Computer Security Symposium 2017
23 - 25 October 2017

レジストリの変化量に着目した未知のマルウェアの検知に関する研究

向野 賢人† 岡村 耕二†

概要：近年サイバー犯罪による被害が深刻化しており，サイバーセキュリティの技術が重要になってきている．従来のアンチウイルスソフトのマルウェア検知手法のほとんどは，マルウェアのデータ構造を記述した定義データとの単純比較である．しかし，この手法では，定義データが対応しない未知のマルウェアの検知が困難である．そこで，解析前の未知のマルウェアによる攻撃を防止するための技術として，振る舞い検知が有効である．本研究では，様々なツールを用いて，マルウェア実行時のファイルやレジストリへのアクセス状況を解析した．その解析結果に基づき，レジストリの変化量に着目した未知のマルウェアの検知手法について検討する．

Research on detection of unknown malware focusing on registry change

Kento Kono† Koji Okamura†

Abstract: In recent years, the damage caused by cybercrime is getting worse, and the technology of cyber security is getting important. Most of conventional malware detection methods of anti-virus software are simple comparisons with definition data that describes the data structure of malware. However, with this method, it is difficult to detect unknown malware that does not correspond to definition data. Therefore, behavior detection is effective as a technique to prevent attacks by unknown malware before analysis. In this research, we analyzed access status of files and registries at the time of execution of malware using various tools. Based on the analysis result, we will examine the detection method of unknown malware focusing on the change amount of registry.

1. はじめに

近年，インターネットの急速な普及に伴いサイバー犯罪による被害が深刻化しており，サイバーセキュリティの技術が重要になってきている．従来の対策は，パターンマッチングタイプのアンチウイルスソフトウェアや，パケットモニタリングなどの人手によるものであった．しかし，近年は毎日のように新種の攻撃が出現してきており，人手による対策には限界がある．このような要因から，新たな対策手法が必要とされている．

本研究では，サイバー犯罪の中でも，マルウェアによるものに着目する．近年，電子メールや Web サイトの閲覧を通じてコンピュータウイルスに感染する被害が増えている．オリジナルを一部改変した多くの亜種ウイルス，自らのプログラムの一部を自ら変更するウイルス等があり，これらについては，従来のワクチンソフトが採用している単純な定義ファイルとの比較を主としたパターンマッチングによる手法では即時の対応・発見が困難である．そのため，定義ファイルが作成されていないウイルスの感染が瞬時に拡大する危険性がある[1].

マルウェアの作成者は，解析や検知を逃れるための新たな方法を常に探している．新種のマルウェアには，OS の機能を悪用し，ディスクにファイルを残すことなく悪意のあるコードをメモリや OS のレジストリに埋め込むものがある．ファイルを利用しない（ファイルレス）マルウェア攻撃では，レジストリにコードを埋め込む方法が現在の主流となっている．ファイルレス攻撃は多くの場合，電子メールメッセージ内のファイルまたはリンクとしてシステムに侵入する．リンクまたは添付ファイルがクリックされると，マルウェアはレジストリにペイロードを書き込み，そのあと消失する．レジストリに書き込まれたペイロードには，何層ものトリックに隠されたスクリプトが含まれている．このスクリプトは正当な Windows プログラム（PowerShell など）を呼び出し，svchost，dllhost，regsvr32 など標準的な Windows プロセスのメモリ領域に悪意のあるコードを埋め込む．このため，悪意のあるプロセスを探してスキャンを実行しても，このコードを検出することはできない．ウイルス定義ファイルの更新だけでは，このようなファイルレス攻撃や，さらに新型のファイルレス攻撃に対する防御として不十分である[2].

レジストリは，Windows で使用される中央階層型データ

† 九州大学
Kyushu University

ベースで，1 人または複数のユーザ，アプリケーションおよびハードウェア装置を構成するのに必要なシステム情報を格納するために使用さる．レジストリには，Windows が実行中に絶えず参照する情報（各ユーザに関するプロファイル，コンピューターにインストールされているアプリケーションおよび各アプリケーションで作成可能なドキュメントの種類，フォルダおよびアプリケーションアイコンについてのプロパティシート設定，システム上に存在するハードウェアの種類，および使用中のポート，など） が格納される[3].

