L-016

ランサムウェア対策としてのクラウドストレージバックアップシステムの提案
A Proposal of the backup system with Cloud Storage System for ransomware countermeasure

播磨尚希 † 小林孝史 ‡
Naoki Harima Takashi Kobayashi

1. はじめに

近年その活動が問題視されているマルウェアとしてランサムウェアが存在する．IPA[1] が発表した情報セキュリティ10 大脅威 2018 においては，2017 年に続いて 2 年連続で個人と組織の両区分において，2 番目に社会的に影響が大きかった事案として選出されている [2][3]．ランサムウェアは情報資産を利用不能とし，その復旧と引き換えに身代金を要求するマルウェアである．情報資産を利用不能とする手法は様々であるが，近年では特にシステム内に存在するファイルを高度な暗号技術を用いて暗号化する暗号化型ランサムウェアが流行している．

ランサムウェアによる被害にあった際には，身代金を支払ったとしても情報資産が復旧するとは限らないばかりでなく，支払った身代金が資金源となる事で犯罪を促進する動機づけとなり得てしまうため，公的機関や大手セキュリティベンダは身代金を支払わないように呼びかけており，バックアップをとることでランサムウェアによる攻撃から情報資産を保全することが重要であるとしている [4][5]．

暗号化型ランサムウェアには，感染した情報端末のみならず，それにマウントされたファイルシステムに対しても暗号化を行うマルウェアファミリが存在することが知られている．そのため，バックアップデータを保全するためには，バックアップの取得後にバックアップメディアを切り離す必要がある．クラウドストレージへの非透過的なアクセスはこれを実現するものであるが，一般的な同期プログラムによるクラウドストレージへのバックアップには，暗号化型ランサムウェアによって生成された暗号化ファイルをも複製してしまうことで，暗号化前のバックアップファイルが上書きされてしまう問題が存在する．本稿では，非透過的にアクセス可能なメディアであるクラウドストレージをバックアップメディアとして利用し，ファイル複製時にファイル識別を行う事でランサムウェアの活動を検知してファイルの複製を停止し，ランサムウェアによる暗号化からバックアップデータを保全するシステムを提案する．

2. ランサムウェア

ランサムウェアとは，情報資産を利用不能とし，その復旧と引き換えに身代金を要求するマルウェアである．情報資産を利用不能とする方法は様々であり，端末の画面をロックすることで操作を不能とする画面ロック型ランサムウェアのほか，システム内に存在するファイルを暗号化する暗号化型ランサムウェアなどが存在する．金銭を獲得する手法としてのランサムウェアの有用性が多くのサイバー犯罪者に認知されたことに加え，ダークマーケットを通じて容易にランサムウェアを手に入れることが可能となったことや，多言語に対応したマルウェアファミリが登場したことなどから，ランサムウェアの脅威は高まる一方である．本稿では，ランサムウェアの中でも特に暗号化型ランサムウェアを対象とし，対抗手段としてのバックアップ方式を検討する．

2.1. 暗号化型ランサムウェア

マルウェアファミリによって細部の挙動は異なるが，ここでは暗号化型ランサムウェアの大まかな挙動を説明する．

1. 侵入.
   システムの脆弱性やメールの添付ファイルなどを利用してシステムに侵入する．
2. ファイル検索.
   侵入したシステムと，そのシステムから透過的にアクセス可能なメディアに対して，暗号化の対象となるファイルを検索する．
3. 鍵の入手.
   C&C サーバと通信し，ファイルの暗号化に用いた鍵を暗号化するための公開鍵を取得する．公開鍵を入手できない場合，暗号化に用いた鍵を平文で残してしまう可能性が存在するため，C&C サーバから鍵が入手できてから，次のステップに進む．
4. 鍵の生成.
   ファイルを暗号化するための対称鍵暗号の暗号鍵と初期化ベクトルを生成する．
5. ファイル暗号化.
   対称鍵暗号によってファイルを暗号化する．
6. 鍵の暗号化.
   ファイルの暗号化に用いた対称鍵暗号の暗号鍵を，事前に C&C サーバから取得しておいた公開鍵で暗号化する．
7. 鍵の送信.
   公開鍵暗号によって暗号化された，ファイルの暗号化に用いた対称鍵暗号の暗号鍵を C&C サーバに送信する．
8. 身代金の要求.
   情報資産が暗号化された旨を利用者に通知し，そ

