

## Back to the Future – Fresh Approaches to Malware Management

*Andrew J. Lee*

*Dorset County Council, United Kingdom*

*David A. Harley*

*National Health Service Information Authority, United Kingdom*

### About the Authors

*Andrew Lee is a Systems Administrator responsible for 3000 networked users. His main responsibilities are the administration, support and deployment of virus prevention systems. In fulfilment of this duty he has designed and implemented many innovative security and anti-virus policies and procedures. He maintains an informational Intranet site dealing with malware issues including the debunking of many common hoaxes. He is a founding member of AVIEN (Anti-Virus Information Exchange Network), and Chairman of its Disciplinary Committee. He is a reporter for the Wildlist Organization and a member of Team Anti-Virus, a group of professional business people dedicated to providing quality Anti-Virus Information and Education.*

*Mailing Address: Dorset County Council, IT Services, Countyhall, Colliton Park, DORCHESTER, Dorset, United Kingdom, DT1 1XJ; Phone: +44 1305 225139; Fax: +44 1305 224391; Email: [virusresearch@aomr.co.uk](mailto:virusresearch@aomr.co.uk) , [a.j.lee@dorset-cc.gov.uk](mailto:a.j.lee@dorset-cc.gov.uk)*

continued on page 2

### Descriptors

virus, worm, malware, safe hex, policy, safe working practices, responsibility, integrity management, policy

**Reference** to this paper should be made as follows:

Lee, A. & Harley, D. (2002). Back to the Future – Fresh Approaches to Malware Management. In U.E. Gattiker (Ed.), EICAR Conference Best Paper Proceedings (ISBN: 78-987271-3-3) pp. 76-109. Copenhagen: EICAR.

*David Harley holds a senior management post in the UK's National Health Service Information Authority, where his role includes threat assessment, incident response/tracking/analysis. In what there is of his spare time, he is an active member of AVIEN (Anti-Virus Information Exchange Network) and EICAR (European Institute for Computer Anti-virus Research). He works from time to time with the Wildlist Organization and TruSecure on Macintosh-related projects, and maintains several Internet information resources, including the Mac Virus web page, at <http://www.sherpasoft.org.uk>. His recent publications include "Viruses Revealed" with Robert Slade and Urs Gattiker and chapters for "Maximum Security" (3<sup>rd</sup> Edition) and the Computer Security Handbook (4<sup>th</sup> Edition, for publication in 2002).*

*Mailing Address: NHS Information Authority, Aqueous II, Aston Cross, Rocky Lane, Birmingham, United Kingdom B6 5RQ; Phone: 44 121 333 0126; Email: [david.harley@nhsia.nhs.uk](mailto:david.harley@nhsia.nhs.uk), [macvirus@dircon.co.uk](mailto:macvirus@dircon.co.uk)*

## Back to the Future – Fresh Approaches to Malware Management

### Abstract

*There is coming a generation of people to whom the word ‘computer’ will bear the same weight in language as “leg” or “arm” and it’s loss or damage be seen in similar terms to the loss or damage to one of those limbs. The computer will be a device without which life will be tremendously difficult. The introduction of malware into such an environment will have enormous repercussion both psychologically and perhaps physically. Even in today’s world the psychological impact of malware cannot be underestimated. Bearing in mind that traditionally, security has been secondary to functionality, the holy grail of the Anti-malware and security worlds will be systems that are not only highly functional and user friendly, but are intrinsically secure and hardened against attacks and other compromises of data integrity, availability, and confidentiality.*

*In this paper we seek to explore the traditional methods of malware management, their congruent weaknesses and strengths, and to propose some ways in which these might be made more successful.*

*Over time there have been many different ideas and theories proposed, some far more useful than others. We have sought to draw on all of these and to extend our own. It is our hope that this examination of current trends and recent history in malware management will throw new light on the issues, and serve as a catalyst to thought and movement towards better future systems.*

### Introduction

The most important part of a computer is neither its hardware nor its software, but the data stored within it. With that in mind, our primary aim should be preserving the following core attributes of our data/information:

- Integrity
- Confidentiality
- Availability

There are two main approaches to the implementation of security measures designed to fulfil the above needs. Allowing everything within reason, or forbidding everything that isn’t necessary. Or, from an administrative point of view:

- Allow everything that isn’t specifically forbidden
- Deny everything that isn’t specifically permitted.