未知のマルウェアによる攻撃を防止するための技術として，振る舞い検知が有効である．本研究では，未知のマルウェアを検知できる振る舞い検知手法の実現を目的とし，様々なツールを用いて，マルウェア実行時のファイルやレジストリへのアクセス状況を解析した．解析結果から，通常マルウェアが実行されると，レジストリに何らかの変化をもたらすことが確認された．レジストリに関する挙動から，マルウェアのみに共通する挙動を捉えることができれば，パターンマッチングによる手法では検知できない未知のマルウェアであっても検知できる可能性がある．

本稿では，マルウェアの解析に使用するツールと解析手順について述べ，解析結果を示す．最後に，レジストリアクセスに基づいたマルウェア検知手法を検討する．

2. 関連研究

マルウェアによる被害が深刻化している中，未知のマルウェアへの対策に関する多くの研究が行われている．本章では，マルウェア検知手法に関する研究のうち，マルウェアを実際に動作させ，その挙動に基づいて検知手法を提案している研究について述べる．

論文[4]では，検査対象の実行ファイルを複数回実行して得られた API ログを比較することで，実行毎の挙動の差異を判断しマルウェアの検知を行う手法を提案している．この手法は，解析や検知への耐性を持たせて作成されたマルウェアのうち，実行毎に挙動に変化を生じさせるマルウェアに着目している．実行毎の挙動の変動を，正規のプログラムには必要がなくマルウェアのみが有する機能と仮定し，ある実行ファイルがマルウェアか否かを判断するための指標として用いている．

論文[5]では，マルウェアの攻撃対象となるシステム上におとりのプロセスを用意し，そのおとりのプロセスを強制終了させようとするプロセスを検知した場合，そのプロセスをマルウェアと判断し停止させることで，マルウェアの活動を抑止するという手法を提案している．この手法は，マルウェアの自己防衛機能の 1 つである，セキュリティ機能を無効化する攻撃に着目している．

3. マルウェアの解析

本章では，まず本研究でマルウェアの解析に使用するツールと解析手順について述べる．最後に解析結果を示す．

3.1 使用するツール

- ToolWiz Time Freeze

ToolWiz Time Freeze は，システムドライブを仮想化して不要な変更や不正な活動からシステムを保護することができるソフトウェアである. システムドライブを仮想化して，それ以降に加えられた変更を，再起動すれば仮想化前の状態に戻すことができる．

- Sandboxie

Sandboxie は，OS 上に仮想領域を作成しプログラムを安全に実行することができる. Sandboxie 上で実行されたプログラムはハードディスクにデータを書き込むことなくプログラムを実行することができる．

- Process Monitor

Process Monitor は，OS 上のすべてのプロセスが行った処理（ファイル，レジストリ，プロセスおよびスレッドの活動）をリアルタイムで表示するツールである．システムのトラブルシューティングやマルウェア検知などに役立てることができる．例えば，アプリケーションがアクセスしているファイルやレジストリキーを特定したり，行われた処理が成功しているか失敗しているかを確認したりすることができる．

- regshot

regshot は，2 つのレジストリを比較して変更を確認することができるツールである．例えば，アプリケーションを追加・削除した場合などシステムに変更を加えたときに，レジストリのどこが変更されたのかを調べるといった用途に利用できる．

3.2 手順

解析には Windows10 の PC を用いる．この解析用 PC には，前述のツールを含め様々な解析ツールをインストールしている．また，組織内に配送された脅威メールに含まれるマルウェアを解析対象とする．

(1) **ToolWiz Time Freeze を起動**

PC を起動後，まず ToolWiz Time Freeze を起動し，それ以降に加えられた変更を, PC を再起動することで元に戻せる状態にする．

(2) **regshot を実行**

regshot を実行してマルウェア実行前のレジストリの状態を保存する．regshot は解析後に再度実行し，マルウェア実行後の状態との差分を取得する．

(3) **Process Monitor を起動**

Process Monitor を起動してプロセスのキャプチャを開始

する．

(4) **マルウェアを実行**

Sandboxie 上で対象のマルウェアを実行する．

(5) **ログを保存**

各ツールで取得したログを保存する．解析終了後，取得したログに基づいて，マルウェア検知手法を検討する．

3.3 解析結果

Process Monitor で取得したログには，OS 上のすべてのプロセスが行った処理が含まれる．本研究では，マルウェアのプロセスのうち，レジストリアクセスに着目した．

