L-032

ファイルバックアップとアクセスモニタの統合による自己修復機構の設計

A Self-Repair System by Integrated File Backup and Access Monitoring

打田 悟志† 西山 裕之† 大和田 勇人†
Satoshi Uchita Hiroyuki Nishiyama Hayato Ohwada

1 はじめに

コンピュータの普及やストレージの大容量化により個人が扱うデータが増加する傾向にある．それに伴い，コンピュータウィルスに感染した際のデータの改ざんやユーザ自身の誤操作によるファイルの消失のリスクが高まっている．

そのようなリスクを回避するためには，バックアップソフトを使うことが有効である．バックアップソフトを利用することで，データ消失等の不慮の事態が起きた場合でも復元することができる．Apple の Mac OS X Leopard に搭載された自動バックアップ機能 Time Machine[1] は，ファイルを世代別にバックアップできる．保存された内容は Time Machine ブラウザで閲覧することができ，指定した日や時刻の状態に丸ごと戻したり特定のファイルを検索して呼び出したりすることができる．このような機能により，ファイルの改ざんや消失が起こった時間が分かればファイルを復元する事が可能となる．しかし既存のソフトでは，ウィルスによる改ざんと正規の更新の識別ができないため，未知のウィルス等による改ざんからデータを自動的に復元することは不可能である．

ウィルスへの対策においては，セキュリティソフトが最も有効である．しかし一般的なルールベースのセキュリティソフトではルールの更新が間に合わず，新種のウィルスが 1 週間以上放置される危険性もある．この間にコンピュータ内部のファイルが改ざんされた場合，ルールが適用されてもウィルスによる被害を完全に復元することは困難である．未知のウィルスに対して既知のウィルスコードを元に類似点を検索し亜種を探す研究 [2] 等がある．しかしこの方法でも完全に検知することはできない．

そこで本研究では，未知のウィルス等によるコンピュータ内のファイルの改ざんや，ユーザ自身の誤操作から自動的にファイルを復元できるシステムの実装を目的とする．既存のバックアップソフトとの最大の違いは，コンピュータ内部のファイルアクセスを監視し，ファイルを更新したプロセスを検出する点にある．ファイルの更新が行われるたびにプロセスを確認し，プロセス情報と共にバックアップを作成する．悪意のあるプロセスによる更新が行われた場合は，自動的に更新前のファイルに復元する．未知のウィルスにファイルを改ざんされた場合でも，ウィルスと認識した時点で改ざんされたファイルのみを自動的に改ざん前の状態に復元できる．ユーザ自身の誤操作によるファイルの消失が生じた際でも，従来のバックアップのようにファイルの特定や，加えて使用したプロセスが特定できれば以前の状態に復元することが可能である．

† 東京理科大学大学院理工学研究科, Tokyo University of Science

2 関連研究

悪意のあるプロセスによるファイルの改ざんから保護を行う手段として，アクセス制御が考えられる．SELinux(Security-Enhanced Linux)[3] はカーネルレベルでプロセスごとのアクセス制御が可能な，Linux 上で動作するシステムである．従来のシステムは root に権限が集中しているため，root パスワードが漏えいするとシステムに致命的な障害を及ぼす恐れがある．そのため SELinux は，強制アクセス制御 (MAC)，リソース（プロセス，ファイル，ソケット）ごとのアクセス制御 (TE)，root を含む全てのユーザに対するプロセス制御 (RBAC) によりセキュリティの高いシステムの構築を目的としている．ユーザによるセキュリティポリシーの設定を支援するためのツールに関する研究 [4] があるが，設定項目が膨大なため一般のユーザには設定が困難であり，未知のウィルス等への対応は不可能である．Windows 上のアクセス制御システムとしては，喜田らの研究 [5] が挙げられる．喜田らの研究はルールベースのアクセス制御により，P2P ソフトをセキュアに利用することを目的としている．コンピュータのファイルアクセス権を管理するエージェントを作成することにより，情報漏えいにつながるアクセスを遮断し，アクセス制御を行う．しかし，SELinux 同様未知のウィルス等への対応は不可能であり，一度改ざんによる漏えいが行われた場合，復元できないという問題点を持つ．

