

ヒューリスティック手法詳説： シマンテックの Bloodhound 技術

目次

ヒューリスティック手法の理解 : シマンテックの Bloodhound 技術 -----	1
-----	1
はじめに -----	1
麻ひもセールスマンの Englebert とヒューリスティック手法 -----	1
ウイルス検知 : 不正確な科学の世界 -----	1
ヒューリスティック手法のロジック -----	2
ヒューリスティック ・ スキャナの動作様式 -----	4
スタティック / ダイナミック ・ ヒューリスティック ・ スキャナ -----	6
Bloodhound: 次世代のヒューリスティック手法 -----	10
Bloodhound: 新出および未知の実行ファイル感染型ウイルスを最高80%の確率で検知 -----	10
Bloodhound-Macro: 新種および未知のマクロウイルスの90%以上を検知、修復 -----	12
まとめ -----	16
シマンテックについて -----	16

* このホワイトペーパーには、セキュリティ情報ページに記載されていた情報が一部含まれています。
また、このホワイトペーパーは1998年5月に作成されたものを翻訳いたしました。

ヒューリスティック手法の理解 : シマンテックのBloodhound 技術

はじめに

“ heuristic(ヒューリスティック)”という言葉の語源は、heuristiskeinというギリシャ語で、元々の意味は「発見する」という意味でした。今日、この言葉はほとんどの場合、複雑な問題を素早く解決するためには効果的であり、速度改善にはなるが、完璧とは言えないアルゴリズムを説明するときのコンピュータ用語として使われています。以下は、この概念をわかりやすく説明するために、コンピュータ・サイエンスの専門家達が多くのヒューリスティック・ソリューションの発見に至った難解な問題を説明するとき定番の例え話として使用されている、出張セールスマンの問題を例に説明します。

麻ひもセールスマンの Englebert とヒューリスティック手法

麻ひもメーカーに勤務する Englebert というセールスマンがいたとします。彼には、世界中のさまざまな都市を1箇所につき1回ずつ訪問して営業する任務が与えられています。彼は、飛行機に乗っている時間をできるだけ短くできるように移動時間を最低限に抑えたいと考えています。

Englebert がロサンゼルス、オレゴン、パリ、ローマに出張する必要がある場合、最初にロサンゼルス、次にオレゴン、パリ、ローマの順に移動すれば、移動時間を最低限に抑えることができます。他方、ロサンゼルスからローマへ行き、オレゴンに戻ってパリへ行った場合、飛行機に乗っている時間は前述の場合に比べ非常に長いものになってしまいます。この2つの経路を比較した場合は明らかに最初の方法が好ましいということになります。Englebert の移動行程を計画するのは一見、取るに足らないことのように見えます。

しかし、もし Englebert が500都市を巡らないといけないとしたらどうでしょうか？このような問題に最も効果的な方法を見つけるには、現時点で最速のコンピュータを駆使しても膨大な年月を要することでしょう。コンピュータによる計算にこのような膨大な時間をかけて計算させるのは非現実的なことから、コンピュータの専門家達は、このような問題をより短時間で解決するのに効果的なヒューリスティック・アルゴリズムを創出しました。ヒューリスティック・アルゴリズムは、与えられた問題の解決方法に関する特定の仮定を立てます。これらの仮定を使用することで、コンピュータによる演算処理時間を削減する一方で比較的良好な結果が得られます。ただし、この方法を使って生成される結果は、その本質的な性質により、それが常に完璧な解決方法となり得るとは言い切れません。どのようなタイムプログラム（または人間）でも、ミスを犯すことはあります。多くの場合、ヒューリスティック・アルゴリズムは、与えられた問題に対する完璧な解答に非常に近い解答を出しますが、常にそうであるという保証はありません。Englebert は、移動時間を最高で10万マイルまで短縮可能なことを知っていたら、そしてさらに、最適な旅程の算出に数百年待たなくても良いとしたら、実際の移動距離が10万55マイルになったとしても十分に満足するに違いありません。

コンピュータ科学の理論と麻ひものセールスマンと何の関係があるのか疑問に思われる方もいることでしょう。弊社では、ウイルス対策分野において、弊社独自の解決が困難な問題セットを持っています。これらの問題の幾つかは、前述の出張セールスマンの問題よりもさらに難解です。事実、ウイルス対策研究者達が取り組んでいる問題には、解決不可能とみなされているものも存在します。言い換えると、ある問題に対する確実に正確かつ最高の解答を限られた時間内で得ることは不可能です。ヒューリスティック・アルゴリズムは、このような問題に対する唯一の解決方法と言えます。

ウイルス検知 : 不正確な科学の世界

あるコンピュータ・プログラムがウイルスかどうかを判定する作業は、解決不可能な問題です。これまでに確認されているウイルス、そして、今後作成されるウイルス全てを検出し、特定のプログラムがウイルスに感染しているかどうかをユーザに100%正確に知らせる能力を備えたウイルス検出プログラムを作成することはまず不可能です。もしこのような問題の解決が可能だったら、ウイルス駆除ソ

フトメーカーやMIS（管理情報システム）の管理者は皆、歓喜に酔いしれるでしょう。また、そうなれば、開発に経費がかさみ、配布が困難な月間ウイルス最新情報も、わずらわしいウイルス誤認情報も終焉を迎えることになるでしょう。

しかし現実には、残念ながら、ウイルスと非ウイルスの判別における問題はまだ解決に至っていません。このような現状を踏まえ、ウイルス研究家は、膨大な数に昇るコンピュータウイルスの検出に役立つ革新的なヒューリスティック手法を生み出してきました。その中でも最も有名な技術がシグニチャ・スキャンと呼ばれる技術です。シグニチャ・スキャンがヒューリスティック技術であると考えている人は数多くいませんが、実際にはヒューリスティック技術の一つです。

シグニチャ・スキャン機能を持つウイルス駆除ソフトは、シグニチャのデータベースを維持し、ユーザのコンピュータ上に存在するすべてのプログラム内で前述のデータベースに登録されているシグニチャを探します。シグニチャとは、特定のウイルスの本体から抽出される短いバイト配列のことで、ウイルス駆除ソフトは、検出可能なウイルスごとに異なるシグニチャ情報を持っています。

通常、この連続したバイト情報は、そのウイルス固有であり、また、ウイルスのロジックを構成するバイトセット全体からみた割合は非常に小さいという特性があります。前者の特性により、ウイルス駆除ソフトが感染していないプログラムを感染していると誤認してしまう可能性は低いものになっています。後者の特性は、ウイルス駆除ソフトにとって不可欠な条件となります。後者の特性がなければ、（毎月更新されるウイルス定義ファイルで検出される全ウイルスの完全コピーを含める必要が生じるため）ウイルス駆除ソフトがウイルス検出に使用するデータファイルは数百メガバイトにもなりかねない非常に巨大なものになっていたでしょう。

シグニチャ・スキャン技術を使用するウイルス駆除ソフトには、あるプログラムに対して多数存在するシグニチャのうちいずれか1つが含まれているかどうかを識別する能力はありますが、そのプログラムが実際に、それに関連したウイルスに感染しているかどうかをユーザに確認することはできません。通常、ウイルス駆除ソフトが前述のような評価を下した場合、ユーザの多くは、ウイルス駆除ソフトの推測に頼ります。しかし、ウイルス駆除ソフトがウイルスに感染していると判断したプログラムが実際には、ウイルスに似たランダムデータを偶然含んでいたり、ウイルスシグニチャとたまたま同じバイト配列を含む正規のプログラムコマンドが含まれることもあり得ます。つまり、シグニチャ・スキャナがウイルスとして識別したプログラムが本当にウイルスである確率は非常に高いものの、それが絶対であるとは言い切ることはできません。シグニチャ・スキャナがヒューリスティックアルゴリズムの1つといえるのは、このことによるものです。