表 1 及び表 2 は，通常正当な windows プロセスである regsvr32.exe の一部であるが，マルウェア実行時に悪意のあるコードを埋め込まれている．表中の RegOpenKey は，指定したレジストリキーを開く操作である．

表 1 のプロセスは，Path の最後に Fiddler.exe, Fiddler2.exe, Fiddler2 のいずれかを指定し，そのレジストリキーを開こうとしている．しかし，解析用 PC にそれらのレジストリキーは存在しないため，処理が失敗している．Fiddler とは，HTTP（S）トラフィックに特化した，無料のネットワークキャプチャツールである．

表 2 のプロセスは，Path の最後に HTTPAnalyzerAddon, IEHTTPAnalyzer.HTTPAnalyzerAddOn ， HTTPAnalyzerStd.HTTPAnalyzerStandAlone のいずれかを指定し，そのレジストリキーを開こうとしている．しかし，解析用 PC にそれらのレジストリキーは存在しないため，処理が失敗している．HTTP Analyzer とは，HTTP パケットを調査することができるツールである．

これらのプロセスは，攻撃対象の PC に Fiddler や HTTP Analyzer といった解析ツールがインストールされているかを確認することを目的としたレジストリアクセスであると考えられる．

表 1　regsvr32.exe のプロセス 1

Operetion	Path	Result
RegOpenKey	HKLM¥SOFTWARE¥WOW6432Node¥Microsoft¥Windows¥CurrentVersion¥App Paths¥Fiddler.exe	REPARSE
RegOpenKey	HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥App Paths¥Fiddler.exe	NAME NOT FOUND
RegOpenKey	HKLM¥SOFTWARE¥WOW6432Node¥Microsoft¥Windows¥CurrentVersion¥App Paths¥Fiddler2.exe	REPARSE
RegOpenKey	HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥App Paths¥Fiddler2.exe	NAME NOT FOUND
RegOpenKey	HKLM¥SOFTWARE¥WOW6432Node¥Microsoft¥Windows¥CurrentVersion¥Uninstall¥Fiddler2	NAME NOT FOUND
RegOpenKey	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Uninstall¥Fiddler2	NAME NOT FOUND
RegOpenKey	HKLM¥Software¥WOW6432Node¥Microsoft¥Fiddler2	NAME NOT FOUND
RegOpenKey	HKLM¥SOFTWARE¥WOW6432Node¥Microsoft¥Fiddler2	NAME NOT FOUND
RegOpenKey	HKCU¥Software¥Microsoft¥Fiddler2	NAME NOT FOUND
RegOpenKey	HKCU¥SOFTWARE¥Microsoft¥Fiddler2	NAME NOT FOUND

表 2　regsvr32.exe のプロセス 2

Operetion	Path	Result
RegOpenKey	HKCR¥SOFTWARE¥IEInspectorSoft¥HTTPAnalyzerAddon	NAME NOT FOUND
RegOpenKey	HKCU¥SOFTWARE¥Classes¥SOFTWARE¥IEInspectorSoft¥HTTPAnalyzerAddon	REPARSE
RegOpenKey	HKCU¥Software¥Classes¥SOFTWARE¥IEInspectorSoft¥HTTPAnalyzerAddon	NAME NOT FOUND
RegOpenKey	HKCU¥Software¥Classes¥Software¥IEInspectorSoft¥HTTPAnalyzerAddon	NAME NOT FOUND
RegOpenKey	HKCR¥IEHTTPAnalyzer.HTTPAnalyzerAddOn	NAME NOT FOUND
RegOpenKey	HKCU¥SOFTWARE¥Classes¥IEHTTPAnalyzer.HTTPAnalyzerAddOn	REPARSE
RegOpenKey	HKCU¥Software¥Classes¥IEHTTPAnalyzer.HTTPAnalyzerAddOn	NAME NOT FOUND
RegOpenKey	HKCR¥HTTPAnalyzerStd.HTTPAnalyzerStandAlone	NAME NOT FOUND
RegOpenKey	HKCU¥SOFTWARE¥Classes¥HTTPAnalyzerStd.HTTPAnalyzerStandAlone	REPARSE
RegOpenKey	HKCU¥Software¥Classes¥HTTPAnalyzerStd.HTTPAnalyzerStandAlone	NAME NOT FOUND