そこで本研究では，ファイルの改ざんが起きた場合でも改ざん前の状態に復元可能にするために，バックアップシステムを実装する．既存のバックアップソフトには，Windows の Backup and Restore Center[6] のように特定のイベントや時間に応じてシステム全体のバックアップを作成するソフト（図 1）と，Time Machine のようにファイルの更新に応じてファイル単位でバックアップを作成するリアルタイムバックアップ（図 2）の 2 種類がある．前者はバックアップの頻度が少ない場合，図 1 のようにバックアップデータの鮮度が失われてしまう．また，悪意のあるプロセスによるデータの改ざん等が起きた際，ユーザが認識できた場合のみ任意の復元ポイントに巻き戻す事ができる．しかし改ざんされたファイルを特定しなければ，ユーザが意図して更新したファイルを含め，システム全体を復元ポイントの状態まで戻す必要がある．後者は図 2 のようにファイルの更新が行われるたびに即時にバックアップを作成する．そのため，ユーザ自身の誤操作によるファイルの消失のように，発生した時間と対象ファイルを特定できるトラブルからの復元に有効である．しかしユーザが特定する事が困難なウィルスによる改ざん等には効果が薄い．

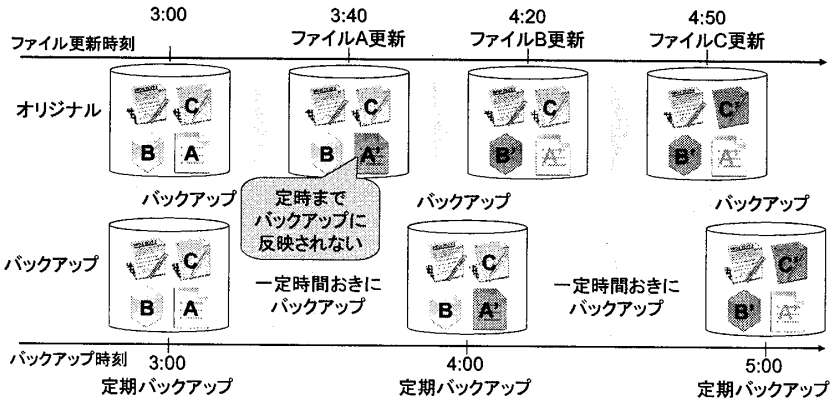

図 1: 従来のバックアップ — 一定時間おきにバックアップを作成するため，バックアップが最新とは限らない．

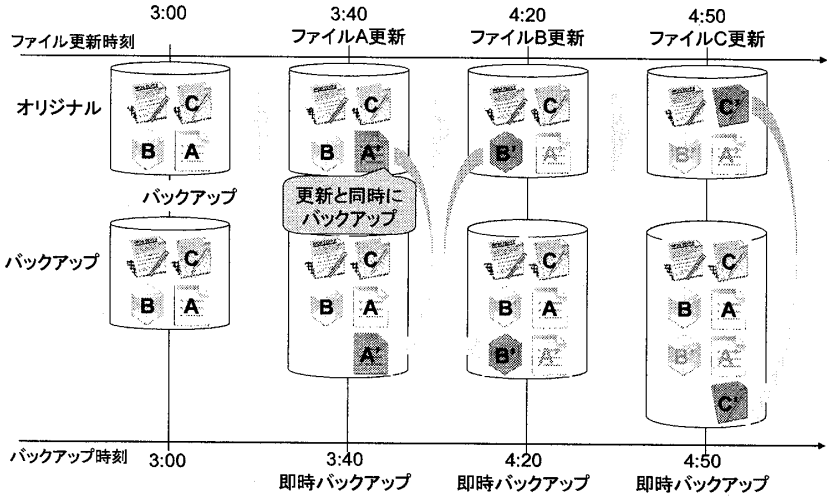

図 2: リアルタイムバックアップ — ファイルの更新と同時にバックアップを作成する．

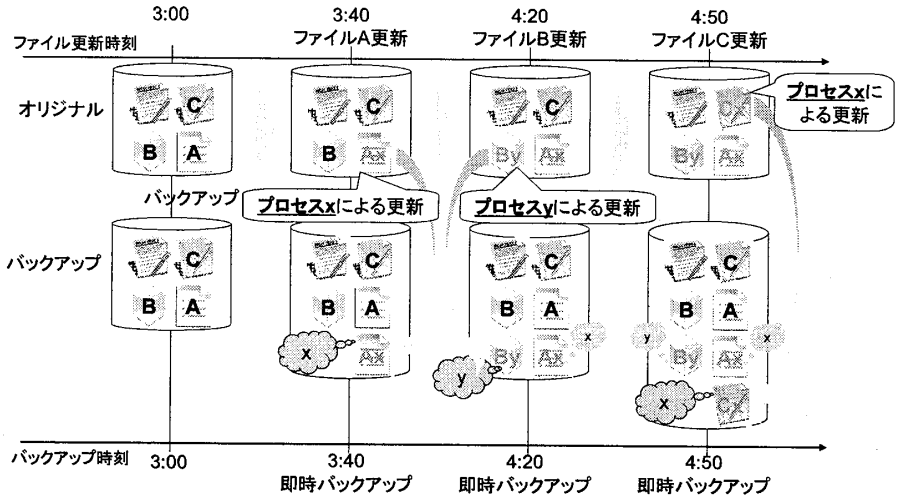

図 3: 本システムにおけるバックアップ — ファイルの更新と同時に，プロセス情報の保存と共にバックアップを作成する．

これらの問題を解決するために，本研究では通常のリアルタイムバックアップと改ざん検知の連携を行う．図 3 のようにリアルタイムバックアップに加え，ファイルを更新したプロセスの情報を取得する．そして悪意のあるプロセスによる更新は改ざんと判断し，任意のプロセスが悪意のあるプロセスと判断した段階で改ざん前の状態に復元することができる．