シグニチャ・スキャンは現在利用可能なウイルス駆除ソフトで今日最も広く採用されている技術です。シグニチャ・スキャンには幾つかの短所があるものの、ウイルス駆除プログラムがデータファイル内にシグニチャ情報を持っているウイルスを識別するためには非常に効果的な方法です。残念ながら、ウイルス作成者達はますます用心深くなり、新たなウイルスを次々と作り続けています。ほとんどの場合、ウイルス検出データファイルに古いウイルスの検出用として含まれていたシグニチャは、新たなウイルスの検出には役に立ちません。その上、インターネットの普及にともない、最近出没している新種ウイルスは、ほとんどどんなユーザにでもわずか数分でアクセスできてしまう可能性があります。シグニチャ、および、時間もコストもかかるウイルス解析プロセスを経ずにウイルスを検出可能なウイルス検知・駆除技術の必要性が出てきたのは、このような要因によるものでした。

ヒューリスティック手法のロジック

前述で概説した通り、今日一般的に使用されているウイルス対策製品の多くは、何らかの形のヒューリスティック・ロジックを採用しています。しかし、ウイルス駆除業界では常に、「ヒューリスティック手法」という単語（名詞）は、特定タイプのウイルス検知技術に言及するときに使用されています。具体的に説明すると、ヒューリスティクスは、ウイルス対策研究者が、ウイルスプログラムのシグニ

チャを探す代わりに、構造、動作、その他の属性を解析することによって、ウイルスを検知するウイルス検知・駆除プログラムのことを説明する目的で作出した用語です。次項以降では、「ヒューリスティック手法」または「ヒューリスティック・スキャナ」は、この技術について説明するための用語として記述されています。

まず、犯罪者を捕まえる方法を例にとって説明しましょう。犯罪者を捕まえる方法には2通りあります。警官はまず、犯行現場へ行き、指紋が残っていないか調べます。指紋が見つかった場合、採取した指紋を警察署に持って帰り、前科がある犯罪者の指紋データベースを調べて照合します。入手した指紋と一致する指紋データが見つければ、その犯罪者の記録を調べて探し出し、逮捕することが可能です。

しかし現実には、犯罪者の多くは初犯です。前科がない人の指紋は署の指紋データベースには記録されていません。その場合は、犯罪者の捜査には別の方法がとられます。警官は、接触した人ひとりひとりを観察・評価し、容疑者を推測します。防弾チョッキを着用し、ショットガンを携帯している人が警官の前を通りかかった場合、警官はその人を不審者と判断し、逮捕します。逆に、元気な赤ちゃんを抱いて歩いているベビーシッターを見かけても何も不審に思わないでしょう。もちろん、無実の人を誤って逮捕してしまったり、真犯人を取り逃がしてしまうこともあります。優れた警官であれば有罪犯罪者の逮捕成功率は高いと思われます。

ヒューリスティック手法を採用しているウイルス駆除プログラムは、上記の例と同様のアプローチを使ってコンピュータ・ウイルスを検出します。ヒューリスティック手法を使うウイルス駆除プログラムがある実行ファイルをスキャンするとき、まず、その実行ファイル全体の構造、プログラミング・ロジックあるいは命令セット、ファイル内に含まれているすべてのデータ、その他の幾つかの属性を調べます。次に、その実行ファイルがウイルスに感染している可能性を査定します。警官の場合と同様に、ヒューリスティック手法の場合も、ウイルスを検出し損ねたり、ウイルスに似た動作を見落としたり、実際には正規のプログラムをウイルス感染ファイルとして誤認してしまったりすることがあります。

ウイルス対策業界の専門家によれば、今日最先端のヒューリスティック・スキャナによる新出または未知のウイルス検知率は70%～80%とされています。問題の難解さからみて、70%～80%という確率は十分妥当であると思われます。ヒューリスティック・スキャン技術のほとんどは上記と同様のウイルス検知率を維持していますが、クリーンなプログラムに対する誤認傾向は一定していません。現在一般的なウイルス駆除製品の中には、しょっちゅうクリーンなプログラムをウイルスと誤認してしまう傾向があるヒューリスティック・スキャナを採用していることが原因で、ヒューリスティック手法に対するいわれのない不評を導いているものがあります。そのような誤認情報を通知する一部のヒューリスティック・スキャナ採用ウイルス駆除プログラムが抱える問題点については、後ほど詳しく説明します。

ヒューリスティック手法の大きな長所は、ウイルスが起動してユーザのコンピュータに感染してしまう機会を獲得する前の段階で、ファイルやブートレコード内に潜むウイルスの検知が可能なることにあります。従来のシグニチャ・スキャナと同様に、ユーザは新規のプログラムやフロッピーやCDを使用する前に、オン・デマンド(手動)でヒューリスティック・スキャンを開始することができます。また、ヒューリスティック・スキャン技術を採用したオン・アクセス式のウイルス駆除プログラムを実行している場合には、インターネットからダウンロードしたり、電子メールの添付ファイルを保存する際、新規のウイルスでも高い確率で検出することができます。

動作ブロックや整合性チェックなど、その他のウイルス検知技術の場合、ウイルスが実際に標的のコンピュータ上で起動したり、不審あるいは有害と思われる動作を実行しない限り、ウイルスを検知し、その動作を停止させることはできない仕様になっています。ウイルスが標的となったコンピュータに危害を与える機会を得る前の段階で、ウイルスの実行を停止させる能力に関しては、ヒューリスティック・スキャンもシグニチャ・スキャンも同等に持っています。

これまでにヒューリスティック手法に対して行われてきた調査研究は、DOS ウイルスの検知に焦点を当てたものがほとんどです。したがって、本書では、ウイルス対策研究者達が DOS ウイルスに対するヒューリスティック手法の問題にどのように取り組んできたかを分析することによって、ヒューリスティック手法の基調をなす概念について説明します。ただし、以下で述べている技術はブートウイルスの検知にも使用されていますし、後述で説明しますが、企業に大きな危害を加えている新世代のマクロウイルスの検知にさえも応用することが可能です。

ヒューリスティック・スキャナの動作様式

ウイルス対策研究者達が調査してきたヒューリスティック・スキャン・アーキテクチャには、スタティック方式とダイナミック方式の2種類あります。この2つの方式の主な相違点は、ヒューリスティック・スキャナがウイルスのような動作を探すときにCPU エミュレーションを使用するかどうかです。ここでは、相違点は無視し、両方のアーキテクチャに共通の属性について説明します。

ヒューリスティック・スキャナは通常、実行形式ファイルにウイルス・スキャンを行う際、2段階の操作を行います。第一段階で行う操作は、スキャン対象のプログラムが実行可能な活動のカタログ(目録)を作成することです。ヒューリスティック・スキャナはまず、ウイルスが実行形式ファイルに感染する場合に自分自身を実行形式ファイルに付着させる確率が高い場所を判断します。実行形式ファイルの中には数百キロバイト、あるいは数メガバイトにのぼるものがあるため、これは非常に重要なステップとなります。サイズが巨大なプログラムに綿密なヒューリスティック分析を行った場合、処理速度が非常に遅くなります。しかし DOS ベースのコンピュータ・ウイルスのほとんどは、わずか数千キロバイトしかないため、効率よく設計されたヒューリスティック・スキャナは、詳細に調べる領域を明確に限定することが可能になっています。多くの場合、その領域はファイルの先頭および末尾の数千キロバイト分の領域に限定されます。

