Vol.2022-DPS-190 No.6
Vol.2022-CSEC-96 No.6
2022/3/10

悪性コードの分割によりマルウェア検知を回避するランサムウェアシステム

志賀 辰輝[1,a)]　宇田 隆哉[1,b)]

概要：近年，ランサムウェアによる被害が急増している．コンピュータ上のデータを暗号化などによって使用不能にし，復元のために仮想通貨を要求するこの手法は，金銭を集めるのに有効であり，多くの犯罪組織がランサムウェアを利用している．セキュリティソフトがランサムウェアなどのマルウェアを検知する際，マルウェア検体のハッシュ値を取ったパターンマッチングや，悪意のあるコードの特徴が検体内にある調べる静的解析手法がある．しかし，これらの手法は一つのファイルに対してコードを読み取るものであり，コードが参照しているファイルを読んで悪性かどうかを判断することができない．そこで我々は，悪性コードを複数の DLL ファイルに分割し，一つ一つの DLL ファイルが良性プログラムでもあり得る挙動を取る場合，マルウェアと断定できないのではないかと仮説を立て，ランサムウェアを作成した．その結果，マルウェア検出エンジンによってはマルウェアと判定されず，ファイルの破壊，復元が可能であった．

Detection Evasion Ransomware System by Dividing Malicious Codes

1. はじめに

近年，ランサムウェアによる被害が急増している．IPA が毎年公開している情報セキュリティ 10 大脅威 [1] では，2017 年から 2021 年までででランサムウェアの脅威が 10 位以内に選ばれている．ランサムウェアなどのマルウェアへの対策として，現在主流のウイルス対策ソフトでは，パターンマッチングや静的解析によってマルウェアを識別する．しかし，パターンマッチングでは未知のマルウェアを検出できず，静的解析では一つのファイルでしか判定できないため，独自に作成した DLL ファイルなどを参照する場合，参照先の処理を見ることはできない．そのため，悪性コードを複数の DLL ファイルに分割し，それぞれが悪性と断定できないコードである場合，実行ファイルや各 DLL ファイルを検証してもマルウェアと認識できないと考えられる．本論文では，これが静的解析における脆弱性となる可能性について言及する．

[1] 東京工科大学
東京都八王子市片倉町 1404-1, 192–0982
[a)] c01181356a@edu.teu.ac.jp
[b)] uda@stf.teu.ac.jp

2. 関連研究

2.1 ライブラリの検知

現在，ウイルス対策ソフトウェアにより，既知のランサムウェアは検知し対処できるが，プログラムコードを書き換えた亜種マルウェアなど，パターンマッチングに検知されないランサムウェアが存在する．このようなランサムウェアに対処するため，ランサムウェアの実行後に行われる暗号化やファイルの上書き・削除などを検知して対処する手法が研究されている．

重田らはランサムウェアでよく使われるライブラリや暗号化アルゴリズムを検知して対処する手法を提案している [2]．これは，2016 年時点で多くのランサムウェアがライブラリに CryptoAPI か OpenSSL を，暗号化アルゴリズムに AES を用いているものが多いことから，これらのプロセスが確認されたらプロセスを中断し，プロセスを再開させるか強制終了させるかをユーザに選択させるものである．しかし，CriptoAPI や OpenSSL を用いない暗号化アルゴリズムを用いたランサムウェアには対処できないという問題点がある．

2.2 編集前後のファイルの比較

Scaife らは，CriptoDrop と呼ばれる，編集前後のファイルの違いでランサムウェアを検知する手法を提案している [3]．これは，ランサムウェアの暗号化がファイルの内容を大きく変えることに着目し，検知するものである．主な指標に，マジックナンバーによるファイルタイプの確認や，ファイル内容の類似度の確認，エントロピーの上昇の確認を行い，補助的な指標として，ファイルの削除やプロセスが参照しているファイルタイプの数を確認し，これらの全体的な評価をスコア化し，閾値以上のプロセスをランサムウェアとして検知・停止を行う．実験では，492 個のランサムウェアサンプルに対し，中央値で 10 個程度のファイル損失でランサムウェアを停止できた．これにより，どのような暗号化ライブラリを用いられた場合でも，ファイルの暗号化や破壊を検知してランサムウェアを停止できるというメリットがある．しかし，類似度やエントロピーをある程度維持した小さな編集を行い，一部の暗号化やファイルの分割・並べ替えを行うことでファイルの破壊ができる可能性がある．