3 システム設計

本システムは，改ざんからの自動復元を行うためにリアルタイムバックアップと，ファイルの改ざん検知を統合する．ファイルの改ざんを検知する一般的な方法は，Tripwire[7] のように対象となるファイルのハッシュ値やファイルサイズ，更新時刻を事前に保存しておき，それらに変化がないか照合する手法である．この方法は，ファイルの更新頻度が低いアプリケーションの実行ファイルや，Web サーバ上の静的なコンテンツを保護するには有効である．しかしユーザが意図した正規の更新の場合でも改ざんと判断してしまう．そのため，日常的にユーザが更新するアプリケーションの設定ファイルを改ざんする，Antinny[8] のようなウィルスによる改ざんは検知できない．

そこで本システムにおける改ざん検知は，ファイルを更新したプロセスに応じて，通常の更新と悪意のある改ざんの判別を行う．ファイルを更新したプロセスの識別において，実行中のプロセスに割り振られるプロセス ID だけを用いる場合，同じプロセスでも実行するたびに異なる ID が割り振られたり，同じ ID の再利用が行われたりするため情報が不十分である．そのため本システムでは，プロセス ID に加えて，プロセスの実行ファイルから一意に求まるハッシュMD5*を用いる．各プロセスのハッシュは，悪意のあるプロセス以外はハッシュリスト，悪意のあるプロセスはブラックリストにより管理する．

ファイルの更新を検知した際，ブラックリストに登録されていなければバックアップを作成する．ブラックリストに登録されたプロセスによる更新の場合は，ファイルが改ざんされたと考えて，改ざん前の状態に復元する．未知のウィルス等によって改ざんされた場合，プロセスがウィルスと判明した段階で，バックアップログファイルを参照する．ログの参照により，どの段階で改ざんされたかが分かるため，この情報を元にウィルスによって改ざんされたファイルのみを改ざん直前の状態に復元できる．