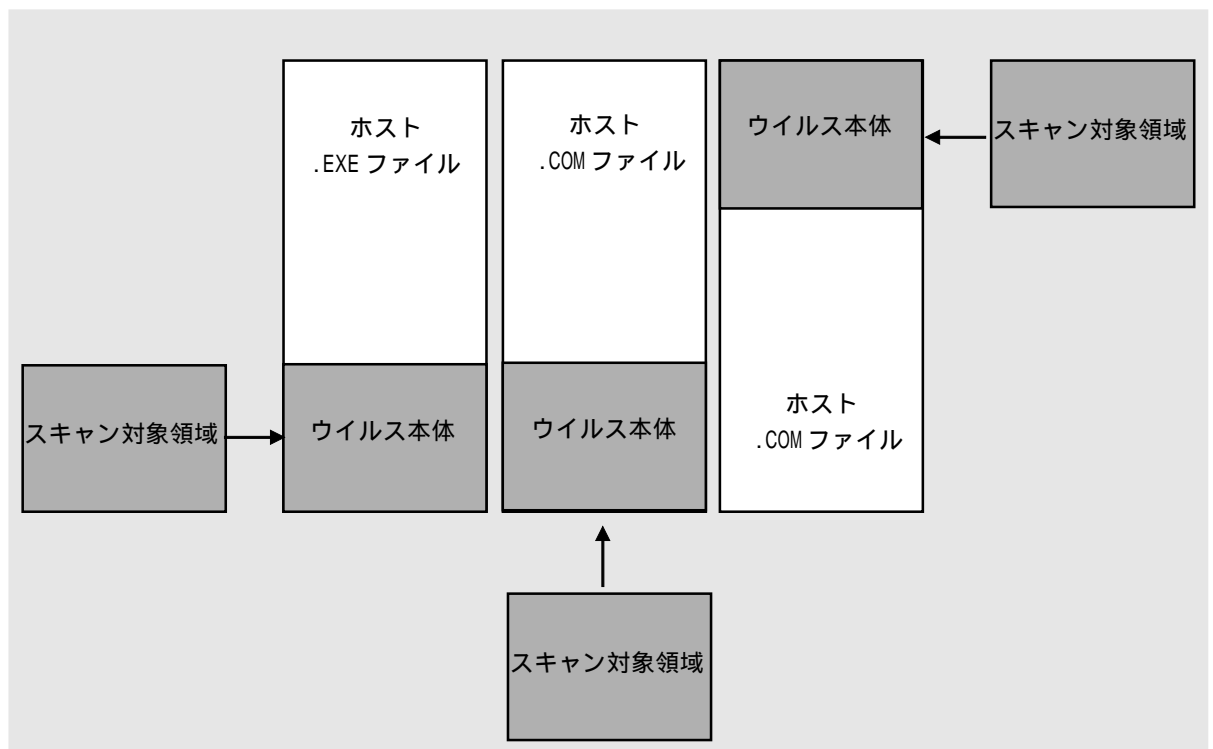


図 1. DOS ファイルウイルスは通常、DOS 形式の .EXE ファイルの末尾に自分自身を付着して寄生します。DOS 形式の .COM ファイルに感染する場合は、そのファイルの先頭または末尾に自分自身を付着させます。その他の感染形態をとることもありますが、さほど一般的ではありません。

ヒューリスティック・スキャナは、ウイルス感染していそうな領域を識別すると、そのコンピュータ・インストラクションで実行可能な動作を判断するために、その領域に含まれているプログラムロジックを解析します。コンピュータ・プログラムの書き方には非常に多くの方法があるため、この操作自身、非常に難解です。

次の2つのシーケンスのインストラクションを例にとって説明しましょう。

例 1	
マシン語のバイトコード(16進数)	インストラクション・コード
B8 00 4C	MOVAX, 4C00
CD 21	INT 21

例 2	
マシン語のバイトコード(16進数)	インストラクション・コード
B4 3C	MOV AH, 3C
BB 0000	MOV BX, 0000
88 D8	MOV AL, BL
80 C410	ADD AH, 10
8E C3	MOV ES, BX
9C	PUSHF
26	ES:
FF1E 84 00	CALL FAR [0084]

図 2.

上記は、パソコン上でプログラムを終了し、DOSプロンプトに制御を返すプログラムロジックの記述方法の例です。左側は、16進数で表されたマシン語のバイトコードで、コンピュータのマイクロプロセッサが理解する言語です。右側は、プログラマがコンピュータに入力するインストラクション・コードです。

どちらの例も、実行させる動作は同じで、プログラムに自分自身を終了させ、DOSプロンプトに制御を返すというものです。しかし、インストラクションを構成するマシン語のバイトコードのシーケンスは全く違うものであることがわかります。例1のコードシーケンスは、単純で一般的な技法を使ってOSを呼び出しますが、例2のシーケンスは、同じリクエストをOSに出すために、例1よりもはるかに遠回りのアプローチを使っています。

例 1 のマシン語のバイトコード : B8 00 4C CD 21
例 2 のマシン語のバイトコード : B4 3C BB 00 00 88 D8 80 C4 10 8E C3 9C 26 FF 1E 84 00

図 3.

コンピュータにプログラムを終了するように命令するマシン語のバイトコードを2通りのシーケンスで示した例。どちらの例も、コンピュータに同じタスクを実行するよう命令するものですが、シーケンスの内容は全く異なるように見えます。

プログラムの実行コードを書く方法は事実上、無数にあるため、コンピュータプログラムを使って、あるプログラムを構成するバイトのシーケンスを調べることで、何らかの情報を突き止めることは不可能に近いように見えます。しかし幸運にも、ほとんどのDOSウイルスは、そのタスクのほとんどを上記で挙げた例1に似た明快なテクニックを使って実行しています。いずれにせよ、ヒューリスティック・スキャナはこのような様式の動作をどのようにして検出するのでしょうか？ヒューリスティック・スキャナは、スタティック方式とダイナミック方式とは明らかに異なるテクニックを使ってこの操作を行います。

スタティック / ダイナミック・ヒューリスティック・スキャナ

スタティック・ヒューリスティック・スキャナは、様々なプログラムを数種類の方法で認識します。スタティック・ヒューリスティック・スキャナの第一の特徴として、前述の通り、バイト・シーケンス（シグニチャ）の巨大なデータベースを維持することが可能であり、そのデータベース内に保存している各バイトシーケンスをその動作と関連付けています。この方式のスキャナは、単純なワイルドカードを使うことによって、ウイルスからウイルスへ増殖するときに変化する可能性がある情報のマッチングを行います。

#	バイト・シーケンス	関連のある動作
1.	B8 ?? 4C CD 21	プログラムの終了（順列置換 1）
2.	B4 4C CD 21	プログラムの終了（順列置換 2）
3.	B4 4C B0 ?? CD 21	プログラムの終了（順列置換 3）
4.	B0 ?? B4 4C CD 21	プログラムの終了（順列置換 4）
	0	
100.	B8 02 3D BA ?? ?? CD 21	ファイル・オープン（順列置換 1）
101.	BA ?? ?? B8 02 3D CD 21	ファイル・オープン（順列置換 2）
	0	

図 4.

ヒューリスティック・スキャナは、動作シグニチャのデータベースを持っています。上記のバイト・シーケンスの1つがプログラム内部で見つかった場合、そのプログラムはおそらく発見したシーケンスに関連付けられた動作を実行する能力を持っていることとなります。シーケンス内の ?? の部分は「ワイルドカード」と言い、任意のバイト値を表します。

スキャン対象プログラムのバイト: B4 09 BA 20 01 CD 21 B8 02 3D BA 12 34 CD 21 CC B8 FF 4C CD 21	
	シグニチャ 100 シグニチャ 1
一致するシグニチャ:	100. B8 02 3D BA ?? ?? CD 21 このプログラムはファイルを開きます。
	1. B8 ?? 4C CD 21 このプログラムは自身を終了します。

図 5.