表 3 も，通常正当な windows プロセスである regsvr32.exe の一部であるが，マルウェア実行時に悪意のあるコードを埋め込まれている．HKLM¥SOFTWARE¥WOW6432Node¥Microsoft¥Windows¥CurrentVersion¥AppPaths¥Wireshark.exe というレジストリキーを開くことに失敗しているが，HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥AppPaths¥Wireshark.exe を開くことに成功している．これは，解析用 PC に Wireshark というソフトウェアをインストールしており，このレジストリキーが存在するためである．指定したレジストリキーを開くことに成功した後は，そのレジストリキーを指定して，RegSetInfoKey, RegQueryKey, RegCloseKey という操作を行っている．RegSetInfoKey は，指定したレジストリキーに関する情報を書き込む操作である．RegQueryKey は，指定したレジストリキーに関する情報を取得する操作である．RegCloseKey は，指定したレジストリキーを閉じる操作である．

表 3　regsvr32.exe のプロセス 3

Operetion	Path	Result
RegOpenKey	HKLM¥SOFTWARE¥WOW6432Node¥Microsoft¥Windows¥CurrentVersion¥App Paths¥Wireshark.exe	REPARSE
RegOpenKey	HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥App Paths¥Wireshark.exe	SUCCESS
RegSetInfoKey	HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥App Paths¥Wireshark.exe	SUCCESS
RegQueryKey	HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥App Paths¥Wireshark.exe	SUCCESS
RegCloseKey	HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥App Paths¥Wireshark.exe	SUCCESS

表 4，表 5 及び表 6 は，複数のマルウェアで確認されたプロセスである．表中の RegCreateKey は，指定したレジストリキーを作成する操作である．表 4 では，マルウェアが，HKU¥Sandbox_{username}_DefaultBox¥user¥current¥SOFTWARE¥AppDataLow¥Software¥Microsoft¥173C91EB-8A2A-6118-4C3B-5E25409F7229 というレジストリキーを作成しようとしている．解析用 PC に HKU¥Sandbox_{username}_DefaultBox¥user¥current¥SOFTWARE は存在するが，そこに AppDataLow が存在しない．そのためマルウェアは，目的のレジストリキーから Path を一つずつさかのぼって作成を試み，作成が成功すれば一つずつ Path を追加することで目的のレジストリキーを作成している．表 5 及び表 6 のように同様の挙動が確認された．

表 4　マルウェアのプロセス 1

Operetion	Path	Result
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥user¥current¥SOFTWARE¥AppDataLow¥Software¥Microsoft¥173C91EB-8A2A-6118-4C3B-5E25409F7229	NAME NOT FOUND
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥user¥current¥SOFTWARE¥AppDataLow¥Software¥Microsoft	NAME NOT FOUND
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥user¥current¥SOFTWARE¥AppDataLow¥Software	NAME NOT FOUND
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥user¥current¥SOFTWARE¥AppDataLow	SUCCESS
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥user¥current¥SOFTWARE¥AppDataLow¥Software	SUCCESS
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥user¥current¥SOFTWARE¥AppDataLow¥Software¥Microsoft	SUCCESS
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥user¥current¥SOFTWARE¥AppDataLow¥Software¥Microsoft¥173C91EB-8A2A-6118-4C3B-5E25409F7229	SUCCESS

表 5　マルウェアのプロセス 2

Operetion	Path	Result
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥MACHINE¥SOFTWARE¥WOW6432Node¥Microsoft¥Windows¥CurrentVersion¥Explorer¥SyncRootManager	NAME NOT FOUND
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥MACHINE¥SOFTWARE¥WOW6432Node¥Microsoft¥Windows¥CurrentVersion¥Explorer	NAME NOT FOUND
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥MACHINE¥SOFTWARE¥WOW6432Node¥Microsoft¥Windows¥CurrentVersion	NAME NOT FOUND
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥MACHINE¥SOFTWARE¥WOW6432Node¥Microsoft¥Windows	NAME NOT FOUND
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥MACHINE¥SOFTWARE¥WOW6432Node¥Microsoft	NAME NOT FOUND
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥MACHINE¥SOFTWARE¥WOW6432Node	SUCCESS
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥MACHINE¥SOFTWARE¥WOW6432Node¥Microsoft	SUCCESS
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥MACHINE¥SOFTWARE¥WOW6432Node¥Microsoft¥Windows	SUCCESS
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥MACHINE¥SOFTWARE¥WOW6432Node¥Microsoft¥Windows¥CurrentVersion	SUCCESS
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥MACHINE¥SOFTWARE¥WOW6432Node¥Microsoft¥Windows¥CurrentVersion¥Explorer	SUCCESS
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥MACHINE¥SOFTWARE¥WOW6432Node¥Microsoft¥Windows¥CurrentVersion¥Explorer¥SyncRootManager	SUCCESS