2.3 ユーザ操作以外のファイル編集の検知

重田らや Scaife らの手法の課題を解決するため，本田らはユーザによる操作以外での文書ファイルや画像ファイルの編集をランサムウェアとしてリアルタイムで検知し無効にする手法を提案している [4]．これは，ファイル編集用ウィンドウが画面上に十分見える大きさで表示されていることと，ファイル編集用ウィンドウ内に内容が表示されている事の二要素をもってユーザによる文書編集とし，それ以外の文書編集をランサムウェアとして，ファイルの削除・上書きを禁止することで，ランサムウェアによる文書の暗号化を防ぐものである．実験では，この手法を実装した PC に Cerver などのランサムウェアに感染させた場合，多くのランサムウェアにおいて，文書の保護ができていた．これにより，多くのランサムウェアの動作である，ユーザに見えないバックグラウンドでのいかなる暗号化やファイル破壊も無効化できる．また，もしこの手法を回避するアルゴリズムで暗号化を試みても，文書の改変を文書編集用ウィンドウが表示されている状態で，かつ人間的な低速度で行わなければならないため，ユーザがランサムウェアの動作を知覚しやすくなると考えられる．

2.4 デコイファイルの監視

ランサムウェアの実行をリアルタイムで検知するために，囮となるファイルをあえてランサムウェアにアクセスさせ，囮にアクセスするプロセスをランサムウェアとする手法が研究されている．萩原らは囮となるデコイファイルを用意してランサムウェアをリアルタイムで検知し，プロセスを特定して停止する手法を提案している [5]．これは，ランサムウェア行うファイル操作の特徴に着目し，ランサムウェアが暗号化する場所にデコイファイルを作成し，デコイファイルの暗号化操作を監視して，暗号化を行うプロセスを停止することで，ランサムウェアをリアルタイムで検知し，実害なく停止させるものである．実験では WannaCry，Cerver など 14 種類のランサムウェアに対して動作停止が可能であり，OneDrive や 7-ZIP など 5 種類の良性プログラムを実行させた場合の誤検知や誤停止も無かった．これにより，重要なファイルが暗号化されることなく，ランサムウェアを停止することができる上，デコイファイルが暗号化されている時のみプロセスの検出が行われるため，それ以外の時に PC の負荷が軽減できるというメリットがある．

2.5 機械学習によるマルウェア識別

Kim らは静的解析をベースとした機械学習によるマルウェアの識別手法を提案している [6]．これは，既知のマルウェアやその亜種マルウェアを類似性のハッシュアルゴリズムを用いて検出し，残りのファイルを機械学習によってマルウェアか良性ソフトウェアかを識別するものである．学習データとテストデータに対し，DLL や API，エントロピーといった特徴を抽出し，類似性のハッシュアルゴリズムである ssdeep，TLSH，DHASH を用いてハッシュ値を算出する．これにより，似ているデータに対して類似のハッシュ値を出力するため，学習データと同一のマルウェアと，類似する亜種マルウェアを検出できる．それぞれのアルゴリズムで判定した結果を，最終的に多数決で決定させた場合，約 47%の有効データ内で 99%以上の精度でマルウェアを識別できた．また，これにより検出できないデータに対しては，SVM や DNN，Dicision Tree，K-NN といった機械学習によってマルウェアの識別を行う．機械学習によるマルウェア識別実験では，それぞれ 92%から 94%の精度で識別でき，類似性のハッシュアルゴリズムと組み合わせた場合，約 95%の精度でマルウェアを識別できた．

3. 提案手法

3.1 コンセプト

本論文の目的は，以下の 3 つの要件を満たすランサムウェアのシステムを提案することである．

(A) ユーザのファイルを暗号化し，攻撃者のみが復号できるようにする．

(B) 2 章で紹介された手法を用いて検知，プロセスの停止が理論上できない．

(C) 既存のウイルス対策ソフトウェアのマルウェア検知，プロセス監視などで検知できない．ただし，良性ソフトウェアと同じ警告を発する場合を除く．

これらの要件を満たすシステムとして，本論文では，オンラインバックアップサービスを応用したランサムウェア

システムを提案する．これは，静的解析やパターンマッチングにおいて一つのファイルのみを参照して判定を行うなら，悪性コードを分散させたマルウェアは検知できないと仮説を立てたものである．概要として，攻撃者はまず無償で大容量のオンラインストレージを使えるバックアップサービスを提供し，ファイル送信用のソフトウェアをインストールしたユーザがバックアップを行うディレクトリを選択し，継続的にサーバへ送信する．そして攻撃者が任意のタイミングでソフトウェアのアップデートを行い，サーバから悪性コードを含む DLL を受信する．受信した直後に DLL を読み出し，ユーザ PC 上の指定ディレクトリを破壊し，サーバ内のファイルを全て暗号化する．その後，攻撃者はサーバ上の暗号化したファイルを人質に，ユーザへ身代金を要求する．これにより，マルウェアの特徴を持たずに悪意ある挙動を引き起こせる．