上記のプログラムのバイト・シーケンスは、ウイルスの可能性のある単純なプログラムを表しています。ヒューリスティック・スキャナが、このプログラムに対し、図 4 に表示されているシグニチャのいずれかに一致する部分を検索した場合、シグニチャ 1 とシグニチャ 100 を発見します。100 番のシグニチャは、このプログラム内の 12 および 34 をワイルドカードに一致するバイト値として扱っていることに注目してください。

これらの文字列は、長年にわたりウイルス駆除製品が十分な根拠に基づいて使用してきた標準のウイルス・シグニチャによく見られるパターンです。事実、これらの文字列はウイルスに使用されていたものです。しかし、従来のウイルス検知スキャナが使用してきた標準のウイルス・シグニチャは、特定のウイルスの系統を明確に識別する目的で使用されています。上記で示したシグニチャは、スキャン対象のプログラムに、何らかのウイルス動作を提示するプログラム・ロジックが含まれているかどうか、また、それ自体に感染性があるかどうかを判別するために使用されています。ヒューリスティック・スキャナがあるプログラム内で前述のような文字列を発見したからといって、必ずしもそのプログラムに感染性があるというわけではなく、そのプログラムがある特定の動作の実行能力があるかもしれないということしかわかりません。

ウイルス・シグニチャの典型的なパターン (Virus Bulletin 誌から引用):

Paulus.1804: B9 D5 00 8B DE ?? ?? 27 06 53 ?? ?? 07 86 CA ?? ?? 86 CA 2E 88 07 4A ??

V.974: 9C 80 FC AA 75 04 B4 BB 9D CF 80 FC 4B 74 0B 80 FC AB 74 06 9D 2E FF 2E

図 6.

上図のウイルス・シグニチャは、ファイルオープンやハードドライブのフォーマットといった具体的な動作ではなく、特定のコンピュータ・ウイルスの系統を識別します。

スタティック・ヒューリスティック・スキャナには、このように単純な動作シグニチャのデータベースに加え、より詳細なハードコード化されたプログラム（ウイルス対策の専門家によって、C++ やアセンブラのようなプログラミング言語で書かれたプログラム）を使って、より複雑なウイルス動作を検索、認識できるようになっているものもあります。例えば、暗号化され、自己変異を行うタイプのDOSウイルスの多くは、感染プログラムの起動時に自分自身を逆スクランブル（復号化）するためのインストラクションを持っています。そのインストラクションを構成するバイト・コードは、外見上は千差万別ですが、ウイルス対策の専門家がウイルスの復号化ルーチンの大半を認識する、かなり単純なサブルーチンを書くことは可能です。ヒューリスティック・スキャナの検知サブルーチンが復号化ルーチンと思われる部分をたまたま発見した場合、その動作も動作カタログに記録します。

スタティック・ヒューリスティック・スキャナは、プログラムの動作のカタログ作成に単純なシグニチャとコード解析サブルーチンに頼りますが、ダイナミック・ヒューリスティック・スキャナは、同様の情報の収集手段としてCPUエミュレーションを使います。ダイナミック・ヒューリスティック・スキャナは、幾つかの初期チェックを実行した後、不審な実行形式ファイルを仮想コンピュータに読み込み、その実行形式ファイルの実行をエミュレートします。エミュレート対象のプログラムは、そのとき自分を実行しているのがシミュレートされたコンピュータ内部とは気づかないため、本物のシステム上で実行された場合と全く同じ動作を実行します。その動作がダイナミック・スキャナによって動作カタログに記録されることとなります。

プログラムが仮想コンピュータ内で動作中、ダイナミック・ヒューリスティック・スキャナは、そのプログラムがオペレーティング・システムに行うリクエスト（割り込み呼び出し）をすべて監視します。仮想オペレーティング・システムがエミュレート中のプログラムによって呼び出されるたびに、スキャナは、その動作を記録し、その後、エミュレート中のプログラムに実行を続行させます。DOSウイルスの大半は、新規プログラムへの感染活動を行う際にオペレーティング・システムに大きく依存するため、このような監視方法は、ウイルス動作の特定には非常に効果的です。この点においては、ダイナミック・スキャナはスタティック・ヒューリスティック・スキャナよりも優れていると言えます。

ここで、スタティック方式とダイナミック方式のヒューリスティック・スキャナの相違点によく似た、ある例え話をご紹介します。まず、私がVotsloviaのアメリカ大使館に勤務する大使である一方で、情報スパイだとしみましょう。私は、大使館から4ブロック離れた研究所で働いているVotsloviaの核科学者から情報を盗み出そうとした場合、その研究所への行き方は何通りもあります。例えば、1ブロック北へ行った後、2ブロック東へ行き、その後1ブロックさらに北へ行く方法をとることができます。あるいは、1ブロック西へ行った後、2ブロック北へ行き、その後3ブロック東へ行く方法をとることもできます。いずれの行き方でも、その科学者の研究所へたどり着くことができます。このように、研究所への行き方が何通りもあるのと同様に、同じ結果を呈するコンピュータ・インストラクションのシーケンスを作成する場合も様々な書き方が考えられます。

Votsloviaの諜報員は、私が数通りの方法を使ってスパイ活動をしていると判断するかもしれません。その場合、諜報員は大使館から科学者の研究所へ行く際に使われそうなルートを確認し、私をそのルートで目撃した場合、当局に通報するでしょう。しかし、私が別のルートを使った場合は？その場合、諜

報員は私の動きを見失うことになります。もちろん、他の諜報員を雇って別のルートを見張らせることもできますが、研究所への行き方は何通りも考えられるため、そのすべてをカバーすることは不可能かもしれません。

別の監視手段として、諜報員が科学者の研究所で張り込みを行うことも考えられます。その場合、諜報員は私が核研究所にどのようにして行くかを考える必要はなく、私が研究所に忍び込んだときに証拠写真を撮るだけですみます。

前者の方法は、スタティック・ヒューリスティック・スキャナがプログラム解析段階で行う方法に似ています。スタティック・ヒューリスティック・スキャナは、様々な動作を探り、検知成功率は不審なプログラムがそのロジックをどのように実行するかに大きく左右されます。解析対象のプログラムが複雑な方法を使ってオペレーティング・システムを呼び出している場合、スタティック・スキャナはその動作の検知に失敗する可能性があります。これは、諜報員が科学者の研究所までのルートを1つまたは少数しか見張っていない場合に私のスパイ行為を見落とすかもしれないのと同じ論理です。

後者の方法は、ダイナミック・ヒューリスティック・スキャナが使用する方法に似ています。ダイナミック・ヒューリスティック・スキャナは、プログラムを仮想マシン上で無制限に動作させます。プログラムは任意のロジックを使って処理を実行できますが、最終的にはオペレーティング・システムを呼び出すことになり、それが行われた時点で、そのプログラムがそれまでに実行したあらゆる演算処理と陰謀の結果が明るみに出てしまうことになります。

プログラムの動作の解析と識別における点では、ダイナミック・ヒューリスティック・スキャナの方がスタティック方式のスキャナよりも数段優れているように見えます。多くの場合、それは当たっているといえます。しかし、処理速度の面では、ダイナミック方式のスキャナはスタティック方式のスキャナよりも遅くなりがちです。CPU エミュレーションは比較的処理に時間がかかりますし、一般的に、限られたメモリ領域上にある文字列をスキャンした場合よりもはるかに処理速度は低下します。さらに、CPU エミュレーションは、エミュレーション対象のプログラムのロジックと気まぐれな行動の影響をもろに受けてしまうという短所があります。その一例として、次のような動作をするウイルスをダイナミック・ヒューリスティック・スキャナを使用して検出したいとします。

擬似コード内のウイルスの動作：

1. 現在の時刻が偶数の場合、インストラクション 3 へスキップする。
2. ステップ 2 へ進む。
3. 単純で識別可能なコンピュータ・インストラクションを使って、新規のプログラムに感染する。
4. ...