†関西大学大学院 総合情報学研究科
‡関西大学総合情報学部

の復旧と引き換えに身代金を要求する．攻撃者は Tor などの匿名化技術を用いて被害者と連絡を取り，暗号通貨など匿名性が高い手法での身代金の支払いを求める．

9. 復号鍵の送信．
攻撃者は，被害者からの身代金の支払いを確認すると，C&C サーバに送られてきた暗号化済みの暗号鍵を手元の秘密鍵で復号し，被害者に送信する．

10. ファイルの復号．
攻撃者から送信された鍵を用いて，暗号化されたファイルを復号する．

3. バックアップ

暗号化型ランサムウェアには，感染した情報端末のみならず，それにマウントされたファイルシステムに対しても暗号化を行うマルウェアファミリが存在する．そのため，取得したバックアップを暗号化型ランサムウェアの暗号化から保護するためには，非透過的なアクセスを実現するバックアップメディアを利用する必要がある．この実現には，クラウドストレージなどの非透過的アクセスを実現するリモートストレージを利用する手法のほか，透過的にアクセス可能なメディアにバックアップを取得した後にメディアの切り離しを行う手法，上書き不可能なメディアを利用する手法などが考えられる．

しかし，バックアップメディアに非透過的にアクセスしていた場合でも，バージョン管理を行わないバックアップ方式を採用していた場合，ランサムウェアによって暗号化されたファイルでバックアップファイルを上書きしてしまうと，リストアが不可能となる．

この問題を解消するためにバージョン管理システムを持つバックアップ方式を採用する場合，バックアップの取得に要するデータ容量とリストアに要する時間の面で不利な点が生じる．フルバックアップを採用すれば，リストアに要する時間は短時間だが，要するデータ容量は肥大化する．一方で，増分バックアップや差分バックアップを採用すれば，バックアップに要するデータ容量は少なく済むが，リストアにより多くの時間を要する．

この問題を解消する手法としては，バックアップ頻度を低下させる手法と，管理する世代数に制限をかける手法が存在する．両者ともに管理するバージョンの数を減らす試みであるが，バックアップ頻度を低下させる手法ではリストア時に失われる変更が多くなり，管理する世代数に制限を掛ける手法では，悪意ある変更が繰り返し行われた際にリストアが不可能となる問題が存在する．

4. クラウドストレージ

クラウドストレージはリモートストレージの一種であるが，ブラウザや API，専用のアプリケーションを通じてファイルをアップロードすることで，ファイルを複製する．アプリケーションによる自動同期が行われる場合，透過的なアクセスと同様の挙動を示すが，それ以外の場合では，非透過的なアクセスとなる．また，API が公開されている場合には，任意の処理を追加した同期プログラムの開発が容易に行える．

5. ファイル識別

未知のファイルのファイルフォーマットを調べる手法には，大きく分けて二つの手法が存在する．一つは，ファイルの拡張子を用いる手法．もう一つは，ファイルのバイナリデータを用いる手法である．バイナリデータを用いる手法では，ファイル全体のデータ分布を用いる手法と，ファイル内の特定位置に存在するマジックナンバーなどの一部のデータを用いる手法が存在する [6]．ファイル全体のデータ分布を用いる手法には，識別精度が低いという問題と，ファイル識別に要するシグネチャを生成する際に一定以上のファイルサイズを要することから，極端に小さなファイルサイズのファイルに対しては利用できないという問題が存在する．一方で，ファイルの特定の一部のデータを用いる手法では，ファイル識別に用いるデータ部分を書き換えることで容易にファイル識別をあざむくことが可能になる問題が存在する．また，拡張子によるファイル識別手法においても同様に，識別に用いる情報を容易に改ざん可能な問題が存在する．

