解　説

コンピュータウイルスとワクチンとの戦い

遠藤　基／日本コンピュータセキュリティリサーチセンター
星澤裕二／（株）シマンテック

はじめに

コンピュータウイルス．私がこの不可思議な名称のプログラムと初めて出会ったのは1989年12月のクリスマスインジャパンウイルスが最初だった．

すでに説明するまでもなかろうが，コンピュータウイルス（以下，単にウイルス記述する）はれっきとした人為的なプログラムであり，自動的に自己複製する機能を持ち，広範囲に蔓延するというプログラムである．

この頃は，世界的視野でみると，1986年のパキスタンブレインウイルスを皮切りに，1987年アメリカでリーハイウイルス，イスラエルのエルサレムウイルス（13日の金曜日），ドイツのカスケードウイルス，ニュージーランドのストーンドウイルスなど，多くのウイルスがウイルス作者たちによってすでに産み出されていた頃である．

当初，その目的から，研究レベルの機能しか持ち合わせていなかったウイルスは，エルサレムウイルスあたりからペイロード（発病ルーチン）を持つようになり，パソコンの運用に実害をもたらすようになっていった．

エルサレムウイルスは，13日の金曜日に発病し，実行しようとしたプログラムファイルを削除してしまう．ユーザは利用したかったアプリケーションを次々と失ってしまうのである．海外で実際にこのウイルスの実態が初めて明らかになったときには，すでに広範囲にウイルス感染が広がった後であり，しかも，発病日が目前に迫っており，多くの人々が対策プログラムを時間との戦いの中で作っていた．

当時，日本国内で利用されていたパソコンの大部分はNECのPC-9800シリーズで，PC-AT互換機をターゲットとしたウイルスは正常に動作しないものが多かった．日本では誰もそうと気づかないうちにウイルスに対する防御ができていたのだ．

しかし，国内でもクリスマスインジャパンのようなウイルスが作られ，またBBSにアップロードされたプログラムがMS-DOS汎用のウイナーウイルスに感染していた事件など，日本でもウイルスの被害が現実のものとなろうとしていた．

ウイルスへの最初の対抗策

1990年の4月，日本では，警察庁が「コンピュータウイルス等不正プログラム対策指針」を，通産省では「コンピュータウイルス対策基準」をそれぞれ作成．通産省では「情報処理振興事業協会」をコンピュータウイルス被害届け出の公的機関として指定し，ウイルス被害の拡大，再発の防止に乗り出した．

さて，ウイルスの脅威を最も間近に感じている人々，それはBBSにアクセスする，ネットワーカたちであった．BBS上で配布されるフリーソフトを使うものにとっても作る側にとっても，ウイルスは厄介な問題であった．特にクリスマスインジャパンウイルスはそのバグ（仕様かもしれないが）のために，感染対象のプログラムを破壊してしまうという実害もあったため，これらウイルスの対策は必須だったのだ．

そういった経過があり，ネットワーカの一部では，ウイルス対策のユーティリティ，いわゆる「コンピュータウイルス用ワクチン」を作成する動きがあった．

ウイルスに感染したファイルやシステムを検出するという目的に対し，さまざまな試みが行われた．これには大きく分けて2種類があった．1つには，ウイルスコードの一部を登録しておき，ファイルを検索・読み出して比較して，一致すればウイルスと判定する，一種GREPプログラムにも似たものである（図-1）．

しかし，このタイプのワクチンは，ウイルスのサンプルを入手し，最適な比較用データを抽出しておく必要がある．ものがモノだけに，サンプルの入手はそう簡単にできることではなかった．

もう1つは，ウイルスそのものではなく，ウイルスがプログラムファイルに感染を行うに際して発生する，ファイルタイムスタンプ，サイズ，そしてファイル属性の変更などを検出するチェックユーティリティであった．ただし，これは問題がないわけではなかった．極端な話，DOSのCOPYコマンドですら属性変更などは行うからである．ウイルスだけに反応するように作るのは至難の業だった．