以上より，本システムで必要となる機能は以下の通りである．

・実行中のプロセス群監視機能
　コンピュータ内で実行しているプロセスを確認し，悪意のあるプロセスかどうかを確認する機能．

　要求 1 実行中のプロセスの実行ファイルを取得できるようにする

　要求 2 プロセスのハッシュリスト，ブラックリストを管理する

　要求 3 悪意のあるプロセスの削除を行う

・ファイルアクセス監視機能
　プロセスによるファイルのアクセス（読み込み・書き込み・新規作成）を確認する機能．

　要求 1 プロセスのファイルへのアクセスを取得できるようにする

・ファイルバックアップ機能
　悪意のあるプロセスによって改ざんされたファイルはバックアップから自動復元を行い，それ以外のファイルは逐次バックアップを作成する機能

　要求 1 既知の悪意のあるプロセスによる更新は自動で復元する

　要求 2 上記以外のプロセスによる更新は逐次バックアップを作成する

　要求 3 未知のプロセスが，悪意のあるプロセスと判明した段階で，過去に遡って復元する

機能詳細を表 1 に示し，システム構成を図 4 に示す．「実行中のプロセス群監視機能」はコンピュータ内部のプロセスをモニタリングして，悪意のあるプロセスの情報はブラックリストに登録し，それ以外のプロセスはハッシュリストに登録する．「ファイルアクセス検知機能」はファイルのアクセスを動的にモニタリングして，ファイルの更新を検知すると「実行中のプロセス群監視機能」にプロセスの情報を問い合わせる．ブラックリストにない場合は，「ファイルバックアップ機能」により世代順に逐次バックアップを作成し，プロセス情報と共にログに保存する．ブラックリストにある場合改ざんされたと判断して，「プロセス削除機能」により実行されたプロセスの排除を行い，改ざんされたファイルのみを「自動復元機能」によりバックアップから復元する．

新たなプロセスがブラックリストに追加された場合ハッシュリストから削除し，過去のバックアップのログを参照して該当プロセスがないか照合する．ブラックリストに新たに追加されたプロセスによる，ファイルの更新が過去にされていた場合，改ざんされた可能性があると考えて，更新されたファイルのみを更新前の状態に復元する．この事後承諾システムによって，悪意のあるアプリケーションによって，重要なファイルが改ざんされても修復可能である．

また，本システムはファイルが更新される度に世代別にバックアップを管理しているため，ユーザの誤操作によりファイルが消失した場合でも，元の状態に復元することができる．

表 1: 本システムの機能一覧

機能大分類	機能中分類
実行中のプロセス群監視	実行中プロセスの把握
	リスト管理
	プロセス削除
ファイルアクセス監視	ファイルアクセス検知
	アクセス情報のフィルタリング
ファイルバックアップ	ファイルバックアップ
	自動復元