表 6　マルウェアのプロセス 3

Operetion	Path	Result
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥user¥current¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Internet Settings¥ZoneMap	NAME NOT FOUND

RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥user¥current¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Internet Settings	SUCCESS
RegCreateKey	HKU¥Sandbox_{username}_DefaultBox¥user¥current¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Internet Settings¥ZoneMap	SUCCESS

4. 提案手法

本章では，マルウェアの解析結果に基づき，レジストリの変化量に着目した未知のマルウェアの検知手法を検討する．レジストリに関する挙動から，マルウェアのみに共通する挙動を捉えることで，パターンマッチングによる手法では検知できない未知のマルウェアの検知が期待できる．

本研究では，マルウェア実行時のレジストリへのアクセス状況に着目した．解析で確認された，攻撃対象の PC に解析ツールがインストールされているかを確認することを目的とした挙動や，目的のレジストリキーを作成するために Path を一つずつさかのぼって作成を試み失敗を繰り返す挙動を，正規のプログラムには必要がなくマルウェアのみが有するプロセスであると判断し検知する手法を提案する．

また，マルウェアによるレジストリアクセスには，その処理が失敗しているものが多く確認された．指定したレジストリキーを開く操作や指定したレジストリキーを作成する操作といったレジストリアクセスの種類や，特定の Path でプロセスにフィルタをかけ，処理の失敗率を定義することでマルウェア検知の指標にできると考える．

提案手法の有効性を評価する指標として，従来のパターンマッチングタイプのアンチウイルスソフトウェアでは検知できないマルウェアに適用した際の検知率を用いる．この検知率が一定以上であれば，未知のマルウェアの検知手法としての提案手法の有効性が示される．また，正規のプログラムに適用した際にマルウェアとして誤検知する確率を抑えられているかも有効性を評価する指標として用いる．

5. まとめと今後の課題

本研究では，未知のマルウェアを検知できる振る舞い検知手法の実現を目的とし，様々なツールを用いて，マルウェア実行時のファイルやレジストリへのアクセス状況を解析した．取得したマルウェア実行時のプロセスのうち，レジストリアクセスに着目し，解析結果を示した．解析で確認されたマルウェアによるレジストリアクセスを，正規のプラグラムには必要がなくマルウェアのみが有するプロセスであると仮定しマルウェアを検知する手法を提案した．

提案手法の有効性を示すことが今後の課題である．提案手法の評価実験を行う前に，誤検知を防ぐために，正規のプログラムによるレジストリアクセスを把握する必要がある．また，本稿で示したマルウェアによるレジストリアクセスのほかに新たな挙動を捉えられれば，検知の精度を向上させることができる可能性がある．

レジストリアクセスの種類や特定の Path でプロセスにフィルタをかけ，処理の失敗率からマルウェアを検知する手法では，フィルタのかけ方や処理の失敗率の定義の仕方が今後の検討課題である．

参考文献

[1] “未知ウイルス検出技術に関する調査“. https://www.ipa.go.jp/security/fy15/reports/uvd/documents/uvd_report.pdf , (参照 2017-08-09).

[2] “高度化するファイルレス攻撃に対処せよ”. http://blogs.mcafee.jp/mcafeeblog/2016/02/post-e8b5.html , (参照 2017-08-09).

[3] “上級ユーザー向けの Windows レジストリ情報”. https://support.microsoft.com/ja-jp/help/256986/windows-registry-information-for-advanced-users , (参照 2017-08-08).

[4] 笠間貴弘, 吉岡克成, 井上大介, 松本勉. 実行毎の挙動の差異に基づくマルウェア検知手法の提案. Computer Security Symposium 2011, 2011.

[5] 松木隆宏, 新井悠, 寺田真敏, 土居範久. セキュリティ無効化攻撃を利用したマルウェアの検知と活動抑止手法の提案. 情報処理学会論文誌. 2009, vol. 50, no. 9, p. 2127-2136.