ウイルスの種類が増え，また高度化していくにつれ，個人でそれに対応するには限界があり，フリーソフトのワクチンソフトは，1つ，また1つと消えていった．

ブートセクタに隠れるウイルス

1992年3月，米国では3月6日に発病してハードディスク内容を消去するミケランジェロウイルスの警戒を呼びかけるニュースで騒然としていた．

その慌てぶりに同調して，日本国内のマスコミが

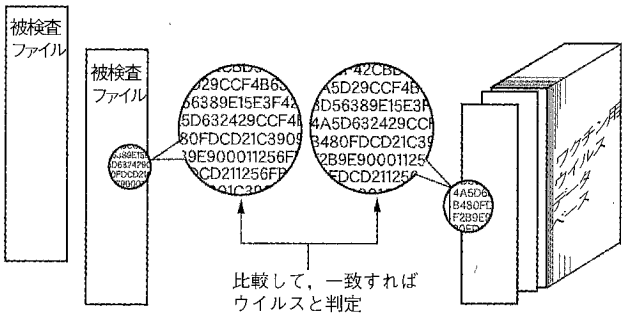

図-1　ワクチン動作の概念

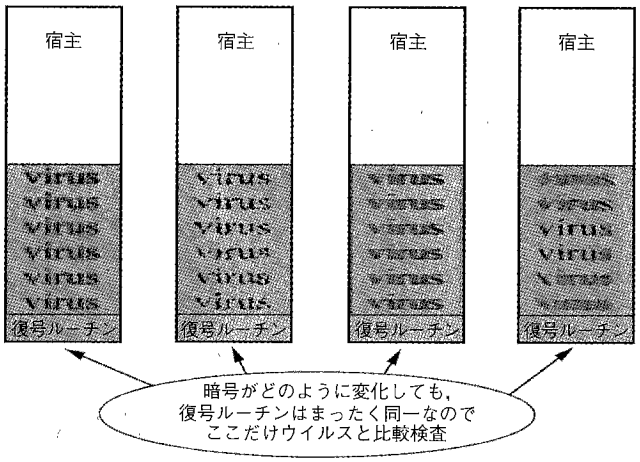

図-2　暗号化ウイルスをワクチンで発見する方法

「3月6日にデータを消す電子ウイルス」という記事を書いたとしても，それは無理からぬ話である．

しかし，日本国内では，ミケランジェロウイルスの被害は数件程度であった．ミケランジェロウイルスや，カンパナウイルスを解析し，ブートセクタウイルスがどのようなウイルスか知っている我々のようなアンチウイルス技術者には予想できたことだ．理由は，当時日本で主流だったパソコン，NEC製のPC-9800シリーズではブートセクタウイルスは絶対に動作しなかったからである．

ブートセクタウイルスが他の種類のウイルスと最も異なる点は，OSに依存しないという点である．これは，ブートセクタウイルスが動作するに際して，OSの機能をまったく利用しないということを意味する．このウイルスは，IBMパーソナルコンピュータ（AT互換機，もしくはDOS/V機と呼ぶ方が分かりやすいかもしれないが）のハードウェアアーキテクチャを利用して動作するのだ．

バイオスワークエリアのメモリ実装量値を必要な量だけ減らして書き換え，ディスクバイオスを横取りしてハードディスクやフロッピーディスクのブートセクタを書き換えることで感染する．パソコンがブートするときに最初に実行されるのはこのウイルスなのだ．

IBM-PCとNEC-PCはそのアーキテクチャもバイオスもまったく異なるので，感染どころか動作すらしない．1992年当時にはそれを知る者は少なかった．それどころか，日本国内にはAT互換機のアーキテクチャに関する資料など，一般には出回っていなかったから，そのほとんどを自力で解析していかねばならない．わずか512バイトのウイルスを解析するのに，信じられないような労力が必要だった．

