

ダミーファイルを用いた暗号化型ランサムウェアの検出と防御に関する検討

荻原 拓海^{1,a)} 小林 良太郎¹ 加藤 雅彦²

概要: 近年ランサムウェアは大きな脅威となっており、被害に遭うケースが後を絶たないがその侵入経路は様々である。例えば、暗号化型ランサムウェアの Wannacry は 2017 年頃に世界中で大きな被害を及ぼした。当時、Wannacry の感染経路となった脆弱性は事前に修正されていたが、パッチを適用していないシステムが被害を被ったことが判明している。したがって本研究では、暗号化型ランサムウェアの動作に着目し、パッチを適用していなくても対応可能な予防策を提案する。暗号化型ランサムウェアが暗号化するパスにダミーファイルを設置し暗号化検知を行い、重要なファイルが暗号化されるまでの時間を確保するシステムの実装を行う。また本システムの有効性を確認するため Wannacry と Cerber を仮想環境上でそれぞれ実行し耐久時間を計測する。

キーワード: ランサムウェア, 暗号化, ダミーファイル, 検知, 防御

A study on detection and defense of cryptographic ransomware using dummy files

OGIWARA TAKUMI^{1,a)} KOBAYASHI RYOTARO¹ KATO MASAHICO²

Abstract: In recent years, ransomware has become a major threat. There's no shortage of cases of victimization. In addition, there are many route of attack. For example, the cryptographic ransomware, known as Wannacry, did significant damage around the world in 2017. In those days, the vulnerability that uses for intrusion had been fixed in advance. However, the computers using unpatched systems were affected. Therefore, I would suggest a precautionary measure that can be used after infection, focusing on encryption operation. In this study, we implement a system that detects encryption by placing dummy files in the path where the cryptographic ransomware encrypts. Thereby, it ensures the time it takes for important files to be encrypted. After that, we evaluate the effectiveness of the system by Wannacry and Cerber in a virtual environment and measured the endurance time.

Keywords: Ransomware, Encryption, Dummyfile, Detection, Defense

1. はじめに

近年ランサムウェアは大きな脅威となっている。IPA が発表している情報セキュリティ 10 大脅威 [1] では、ランサムウェアは 2016 年からランキングされており、2020 年

現在でも 5 位となっている。ランサムウェアの脅威はいまだ継続しており、対策が急務である。ランサムウェアが感染したのち、その動作は大別して次の 2 パターンに分かれる。1 つ目は端末内のファイルを暗号化する「暗号化型ランサムウェア」、2 つ目はユーザーが端末を操作できないように画面をロックする「画面ロック型ランサムウェア」であり [2]、近年は前者が主流となっている [3]。

暗号化型ランサムウェアである Wannacry は 2017 年に世界中で猛威を振るった。Wannacry はファイルを暗号化す

¹ 工学院大学 Kogakuin University 1-24-2, Nishi-shinjuku, Shinjuku-ku, Tokyo, Japan

² 長崎県立大学 University of Nagasaki 123, Kawashimo-chou, Sasebo-shi, Nagasaki, Japan

^{a)} j117058@ns.kogakuin.ac.jp

る機能に加え自己増殖するワーム機能を持ち、ネットワークを介して感染させることで広範囲に被害を及ぼす。このマルウェアは Windows の共有プロトコルである SMBv1 の脆弱性を利用しユーザーが気付かないまま感染を広げる [4]。2017 年の時点で Wannacry が利用する SMB の脆弱性は修正パッチが存在したものの、パッチを適用していないコンピュータが数多く存在したために、感染が広がった。また、日本では 2019 年から Emotet による被害が拡大している。Emotet はメールに添付されたファイルや URL から感染し、パスワード等の認証情報を窃取するマルウェアであるが、同時にランサムウェアを感染させる場合があることが報告されている。Emotet もパッチが存在するが、未適用の場合は攻撃者の踏み台となるだけではなく、ランサムウェア感染により重要なデータが暗号化され使用できなくなる可能性がある。様々な理由からパッチを適用していないコンピュータは数多くあり、修正済みの脆弱性を悪用した攻撃でもそれらが大きな被害を与えるケースがある。