5.1. file コマンド

file コマンドは，与えられたファイルのファイルフォーマットを識別する Unix コマンドである．file コマンドはまずパラメータで指定したファイルを読み取り，ファイルシステムテスト，マジックナンバーテスト，言語テストの順にテストを実行し，ファイルの分類結果を標準出力に書き出す．

ファイルシステムテストでは，stat システムコールの結果の検討に基づき，空ファイルか特殊ファイルであるかの確認を行う．続くマジックナンバーテストでは，標準インクルードディレクトリの a.out.h や exec.h，/usr/share/file/magic.mgc などを用いて，マジックナンバーをもとにしたファイル識別を行う．ファイルがすべてのマジックナンバーのエントリに符合しなかった場合，テキストファイルであるかの確認を行う．テキストファイルであるかの判定には，ファイル内の表示可能なテキストを構成するバイト列の範囲を確認することによって文字コードを識別し，可読文書であるかの確認を行う．また，テキストファイルだと判別すると，最後に言語テストが行われる．これは，ファイルの最初の数ブロックに現れる特定の文字列を探し，格納されているコードの言語を特定しようとするものであり，tar ファイルなど，特定のバイナリ形式のファイルの特定にもこれが用いられる．これらの過程を経て，どのテストにおいても該当するエントリが存在しなかった場合，ファイルフォーマットは単に “data” と示される [7]．このため，ヘッダーの挿入などの何らかの改ざんが行われていない場合，現代暗号によって暗号化されたファイルはマジックナンバーを持たないため，偶然何らかのファイルフォーマットのマジックナンバーと等しいマジックナンバーを持つファイルが生成されな

い限り，“data” と識別される．

6. 本研究のシステム

本システムは，ローカルストレージ内の管理対象とするファイルへの変更をバックアップメディアに同期するシステムとして動作する．この時，同期工程にファイル識別工程を加えることで，暗号化型ランサムウェアの活動を検知して同期を切断する．これにより，バックアップファイルの保護を実現する．なお，本システムではバックアップメディアにクラウドストレージを採用し，バックアップに要するデータ容量が肥大化するのを防ぐため，管理世代数に制限をかけたファイルバックアップとして利用する．

システム利用の流れを従来手法のファイルの同期処理のシステムのそれとともに図 1 に示す．

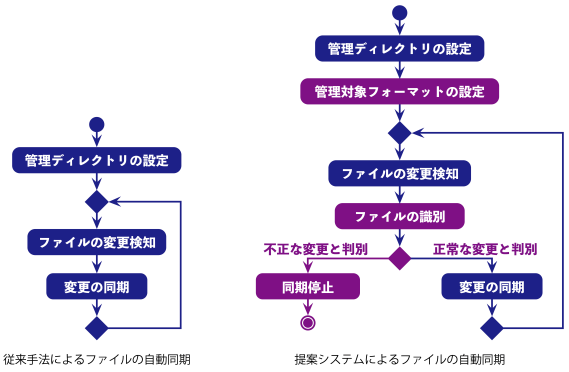

図 1: 従来手法と提案システムのファイルの同期処理

システム利用者は事前に管理対象とするディレクトリ（以下，管理ディレクトリ）及び管理対象とするファイルフォーマットの拡張子を指定する．ここで，管理対象とするファイルフォーマットは，バックアップ対象のファイルフォーマット及び，バックアップ対象とはしないが管理ディレクトリ内に配置されることを許すファイルフォーマットからなる．管理対象のファイルフォーマットの拡張子を指定することで，独自の拡張子を用いて暗号化ファイルを生成する暗号化型ランサムウェアの検知が可能となる．加えて，拡張子によって示されるファイルフォーマットと，バイナリデータを用いた識別手法によって示されるファイルフォーマットが一致するかを確認することで，暗号化前と同一の拡張子を用いて暗号化ファイルを生成する暗号化型ランサムウェアの活動を検知することが可能となり，暗号化後のファイルで同名のクラウドストレージ上のバックアップファイルを上書きすることを防ぐことも可能となる．なお，バイナリデータを用いたファイル識別には file コマンドを用いる事が可能である．ただし，拡張子と file コマンドによって示されるファイルフォーマットは 1 対 1 では対応付かず，n 対 n で対応付けられるため，対応付けを管理する辞書を作成する必要がある．