ところで，ブートセクタウイルスはOSに依存しないと書いたが，これは実は恐ろしい意味を持つ．なぜなら，OSがいくら世代交代しようとも，IBM-PCアーキテクチャが生き残る限りブートセクタウイルスも根絶しないからである．10年も前に作られたウイルスが現在でも活動可能なのだ．それどころか，Windows NTがインストールされたパソコンをクラッシュさせるなど，思わぬ被害を見せている．

自分自身を暗号化するウイルス

知られているウイルスの中で，最も早く出現した暗号型ウイルスは，カスケード1701である．1987年にドイツで作成されたこのウイルスは，時刻情報から作成した暗号キーを使い，ウイルス自身のプログラムコードをXORで暗号化したものだった．

この暗号化は，フリーソフトのワクチン作者を駆逐するという目的にはかなり効果的だったようである．なぜなら，感染時刻によって何通りもの暗号パターンが発生し，単純なデータ比較でウイルスを発見することができなくなったからだ．

実際，海外のフリーソフトワクチンはこの時点でかなり数を減らしているという．このウイルスには，アンチウイルス研究者に対する挑戦という意味があったのだろう．

しかし，少し考えれば分かるように，このような暗号化が解析や対策の難易度を高めたかというと，実はそうでもない．暗号化されたプログラムコードはそのままでは実行できないため，実行に先立って復号化されなければならない．そしてその復号化プログラムは実行できる形でなければならないため，暗号化しておけないのだ．したがって，暗号型ウイルスの実行開始ルーチンは，通常のアプリケーションにはあり得ないような，怪しげな復号ルーチンとなっており，ウイルスであることがすぐに分かってしまう（図-2）．

結局のところ，ウイルス作者は暗号化によるメリットを，発病時の表示文字をプログラムダンプによるウイルス発見から守る手段として使うようになっていった．

単純な暗号化はウイルス隠蔽に大した効果はなかったが，この「暗号化」という機能は，数年後に思わぬ形でアンチウイルス研究者の前に立ちふさがることとなる．

姿を隠すウイルス

1990年代に入ると，ウイルスの作者の興味は，単にウイルスを作成するという点から，どのようにしてウイルスを長生きさせるかという点に移ってきた．

ウイルスというものの認知度が上がってくるとともに，ウイルスの認識方法や対策方法が少しずつ，浸透

してきた．例をあげれば，ファイルサイズの確認であるとか，デバッガによるプログラムダンプなどである．コンピュータユーザが少しずつ自衛手段を身につけつつあったのだ．

このような状況で，ウイルスには新たな機能が加わるようになってきた．

ファイルサイズがウイルス発見の鍵となるならば，ファイルサイズが増えたように見えなければいい．プログラムダンプしても表示されないようにすればいいというのである．

難しそうに感じられるこの機能も，メモリに常駐するタイプのウイルスならば，比較的簡単に実現できてしまう．ウイルスは，OSの（といっても大抵はMS-DOSなのだが）機能呼び出しを先取りすることによってファイルに自分自身のコピーを取りつける仕組みになっている．プログラム実行やファイルのオープン／クローズが呼び出されたタイミングで，ウイルスは自分自身を複製する．

さて，ファイルサイズやファイルの読み出しを行う際も，これら機能呼び出しは行われるので，このときに現実のファイルサイズから，ウイルス自身のサイズを減算したらどうなるか．システムによって提供されるファイルサイズは，感染したファイルのサイズからウイルスサイズだけ引いた分，すなわち感染前のファイルサイズとなって，サイズ増加分は隠される．ファイルが読み出された場合も，ウイルス自身が読み出される前に読み出しをキャンセルされると，ウイルス部分の読み出しや表示ができない．

このような機能を組み込まれると，ウイルスがメモリに常駐している限り，そのウイルスを検出することができない．このような自分自身を隠す機能は，レーダに映らない戦闘機になぞらえて，「ステルス機能」と呼ばれた．