まず，サーバ上のファイルが暗号化され，鍵を持つ攻撃者のみが復号できるので要件 (A) を満たす．また，この手法であれば，クライアント側のプログラムが CriptAPI や OpenSSL を使用する必要はないので重田らの手法で検知できず，ユーザが良性のオンラインバックアップサービスを使用するためにバックグラウンドのファイル編集を許可しなければならず，良性のファイル編集と悪性のファイル編集を区別できないため，バックアップサービスを使用する場合に Scaife らの手法や本田らの手法で停止できない．デコイファイルのディレクトリ名はすべて “!” であり，名前の回避は容易にできるため，萩原らの手法で停止できず，機械学習において，トレーニングデータにない独自の DLL を用いたマルウェアは検知できないため，Kim らの手法で検知できない．これらより，要件 (B) を満たす．要件 (C) は実際に動作させる必要があるため，評価の章にて検証する．以降では提案手法の解説をしてゆく．

3.2 攻撃手法

3.2.1 潜伏期間

潜伏期間では，良性のオンラインバックアップサービスとしての機能を提供する．この段階での動作はバックアップを行うためのものであり，ユーザが不利益を被ることはない．ソフトウェアはコンソール上で動作するものとする．ここでの手順は以下のようになる．

手順 1．攻撃者はオンラインストレージ用のサーバを用意し，ファイルをサーバへ送信するためのソフトウェアを公開する．ここではインストーラではなく，実行ファイルを公開する。

手順 2．ユーザがソフトウェアをダウンロードし，実行する．ソフトウェアの起動時に ID を入力してログインする．本来ログイン処理は ID とパスワードを用いてユーザを検証するが，今回はランサムウェアを評価するためのプログラムであるため，パスワードは省略する．ID を入力してサーバへ送信し，ID によってサーバ上でユーザを識別する．アップデートがあった場合、自動的にサーバからファイルを受信し、実行ファイルと同じディレクトリへ書き込み，上書きする．

手順 3．アップデート後，ソフトウェア起動時および一定時間毎に自動的に指定ディレクトリ内のファイルを，サーバへ送信する．クライアントはまず，バックアップディレクトリの絶対パスを送信し，サーバ上でテキストファイルに記録する．その後，ディレクトリ内のファイル名とファイルデータを送信し，サーバ上に格納する．サーバでは，ID 毎にディレクトリを用意し、それぞれのディレクトリ内にファイルを格納することで，ユーザ毎にバックアップファイルを管理できるようにする．

クライアント側の，ディレクトリ内のファイルをサーバへ送信するコードは DLL に実装し，AllFileSend.dll とする．アップデート処理の直後に AllFileSend.dll を呼び出すようにすることにより、アップデートで AllFileSend.dll を悪性コードに上書きした直後に悪性コードを実行できる．

手順 4．ソフトウェア起動時のファイル送信の後、現在バージョンなどの情報を文字列で出力する．文字列の出力も DLL で実装し，DisplayMassage.dll とする．アップデートによって DisplayMassage.dll を上書きできるようにすることにより，アップデート毎に出力するバージョン情報を更新すると共に，ランサムウェア化した際に脅迫文を表示させることができるようになる．

手順 5．使用できる機能を表示し，ユーザが番号を入力することでそれぞれの機能を実行する．ユーザが実行できる機能と番号は以下の通りである．

機能 1．バックアップディレクトリ追加：ユーザがバックアップを取りたいディレクトリを設定する．絶対パスを入力してディレクトリを指定し，ディレクトリが存在するかを確認する．存在していた場合はテキストファイルに記録する．ファイル送信，受信の際には，このテキストファイルを参照する．入力された絶対パスに対し，ディレクトリが存在するか確認し，テキストファイルに記録する機能は DLL で実装し，AddDir.dll とする．AddDir.dll 内の関数は戻り値に 0 を返す．これにより，ランサムウェア化した際に，AddDir.dll を呼び出すことで，ファイルを 0 で上書きして破壊できる．

機能 2．バックアップディレクトリ削除：登録されたバックアップディレクトリをテキストファイルから消去する．絶対パスを入力してディレクトリを指定し，テキストファイルにパスが存在するかを確認する．存在していた場合はテキストファイルからパスを消去する．ファイル上書きによるマルウェア判定を避けるため，パスが存在するか確認し，テキストファイルから消去する機能は DLL で実装し，DeleteDir.dll とする．

機能 3．バックアップディレクトリ表示：テキストファ

イルを読み出し，登録されたバックアップディレクトリをコンソール上に出力する．

機能 4．バックアップディレクトリ初期化：登録されたバックアップディレクトリをテキストファイルから全て消去する．ファイル上書きによるマルウェア判定を避けるため，テキストファイルを初期化する機能は DLL で実装し，ClearDir.dll とする．

機能 5．バックアップファイル受信：サーバ上にあるファイルを受信する．サーバ上からバックアップディレクトリの絶対パスを受信し，クライアント側でディレクトリ内を全て消去する．その後，ディレクトリ内のファイル名とファイルデータを受信し，バックアップディレクトリ内に格納する．ディレクトリ内の消去やファイル書き込みによるマルウェア判定を避けるため，ファイルを受信する機能は DLL で実装し，Recieve.dll とする．