*RFC 1321 により定義されているメッセージダイジェスト．

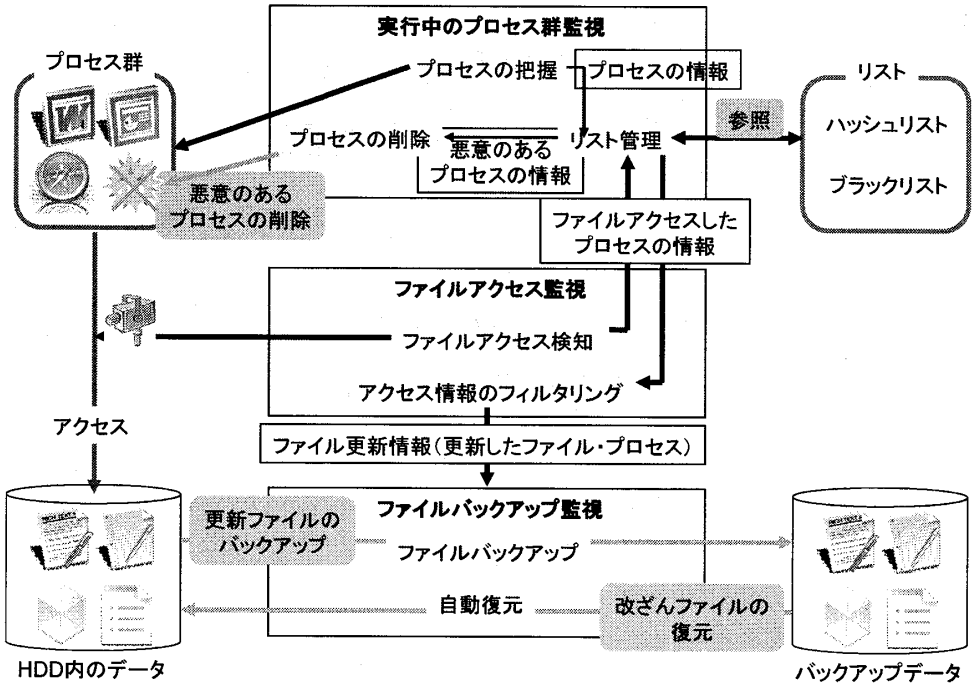

図 4: 本システムの構成

4 実装

4.1 実行中のプロセス群監視機能

本機能は，コンピュータ内部で実行中のプロセスを把握する．実行中の全てのプロセス ID を取得した後，各プロセスがロードしているモジュールから実行ファイルを特定してハッシュMD5 を生成する．ハッシュ生成後ブラックリストとハッシュリストの照合を行う．ハッシュがブラックリストに登録されている場合，該当プロセスを停止して実行ファイルを削除する．ハッシュリストに登録されている場合通常の起動を許可する．どちらのリストにも登録されていない場合は，生成したハッシュをハッシュリストに登録した上で起動を許可する．

本機能は，「ファイルアクセス監視機能」から問い合わせを受けた際，プロセスが各リストに登録されているか照合を行い返答する．問い合わせを受けたプロセスがブラックリストに登録されている場合，起動したプロセスを強制終了する．

4.2 ファイルアクセス監視機能

本機能は，コンピュータ内を監視することでファイルアクセスが行われた際の情報を取得する．取得する必要がある情報は以下の通りである．

- アクセスしたプロセス
- アクセスされたファイル
- アクセスの種類（書き込み・読み込み）

取得したファイルアクセスの情報に対して，アクセスの種類に応じてフィルタリングを行う．アクセスの種類がファイルの更新の場合，プロセスが登録されているリストを「実行中のプロセス監視機能」に問い合わせ，その結果に応じた動作を「ファイルバックアップ機能」が行う．本機能の実装にあたり以下に示す 3 種類の手法を検討した．

4.2.1 Win32API による実装

ファイルへのアクセスが検知された際，ファイル名，アクセスの種類等を検知するモジュールを ReadDirectoryChangesW 関数を用いて実装した．しかし Windows 標準の API 群では，アクセスされたファイル名や種類の検知はできるが，アクセスしたプロセスの検知ができなかった．

4.2.2 DLL Injection による実装

DLL Injection とは標準 DLL†の代わりに，独自に実装した DLL を読み込ませる手法である．この手法により，任意のプログラムを注入することで対象とするソフトウェアのファイルアクセスを取得できる．しかし，DLL Injection は対象が限定され汎用性が低く，対象となるソフトウェアに応じた実装が必要となるため有効とはいえない．

4.2.3 フィルタドライバによる実装

ファイルシステムフィルタドライバ（以下フィルタドライバ）とは，ソフトウェアを管理する OS とハードウェアを管理するデバイスドライバの間を流れるデータを監視・変更可能な中間ドライバである．この機能により，WindowsAPI では取得不可能な様々な情報を取得するこ

†ダイナミックリンクライブラリ：複数のプログラムから共通に利用できるように種々の機能をプログラム本体から分離して実装したモジュール

とができる．そこで，本システムでは Sysinternals による Filemon[9] で使用されるフィルタドライバを filem.sys 利用した．filem.sys はファイルの読み取り，書き込み，削除，更新等全てのファイルに対する操作と，その操作を行ったプロセスを検知可能である．本機能は起動時にフィルタドライバをサービスに登録し，表 2 に示すデータ型を参照する．フィルタドライバに対する制御は表 3 に示した関数を用いて行い，ファイルアクセスが生じた際情報を取得する．フィルタドライバから取得できる情報は以下の通りである．