この機能は，一般のパソコンユーザだけでなく，アンチウイルス研究者をも弱らせた．最大の問題点は，ステルス機能が動作している限り，ワクチンでさえもウイルスの検出ができないからである．そればかりか，ウイルスによっては検索したファイルすべてにウイルスが感染してしまうことも考えられ，ワクチンが逆に被害を大きくしかねないという点だった．

結局，ワクチンの動作は，起動時にパソコンのメモリ内に，このようなウイルスがいないかどうかチェックすることで解決した．メモリ内部にウイルスを検出した場合は，動作を中止してユーザにその旨を伝えたり，ウイルスにパッチを当てて，ウイルス機能をパススルーさせるようにしたのだ（**図-3**）．

ただパッチを当てるといえば簡単そうに聞こえるが，パソコンが動作中に，OSの中枢に絡みついたウイルスプログラムを切り落とすのだからかなり乱暴な話ではある．ちょっとでも間違えれば，あっという間にパソコンは暴走クラッシュだ．ワクチンを作るときには，機種やOSを変えて，何度もテストしたものである．

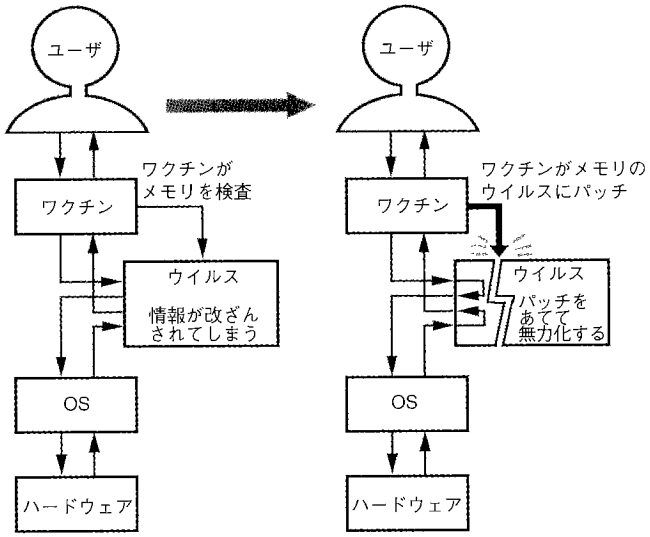

図-3　ステルスウイルスの処置方法

姿を変えるウイルス

1992年頃になると，ウイルス作者はグループ活動を始めるケースが多くなり，産み出されるウイルスもかなり高度なものが見られるようになった．その中で，最もアンチウイルス研究者を困惑させたのは，ミューテーションエンジンをはじめとする，ポリモーフィックウイルスの出現だろう．

ポリ＝多くの・モーフィック＝変形する，すなわち，多変形型ウイルスとは，その名の通り，1種類のウイルスがまったく異なった，しかし同一の機能を有する何億通りの変形を起こしつつ感染していくタイプのウイルスを指す．

ポリモーフィックウイルスを解析してみると，基本的には通常の暗号ウイルスとそう変わりはない．しかし，1つだけ大きく違っている点は，復号化ルーチンのための，プログラム自動作成機能が備わっている点である．

つまり，こういうことである．暗号化ウイルスの最も弱いポイントは，その復号化ルーチンが常に同じ形であるということである．そのためにこの部分を検索・同定されるとウイルスの種類が一発で特定されてしまう．ならば，暗号を解除するプログラムを，まったく新規に，自動的に作成してウイルスの復号ルーチンとしたらどうなるか．感染のたびに異なる形を持つウイルスとなり，ワクチンによる比較検査を無効化できるのではないか．おそらくミューテーションエンジン（MtE）の作者はそう考えたに違いない．