3.2.2 ランサムウェア化

3.2.1 項にて説明した，良性のオンラインバックアップサービスとして振舞っていたものを，攻撃者の任意のタイミングでソフトウェアをランサムウェアに更新し，クライアント側のファイルを破壊し，サーバ内のファイルを暗号化して身代金を請求する．本論文では，仮に 10 万円相当のビットコインを要求し，接続から 3 日経過すると身代金が 2 倍になり，7 日経過するとサーバへの接続ができなくなるものとする．

本論文で作成するランサムウェアは，暗号化はサーバ上で行い，クライアント側のファイルを ‘0’ で上書きして破壊する．サーバ上のプログラムはユーザが検証できないので，クライアント側のプログラムにてファイルの破壊がマルウェア判定されないようにする必要がある．そのため，ファイルの編集は全て DLL を呼び出して行い，上書きする際の ‘0’ の入力は，AddDir.dll 内の関数の戻り値を用いる．ファイルを ‘0’ という文字で上書きする一連のコードが一つの実行ファイルにあれば悪性であることが明白である．しかし，独自ライブラリの関数の戻り値で上書きする場合，参照ライブラリの戻り値を確認しなければ，悪性コードであるかを判断できない．静的解析では一つのファイルでしか判断できないため，このプログラムが良性か悪性かを断定することはできない．

また，サーバでは、常時実行させるプログラムを変更する．接続してくるクライアントの ID に対して、ブラックリストとホワイトリストを設ける．ランサムウェアに更新後，最初の接続から 7 日以内に入金が無かった ID に対して，ブラックリストに登録して接続を拒否する．受信したファイルに対して全て暗号化し，ホワイトリストに登録されていない ID に対してはクライアントへファイルを送信しない．身代金が振り込まれたら，ホワイトリストへ登録してファイルを送信できるようにする．ファイル送信の際はサーバ上でファイルを復号してから送信する．

ここでの手順は以下のようになる．

手順 1．攻撃者が任意のタイミングで，ファイルを送信後に破壊する DLL と脅迫文を表示する DLL をサーバのアップデート用ディレクトリに格納する．クライアントがソフトウェアを起動させると，自動的にアップデートが行われ，ファイルを送信する AllFileSend.dll のコードをファイルを送信後に破壊するコードに，バージョン情報を表示する DisplayMassage.dll のコードを脅迫文を表示するコードにそれぞれ上書きする．

手順 2．アップデート後の自動的なファイル送信のタイミングで，悪性コードに上書きした AllFileSend.dll を呼び出して中のコードを実行する．サーバへファイルを送信した後，クライアント側のファイルを破壊する．ファイルを破壊する際には，3.2.1 項の機能 1 に記述した，バックアップディレクトリをテキストファイルに記録する DLL を呼び出して，その戻り値である ‘0’ で上書きする．

手順 3．手順 2 でのファイル送信が行われた後，サーバにて，クライアントから受信したファイルを全て暗号化する．サーバのホワイトリストに登録されていない ID の場合，クライアントから受信要求があっても応じない．

手順 4．3.2.1 項の手順 4 で記述したバージョン情報を出力するタイミングで，脅迫文を表示するコードに上書きした DisplayMassage.dll を呼び出して，コンソール上に脅迫文を表示する．脅迫文の内容として，以下の情報を出力する．

・PC 内のバックアップファイルを削除し，サーバ内のファイルを暗号化したこと．・ファイルを返すために 10 万円の身代金をビットコインで払う必要があること．また，3 日後に身代金が 2 倍必要となり，7 日後にはサーバから接続を断絶すること．・身代金が 2 倍になる 3 日後の時刻とサーバから断絶される 7 日後の時刻．・身代金の送金先．

手順 5．身代金が振り込まれたら，手動でサーバのホワイトリストに ID を登録する．ID がホワイトリストに登録されていれば，クライアントからの受信要求に応じ，ファイルを復号してからバックアップファイルを送信する．

3.3 オンラインバックアップサービス

ユーザにファイルを送信させるオンラインバックアップサービスには，以下の機能を持たせる．

(1) 無償で 50GB 以上の容量を提供する．

(2) 起動時に ID を入力させ，サーバに登録する．

(3) 平常時，ユーザが指定したディレクトリ内のファイルを，起動時および一定時間毎にサーバへ自動的に送信する．また，ユーザの任意のタイミングでサーバ上のファイルを受信できる．