これらの過程を経て，変更が加えられた管理ディレクトリ内のファイルのファイルフォーマットがバックアップ対象のものであると識別された場合，ファイルの変更内容をクラウドストレージに反映させる．一方で，変更が加えられた管理ディレクトリ内のファイルのフォーマットが管理対象でなかった場合，暗号化型ランサムウェアによる不正な暗号化活動が行われていると判断し，クラウドストレージへの変更の自動同期を停止する．

本システムは，クラウドストレージへの変更内容の反映作業を担うものであるため，システムが動作している状態においてのみ，ファイルの変更がクラウドストレージに同期される．この事から，ファイルシステム自体が暗号化された場合など，システムが動作しなくなるような攻撃が起こった際には，その時点でクラウドストレージへの同期が停止される．また，file コマンドでのマジックナンバーテストに用いられる magic.mgc が暗号化された場合には，空ファイルなどの特殊ファイルでない限り，file コマンドは全てのファイルフォーマットのファイルを “data” として識別する．このため，このケースにおいても，システムの動作する環境に対する暗号化型ランサムウェアの活動を検知し，クラウドストレージへの同期の停止を実現することができる．これらのことより，本システムはフェールセーフの挙動を示すといえる．

なお，管理ディレクトリ内に存在する全てのファイルに対する暗号化を検知する場合，ユーザが意図的に行った暗号化に関しても，暗号化型ランサムウェアによる意図しない暗号化と同様に検知してしまう．これによってユーザがシステム内で暗号化処理を行えない問題を解決するため，暗号化ファイルの設置を許す特別なディレクトリを予め規定する．

7. 評価実験

提案システムを実装し，評価実験を行った．暗号化型ランサムウェアの行う活動のうち，本研究においてその活動の検知に用いる活動は，暗号化型ランサムウェアがユーザの意図しない暗号化を行う活動である．このことより，ファイルの暗号化を行うプログラムを作成し，これを用いたファイルの暗号化と，正規の変更を管理対象のファイルシステムに加え，正規の変更がクラウドストレージに反映されるかの確認と，ファイルの暗号化を変更を検知してクラウドストレージへのファイルの同期が停止されるかの実験を行った．この際，ファイル暗号化に用いるプログラムは，管理対象のファイルフォーマットに定義されていない未知の拡張子を用いて暗号化ファイルを生成するものと，暗号化前のファイル名で暗号化ファイルを生成する二種類のプログラムを用意した．また，システムのプログラムファイル自身と，それを実行する上で利用されている file コマンドの参照する magic.mgc の暗号化を行い，フェールセーフが実現されているかの確認を行った．

7.1. 実験環境

クラウドストレージへの同期とファイルの暗号化，ファイルの識別が実現できる環境であればいかなる環境でも実験を行うことができると考えられる．ここで

は，実装が容易であることから，Linux 系 OS である Ubuntu 環境にて実験を行った．実験環境が Linux 系 OS 上であるため，ファイルの変更検知にはファイルシステムへの変更を高速に通知するカーネルサブシステムである inotify を用いることができる．また，実験システムは Python を用いて実装したため，Python から inotify を利用可能にしたモジュールである pyinotify[8] を用いてファイルの変更を検知した．バックアップ先のクラウドストレージには，コマンドラインからの操作のみでホスティング元より提供される API の認可プロセスが実現できることから Dropbox を利用した．

バックアップ対象のファイルフォーマットとして，jpg・png・pdf・txt・html・mp3・mp4・zip・exe の拡張子を設定し，管理ディレクトリ配下に，インターネット上から無作為に取得したこれらのファイルタイプのファイルを配置した．これらのファイルに対し，ファイル内容を更新する正規の変更を加えるとともに，AES・Blowfish での暗号化を行った．