実際にポリモーフィックウイルスがどのように変形していくかといえば，排他，ローテート，シフト，加算，減算といった何種類もの暗号方法に加え，レジスタの割り振り変更，ジャンプ挿入，無意味な命令（＋－で0のような）を組み合わせて，機能的には等し

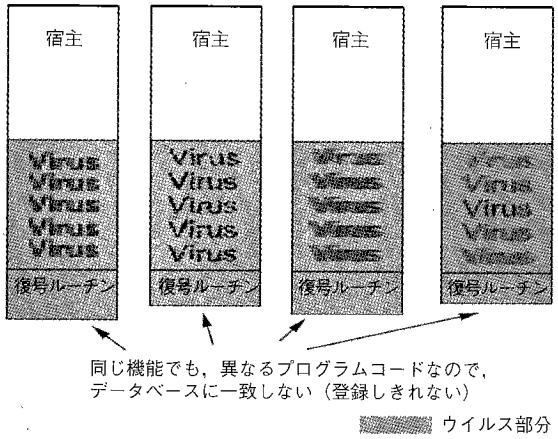

図-4（a）ポリモーフィックウイルスの検査（1）

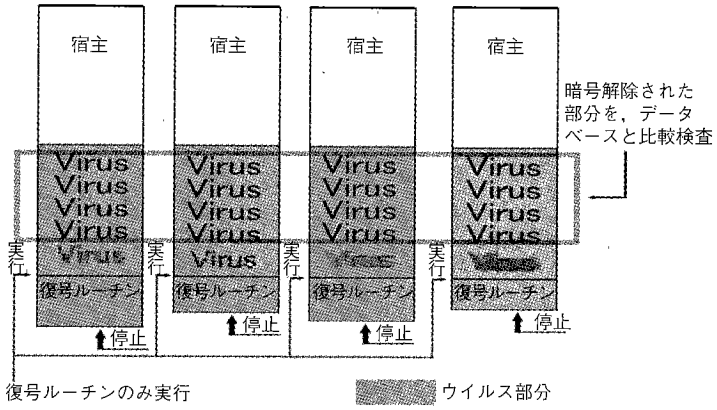

図-4（b）ポリモーフィックウイルスの検査（2）

い，しかしまったく見かけが異なる復号プログラムを自動的に作成するのだ．もちろん本体はその復号プログラムにあわせて暗号化されている（**図-4（a）**）．

実際に実験してみたが，1000「個」の感染サンプルを作成すると，恐ろしいことに，ちゃんと1000「種類」のサンプルができてしまった．変形の数だけ比較データを作ろうものならデータベースがどのような容量になるか見当もつかない．

だが，よく考えてみると，正しい復号ルーチンはウイルス自身が持っているのだから，それを流用して暗号解除すればよいのだった．つまり，ウイルスを途中まで実行させてやって，暗号が解けたところで止める，というコロンブスの卵的発想である（**図-4（b）**）．ただし，ウイルス全体が実行されてしまっては元も子もないので，どのように暗号解除だけをさせて止めるかが大問題となり，言うほど楽ではない．どのくらい大変かというと，この部分のアルゴリズムは，ワクチンメーカが今でもトップシークレットにしているということで，お分かりいただけると思う．

データに侵入するウイルス

1996年，マイクロソフトワードで動作するワードベーシックマクロプログラムの中に，妙なマクロプログラムが発見された．文書ファイルに含まれているマクロプログラムが，文書を開いたときに自動的に実行され，標準テンプレートというデフォルトテンプレートにコピーされるのだ．そして新しい文書ファイルを作成・編集すると，そのマクロプログラムも同時にコピーされて，次々と増えていくのだった．

プラットフォームはアプリケーションであったものの，これは間違いなくウイルスであった．すなわち，マクロウイルス．

実は，マクロウイルスの発生は予見されていた．学術的に，である．しかし理論上のものであり，あまり重要視されていなかった．ところが，最初のマクロウイルス（WM/Concept）が発見されるや否や，類似種が次々と作られ，あっという間にDOSやブートセクタウイルスの被害数を追い抜いて，最も警戒を要するウイルスとなってしまったのだ．