(4) サーバからプログラムをダウンロードし，ソフトウェアを更新できる．

(1) は，有償にした場合に金の流れが発生してしまい，そ

れを根拠に攻撃者が特定される可能性があるためである．サービスの開発者を匿名もしくは他人に転嫁させるために，サービスは無償である必要がある．また，現在において無料かつ無条件で利用できるオンラインストレージサービスの内，容量が最も大きいものが，50GB を提供する Mega Limited による MEGA[7] である．無期限で 50GB 以上のストレージ容量を提供すれば，最もストレージ容量の多いオンラインストレージサービスとしてユーザが増えると考えられる．(2) は，オンラインバックアップサービスを展開する上でサーバ側でユーザの識別を行う必要があるためである．(3) は，バックグラウンドでのファイルアクセスをユーザに承諾させるためである．ユーザの PC 上で行われるファイル編集は，ウイルス対策ソフトや研究手法によって検知され，悪性のものであれば停止できてしまう．しかし，良性のソフトウェアの段階でユーザにファイルを指定して編集を許可させることで，悪性の挙動を検知されなくさせる狙いがある．(4) は，アップデートによって悪性コードを追加するためである．ソフトウェアをダウンロードする際に，一つのファイルに悪性コードが含まれていた場合は，ユーザによるマルウェア検知によって発覚する可能性がある．しかし，悪性コードを複数の DLL ファイルに分割して，組み合わせることで悪性の挙動を引き起こす場合，実行ファイルやそれぞれの DLL ファイルからマルウェアと判断される事は無いため，アップデートによる悪性コードの追加は検知されないと考えられる．

4. 実装

開発環境は Visual Studio 2019 で C++を用いた．実験環境のため，クライアントは 1 台としてサーバを設定した．

4.1 オンラインバックアップサービス

3.2.1 項のオンラインバックアップサービス機能の実装について説明する．通信は winsock2.h を用いたクライアントサーバ方式で，クライアントとサーバの宛先は IP アドレスで指定し，通信方式に TCP を用いた．サーバからアップデートの有無が送信され，アップデートがあった場合，クライアント側のソフトウェアの更新を行えるようにした．

クライアント側では，バックアップを行うディレクトリパスを設定する必要がある．

機能 1 の AddDir.dll では，ユーザに std::cin で登録するディレクトリの絶対パスを入力させディレクトリを選択し，ディレクトリが存在するかを stat を用いて判定する．ディレクトリが存在していれば，「ディレクトリパス+改行コード」の形で dirPath.txt に追記する．また，悪意ある挙動に用いるため，バックアップディレクトリを登録する関数 AddDir は整数型の戻り値で 0 を返すようにする．

機能 2 の DeleteDir.dll では，ユーザに std::cin で登録を消去するディレクトリの絶対パスを入力させディレクトリを選択し，dirPath.txt を一行ずつ読み出して入力したパスが存在するかを判定する．パスが存在していれば，入力パス以外を再度 dirPath.txt に出力して上書きする．

機能 3 の登録されたディレクトリの表示では, dirPath.txt の内容を読み出し，std::cout でコンソール上に出力する．

機能 4 の ClearDir.dll では, dirPath.txt を std::ios::trunc モードで読み込み, 閉じることで dirPath.txt を初期化する．

手順 3 で行うファイル送信では，dirPath.txt を読み出し，該当するディレクトリの全てのファイルをサーバへ送信するようにした．具体的には chrono.h で現在時刻を取得し，プログラムが起動している間は 1 時間毎に上記の操作を自動的に行う．これにより，クライアント側で指定ディレクトリ内の自動的なファイルアクセスとファイル削除を許可できる．また，アップデートにより悪性の処理を埋め込めるよう，この部分の関数 AllFileSend を DLL で作成し，実行ファイルと同階層に用意した．

さらに，ランサムウェア化した際に脅迫文を表示できるよう，任意のメッセージを std::cout で出力する関数 DisplayMassage を DLL で実装した．

4.2 ランサムウェア

3.2.2 項にて述べたランサムウェアの実装について説明する．なお，ビットコインによる送金は行わないものとした．

アップデートを受信したら，AllFileSend.dll，DisplayMassage.dll に悪性コードを追加する．クライアント側のソフトウェアが起動している時，アップデートを検知したらユーザに通知し，サーバから悪性コードを含む DLL ファイルを受信し，DLL ファイルに上書きする．

アップデートによって上書きする DLL とその機能は次のとおりである．

・AllFileSend.dll 各ファイル送信の後，参照ファイルを out 形式で開き，AddDir.dll を呼び出す．関数 AddDir の戻り値をファイルに書き込み，ファイルを閉じる．関数の戻り値は 0 なので，これにより参照ファイルを 0 で上書きすることになる．

・DisplayMassage.dll 脅迫文を表示する．

アップデートによって悪性コードを含んだ DLL に上書きした後，ソフトウェアが AllFileSend.dll を読み出してランサムウェアを実行する．クライアント側で，dirPath.txt に存在するファイルパスとサーバからのバックアップディレクトリ全てに対し，AddDir の戻り値である ‘0’ で上書きして破壊する．

サーバでは，クライアントから送信されたファイル全てを暗号化し，攻撃者以外に復号できないようにする．ファイルの暗号化・復号には，AES のサンプルコードを配布している Web サイト [8] から入手した C 言語の AES コードを基に実装した．サンプルコードは 1 ブロック 16bit の