The first option is what we might call a service-oriented approach. We allow everything we possibly can, but introduce some restrictions if a particular service entails a significant enough risk. This approach is often associated with academic environments, and may be system-specific. “Big Iron” systems requiring a minimum level of access control and other security measures (Unix, VMS, MVS) and network operating systems such as NetWare are implemented accordingly, though the degree of additional “hardening” introduced may vary widely. Desktop systems, which are historically insecure, are likely to be reliant on access to the network being controlled rather than security implemented directly on the end-user’s machine. Organizations with this mindset may overlook the fact that the distinction between a modern desktop operating system and its server equivalent may be very tenuous indeed.

Windows NT/2000/XP, Linux and Mac OS X may be found on desktop- and server-class machines, with all that entails in terms of functionality and vulnerability. In fact, the distinction between desktop and server machines has always been somewhat illusory. PCs unable to run modern versions of Windows, which has allowed limited server functionality since version 3.11 or thereabouts, may nevertheless be able to run older versions of Unix or NetWare.

The second option prioritizes security over service. We deny everything unless it is requested, and justify this in terms of the tradeoff between risk and functionality. This approach tends to be associated with organizations where a highly restrictive lockdown of the desktop is favoured. Desktop-oriented operating systems such as Windows 9x and Me are locked down as far as they allow, using the system policy editing tools provided, or replaced with more "grown-up" and potentially restrictive operating systems such as XP. "Career" security administrators and managers (as opposed to system administrators who manage some local security, but may not be extensively trained in security management) usually favour this approach. In some cases, where confidentiality and integrity are of paramount importance to the organization, this mindset is undoubtedly appropriate. In other contexts, the issues may be less clear-cut, and the need for security may be used to justify politicking and empire-building, and the tail may wag the dog to such an extent that the organization is more handicapped than helped by its own security measures.

In practice, of course, many organizations are somewhere between these two extremes. While such a position may often be more accidental than planned, we can nevertheless refer to this as a hybrid approach.

Given those positions, can we achieve a position where a computing environment offers all the functionality a user could want or need, is easy to use, and incorporates viable security? (That is, defense against software/hardware problems, user errors, environmental dangers, and hostile action from people in black hats.) This is a major, complex issue, and rather than attempt to redefine the whole field of information security, we will focus on the significant and highly publicized area of malicious software.

## Literature Review

This is a brief survey of some of the more useful resources that specifically address the malware management issues with which we are concerned.

There is woefully little recent printed literature of high quality on the subject of malware in general, and even less on the effective management of it. Currently most of the recommendations say "Use AV software": a reasonable suggestion as a high-level strategy, but not sufficient when so many, inside and outside the corporate environment, fail to understand either the problem or the solution fully. This results in continuing anomalies such as corporations' insistence on the use of "sheepdip" systems (Solomon & Gryaznov, 1995) and routine on-demand scanning (Overly, 1999; Schmauder, 2000; Mansfield, 2000), despite the high overheads introduced by such measures, with comparatively little gain (Grimes, 2001; Harley, Slade, Gattiker, a 2001).

## Printed Resources

"*Viruses Revealed*", (Harley, Slade, Gattiker, 2001) is a recently published and fairly comprehensive guide to malware and defensive measures. It has a strong practical bias: it includes a significant amount of guidance on safe practice, but leaves the question of where

to trade off functionality against security largely to the judgement of the reader. It has been accused of a degree of anti-Microsoft bias.

"*The Enterprise Anti-Virus Book*" (Vibert 1999) is a comprehensive guide to evaluating and purchasing Anti-viral solutions in corporate situations, but requires a certain amount of knowledge on the part of the reader. (This book is currently pending a second edition that should update and expand many of the sections). It is probably of most use to people evaluating and implementing new anti-virus solutions in large networks.

"*Malicious Mobile Code*" (Grimes, 2001), is useful for those using predominantly Microsoft products, and contains some generally good information about Macro and email aware viruses, but is disconcertingly sparing in reference to ethical issues or to other resources and research work.

The series "*Safe Hex in the 21st Century*" by Martin Overton, published in the June and July 2000 editions of *Virus Bulletin*, contains a good summary of safe computing practices.