困ったことに，マクロウイルスの特徴は，作りやすく，対策しにくいということだ．ワードやエクセルのマクロはソース状態のままデータファイルの中に取り込まれ，そのまま流通する．したがって，マクロウイルスに感染したファイルを手に入れた者は，そのままウイルスのソースコードを入手したことになる．しかもウイルスの改造／開発は，ワードやエクセル自体でできてしまう．

逆にこのウイルスを文書ファイルから直接に発見しようとすると，マイクロソフト社独自の謎のデータ構造に悩まされることになる．このデータファイルは，ファイルの中にまたファイルが書き込まれた状態になっており，しかもOLE2フォーマットによって，さらに複雑なファイル構造となっている．ウイルスを消去しようとうかつに編集すると，データそのものを破壊しかねない．

アンチウイルス研究者はなんとかデータ構造を解析してマクロウイルスの発見・駆除を行ってはいるが，決して楽な仕事ではない．改造種がめちゃくちゃに多く，日々増えていく．これに次々と対応していかなければならないのだ．

さて，日本国内でもマクロウイルスの被害は相当数にのぼるが，実はこれでもかなり少ないほうである．なぜなら，世界中で発見されているマクロウイルスのごく一部（おそらく1割にも満たない）しか，日本語環境で感染しないからである．

ブートセクタウイルスのときのように，日英の非互換性に助けられて被害が少ないだけなので，もし，日本語環境で感染するマクロウイルスが少し増えれば（あるいは，日英のワード／エクセルの仕様の差がもっと少なくなれば）この数倍から十数倍のウイルス被害が発生するであろう．

現状の問題点と今後のワクチン技術

現在，月平均200から300種類の新種ウイルスが発見されている．インターネットや電子メールが普及し，ウイルスが蔓延する速度も増すばかりである．従来のウイルス検出技術（ウイルスの指紋が登録されている

番号	バイト列	ニモニック	意味
#1	B8 4C 00 CD 21	MOV AX, 004C INT 21	プログラム終了

#2	BB 4C 00 89 D8 CD 21	MOV BX, 004C
		MOV AX, BX
		INT 21
			プログラム終了

#3	31 CD 05 4C 00 CD 21	XOR AX, AX ADD AX, 004C INT 21	プログラム終了

バイト列，ニモニックがまったく異なっても，プログラムとして同じ意味を持つパターンを番号をふって登録しておく．

図-5（a）ヒューリスティック法の概念（1）

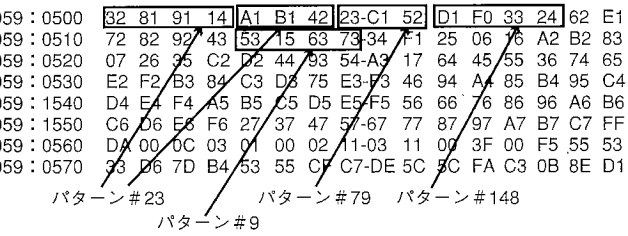

図-5（b）ヒューリスティック法の概念（2）

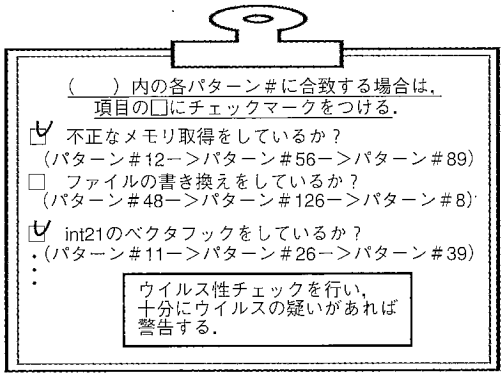

図-5（c）ヒューリスティック法の概念（3）

データベースと比較し，発見する方法）では，新しいウイルスが発見されるたびに，ウイルスの指紋がデータベースに登録されなければならない．したがって，すべての新種ウイルスを見つけるためには，このデータベースを頻繁に更新する必要がある．これは，面倒な更新作業を余儀なくされるユーザだけでなく，常に最新のウイルスデータベースを提供し続けなければならないワクチンメーカにとっても非常に負担である．