- プロセス名，プロセス ID
- アクセスするファイルのパス
- アクセスの種類（書き込み・読み込み）
- アクセスが成功したか

フィルタドライバによって取得したアクセス情報からファイルの更新情報のみ抽出する．その後プロセス ID を用いて「実行中のプロセス群監視機能」に問い合わせる．ファイルに更新を与えたプロセスがリストに登録されていないか照合した後，その情報を「ファイルバックアップ機能」に受け渡す．

4.3 ファイルバックアップ機能

「ファイルアクセス監視機能」から，ファイルを更新したプロセスに関する情報を受け取ると，プロセスが登録されているリストに応じてファイルバックアップ機能は以下に記す 2 種類の処理を行う．

4.3.1 ハッシュリストに登録されている場合

ハッシュリストに登録されたプロセスによる更新が行われた場合,「自動バックアップ機能」のスレッドを生成する．本機能は，あらかじめ指定したバックアップ用ディレクトリに，オリジナルのディレクトリ構造を維持して更新ファイルのバックアップを作成する（図 3）．バックアップファイル名に，バックアップを作成した時刻を用いることでバックアップファイルを管理する．この際，更新を与えたプロセスの実行ファイルの MD5 と，バックアップを作成したファイル，時刻をログに書き込む．このログファイルを参照することで，特定のプロセスに更新されたファイルのみ復元するといった処理が可能となる．

表 2: DeviceIoControl のデータ型

データ型	パラメータ	意味
HANDLE	hDevice,	デバイス，ファイル，へのハンドル
DWORD	dwIoControlCode,	実行する動作の制御コード
LPVOID	lpInBuffer,	入力データを供給するバッファへのポインタ
DWORD	nInBufferSize,	入力バッファのバイト単位のサイズ
LPVOIDDWORD	lpOutBuffer,	出力データを受け取るバッファへのポインタ
DWORD	nOutBufferSize,	出力データのバイト単位のサイズ
LPDWORD	lpBytesReturned,	バイト数を受け取る変数へのポインタ
LPOVERLAPPED	lpOverlapped	非同期動作を表す構造体へのポインタ

4.3.2 ブラックリストに登録されている場合

ブラックリストに登録された悪意のあるプロセスによる更新が行われた場合，ファイルが改ざんされたと考え，「自動復元機能」のスレッドを生成する．本機能は，プロセスの強制修了と実行ファイルの削除を行い，改ざんされたファイルを削除した後，バックアップファイルを用いて改ざんされたファイルのみを改ざん前の状態に復元する．

4.3.3 ブラックリストに新たに追加する場合

ハッシュリストに登録していたプロセスが悪意のあるプロセスと判明した場合，ハッシュリストから削除してブラックリストに登録する．その後，バックアップログを照合する．該当プロセスによるファイルの更新が行われていた場合の処理を図 5 に示す．図 5 においてコンピュータ内に存在するファイルは図 3 のバックアップ終了後の状態とし，プロセス x が悪意のあるプロセスと判明したとする．この時，図 3 においてプロセス x に更新されたファイル Ax と Bx は改ざんされている危険性があるため削除し，改ざんされる直前のファイル A と B のみを復元する．この機能により，たとえ未知のウィルス等によってファイルが改ざんされても，ウィルスと判断した段階で改ざんされたファイルのみを復元できる．

5 実証実験

5.1 実験環境

本システムの有効性を示すため，実際のコンピュータウィルスを用いて実証実験を行った．実験に用いたウィルスは以下の通りである．

・Winny ウィルス

Antinny[8] は P2P 型ファイル共有ソフト Winny を媒介として感染するウィルスである．Antinny は起動すると，Winny の設定ファイル UpFolder.txt を不正に改ざんする．その後，一定時間おきにマイドキュメントやメールの情報を収集し，圧縮した上で Windows ネットワーク内に漏えいする．亜種によっては，特定のサーバに対して Ddos 攻撃をするものもある．

・qvimi.exe / mmvo0.dll

qvimi.exe / mmvo0.dll[10] は図 6 のように，USB メモリをはじめとしたリムーバブルメディアを経由して感染する．感染した場合，オンラインゲームのアカウント情報をインターネット上に漏えいする．USB メモリ内にウィルス本体と不正な autorun.inf ファイルを生成することで，コンピュータに接続した USB メモリを開く時（図 6②），Windows のオートラン機能によりウィルスが実行される．USB メモリ内のウィルスが実行されると，ウィルス本体をコンピュータに接続した全てのドライブ