## Online Resources

The safe-hex and anti-virus pages at <http://www.claymania.com/nav-map.html> constitute a collective effort by various alt.comp.virus participants.

A number of anti-virus vendors include somewhat similar material, but tend to favour a resolutely "security before functionality" approach.

The Sophos Safer Computing Guidelines at <http://www.sophos.com/virusinfo/articles/safehex.html> are fairly typical, but offer good practical guidance.

## Ethical and Moral issues related to Malware

The most widely published author and researcher in this area is Sarah Gordon, and a number of relevant papers are available or linked from her site at <http://www.badguys.org/papers.htm>

Particularly useful are:

The Generic Virus Writer – Sarah Gordon

<http://www.research.ibm.com/antivirus/SciPapers/Gordon/GenericVirusWriter.html>

The Generic Virus Writer II – Sarah Gordon

<http://www.research.ibm.com/antivirus/SciPapers/Gordon/GVWII.html>

Robert Slade's *Viral Morality: A Call For Discussion* - <http://victoria.tc.ca/int-grps/books/techrev/virethic.txt> - covers in more detail some of the ethical issues raised in this paper.

Dr Vesselin Bontchev has written an excellent paper discussing the issues surrounding the possible use of viruses for "good" or "useful" purposes. The paper is available online at <http://www.f-prot.com/~bontchev/papers/goodvir.html>

## Security is Never Convenient

The section title, taken from a sign on a Huntsville (TX) prison cellblock summarizes neatly a crucial problem in security administration, including malware management. It has been said that if the choice is between security and ease of use, ease of use will always win.

If this is true for computing now, and continues to be so in the future, does it mean that security will never exist for the everyday user, because he or she will not understand the issues and the technologies involved? Possibly not, but the most obvious alternatives are:

- To educate individual computer users or the online community in general, well enough to ensure that they adhere to reasonably safe security practices. It has sometimes been said in the anti-virus community that education doesn't work.

"Any attempts to educate the users about computer security and other things which aren't their job are doomed to fail - I know, I've been trying to do it for years. Sure, you can educate some. But the masses do not \*want\* to be educated. They want to be left alone to do their job. They say that its \*our\* job to protect them, not theirs." (Bontchev, 1999)

"Education of computer users is not very effective...nobody can really rely on the education and discipline to reduce treats [sic] from the Internet" (Muttik, 1998)

Actually, these comments make a very valid point. Users probably don't want to be educated. The fact is, just as we attempt to force them to work in safe ways using technological methods, we should also force them to become educated. This should be part of a wider anti-virus policy with a strong commitment to enforcement behind it.

Our experience, and the experience and studies (Gordon, 1993) of others, show that education does work and is important, if it's done well enough, but cannot be relied upon as a panacea (Harley, 1998). It is especially important that the policy of education is maintained as strongly and vigorously as the technological aspects of the wider policy.

- To lock down systems to such an extent that it's easier to adhere to safe practice than to evade it, or to build security into systems so that "you do it because that's how it's done". We don't usually associate security with transparency, but generations of computer users who are conditioned to regard safe computing practices to be as "natural" as not sharing credit card personal identification numbers (PINs), or walking down dark alleys at midnight, may see it differently. (Gordon & Chess, 1998)
- To reconstruct online society through global education so that people act responsibly and ethically and obey the law. (Well, we said obvious, not necessarily practical.)

## The Biological Model of Virus Management

The paradigm of physical security in terms of protection of human beings from themselves or others can never truly be applied to the world of computing. Nonetheless, there are clearly many parallels. One commonly drawn is the medical analogy of replicative computer malware and human viruses.

However, if the healthcare industry operated in the same way as the anti-virus industry, each of us would be immunised against all viral diseases as and when they were discovered. We suspect that this would have a negative long-term impact on species viability as natural immune response systems atrophied (Harley, 2000).

Is this happening with anti-virus software? Not exactly: most antivirals are as capable of detecting and dealing with a given virus today, as they were at the time that virus was first discovered. They may even deal more effectively with some more recent strains ('mutations', if we follow the populist fad for overextending the biological metaphor). Detection of known viruses could be said to entail the quasi-medical introduction of denatured pathogens to excite an immune response, though this is a somewhat forced analogy, even when it concerns 'classic' virus signatures rather than the more abstract string-searching algorithms employed by competent modern software.