これらの対策として、暗号化を解除するソフトウェアがリリースされているが、一部のランサムウェアに対応するにとどまる。頻繁にバックアップを取るといった対策も推奨されているが、定期的に自動バックアップしているような場合、最終バックアップ後に作成したデータの復元は保証されない。様々な理由でパッチが適用できず、バックアップも頻繁に取れないような環境で重要なデータが保存されているような場合、ランサムウェアに対して有効な対策を取ることは困難である。

そこで本研究では暗号化型ランサムウェアの動作に着目し、ダミーファイルを用いてランサムウェアの検知とファイルの保護を行う手法を提案する。

第 2 章では提案手法の説明を行う。第 3 章では実装について説明し、第 4 章で評価を行う。第 5 章では評価の考察を行う。第 6 章では今後の課題について示し、第 7 章で本論文をまとめる。

2. 提案手法

ランサムウェアは、特定のフォルダ下において、特定の拡張子を持つファイルを暗号化する。提案手法はこの特徴を利用導入する。ダミーファイルはシステムからもユーザからも利用されることがないファイルで、かつ、ランサムウェアによる暗号化の対象となるファイルである。提案手法ではランサムウェアによる暗号化対策として、次の 2 点を行う。

- (1) 本来変更されるはずがないダミーファイルの変更を監視することで、ランサムウェアの暗号化動作を検知する。
- (2) ランサムウェアが、重要なファイルよりも先にダミーファイルを暗号化することにより、重要なファイルを暗号化する動作を妨害する。

先行研究 [6] より、ランサムウェアが暗号化するファ

表 1 ファイルサイズ

Table 1 Size and number of dummyfiles

サイズ	ファイル数
1KB	4
9MB	26
10MB	20
65MB	45
260MB	5

表 2 ライブラリ一覧

Table 2 List of Libraries

ライブラリ	主な用途
time	sleep() による周期の管理
shutil	copy() によるファイルの上書き
os	ファイルリストの取得
subprocess	プロセスの呼び出し
sys	引数の取得

イルサイズには、下限あるいは上限が存在するものがある。そこで、本研究では表 1 のようにサイズの異なる複数のダミーファイルを用意する。ダミーファイルの拡張子を含むファイル名が変更されることを監視することによって、ランサムウェアの暗号化動作を検知する。また、同時にランサムウェアが暗号化するファイルサイズの下限あるいは上限を推測する。初めに 100 個のダミーファイルを置くことで、システムが暗号化の検知を行い妨害を始めるまでの時間を確保する意味も持つ。

ランサムウェアの動作を妨害するためには多数のダミーファイルを用意する必要がある。しかし、検知時に推測したサイズのダミーファイルを多数生成するとディスク容量を圧迫してしまう。そこで、最初はサイズが 0 byte のダミーファイル (空ファイルと呼ぶ) を大量に生成し、ランサムウェアの検知後に随時空ファイルをランサムウェアが暗号化対象と推測したサイズのダミーファイルに変換する。これによりディスク容量を圧迫せず大量のダミーファイルを用意することができる。

3. 実装

プログラムは Python3.7.7 で記述し IDE は Spyder4.1.4 を使用した。このシステムを作成する上で使用したライブラリ用途を表 2 に示す。

3.1 システム構成

このシステムは以下の 3 つのプログラムによって構成されている。これに関して動作の説明を行う。

- (1) 設定プログラム
- (2) 監視プログラム
- (3) 妨害プログラム

設定プログラムはシステムの開始時に実行され監視プロ

グラムや妨害プログラムが動作するために必要な各種設定を行う。監視プログラムはシステムの開始後、常に動作し続け、ダミーファイルを用いてランサムウェアの検知を行う。妨害プログラムはランサムウェアの検知後に実行され、ダミーファイルを用いてランサムウェアによる重要なファイルの暗号化を妨害する。