図 7.

ダイナミック・ヒューリスティック・スキャナから自分自身の感染インストラクションを隠蔽するロジックを使っているウイルスの一例。このウイルスは、特定の状況下でのみ新たなプログラムに感染します。このようなウイルスが仮想コンピュータ上でエミュレートされた場合、そのロジックが原因で、ダイナミック・ヒューリスティック・スキャナは、感染ロジックを見落としてしまう可能性があります。

ダイナミック・ヒューリスティック・スキャナが、上図のプログラムを仮想コンピュータ内部でエミュレートした場合、現在の時刻をチェックする最初のインストラクションに遭遇した時点でつまづいてしまいます。エミュレーション実行時の時刻がたまたま奇数（1 p.m や 3 p.m. など）だった場合、エミュレートされているプログラムは延々とインストラクション 2 を実行し続けるため、ダイナミック・ヒューリスティック・スキャナはそれ以外のウイルス動作（上図の 3 行目以降）を見

落としてしまうこととなります。一般的に、CPU エミュレータ がウイルスが要求しているものを提供し損なった場合、ウイルスは、そのロジックを使い、本来意図されている動作の実行を停止し、検出から逃れる可能性があります。残念ながら、CPU エミュレータは行われていない動作を言い当てる能力を持っていません。

一方、弊社のスタティック・ヒューリスティック・スキャナは、プログラムのロジックによる影響を受けないため、上図のプログラム内のすべての動作を検知します。スタティック・スキャナは、ウイルス本体に含まれる動作をすべて探します(検索対象の動作がウイルスの通常の実行中に遭遇する動作かどうかは関係ありません)。このように、スタティック方式もダイナミック方式のどちらのスキャナもそれぞれ、長所と短所があります。

スタティック、ダイナミックのいずれのヒューリスティック・スキャナも、プログラムのロジックとインストラクションの解析完了後、ウイルスが潜んでいそうな領域内に保存されている不審なバイト・コードあるいは不審なデータを探します。ウイルス内にはウイルス作成者によって何らかのメッセージや "virus" という単語が含まれることがよくあります。通常、ほとんどの正規プログラムは、このタイプのデータは持っていないため、文字列の存在は、プログラムがウイルスに感染していることを示す強力な証拠となり得ます。ヒューリスティック・スキャナは、スキャンプロセスの第二段階でこれらの文字列や感染を示すその他の兆候を探して記録します。

ヒューリスティック・スキャナは、スキャン対象のプログラムから一組の動作と属性情報を入手すると、観察した動作の解析段階に入ります。通常、この第二段階で行う操作は、ダイナミック方式でもスタティック方式でも同じです。ヒューリスティック・スキャナはこの時点で、観察した動作、ウイルスの疑いがある属性、その他、スキャン対象のプログラムから収集した情報の一覧を入手していることとなります。その後、検出した動作セットがウイルスの疑いがあるかどうかを査定します。例えば、.COM ファイルの末尾に自分自身を付加するタイプのウイルスの場合、ヒューリスティック・スキャナは、そのタイプのウイルスは新たな実行ファイルに感染する際に次の動作を行うことを知っておく必要があります。

オペレーティング・システムに次の操作を要求する：

1. カレントディレクトリ内で最初の COM ファイルを見つける。
2. そのファイルを開く。
3. ファイルの末尾へ行く。
4. ファイルの先頭へ行く。
5. ファイルの先頭に 3-4 バイトを書き出す。
6. ファイルの末尾へ行く。
7. ファイルの末尾に数百バイトを書き出す。
8. ファイルを閉じる。

図 8.

ヒューリスティック・スキャナが、第一段階で上図のすべての動作を観察していた場合、第二段階ではかなり高い確信を持ってウイルスを検知したと報告するでしょう。しかし現実には、第一段階で完全な動作リストが得られることはほとんどありません。スタティックかダイナミックかに関係なく、動作の識別とカタログ作成技術には欠点があり、スキャン対象プログラムに含まれる特定の動作を検出できないことが時々あります。したがって、第二段階では、第一段階で観察した(完全性に欠ける可能性がある)動作セットに基づいて、プログラムの「ウイルス性」に関して高度な推測が行われる必要があります。

例えば、上図で示されている動作のうち、動作カタログ作成段階で 4 番目 ~ 8 番目の動作のみが観察されていた場合はどうでしょうか？ほとんどのウイルス対策専門家は、このサブセットの動作だけでも大きな危険信号を提示していると主張しています。しかし、スキャン対象の実行形式ファイルの検査やス

キャナによる査定を検証する技能を備えた専門家にとっては少しでも感染の疑いがあることを通知するのは有益な情報かもしれませんが、ウイルス解析スキルを持たないエンドユーザに通知する情報がこれだけというのは別問題として考える必要があります。

ヒューリスティック・スキャナが観察により入手した動作セットは全部ではない可能性があるため、ヒューリスティック・スキャナの動作解析コンポーネントは、そのことを注意深く考慮した仕様にする必要があります。このコンポーネントがその条件設定に厳格すぎる場合は、大量のウイルス検知が困難になります。逆に動作条件があまりにも緩く設定されている場合、そのウイルス検知・駆除製品は、誤認傾向が高くなってしまいます。これで、ヒューリスティック技術を採用したウイルス駆除ソフトの中には頻繁に誤認問題を起こしている製品が一部存在する理由がおわかりになっていただけたかと存じます。そのようなウイルス駆除ソフトの設計者は、誤認の可能性が出ることを犠牲にしても、チェックする動作条件を少なくして、検知率が高くなる方法を選んだのだと思われます。

これまで、ウイルス動作の評価には数多くのアプローチが採られてきました。例えば、IBM AntiVirus は、ヒューリスティック技術を採用したブート・ウイルス・スキャナの動作関連の情報の解析にニューラル・ネットワークを使っています。また別の例として、シマンテックのBloodhound技術があります。Bloodhoundは、カタログ化された動作の解析とウイルス感染の徴候査定にエキスパートシステムを採用しています。おそらく、動作解析方法は、現在存在するヒューリスティック・スキャン技術を採用している製品の数だけ存在し、今後も大きく進化し続けるであろうと思われます。

Bloodhound: 次世代のヒューリスティック手法

今後リリースされる Norton AntiVirus 製品にはすべて、シマンテックのBloodhound技術(特許出願中)が装備される予定です。シマンテック・セキュリティ・レスポンス(旧 SARC)のウイルス対策研究者チームは、Norton AntiVirus用に2種類のヒューリスティック技術を開発しました。1つは新出および未知の実行ファイル感染型ウイルスの検知率80%を誇るBloodhoundで、もう1つは新出および未知のマクロウイルスを90%以上の確率で検出、修復するBloodhound-Macroです。

Bloodhound: 新出および未知の実行ファイル感染型ウイルスを最高80%の確率で検知

シマンテックのBloodhound技術は、幾つかの重要な点において、従来のヒューリスティック・スキャナから大きく進展した特長を持っています。Bloodhoundシステムは、従来のスタティックあるいはダイナミック方式のスキャナが使用しているカタログ作成アルゴリズムを使わず、代わりにスタティック方式とダイナミック方式両方の長所を生かしたハイブリッド技術を業界で初めて採用したヒューリスティック・スキャナです。