データを暗号化・復号するものであったが，そこから任意のファイルのバイナリデータに対してブロック単位で暗号化・復号できるようにコードを追加した．

クライアント側でファイルの削除が終わったら，アップデートによって上書きされた DisplayMassage.dll を読み出し，脅迫文を出力する．

5. 評価

3.1 節で定義した要件を評価した．実装したランサムウェアを実際に OS 上で動作させて，有効な被害を与えられるか，ユーザにとって不審な挙動を示さないかを確認し，要件 (A) を評価した．また，ランサムウェアに対してウイルススキャンを行い，マルウェアとして認識されるかを実験し，要件 (B) を評価した．

5.1 動作実験

4.2 節の手法で実装したランサムウェアを仮想マシン上で実行し，ファイルの視覚的な変化を確認したり，実行が可能であるか，修復が可能であるかを確認したりすることで，マルウェアとしての有効性を評価することを目的とした．

5.1.1 実験方法

動作実験は仮想化ソフトウェアである virtualbox を用いて，仮想化した Windows10 上で行った．仮想マシン上にて，実装したランサムウェアによる編集結果を確認するためのファイルを複数用意した．種類と数を以下に示す．

・テキストファイルとして，.txt ファイルを 100 個

・画像ファイルとして，.bmp ファイルを 20 個，.png ファイルを 20 個，.jpg ファイルを 20 個．

・音声ファイルとして，.wav ファイルを 30 個，.mp3 ファイルを 30 個

・動画ファイルとして，.mp4 ファイルを 10 個

・office ソフトウェアによるファイルとして，.pptx ファイルを 30 個，.docx ファイルを 30 個，.xlsx ファイルを 30 個

ファイルを用意した仮想マシン内で，実装したソフトウェアを実行し，ランサムウェアとなった際に実際にファイルが破壊されるかを確認した．実験手順を以下に示す．

1. ソフトウェアを起動し，上記ファイルが存在するディレクトリを設定する．

2. サーバからアップデートを受信し，継続してソフトウェアを実行する．この時，良性のソフトウェアでも起こりうる警告が表示された場合は，ユーザが不審に思わないものとして，手作業で許可し実行を継続する．それ以外の警告が表示された場合は，ランサムウェアであるとユーザに判明したとして，実行を中断し，以降の手順を行わない．

3. 身代金の要求文を確認した後，設定したディレクトリ内のファイルが破壊されていることを確認する．

表 1 ファイル破壊数と復元数
Table 1 Number of destroyed and recovered files.

	ファイル数	破壊ファイル数	復元ファイル数
.txt	100	100	0
.bmp	20	20	0
.png	20	20	0
.jpg	20	20	0
.wav	30	30	0
.mp3	30	30	0
.mp4	10	10	0

5.1.2 復元可能性の実験

ランサムウェア実行後に削除されたファイルが復元可能であるかを確認した．修復ソフトウェアはフリーソフトウェアである DataRecovery と Recuva を用いた．これは，ディスクをスキャンして，紛失データを検出・分析し，復元できるデータ復旧ソフトウェアである．ランサムウェアを実行した後にそれぞれのソフトウェアをインストールし，破壊されたファイルの復元を試み，復元できたファイルの種類と数を確認した．結果を**表 1** に示す．表より，すべてのファイルが破壊され，復元不可能となった．

5.2 ウイルススキャン実験

4.2 節の手法で実装したランサムウェアに対してウイルススキャンを行い，マルウェアの可能性があると認識されるか否かを確認することによって，マルウェア検知のシステムに対しての有効性を評価した．

5.2.1 実験方法

マルウェアスキャンには，オンラインのマルウェア検査サービスである VirusTotal[9] を用いた．VirusTotal は，検査対象のファイルをアップロードすることで，60 種類以上のアンチウィルス製品を使用して検査を行い，それぞれのアンチウイルス製品の結果が一覧表示される．また，アップロードしたファイルに対して，リンクされているライブラリ名の取得などが行われ，それらの結果も取得できる．実装したソフトウェアを VirusTotal にアップロードし，何個のアンチウイルス製品にマルウェアと判断されるか確認した．

また, 実装したランサムウェアがセキュリティソフトウェアによって被害を予防できるかを評価した．VirtualBox を用いて仮想化した Windows10 上でセキュリティソフトウェアを起動した状態で，ウイルススキャンを実行した後に，ソフトウェアを実行した．ウイルススキャンの結果と，実行の際の警告内容と実行結果を以てセキュリティソフトウェアに対する有効性を評価した．実験に使用したセキュリティソフトウェアは，Windows セキュリティである．実験の手順を以下に示す．

1. Windows セキュリティを起動し，ファイアウォール，ネットワーク保護，評価ベースの保護を有効にする．

表 2 マルウェア判定
Table 2 Malware detection.