3.1.1 設定プログラム

このプログラムは、監視フォルダの生成、ダミーファイルの生成、監視プログラムの監視周期の変更、妨害プログラムの妨害周期の変更、設定ファイルの作成を行うプログラムである。監視フォルダとは、ダミーファイルを生成するフォルダのことである。本研究では、重要なデータを保持していると思われる表5を監視フォルダに設定した。なお、フォルダ名を「!」とした理由は、ランサムウェアがUnicode順にフォルダ、ファイルのリストを作成しているという特性[6]を利用し、重要なユーザのファイルより先にダミーファイルが暗号化されるようにするためである。ダミーファイルは先ほど述べた容量の異なるダミーファイルと空ファイルを監視フォルダ内に生成する。ダミーファイルの名前は「生成順 + dummyfile.txt」とした。したがって、ファイル名は、0dummyfile.txt, 1dummyfile.txt, 2dummyfile.txt, ... となる。監視周期とは、監視プログラムがフォルダを監視する周期であり、監視時間とディレイの和となる。本システムではディレイを設定することで監視周期を間接的に設定する。デフォルトでは、1/監視フォルダ数(sec)となっている。妨害周期とは、妨害プログラムが1つの空ファイルのある特定のサイズのダミーファイルに置き換える周期であり、置き換え時間とディレイの和になる。デフォルトでは、0.1(sec)となっている。

3.1.2 監視プログラム

図1に監視プログラムの処理の流れを示す。このプログラムは設定プログラムで設定された情報を読み取り監視フォルダごとにダミーファイルを参照し、ファイル名と拡張子に変更がないか確認して行く。ランサムウェアにはファイル名と拡張子を変更するものや、拡張子のみを変更するものがあるため、両方確認する必要がある。変更があった場合、ランサムウェアの暗号化動作が検知できたとしてユーザーに通知を行うとともに、変更があったフォルダに対する暗号化を妨害するために、妨害プログラムを呼び出す。なお、監視プログラムは妨害プログラムを呼び出した後もフォルダの監視を続ける。ただし、妨害プログラムが暗号化を妨害しているフォルダは、監視対象となるフォルダから除外される。

3.1.3 妨害プログラム

図2に妨害プログラムの処理の流れを示す。このプログラムは監視プログラムから呼び出され、ファイル名が変更されつつある監視フォルダ内において、ランサムウェアがダミーファイルを暗号化する時間をできるだけ長くする。

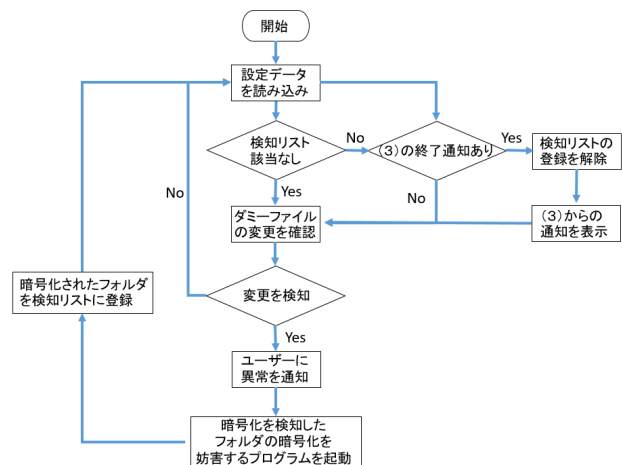


図1 監視プログラムの処理

Fig. 1 Monitoring program processing

ダミーファイルが暗号化される時間は、監視フォルダ内において空ではないダミーファイルを暗号化する時間で決まる。しかし、ディスク容量の圧迫を防ぐため、監視フォルダ内には、少数のダミーファイルと多数の空のダミーファイルが用意されており、このままでは暗号化が短時間で終了してしまう。そこで、暗号化されたダミーファイルを1つ削除し、その直後に n byte のダミーファイルで空のダミーファイルを1つ上書きする操作を繰り返す。ここで、 n の初期値は65Megaとし、暗号化が行われていなければ n を変化させていく。これにより、ディスク容量を圧迫することなく、動的に空のダミーファイルを暗号化対象となる n byte のダミーファイルに変換していく。