これまでに述べた通り、従来のヒューリスティック・システムでは、2通りのテクニックのいずれかを使って動作カタログを作成していました。1つはスタティック方式のカタログ作成アルゴリズムで、短時間で処理を終えるという長所がありますが、わかりにくいプログラムロジックの識別には弱く、わずかに標準から外れたものでも認識に失敗してしまうことがよくあります。ウイルス作成者はますます抜け目がなくなってきているため、新種のコンピュータ・ウイルスに対してはスタティック・ヒューリスティック・スキャナの効果はさらに薄れてゆくと考えられます。

他方、ダイナミック方式の動作カタログ作成アルゴリズムは、わかりづらいプログラムのロジックの識別には優れていますが、スタティック方式を用いた場合よりも処理に時間がかかりがちです。また、ダイナミック方式のアルゴリズムは、ウイルスに組み込まれたロジックのトリックが原因で、動作カタログの作成に失敗することがあります。このような問題は、ウイルス作成者によって意図的にダイナミック・スキャナを混乱させるロジックがウイルスに組み込まれている場合に生じます(図7参照)。しかし、多くの場合、普通のロジックを使用するウイルスでも、ダイナミックスキャナによる検知から逃れています。

例えば、ウイルスの多くは、特定範囲のサイズを持つプログラムにのみ感染する仕様になっています。このタイプのウイルスが新規の寄生先プログラムを探しているときに、ファイルサイズが大きすぎる、あるいは小さすぎる実行ファイルを見つけた場合、そのようなファイルには感染せず、他の適切なサイズのファイルを探します。したがって、この場合、ウイルスは検知に必要とされるウイルス動作を見せません。また、別の例として、タイムスタンプをもとに、そのファイルにすでに感染済みかどうかを判断し、タイムスタンプの秒の値が12のプログラムには感染しないウイルスがあります（各ファイルは、最終更新日を「時:分:秒」の形式で表示するタイムスタンプ情報を持っています）。ウイルスの中には、ファイル名の末尾がAVのファイルには感染しないものもあります。これは、NAV.EXEやTBAV.EXEなど、ウイルス・スキャンソフトのプログラムファイルには、ファイル名がAVで終わっているものが数多く存在し、ファイル名にAVを含むプログラムに感染すると、ユーザに感染が気づかれる可能性があることによるものです。残念ながら、個々のウイルスにはそれぞれ独自の条件セットが設定されていて、このように多様な条件がダイナミック・スキャナによる検知を妨害する原因となっています。

このタイプのウイルスの特徴を把握していただくために、ここでわずかに特殊ながらも正確な類似例をご紹介します。ある条件下でのみ感染動作を呈するウイルスを、研究用のカエルと比較してみてください。科学者は、カエルのジャンプ距離や、カエルの足の筋力を調べたいとします。その場合、科学者はまず、人工的な生息環境（水槽や人工池）を用意し、そこにカエルを入れ、カエルがどれくらい遠くまでジャンプしたかを観察することで、このような属性を調べることができます。しかし、カエルは全くジャンプしないこともあれば、少しずつしか跳ねなかったりすることもあります。その場合に研究者が遭遇するのと同様の問題を、ダイナミック・ヒューリスティック技術を使ったウイルス駆除プログラムは抱えています。仮想上の生息環境（仮想コンピュータ）内で、ウイルスが自分の行動に対して制御力を持ち、何らかの動作を自分で抑制することが可能な状況で、あるウイルスがどんな動作を行う能力を持っているのかを正確に判断するのは困難です。

カエルについての研究をしている神経学者であれば、カエルを水槽に入れて動作を起こすのを待つなどということはせず、代わりに、カエルの脳の様々な領域に電極を装着し、領域ごとに電気で刺激を与えることで、カエルの意思に関係なく、その領域に関係した動作を起こさせようとするでしょう。これは、健康診断のときに反射神経を調べると同じようなもので、医師があなたの膝（ひざ）を何かでトントンと叩いたとき、あなたは脚の反射神経を自分でコントロールすることは全くできませんね。脳を電気で刺激する方法は、カエルや他の生物に適用する場合には残酷かどうかについての議論が起きていくくらい特殊な方法ではありますが、コンピュータ・ウイルスに適用する場合は完全に合法かつ倫理にかなったものですし、非常に高い効果が期待できます。

Bloodhound は、人工知能技術を使って、スキャン対象として指定されたプログラムの様々な論理領域を特定し、その後、カエルの脳の各領域に刺激を与えると同時に、特定した論理領域一つ一つに含まれるプログラム・ロジックを分析して、ウイルスらしき動作を探します。Bloodhound は、スタティックとダイナミックの両方の技術を使って、この分析操作と刺激を与える操作を行なうため、スタティックとダイナミックのいずれか一方のみの方式でスキャンした場合に比べ、解析後はより多くのバリエーションの動作を検知可能になります。Bloodhound はウイルスの論理コンポーネントを個別に識別、検査するため、ロジックのトリックを使った攻撃や一般的なウイルスの条件設定による効果に動じることほとんどありません。さらに、分析にはダイナミック方式を用いるため、非常に複雑でわかりにくいプログラム・ロジックでも識別することが可能です。

Bloodhound はスキャン対象のプログラムに刺激を与えることによってそのプログラムに反射動作を起こさせた後、すぐに、エキスパート・システムを使って観察した動作を解析し、そのプログラムがウイルスかどうかを判断します。完全な情報が得られない場合、エキスパート・システムは、与えられた命題に関する高度な推測を行います。エキスパート・システムという名称はもともと、それが使用するルール設定と決定ロジックは通常、その分野の専門家によって設計、プログラミングされていることに由来しています。エキスパート・システムは、主に医療分野で医師による診断材料の収集に効果的に活

用されていますし、また、クレジットカード会社で不審な購入パターンやクレジットカード詐欺を検知する手段としても使用されています。

シマンテックの研究者は、Bloodhoundで使用する大規模なウイルス検知エキスパートシステムの制作に長年にわたり携わってきました。また、シマンテックのエンジニアは、Bloodhoundのエキスパート・システムを、標準のウイルス定義ファイルの更新を通じて定期的に更新できるように設計しました。これにより、シマンテック セキュリティ レスポンスのエンジニアは絶えず Bloodhound システムにさらに磨きをかけたり改良を加えたりしながら、その成果を製品をインストールしなおす手間をかけずに LiveUpdate を通じてエンド・ユーザの皆様に自動的に提供することができるのです。

Bloodhound システムはまた、シマンテックの次世代型ポリモーフィック・ウイルス検知エンジンである Striker が使用する技術を広範囲に渡って利用しています。Striker エンジンのコンポーネントは、暗号化されたウイルスとポリモーフィック・ウイルスの両方のタイプのウイルスを本来の状態に戻す目的で使用されます。このタイプのウイルスは、自身のコンピュータ・ロジックを暗号化することによって、ヒューリスティック・スキャナによる検出を逃れようとしています。しかし Bloodhound は特許出願中の Striker テクノロジーを使ってウイルスの暗号を解析する能力があるため、ウイルスによる隠蔽は Bloodhound には通用しません。

シマンテックの開発チームは、Bloodhound が未知のウイルスの検出では常に超一流のスキャナになるように日夜尽力しています。また、ウイルス誤認率を事実上ゼロにするための研究・開発に膨大な時間を費やしています。シマンテック セキュリティ レスポンスチームのメンバーは、インターネット、商用ソフトウェアライブラリ、ソフトウェア CD を隈なくチェックしてウイルスの発見に努めたり、Bloodhound をあらゆる状況下でテストしています。同チームはこれまでに、米国、欧州諸国、日本から数ギガバイトにのぼる実行形式ファイルを収集し、Bloodhound を世界最強のヒューリスティック技術にしてきました。