表 3: dwIoControlCode で制御する動作

値	意味
IOCTL_FILEMON_GETSTATS	ファイル更新情報の取得
IOCTL_FILEMON_SETDRIVES	監視するドライブの設定
IOCTL_FILEMON_SETFILTER	フィルターの設定
IOCTL_FILEMON_STARTFILTER	フィルタリングの開始
IOCTL_FILEMON_STOPFILTER	フィルタリングの停止

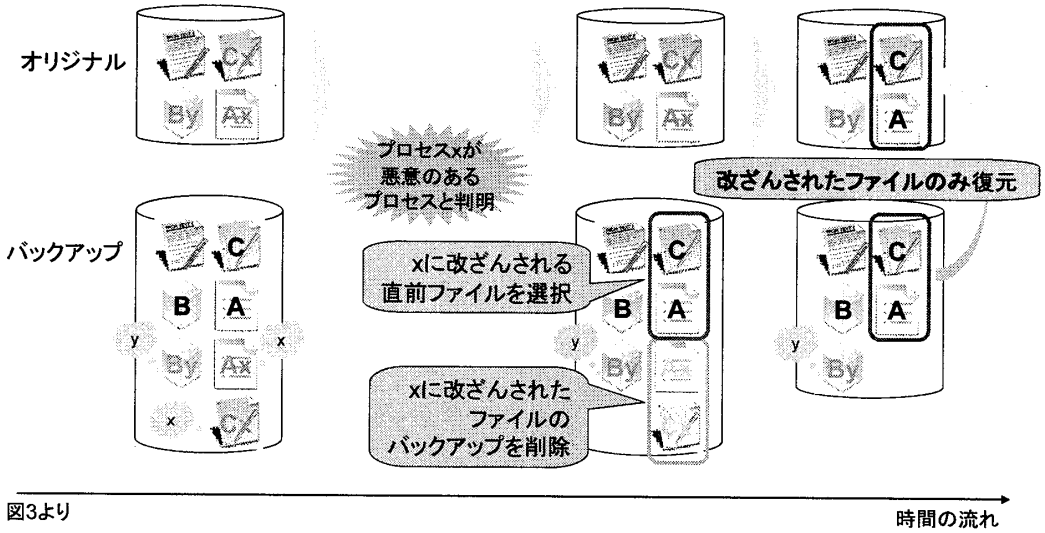

図 5: 本システムの復元処理 — プロセス x が悪意のあるプロセスと判明した際，過去に X により更新されたファイルのみを更新前の状態に復元する．

に autorun.inf ファイルと共に複製する．この際，レジストリを変更することで Windows 起動時に自動的に起動するようになり，新たに接続した別の USB メモリにも感染する（図 6A）．また，隠しファイルを強制的に非表示にすることで自身をユーザから隠蔽する．不正な Web サイトに接続し，自身と異なるウィルスをダウンロードする（図 6B）亜種も存在する．

実験は表 4 に記した環境でハッシュリストに 30 件のプロセス，ブラックリストに qvimi.exe のみ登録，Antinny は未登録の状態でおこなった．改ざんからの復元が正常に行われる事を確認するために，ブラックリストに登録したプロセスの起動を確認しても，ファイル更新が行われない限り削除しないように設定した．ハッシュリストに登録されたプロセスによる更新が行われた際，ブラックリストに登録されたプロセス qvimi.exe が実行された際，リストに登録されていない Antinny が実行した後ブラックリストに追加した際の動作を検証した．