なお、 n byte のダミーファイルの内容は、事前にランダムに生成したデータである。したがって、上書き後のダミーファイルの内容は、容量が同じであれば全て同一となる。 n を変化させる方法は以下のとおりである。ファイルの置き換えをする際に順番通りに暗号化されていない状態が続く場合、暗号化されていないファイルのサイズを参照することによって、ランサムウェアのファイルサイズによる制限を確認し、その制限内で n を調整する。これにより、動的にランサムウェアの暗号化の対象となるダミーファイルを生成する。

4. 評価

本研究では、提案手法によってランサムウェアの暗号化を妨害した時間（以下耐久時間と呼ぶ）を評価する。なお、ランサムウェアの検知に成功したかどうかは、監視プログラムがユーザーに異常を通知したかどうかで確認できる（図1）。

4.1 評価環境

検証環境は表4のとおりである。仮想マシンとして Virtu-

表 5 監視フォルダ

Table 5 Folders to monitor

ラベル	パス
0	C:\Users\%USERNAME%\Desktop\!\
1	C:\Users\%USERNAME%\Downloads\!\
2	C:\Users\%USERNAME%\Favorites\!\
3	C:\Users\%USERNAME%\OneDrive\!\
4	C:\Users\%USERNAME%\Pictures\!\
5	C:\Users\%USERNAME%\Documents\!\
6	C:\Users\%USERNAME%\Videos\!\
7	C:\Users\%USERNAME%\Public\!\
8	C:\Windows\!\
9	C:\Users\!\
10	C:\Users\Program Files\!\
11	C:\Users\Program Files (x86)\!\
12	C:\!\

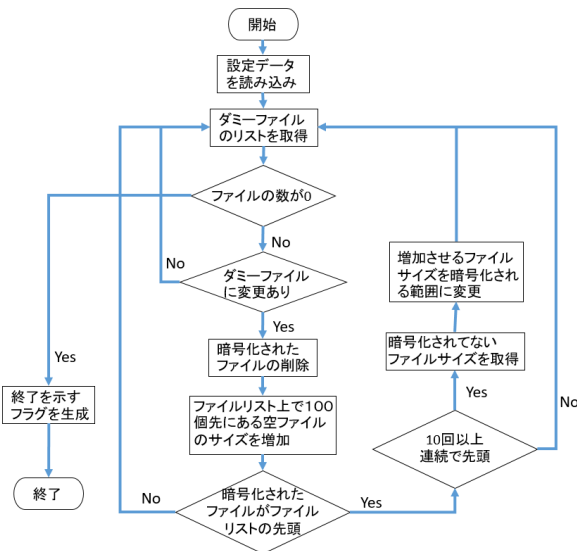


図 2 妨害プログラムの処理

Fig. 2 Disturbance program processing

表 3 MD5 のハッシュ値

Table 3 MD5 hash values

種類	ハッシュ値
Wannacry	84c82835a5d21bbcf75a61706d8ab549
Cerber	8b6bc16fd137c09a08b02bbe1bb7d670

表 4 検証環境

Table 4 Evaluation environment

	ホスト	ゲスト
OS	Ubuntu 20.04LTS	Windows10 home
CPU	Intel® Core™ i5-9400	仮想プロセッサ 8
RAM	16GB	8GB
Storage	256GB	100GB

alBox6.1.10 を使用した. ランサムウェアとして Wannacry と Cerber を用いた. これらの検体の MD5 ハッシュ値は表 3 のとおりである. ダミーファイルは表 5 の各監視フォルダに 15,000 個ずつ生成した. なお, 15,000 個のダミーファイルにおいて, 100 個を空でないファイルとし, 残りをすべて空のファイルとした. これらのファイル数に関しては他の組み合わせでも検証を行ったが, 本章では Wannacry において暗号化を妨害できた組み合わせの 1 つを示す. 本システムがランサムウェアを検知して通知を行ったときに計測を開始し, 各フォルダに設置したダミーファイルの暗号化が終了するまでの時間を耐久時間として計測する.