最後に、Bloodhound は、新種および未知のコンピュータ・ウイルスを約 80% という高い確率で検知する一方、その処理にかかる時間は最低限に抑えています。Bloodhound は、厳格な前提条件に適合した場合にのみプログラムの詳細な解析を実行するように設計されています。ほとんどの場合、Bloodhound は、ウイルスに感染することはまず有りえないようなファイルについては、1 ファイル当たりわずか 100 万分の 1 秒でウイルスかどうかを判別できます。Bloodhound は、安全なファイルと判断した時点で直ちにそのファイルの解析を停止するため、最も効率的なヒューリスティック・システムの 1 つになっています。

Bloodhound-Macro: 新種および未知のマクロウイルスの 90% 以上を検知、修復

過去 2 年の間に、マクロウイルスは史上最も感染力が高く、広範な地域に渡って蔓延するコンピュータ・ウイルスになっています。これらの電子的な悪党は、信じられないくらいに感染力が強く、電子メール、インターネット、フロッピー・ディスク、データ CD、電子掲示板 (BBS) など、様々な経路を使って瞬く間に拡がります。従来のコンピュータ・ウイルスとは異なり、マクロウイルスはアプリケーションファイルやフロッピーディスクには増殖しない代わりに、企業や家庭で使うことの多い文書ファイルやワークシートファイルに増殖します。

マクロウイルスとは、そもそも、どんなものなのでしょうか？マクロウイルスとは、ワードプロセッサの文書やワークシートに感染するウイルスです。Word や Excel など、最近の文書作成プログラムや表計算プログラムでは、マクロと呼ばれる簡単なプログラムを書いて、それを文書やワークシートに添付することが可能になっています。これらのマクロは本来、繰り返し行う作業や表計算操作など一連の操作を自動化して作業効率を上げるために用意されたものです。マクロウイルスは、本来の有益な目的ではなく、文書から文書、あるいは、ワークシートからワークシートへ自分自身をコピーするように設計された悪質なマクロプログラムにすぎません。Windows 版の Word 文書はマクロウイルスの

最も一般的な増殖手段となっていて、このタイプのマクロウイルスの数は現時点で確認されているものだけでも1,500種を超えています。

マクロウイルスは、特に幾つかの理由により厄介なウイルスです。まず、インターネット、電子メール、ワークグループソフトウェアの普及にとともに、人々が内外の他のユーザとやりとりする情報量もこれまでになく増大してきています。以前は、文書やワークシートにマクロを含めることは不可能だったため、マクロウイルスが含まれる恐れもありませんでした。しかし最近になって文書ファイルにマクロを含めることが可能になると、必然的に文書ファイルにマクロウイルスが含まれる可能性が生じたため、電子メールでやり取りされることが最も多い文書ファイルを介してマクロウイルスがはびこる原因となりました。

マクロウイルスが蔓延するもう1つの要因として、世間で広く使われている業務用アプリケーションの可用性です。Word for WindowsやExcelは、ドキュメントの作成環境とは無関係に、同一の文書あるいはワークシートファイルを共有できる仕様になっています。そのため、マクロウイルスに感染したドキュメントをWindowsパソコン上で作成し、それをMacintoshを使っている他のユーザに送りつけることができます。感染ドキュメントを受け取ったMacintoshユーザがそれを開いたり、編集したりすると、ドキュメントに潜んでいたウイルスがそのシステムに寄生してしまいます。他方、従来のDOSウイルスは、ウイルスの作成元と同じタイプのコンピュータ以外では動作しません。

さらに、マクロウイルスは簡単に作成可能ということも、その出現と蔓延を助長する原因となっています。以前、コンピュータ・ウイルスを作成できるのは、アセンブラなどの低級言語でプログラムを組めるプログラマに限られていました。しかし、ユーザ・フレンドリーなアプリケーションマクロの導入に伴い、ちょっとした知恵があるユーザなら誰でも、Microsoft Officeと数冊の参考書さえあれば、わずか半日でマクロウイルスを作成できるようになりました。さらに悪質なユーザになると、既存のマクロウイルスに使われているプログラミング・ロジックを研究し、それに修正を加えることで別の動作をするマクロウイルスに作り変えてしまいます。既存のマクロに何かを少しだけ加えるだけで、データの暗号化、ハードディスクのフォーマット、さらにはワークシートに含まれている数値データの改ざんまで行うマクロウイルスを、さほどの知識や労力をかけなくても作成できる状況になっているのです。

パソコン環境においてマクロウイルスが広く出回っているのは、このような複数の要因が重なったためと考えられます。しかし、他の要因をすべて組み合わせた場合よりも多くの問題を引き起こしている要因があります。Word for Windowsでは、マクロウイルスが文書から文書へ増殖する際にウイルスのマクロ（プログラミング・ロジック）が壊れてしまう可能性があります。この破損現象は、おそらく、Word for Windowsのソフトウェア上のバグが原因と見られています。

具体的には、マクロウイルスが新たな文書に増殖しようとするたびにWord for Windowsがうっかりそれを壊して変異させてしまう可能性があります。この破損現象は、マクロウイルスの他の文書への感染拡大活動を妨げるため、新たなマクロウイルスにとっては致命的になることもあります。逆にこの破損によって、増殖活動を必ず成功させる能力を持つ全く新しいマクロウイルスの作成にもつながる可能性があります。この破損現象は、多くの点で、故意に発生させているランダムな変異動作と類似しています。変異動作の多くは、新たな子孫を即座に死に至らしめますが、一部の子孫を残して繁殖させるものもあります。

このように、ほとんどのウイルス対策製品メーカーは、既知のウイルスマクロのシグニチャのデータベースを管理維持し、その情報を定期的にウイルス情報としてお客様に伝えています。ウイルス駆除プログラムは、文書やワークシートのスキャン中にシグニチャ・データベースと一致するシグニチャを持つマクロを発見した場合、そのファイルがマクロウイルスに感染していることを通知します。このプロセスは、FBIエージェントが指紋データベースをもとに犯人を識別するときのと同じようなものです。犯行現場に残っていたものと同じ指紋がFBIのデータベース上で見つかった場合、FBIは犯人を識別して逮捕することができます。

残念ながら、これらの対応措置はマクロウイルスにはあまり効果的ではありません。マクロウイルスのプログラミング・ロジックが変化して（破損した場合など）新たに変異したウイルスになった場合、その指紋はウイルス駆除ソフトのデータベースに保存されているものとは異なるものになるため、ウイルス駆除ソフトによる検出から逃れてしまいます。従来のコンピュータ・ウイルスは、この種の破損による影響は受けません。つまり、ユーザは同じウイルスに何度も遭遇する可能性が高く、ウイルス駆除ソフトはもともと、この種のウイルスの検知と駆除に焦点を当てて設計されています。

偶然のマクロ破損現象は、ウイルス作成者の支援を全く得ずに、数百にも上るマクロウイルスを生む原因となりました。以前は、ウイルス作成者の手をかけずに新たなウイルスが出現することはありませんでした。しかし現在では、世界中の無数のコンピュータ上で毎日次々と新たなウイルスが発生しています。これらの新しいウイルスを生み出しているのは、ウイルス研究所でも怪しげなウイルス作成者のコンピュータでもなく、実際に危害を被る職場や家庭にいるエンド・ユーザのコンピュータそのものなのです。

さらに厄介なことに、他のマクロウイルスと徒党を組んでいるマクロウイルスまで出現しています。1つの文書に複数のマクロウイルスが組み合わせられ、全く新しいマクロウイルスを形成する症例について書かれた論文が数多く出されています。新規のマクロウイルスの指紋は、元のマクロウイルスとは異なる指紋になるため、多くの場合、従来のウイルス駆除ソフトでは検知不可能です。このようにウイルス問題の性質が根本的に変化してきたことから、必然的にマクロに対するヒューリスティック・スキャン技術の必要性が高まりました。