ファイル名	判定ソフトウェア	マルウェア判定
FileBackupClient.exe	67	4
AllFileSend.dll	65	0
DisplayMassage.dll	65	0

表 3 ファイル破壊数と復元数
Table 3 Number of destroyed and recovered files.

	ファイル数	破壊ファイル数	復元ファイル数
.txt	100	100	0
.bmp	20	20	0
.png	20	20	0
.jpg	20	20	0
.wav	30	30	0
.mp3	30	30	0
.mp4	10	10	0

2. ランサムウェアに対してウイルススキャン機能を使用し，マルウェア検知の可否を確認する．ここでマルウェアであると判断された場合は，ユーザがランサムウェアを実行をしないものとし，以降の手順を行わない．

3. サーバに保存されるファイルを用意した上で，サーバからアップデートを受信し，ランサムウェアを実行する．実行した後，良性のソフトウェアでも起こりうる警告が表示された場合は，ユーザが不審に思わないものとして，手作業で許可し実行を継続する．それ以外の警告が表示された場合は，ランサムウェアであるとユーザに判明したとして，実行を中断し，以降の手順を行わない．

4. 用意したファイルを確認し，破壊されたファイルと復元可能なファイルの種類と，その数を確認する．

5.2.2 結果

手順 2 の VirusTotal によるマルウェア検知の結果を**表 2** に示す．表より，FileBackupClient.exe が 4 種類のセキュリティソフトにマルウェアと判定されたが，悪性挙動を起こす AllFileSend.dll や，脅迫文を表示する DisplayMassage.dll は，セキュリティソフトによるマルウェア判定はされなかった．FileBackupClient.exe がマルウェアの判定をされたが，これは後述の理由から偽陽性である可能性が高いため，実験を続行した．

続いて手順 4 にて確認したファイルの結果を**表 3** に示す．表より，Windows セキュリティが有効であっても，すべてのファイルが破壊され，復元不可能であった．

6. 考察

6.1 動作実験

表 1 より，5.1.1 項にて用意した対象のファイルを全て破壊することができ，インストールした復元ソフトウェアによって一切復元ができないことがわかる．これは，悪性挙動の際に，ファイルを削除するのではなく，ファイルに「0」で上書きしているため，ファイルのデータが破棄されたためであると考えられる．実際に，メモリ内を復元ソフトウェアによって検索した結果，破壊されたファイルを検出することは可能であったが，そのファイルを開いた際には復元不可能である旨のメッセージが表示されていた．そのため，ユーザ PC 上の重要なファイルを使用不可能にし，サーバ上でのみユーザのファイルを保持する状態を作り出すことに成功し，サーバ上のファイルを人質に身代金を請求できる．これより，4 章で実装したプログラムは，ランサムウェアとして十分動作するといえる．

6.2 ウイルススキャン実験

表 2 より，実行ファイルである FileBackupClient.exe が 4 種類のアンチウィルス製品に対してマルウェアの判定をされていた．しかし，これらの判定は偽陽性である可能性が高い．それを実証するため，実験後，良性であることが明白なプログラムをコンパイルした実行ファイルを VirusTotal に判定させた．FileBackupClient.exe は，winsock2.h, ws2tcpip.h, iostream, fstream, vector, filesystem, string, chrono を呼び出して実行する．表 2 でマルウェア判定したセキュリティソフトは，呼び出したヘッダファイルやライブラリを認識してマルウェアの判定を出していると仮説を立てた．そのため，上記のヘッダファイルやライブラリを呼び出して，std::cout で “HelloWorld!” を出力するのみの，明らかに無害なソースコードをコンパイルした実行ファイルを VirusTotal に判定させた．

本論文で実装したクライアントプログラム FileBackupClient.exe を VirusTotal で判定させた結果を**図 1** に，“HelloWorld!” を出力するプログラムを VirusTotal で判定させた結果を**図 2** にそれぞれ示す．図 1 から，FileBackupClient.exe に対し Bkav Pro，MaxSecure，Microsoft，SecureAge APEX の 4 種類のセキュリティソフトがマルウェアの判定を出しているが，図 2 から，“HelloWorld!” を出力するプログラムに対しても同じセキュリティソフトが同じメッセージでマルウェアと判定していることがわかる．このことから，これらのセキュリティソフトは，FileBackupClient.exe に対し悪性の挙動を読み取ったのはなく，ソケット通信や DLL の参照，インクルードしたヘッダファイルなどの一部の機能に対し，良性悪性に関わらずマルウェアであると判定しているものと考えられる．即ち，これらのセキュリティソフトは良性のソフトウェアであってもマルウェアと判定するため，FileBackupClient.exe のマルウェア判定は偽陽性であるといえる．

また，5.1.1 項にて VirusTotal に判定させたプログラムの内，FileBackupClient.exe は単体では悪性の挙動を行わず，実際にファイルの破壊をもたらす書き込みを行うコードが書かれているのは AllFileSend.dll である．実際に悪性