4.2 検証結果

Wannacry の検証を行った結果, 暗号化の検知に成功し, 耐久時間は 14 時間 56 分となった. 表 6 に Wannacry が暗号化した監視フォルダを示す. この表においてラベルは表 5 の監視フォルダのパスに着けられたラベルであり, 順番は Wannacry が監視フォルダを暗号化した順である. ラベ

表 6 Wannacry が暗号化したフォルダと順番

Table 6 Wannacry encrypted folders and order

順番	ラベル	パス
1	0	C:\Users\%USERNAME%\Desktop\!\
2	0	C:\Users\%USERNAME%\Desktop\!\
3	5	C:\Users\%USERNAME%\Documents\!\
4	12	C:\!\
5	5	C:\Users\%USERNAME%\Documents\!\
6	1	C:\Users\%USERNAME%\Downloads\!\
7	2	C:\Users\%USERNAME%\Favorites\!\
8	3	C:\Users\%USERNAME%\OneDrive\!\
9	4	C:\Users\%USERNAME%\Pictures\!\
10	7	C:\Users\%USERNAME%\Public\!\
11	6	C:\Users\%USERNAME%\Videos\!\
12	12	C:\!\
13	1	C:\Users\%USERNAME%\Downloads\!\
14	2	C:\Users\%USERNAME%\Favorites\!\
15	3	C:\Users\%USERNAME%\OneDrive\!\
16	4	C:\Users\%USERNAME%\Pictures\!\
17	7	C:\Users\%USERNAME%\Public\!\
18	6	C:\Users\%USERNAME%\Videos\!\

ルは表において監視フォルダを識別しやすくするために示している. また, Wannacry は 1 つの監視フォルダに対し 1 回目は Unicode 順に 2KB 以上, 2 回目は Unicode 順に 0KB 以上のダミーファイルを暗号化するため, 表では監視フォルダごとに 2 回ずつ暗号化が行われていることが確認できる.

Cerber の検証を行った結果, 暗号化の検知に成功し, 耐久時間は 7 分となった. 耐久時間は Wannacry に比べ非常に短い. 表 7 に Cerber が暗号化した監視フォルダを示す. 表の形式は表 6 と同じである. Wannacry とは異なり, Cerber では監視フォルダの一部のみが暗号化の対象となり, また暗号化されたとしても, 監視フォルダ内の一部のダミー

表 7 Cerber が暗号化したフォルダと順番
Table 7 Cerber encrypted folders and order

順番	ラベル	パス
1	12	C:\!\
2	5	C:\Users\%USERNAME%\Documents\!\
3	0	C:\Users\%USERNAME%\Desktop\!\

ファイルしか暗号化されなかった。

5. 考察

本研究では仮想環境で Wannacry と Cerber を動かし、有効性の検証を行った。

Wannacry において、はじめに 98 ファイル暗号化した後一旦暗号化を中断し再度すべてのファイルを暗号化する動作を確認した。このとき、1 回目の暗号化では、1KB のファイルは暗号化されなかった。この結果から初めに生成している 1KB のファイルを除く容量のあるダミーファイルの数 96 個と近いことから、1 回目に暗号化されたと記録されている 98 のファイルについて調べた。すると Wannacry が暗号化を始めたフォルダに生成される 2 つのファイルに関してダミーファイルが暗号化されたものであるとカウントしていることが分かった。このことから、初めに暗号化するサイズを絞って暗号化を行い、次に再度すべてのサイズをターゲットに暗号化を行っているのではないかと考えた。

そこで今回の検証では空のファイルとしていたところを 2KB に置き換えて再度検証を行った。また 4 章では Wannacry に管理者権限を与えていない状態で検証したが、今回は管理者権限を与えて暗号化の対象が変化するか、合わせて確認する。さらに、再検証では耐久時間を確認することが目的ではないためダミーファイルの数を 300 として検証する。その結果、1 回目の暗号化でフォルダの中にある 1KB のダミーファイルを除くダミーファイルが暗号化されていることを確認した。この時 Wannacry が暗号化暗号化したフォルダ順を表 8 に示す。

また、この 2 つの動作の違いから Wannacry は、暗号化する段階でファイルのサイズを確認しているのではなく、暗号化対象のファイルリストを取得した段階でサイズも取得していると推測される。