7.2. 実験結果と考察

評価実験の結果，本システムは管理ディレクトリ内のバックアップ対象ファイルに対する正規の変更をクラウドストレージへ複製し，管理対象ディレクトリ内に不正な拡張子のファイルが生成された場合や，暗号化前のファイルの拡張子を用いて暗号化ファイルが生成された際にはその活動を検知し，クラウドストレージへのファイルの同期を停止することが確認できた．

また，システム稼働後に，システムのプログラムファイルとそれを実行する Python のオブジェクトファイルを暗号化した場合でも，すでにプロセスメモリ空間にオブジェクトコードが展開されているため，システムは問題なく挙動することが確認できた．magic.mgc ファイルが暗号化された場合においては，file コマンドによる各種ファイルのファイル識別結果が “data” となることが確認でき，正規の変更であっても暗号化による変更であっても，クラウドストレージへの同期が停止されることが確認できた．

このことより，提案手法によって暗号化型ランサムウェアによる暗号化からクラウドストレージのバックアップを保全することは可能であると評価できる．また，攻撃による影響がシステムに対して現れた際には，クラウドストレージのバックアップファイルを暗号化から保全するようにシステムは動作することが確認できたため，システムはフェールセーフな挙動を示すといえる．

8. おわりに

本システムは，クラウドストレージへの同期プログラムにおけるファイル変更の同期工程にファイル識別の工程を介す事で，ファイルの暗号化を検知して同期を停止することを可能とした．これにより，暗号化型ランサムウェアによるファイルの暗号化からクラウドストレージ上のバックアップファイルを保護し，暗号化型ランサムウェア対策としての安全なバックアップ領域としてをクラウドストレージを活用可能でであることを示した．しかし，本実験のみでは，研究システムの論理的な有効性しか示すことができておらず，実際の暗号化型ランサムウェアが攻撃対象とする Windows 環境でシステムを実装し，暗号化型ランサムウェアに対して有効であるかの実験を行う必要がある．

今後の課題としては，本研究のファイル識別手法では，ファイルヘッダーを挿入するなど，マジックナンバーを改ざんすることで暗号化の検知を回避することが可能である点が挙げあられる．これを解決するため，精度の高いコンテンツベースのファイル識別手法の実現を検討する必要がある．加えて，制約の少ないユーザ行動の実現と，暗号化型ランサムウェアによる暗号を許してしまうディレクトリを排除するため，暗号化を行う主体の判別にプロセス監視を用いることを検討している．また，SSH File Transfer Protocol（SFTP）などを用いることでローカルネットワーク下のファイルサーバに対する非透過的なアクセスを実現し，より機密性の高いシステムの実現を目指す予定である．

参考文献

[1] IPA 独立行政法人 情報処理推進機構，https://www.ipa.go.jp/，2018 年 6 月確認．

[2] 情報セキュリティ10 大脅威 2017：IPA 独立行政法人 情報処理推進機構，https://www.ipa.go.jp/security/vuln/10threats2017.html，2018 年 6 月確認．

[3] 情報セキュリティ10 大脅威 2018：IPA 独立行政法人 情報処理推進機構，https://www.ipa.go.jp/security/vuln/10threats2018.html，2018 年 6 月確認．

[4] 警視庁,【緊急】ランサムウェアに要注意！ 警視庁，http://www.keishicho.metro.tokyo.jp/kurashi/cyber/joho/ransomware.html，2018 年 6 月確認．

[5] Symantec Corporation，インターネットセキュリティ脅威レポート第 22 号，株式会社シマンテック，2017．

[6] Mason McDaniel， M. Hossain Heydari，“Content Based File Type Detection Algorithms”，36th Hawaii International Conference on System Sciences (HICSS ’ 03) 0-7695-1874-5/03，2002．

[7] file man page，http://linuxcommand.org/lc3_man_pages/file1.html，2018 年 6 月確認．

[8] seb-m/pyinotify: Monitoring filesystems events with inotify on Linux.，https://github.com/seb-m/pyinotify，2018 年 6 月確認．