また，データベースと比較するこの方法は，すでに識別されている既知ウイルスにのみ有効であり，当然ながら，未知のウイルスを発見することはできない．つまり，頻繁にデータベースの更新作業を行っていても，ウイルス被害の脅威は依然として消えない．こうした背景の中で，未知ウイルス検出技術は生まれた．

未知のウイルスを検出するために，ヒューリスティック（heuristic）と呼ばれる技法が用いられる．ヒューリスティックとは，「試行錯誤により自分で発見させる」という意味である．つまり，単なる比較法によりウイルスを見つけるのではなく，プログラムコードの中からウイルスに必要な行動パターンを収集し，ワクチンソフト自身がウイルスかどうか判断する方法である．ヒューリスティック技術は，この行動パターンの収集方法の違いにより，スタティックヒューリスティックとダイナミックヒューリスティックに大別される．

スタティックヒューリスティックは，一般にウイルスが複製したりするときに見られるプログラムコード（バイト列）をデータベースに登録する（**図-5（a）**）．プログラムコードとこのデータベースを比較し，行動パターンを収集する（**図-5（b）**）．いくつかのウイルスらしい行動パターンに該当する場合，ウイルスと認識する（**図-5（c）**）．スタティックヒューリスティック技術の問題点は，同じ仕様のプログラムを作成するためにさまざまなプログラム方法が考えられるため，行動パターン，つまり，プログラムコードを数多く登録しなければならないことである．すべてのプログラムコードを登録することは不可能である．

ダイナミックヒューリスティックでは，この問題を解決するため，プログラムを仮想メモリ空間で実際に実行（エミュレーション）し，ウイルスらしい行動パターンを収集する．計算結果は同じでも，計算方法が異なる場合，スタティックヒューリスティックでは，すべての計算方法を登録するのに対して，ダイナミックヒューリスティックでは，計算結果だけを登録すれば済むのである．

ウイルスとワクチンソフトは，泥棒と警察にたとえて説明されるが，スタティックヒューリスティックによる未知ウイルス検出技術は，犯人の人相や，犯行前の行動だけで逮捕してしまうため，ウイルスではないファイルを誤認逮捕することが少なくない．しかし，ダイナミックヒューリスティックは，実際の犯行現場を目撃し，現行犯で逮捕するため，誤認せず確実にウイルスを見つけることができる．

ワクチンソフトのウイルス検出技術は，新種ウイルスの出現とともに進化してきた．しかし，ダイナミックヒューリスティックによる未知ウイルス検出技術が開発されたことによって，ウイルス技術を超える結果となった．

おわりに

コンピュータウイルス対策の最も難しい部分は，コンピュータウイルスを作っているのが，人間であるということだろう．フールプルーフ的なセキュリティは比較的簡単で解決可能ではあるが，相手が人間である場合は，セキュリティを積極的に破ろうとしてくるうえに，ワクチンの裏をかいてくることが十分に考えられるからだ．

最近はウイルスの発病も，フラッシュROMの消去や，ネットワークを使ったセキュリティ突破など悪質・高度化しているのでその被害は無視できない．

また，マイクロソフトワードやエクセルの例のように，新しいプラットフォームを利用したウイルスや有害なソフトウェアを新規に制作してくるかもしれない．APPLEスクリプトやWindowsスクリプト，JAVAも安心ではないということだ．

つまるところ，これはコンピュータ上の見えない戦いに他ならないのだ．先に諦めた方の負けである．

（平成10年10月30日受付）