大量のマクロウイルスに遭遇しても、従来のウイルス駆除ソフトはこのようにして発生した新規のマクロウイルスを検出、駆除する能力を持っていません。そのため、ユーザは現在、検出も修復も不可能なウイルスに以前にも増して数多く遭遇するようになり、ウイルス駆除ソフトメーカーから対応策を得るまでの時間も以前と比較して長くなっています。つまり、問題を一層深刻にしているのは、新規のウイルスが出現した後、ウイルス駆除ソフトがそれを検出、駆除できるようになるまでの時間がかかりすぎるといことです。

シマンテックのBloodhound-Macro技術は、このような検出に関わる根本的な問題に対処します。Bloodhound-Macroは、あらゆる新規または未知のマクロウイルスの90%以上を自動的に検出、修復することで、ウイルス対策メーカーがその都度手を加えるコストと時間の低減に役立っています。では、Bloodhound-Macroがこのような機能をどのようにして提供できるのかについて解説します。

Bloodhound-Macroは、マクロウイルスを検出、修復する特許出願中のハイブリッド・ヒューリスティック・スキャン技術です。Norton AntiVirusが文書ファイルのスキャンする際、Bloodhound-Macroは毎回、仮想上の完全なWord for Windows環境をセットアップし、そこにスキャン対象の文書を読み込み、その文書に含まれているマクロ（プログラム）を、本物のWordで実行した場合と同様に動作させます。

マクロが仮想上のWord for Windows環境で動作を行っている間、Bloodhound-Macroはその活動状況を監視し、元の文書から他の仮想上の文書へ自己複製を行うかどうかを観察します。正規のマクロのほとんどは通常、文書間で自己複製をする必要はないため、この動作だけでも疑わしいと思われる。しかしウイルス感染を通知する根拠としては、この動作だけでは不十分です。

その理由は？現在、様々な開発会社からリリースされている正規のマクロ・パッケージには、インストール時にユーザの文書ファイルにマクロを自己複製させる方式を使ったものが数多く存在します。これらのマクロは、自分自身をWord for Windows環境にインストールした後は、他の文書にさらに自分自身をコピーすることはありません。他方、マクロウイルスの場合、自分自身を文書ファイルにコピーするとき、そのプログラミング・ロジックとさらに他の文書に増殖するために必要となるデータ

も必ず一緒にコピーします。このように活動内容の決定的な違いが明確になった時点で、正規のマクロ・パッケージとマクロウイルスとを区別できるのです。

Bloodhound-Macro は、このような詳細情報を認識する能力を持っています。Bloodhound-Macro は、ホスト文書内のマクロが他の仮想文書にコピーするかどうかをチェックすることに加え、仮想環境上でコピーされたマクロの動作もシミュレートし、第2世代のマクロにさらに増殖活動を行うためのプログラミング・ロジックが含まれているかどうかをチェックします。Bloodhound-Macro はこのようなシミュレーションを行った結果、そのマクロがウイルスらしい動作を繰り返し行う能力を持っていることを確認した時点ではじめて、ウイルスの存在をユーザに通知する仕様になっています。

マクロウイルスの検出にハイブリッド方式のヒューリスティック・スキャン技術を採用している製品は現在、Norton AntiVirus の Bloodhound-Macro 技術のみです。同製品以外でヒューリスティック技術採用と謳っているウイルス駆除製品が実際に採用しているのは、単純なスタティック・ストリング・スキャンのみです。ストリング・スキャンの場合、文書間に渡って自己複製能力があるかマクロを検出することは簡単にできますが、インストール目的で自己複製を1度だけ行う以外は不審な動作はしない正規のマクロと、本物のマクロウイルスとを判別することは非常に困難です。必然的に、このタイプのヒューリスティック技術は実際には感染していないマクロをウイルスと誤認してしまうため、本来の目的とは裏腹に逆に問題の原因になってしまう可能性があります。

Bloodhound-Macro は文書内のマクロを仮想環境上でエミュレートさせ、どのマクロが複製操作に関与したかを記憶します。その情報は、後で感染文書を修復するときに使用されます。Norton AntiVirus は、ウイルスが増殖活動を行うときに完全にコピーしたマクロのみを除去するため、ユーザ定義のマクロは文書内にそのまま無傷で残ることになります。

他方、スタティック方式のヒューリスティック・スキャナは修復を試みる際も Bloodhound-Macro に劣っています。スタティック方式のヒューリスティック・スキャナは、どのマクロが文書間で自己複製するように設計されているかを識別することはできても、それ以外のウイルスマクロを確実に識別する方法は持っていません。ほとんどのマクロウイルスは、複数のマクロで構成されています。ほとんどの場合、これらのマクロのサブセットのみが文書間でコピーするロジックを含んでいます。それ以外のウイルスマクロには悪意を持った動作をするロジックは含まれている可能性はありますが、そのロジックは必ずしも複製プロセスを行うものとは限りません。したがって、スタティック方式のヒューリスティック・スキャナがマクロウイルスの感染部分の修復を試みた場合、ウイルスらしい動作を行う(マクロのコピーを行う)プログラム・ロジックを含む識別可能なマクロのサブセットのみを削除するため、ウイルス駆除プログラムによる駆除処理の実行後も文書内に悪意を持ったマクロが残ってしまう可能性があります。さらに、スタティック・ヒューリスティック・スキャナは、他社製の正規マクロを、コピーロジックを含んでいるという理由だけで誤ってウイルスであると判定し、削除してしまう可能性があります。

一般に、スタティック・ヒューリスティック・スキャン技術の採用は、他の方式のスキャン技術に比べ誤認傾向が高いと言われています。このタイプのヒューリスティック・システムは、修復プロセス中、ウイルスではないマクロを間違えて削除してしまったり、ウイルスマクロを完全に削除しなかったりすることが多いため、文書やネットワークの完全性が損なわれる可能性があります。シマンテックの特許出願中の Bloodhound-Macro システムが新種および未知のマクロウイルスの検出における現在唯一のソリューションであるというのは、このような所以によるものです。

まとめ

Norton AntiVirusが含まれている製品にはすべて、ウイルス対策業界で最も高度かつ効果的なヒューリスティック技術であるBloodhoundとBloodhound-Macroが組み込まれています。シマンテックの研究者達は、これらの技術が目立たず、オーバーヘッドを最小限に抑える一方で、同時に新種や未知のコンピュータ・ウイルスの検出においては最大限の効果を発揮できる仕様になるように長年にわたり絶えず尽力しています。

Bloodhoundによる新出および未知のファイル感染ウイルスの検知率約80%に上ります。Bloodhound-Macroは新出および未知のマクロウイルスを90%を超える確率で検知し、自動的に修復します。

Bloodhound技術(特許出願中)は、このようにして、Norton AntiVirusを現在市販されている企業向けウイルス対策ソリューションの中で最も効果的な製品にするために大きく貢献しています。

Bloodhoundシステムは、シマンテックが自信を持ってユーザの皆様にお届けする21世紀にふさわしい技術です。

シマンテックについて

株式会社シマンテックは米Symantec Corporation(会長兼CEO: ジョン・W・トンプソン、本社: 米国カリフォルニア州クパチーノ)の日本法人として1994年9月1日に設立されました。インターネット・セキュリティのリーダー企業として、個人、企業、サービスプロバイダへウイルス対策、ファイアウォール、リモート管理技術などのネットワークのセキュリティ・ソリューションを提供し、日本市場のニーズや環境に適した製品の開発、販売およびサポートを行っています。詳細は<http://www.symantec.co.jp>をご覧ください。

* Symantec社の名称、ロゴおよび各製品は、米国Symantec Corporationの米国内およびその他の国における登録商標または商標です。

* その他製品名などはそれぞれ各社の登録商標または商標です。

(c) 1995-2002 symantec corporation.all rights reserved.