Cerber において、暗号化された監視フォルダは Wannacry に比べ少なかった。さらに、フォルダの暗号化の途中でそのフォルダに対しての暗号化を終了し、次のフォルダへの暗号化を開始する動作を確認した。その原因に関してファイルがロックされている場合の例外処理として次のフォルダに進むよう設計されているのではないかと考え、Cerber が最初に暗号化するファイルをロックして再度検証を行った。Wannacry と同様に、4 章では管理者権限を与えていなかった。そこで今回は管理者権限を与え暗号化するフォルダに変化があるか同時に検証した。

表 8 Wannacry が暗号化したフォルダと順番 (修正後)
Table 8 Wannacry encrypted folders and order (Fixed)

順番	ラベル	パス
1	0	C:\Users\%USERNAME%\Desktop\!\
2	5	C:\Users\%USERNAME%\Documents\!\
3	12	C:\!\
4	9	C:\Users\!\
5	1	C:\Users\%USERNAME%\Downloads\!\
6	2	C:\Users\%USERNAME%\Favorites\!\
7	3	C:\Users\%USERNAME%\OneDrive\!\
8	4	C:\Users\%USERNAME%\Pictures\!\
9	7	C:\Users\%USERNAME%\Public\!\
10	6	C:\Users\%USERNAME%\Videos\!\
11	12	C:\!\
12	9	C:\Users\!\
13	1	C:\Users\%USERNAME%\Downloads\!\
14	5	C:\Users\%USERNAME%\Documents\!\
15	2	C:\Users\%USERNAME%\Favorites\!\
16	3	C:\Users\%USERNAME%\OneDrive\!\
17	4	C:\Users\%USERNAME%\Pictures\!\
18	7	C:\Users\%USERNAME%\Public\!\
19	6	C:\Users\%USERNAME%\Videos\!\

その結果ロックしたファイルは暗号化されず、そのフォルダ内の暗号化が継続した。したがって、ファイルのロックが原因ではないことが分かる。また管理者権限を付与したことによる暗号化対象の変化は確認されなかった。ここで Wannacry と時と同様に、Cerber が生成したファイルをダミーファイルが暗号化されたものとしてカウント可能性があるため、調査を行った。すると Cerber が生成したファイルを暗号化されたものと認識して削除しており、Cerber も削除されるたびに新しいファイルを生成していた。これにより、実際より多くのファイルが暗号化されているものとしてカウントされていたことが分かった。実際に暗号化されていたのは 1KB 以下のファイルを除く 96 個のダミーファイルだけであった。

上記の結果を踏まえ、Cerber の耐久時間を再計測した。ダミーファイルの数は 5,000 個とし、内訳は表 1 に示したものと 4,900 個の 2KB のダミーファイルとする。また監視フォルダは表 1 に示すとおりである。

この時、耐久時間は 3 時間 25 分であった。Cerber が暗号化したフォルダを暗号化された順に並べて表 9 に示す。

今回の検証ではファイル数を 5,000 個としたが、さらに増やすとより長い間暗号化を妨害することができると考えられる。提案手法の時とは異なり空のダミーファイルを用いているのではなく 2KB のダミーファイルを用いているため、ファイルを増やすごとにディスク容量を圧迫すると考えられるが、その量は表 1 で示したダミーファイルの容量と比較すると十分無視できるものである。

表 9 監視フォルダ (修正後)
Table 9 Folders to monitor (Fixed)

ラベル	パス
0	C:\Users\%USERNAME%\Desktop\!\
1	C:\Users\%USERNAME%\Desktop\!!\
2	C:\Users\%USERNAME%\Desktop\!!!\
3	C:\Users\%USERNAME%\Documents\!\
4	C:\Users\%USERNAME%\Documents\!!\
5	C:\Users\%USERNAME%\Documents\!!!\
6	C:\!\
7	C:\!!\
8	C:\!!!\

表 10 Cerber が暗号化したフォルダと順番 (修正後)
Table 10 Cerber encrypted folders and order (Fixed)

