図１に示す．

・ブロック READ

NetSky. B，NetSky. Z においては，自分自身のファイル（ワーム本体）をブロック（例えば NetSky. Z では 1024 バイト）ごとに次々と READ していることがわかった．観測結果（の一部）を図２に示す．

なお，すべてのワームにおいて，自分自身のファイル（ワーム本体）のすべてのデータが READ されていることが確認された．

今回の実験では変異型ワームを入手することができなかったため，現時点においては変異型ワームに対する検知実験を行うことはできていない．ただし，表１のワームの中で，NetSky. Z は ZIP 圧縮した自己複製を作成するワームである．圧縮も

暗号化や難読化と同様に自分自身のコードに対する処理であるということを勘案すると，NetSky.Zと同様に変異型ワームも本方式で検知できることが期待される．

3．3　誤検知実験

本方式によって正規のプロセスがワームとして誤検知されることがないかを調べる実験を行った．表２に本実験で用いた正規プロセスと実験結果を示す．

表２．検査対象プロセスおよび検知結果

名前	誤検知
MS WORD	×
MS EXCEL	×
インストーラ(sinst1-4-7-0.exe[5])	○
Internet Explorer	○

MS WORD，MS EXCEL においては，自己ファイルREADは観測されなかった．

インストーラ（sinst1-4-7-0.exe）においては，自分自身のファイルをREADするイベントが観測された．しかし，一般的なインストーラは，インストールされるプログラム群が格納されているデータ領域とインストールを制御するプログラム領域から成っていることが一般的であるため，基本的には，自分ファイルREADの対象はデータ領域内のデータのみとなる．よって，自己ファイルREADが検知された際に，READ されたデータがファイル全体のどれくらいの割合であるかということをチェックすることにより，ワームとインストーラを切り分けることが可能であると考えられる．

また，Internet Explorerにおいても，自分自身のファイルを READ するイベントが観測された．READ されたデータのサイズはファイル全体の１％にも満たないものであった．Internet Explorerのソースファイルが公開されていないため，推測の域を出ないが，これは，自身のコードの中に書き込まれているバージョン情報などを読み込んでいるのではないかと思われる．いずれにせよ，READされたデータがファイル全体のどれくらいの割合であるかをチェックすることでワームとの切り分けが可能であると考えられる．

ただし，本方式が公知のものとなった場合には，ある程度のサイズのジャンクコードをワーム本体に付加することにより，「ファイル全体の中の一部に本体が隠されている」というワームが作成されるようになるかもしれない．このようなワームは，１）ファイル全体の中からワーム本体の部分のみをREADし，２）新たなジャンクコードを生成した上で，３）１のワーム本体と２のジャンクコードを合わせたものを他のPCに感染させる，という行動をとることにより，インストーラの振る舞いに偽装させることができる．この問題に対する対策は今後の課題としたい．

4．まとめ

自分自身のファイルを再びREADして自己複製を行うというワーム特有の振る舞いに着目し，プロセスのファイルアクセスを監視することにより未知ワームの検知を行う方法を提案した．本方式は変異型ワームの検知に対しても効果的であると考えられる．

OS のファイルアクセスを観測するモニタツールを用いた基礎実験から，本方式の可能性が確認できた．今後は本方式を実装した上で各種のワームに対する実証実験を行うことにより，実際の検知率，誤検知率，リアルタイム検知を行うにあたってのオーバヘッドなどを計測していく．また，自己ファイルREADをワームらしい振る舞いとして追加することにより，ビヘイビアブロック法の性能がどれくらい改善できるか確かめていきたい．

参考文献

[1] 情報処理推進機構, “未知ウイルス検出技術に関する調査”, http://www.ipa.go.jp/security/fy15/reports/uvd/index.Html

[2] アットマーク・アイティ, “Code Red ワームの正体とその対策［改訂版］”, http://www.atmarkit.co.jp/fwin2k/insiderseye/20010803codered/codered.html

[3] Computer Associates /Taras Malivanchuk, "The Win32 worms: classification and possibility of heuristic detection” , http://www.virusbtn.com/conference/vb2002/abstracts/heuristic.xml

[4] FileMon, http://www.sysinternals.com/

[5] サクラエディタ, http://sakura_editor.at.infoseek.co.jp/