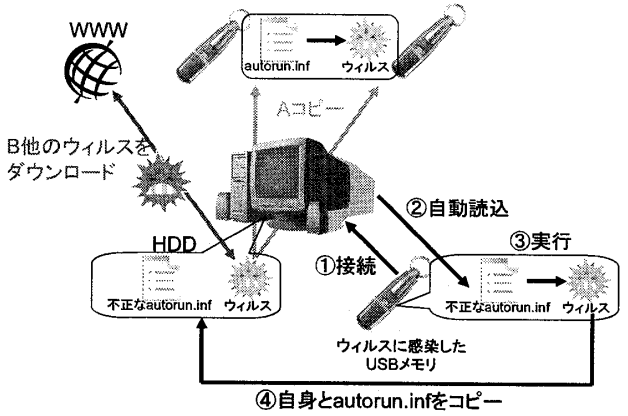

図 6: qvimi.exe の仕組み

表 4: 実験環境

OS	Windows XP Professional SP2
CPU	Core2Duo E6400 2.13GHz
メモリ	1.0GB
開発環境	Java, JDK 5.0 Update 16
	Visual C++ 2005 Express Edition
起動中のプロセス数	30〜60 件

5.2 ハッシュリストに登録されたプロセス

ハッシュリストに登録した Windows 標準のアプリケーションであるメモ帳を用いてテキストファイルの編集を行った．編集されたテキストファイルのバックアップが正常に作成されていることを確認した．

5.3 ブラックリストに登録されたプロセス

ブラックリストに登録したウィルス qvimi.exe を起動した際の動作を検証した．C ドライブ直下にコピーされた qvimi.exe とシステムファイル内に作成された mmvo0.exe, mmvo1.dll, juy.dll, IMM32.dll 等のウィルスファイルを削除することを確認した．

5.4 未知のプロセス

Winny ウィルス Antinny を未登録の状態で起動させた後，ブラックリストに指定した際の動作を検証した．Antinny の起動後ハッシュリストに登録し，改ざんした win.ini や UpFolder.txt 等のバックアップを作成した．その後，ブラックリストに指定することにより，改ざんされたファイルのみを改ざん前の状態に復元し，改ざんされたバックアップファイルも削除されることを確認した．

6 おわりに

本論文では，悪意のあるプロセスによるコンピュータ内のファイルの改ざんや，ユーザの誤操作によるファイルの消失からファイルを復元するシステムについて述べた．本システムではリアルタイムバックアップに，アクセスモニタを統合することで，ファイルを更新したプロセスを特定し，悪意のあるプロセスによる改ざんを検知した場合には改ざんされたファイルのみを，改ざん前の状態に戻すことが可能である．本研究では既存のウィルスを用いた実験により，改ざんされたファイルが正常に復元されることを証明した．

参考文献

[1] “Apple Time Machine” http://www.apple.com/macosx/features/timemachine.html

[2] Vyas Sekar et al. “A Multi-Resolution Approach for Worm Detection and Containment”, Proc. of the 2006 International Conference on Dependable Systems and Networks, pp.189-198, 2006.

[3] Peter Loscocco, Stephen Smalley, “Integrating Flexible Support for Security Policies into the Linux Operating System”, Proc. of the FREENIX Track of the 2001 USENIX Annual Technical Conference, pp.29-42, 2001.

[4] Wenjuan Xu et al. “Visualization based policy analysis: case study in SELinux”, Proc. of the 13th ACM Symposium on Access Control Models and Technologies, pp.165-174, 2008.

[5] 喜田 弘司 他 “ファイルアクセス制御エージェントの提案 : P2P 型ファイル共有システムのセキュアな利用を目指して”, 情報処理学会論文誌, Vol.48, No.1, pp. 200-212, 2007.

[6] “Microsoft Backup and Restore Center” http://www.microsoft.com/protect/yourself/data/backup.mspx

[7] Gene H. Kim and Eugene H. Spafford, “The Design and Implementation of Tripwire: A File System Integrity Checker”, Proc. of Conference on Computer and Communications Security, pp.18-29, 1994.

[8] “Symantec Antinny” http://www.symantec.com/region/jp/avcenter/venc/data/jp-w32.hllw.antinny.html

[9] Mark E. Russinovich, David A. Solomon, “Microsoft Windows Internals: Microsoft Windows Server 2003, Windows XP, and Windows 2000”, Microsoft Press, 2005.

[10] “Trend Micro WORM_ AUTORUN.CEZ” http://www.trendmicro.co.jp/Vinfo/virusencyclo/default5.asp?VName=WORM_AUTORUN.CEZ& VSect=P