順番	ラベル	パス
1	6	C:\!\
2	7	C:\!!\
3	8	C:\!!!\
4	3	C:\Users\%USERNAME%\Documents\!\
5	4	C:\Users\%USERNAME%\Documents\!!\
6	5	C:\Users\%USERNAME%\Documents\!!!\
7	0	C:\Users\%USERNAME%\Desktop\!\
8	1	C:\Users\%USERNAME%\Desktop\!!\
9	2	C:\Users\%USERNAME%\Desktop\!!!\

6. 今後の課題

本研究では暗号化型ランサムウェアの検知と妨害を行った。しかし、今回の検証で使用した Cerber は全てのファイルを暗号化することはせずに広い範囲のファイルを暗号化して動作を終了した。再度条件を見直して計測を行った際には暗号化を長い時間妨害できたが、ランサムウェアの暗号化対象によっては十分な効果を発揮しない可能性がある。そこで、検知後妨害を行っている間に別の対策も取る必要性を認識した。そこで以下の2つの手法を考えた。

- (1) プロセスが開いているファイルを確認し、そのファイルがダミーファイルである場合にそのプロセスはランサムウェアであると分析し、動作を停止させる。
- (2) 妨害を行う過程で暗号化の速度を取得し、それに似た速度でファイルの読み書きを行っているプロセスをランサムウェアであると分析し、動作を停止させる。

実際に上記の手法を用いてシステムの試作を行いテストを行ったが前者の手法は python のライブラリである psutil の open_files() を使用したが開いているファイルの特定は出来なかった。

そこで、後者の手法を用いて読み書きの増加量を見て分析すると、高い確率でランサムウェアを特定することができた。プロセスの特定ができれば停止等対処が可能になり、

長い時間暗号化の引き延ばしを行う必要がなくなるが、この手法はプロセスの特定、対処を行うまでは暗号化の妨害をしておく必要がある。さらに動作を停止させる場合には False Positive の可能性もあるため注意が必要になる。

7. まとめ

本稿では、近年脅威となっている暗号化型ランサムウェアに対して、ダミーファイルを設置することで暗号化の検知と妨害を行う手法の提案を行った。またシステムの実装を行い仮想環境で Wannacry と Cerber をそれぞれ動作させることで、システムの有効性を検証した。その結果をもとに Wannacry, Cerber の動きを推測し条件を変更して再検証を行うことで推測の真偽を確かめた。その過程で、Cerber に関しては大幅に耐久時間を延ばすことに成功した。さらに今後の課題として別の手法を組み合わせる暗号化型ランサムウェアの動作を停止させる必要性に関して検討を行った。ランサムウェア動作は種類によっても検体によっても異なり、汎用性を向上させるためには様々な手法を組み合わせる必要がある。

謝辞 本研究の一部は、JSPS 科研費 20K11818, 19K11968, 19H04108 の支援により行った。

参考文献

- [1] IPA, 情報セキュリティ 10 大脅威 2020, available from <https://www.ipa.go.jp/files/000080871.pdf> (参照 2020-08-12).
- [2] IPA, ランサムウェアの脅威と対策～ランサムウェアによる被害を低減するために～, available from <https://www.ipa.go.jp/files/000057314.pdf> (参照 2020-08-12).
- [3] kaspersky, ランサムウェア (身代金要求型ウイルス) の多様な手口と対策方法, available from <https://www.kaspersky.co.jp/resource-center/threats/ransomware-examples> (参照 2020-08-18).
- [4] IPA, 情報セキュリティ白書, available from <https://www.ipa.go.jp/files/000070313.pdf> (参照 2020-08-12).
- [5] JPCERT/CC, マルウェア Emotet の感染に関する注意喚起, available from <https://www.jpcert.or.jp/at/2019/at190044.html> (参照 2020-08-12).
- [6] 田中 智也, 小池 一樹, 小林 良太郎, 加藤 雅彦, “ダミーファイルを利用した暗号化型ランサムウェア対策システムの実装,” コンピュータセキュリティシンポジウム 2019 論文集, pp. 163-169, 2019.