の挙動を引き起こす DLL はマルウェアとして検出されていないことから，本論文のランサムウェアシステムは，アンチウィルス製品によるマルウェア検知を回避できているといえる．

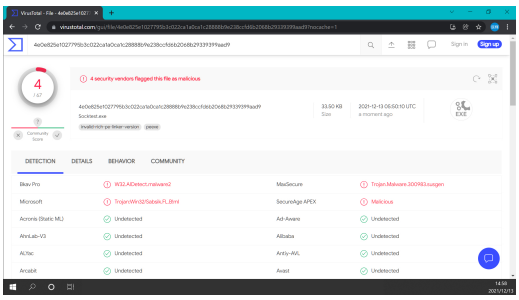

図 1 本論文で実装したクライアントプログラム
Fig. 1 Client programs implemented for evaluation.

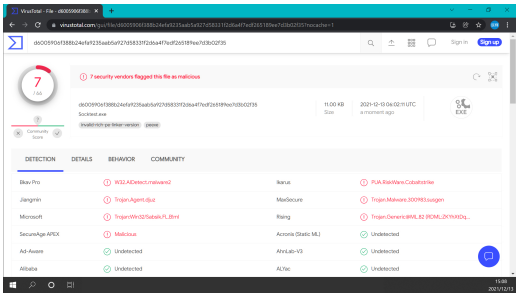

図 2 ”HelloWorld”を出力するのみのプログラム
Fig. 2 Client program for outputting ”HelloWorld”.

6.3 Windows セキュリティのフォルダ保護

Windows セキュリティには，他のセキュリティソフトがインストールされていない場合に，ランサムウェアの防止のためのフォルダ保護機能が存在する．これは，指定したディレクトリ内のファイルを保護するもので，承認されていない変更を無効にする機能である．5.2.1 項の実験にて，ファイル破壊の際に，バックアップディレクトリに対しこのフォルダ保護機能を有効にすることで，ファイルの編集が無効にされ，破壊ができないことがわかった．その際，ファイルの読み取りやバイナリデータの送信が可能であった．

そのため，本論文のランサムウェアのような，ファイルを編集する必要のあるマルウェアの場合，良性の段階でファイル編集をユーザに許可させ，フォルダ保護を有効にすると良性のサービスを受けられなくさせる必要がある．

7. 結論

悪性コードを分割することによって，各実行ファイル，DLL はマルウェア判定されず，ランサムウェアのファイルの破壊・復元と身代金要求といった機能を持たせるシステムを作成することができた．ただし，本論文で実装したプログラムでファイルの破壊ができるのは，ユーザが編集を容認しているファイルのみであり，管理者権限が必要なファイルは編集できない．

また，Wannacry や Cerver など実際のランサムウェアには，身代金を被害を拡大させるための伝染機能が存在する．これらのランサムウェアと同等に機能させるためには，管理者権限の奪取やネットワークの詮索，リモートコントロールによるコードの実行などを行う必要がある．今後の展望として，悪性コードの分割によってこれらの動作を再現できるシステムを作成したい．

参考文献

[1] 独立行政法人 情報処理推進機構 : 脆弱性対策，入手先〈https://www.ipa.go.jp/security/vuln/index.html〉(参照 2021-08-06).
[2] 重田貴成，森井昌克，長谷川智久，池上雅人，石川堤一：ランサムウェアにおける暗号化処理について，情報処理学会コンピュータセキュリティシンポジウム 2016 論文集，Vol.2016, No.2, pp.134–137, (2016).
[3] Scaife, N., Carter, H., Traynor, P., Butler, K. R. B.: Cryptolock (and drop it): Stopping Ransomware Attacks on User Data, In *Proc. 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pp.303–312, (2016).
[4] 本多俊貴，向山浩平，白井丈晴，西垣正勝：ユーザの文書編集操作を考慮したランサムウェア検知に関する一検討，情報処理学会研究報告, Vol.2017-CSEC-77, No.14, pp.1–8, (2017).
[5] 荻原拓海，小林良太郎，加藤雅彦：デコイファイルを用いた暗号化型ランサムウェアの検知とプロセス特定に関する検討，情報処理学会研究報告，Vol.2021-DPS-186, No.50, pp.1–8, (2021).
[6] Kim, S., Ha, J., Oh, S., Lee, T.: A Study on Malware Identification System Using Static Analysis Based Machine Learning Technique, *Journal of the Korea Institute of Information Security & Cryptology*, Vol.29, pp.775–784, (2019).
[7] Mega Limited：MEGA, 入手先〈https://mega.io/〉(参照 2021-05-18).
[8] サンプルプログラム・ソース・ソフトウェアの配布：PJC, 入手先〈https://free.pjc.co.jp/AES/index.html〉(参照 2021-12-14).
[9] Google inc : VirusTotal, 入手先〈https://www.virustotal.com/gui/〉(参照 2021-08-06)