While somewhat peripheral to the scope of this paper, much has been written about biological model comparisons in computer viruses. This research, while now relatively old, still holds great relevance, and can be seen as the basis for much of the thinking behind heuristic approaches and digital immune systems (Kephart, White, Chess, 1993).

Here though, the problem does not usually lie with fragments of virus code in protected (albeit bulky) definitions databases. Rather, it lies in the introduction of processor-intensive scanning software using complex file-parsing and anti-spoofing techniques entailing low-level manipulation of the underlying operating system and file structure. All too often, the use of such esoteric techniques is associated with an inability to share headroom with other software, leading to obscure conflicts and interactions and, in some cases, a noticeable processing overhead in the performance of simple operations.

## **The Past - Let's face it, it's not going to happen again.**

Or: to quote Santayana, "Those who cannot remember the past are condemned to repeat it" (Santayana, 1905).

How often have we heard, perhaps even said, "It never used to be this way", or "It was better in the old days"? In fact, many of our present problems with security in its broadest sense have their roots in the inability of most neophytes to apply 19<sup>th</sup> century morality to the use of 21<sup>st</sup> century technology.

- It isn't stealing; it's just copying.
- It isn't stealing if the victim doesn't know it's been copied.
- Trespassers will not be prosecuted if the violated area is a hard disk.
- Breaking a window is vandalism. Breaking Windows<sup>[TM]</sup> is experimentation - or, better still, ethical hacking.

There is no law that says that progress will not happen, but several perfectly usable ones that say that it inevitably will. Although it may be a rather unpopular and unusual stance, for the purposes of this paper, we will work from the point of view that progress will happen, things will change, and that actually, it's probably not all that bad a thing that they do. "Change", as the saying goes, "is opportunity". (Another saying goes: "Don't fix what ain't broke": the trick is to be able to ascertain which viewpoint is applicable to a current scenario.)

A brief trawl through the relevant USENET groups and security related sites will often turn up advice similar to "use text- only email because it cannot contain viruses", or "send

documents in .RTF, because they cannot contain macros". Unfortunately, these assertions are only true if we define our terms rather carefully to exclude such inconvenient issues as unsafe, unpatched versions of Outlook and Internet Explorer, or the several scenarios where a filename with a .RTF extension can conceal a very dangerous object. Then there are the real hardcore few who say, "don't use Windows, go back to DOS"; forgetting, of course, that there are whole classes of virus, rarely seen or effective in a Windows environment that flourished in DOS. Or the evangelists of Linux who preach the gospel of "If there were no Microsoft, there would be no virus problem."

*"Linux, designed to be a secure, virus resistant, operating system now provides a viable alternative desktop solution"* – NetProject flyer for the Secure Open Desktop Project (<http://www.netproject.com>)

This sort of zealotry ignores a whole series of inconvenient facts, such as:

- Unsecured Linux distributions are less safe than properly secured Windows NT systems. This is true of most operating environments, of course. Out-of-the-box configurations are usually "unsafe" but user-friendly, rather than highly secure by default. This is often true of applications too, including some firewalls and anti-virus software.
- Those same NT systems have the same capabilities for integrity management and sound access control that they admire in Unix, or similar.
- Linux has its own problems with native viruses and worms, an area that has excited increasing attention as the OS has become easier to use and its user base has widened. More users inevitably equals more interest from the 'Black Hat' community, and Unix has always been a 'Bad Boy's (and girls!) Playground'. (This is probably due to widely documented internals and freely available development tools.)

Others argue, "If people really want to send .DOC or .EXE, then they can zip them up." This is not even a technological solution, but collusion in the evasion of security measures. For instance, zipped files are not intrinsically safer than anything else, and some organisations are now recognising this fact by using gateway anti-virus and content analysis software configured to scan the contents of zipped files, and discard or quarantine them if encryption prevents such scanning. 'Solutions' which rely on conspiring to evade Draconian measures are simply about shoving the responsibility back at the customer - usually the person least capable of handling it – via the backdoor. Perhaps that really is where the responsibility belongs, but "sort it out yourself" is not much of a security policy.

All of the progenitors and protagonists of these ideas have seen an aspect of the problem and, instead of rationally tackling it, have given up and decided to go back to some happier age of computing where mail was text and real men used *vi* or *edlin*. Notwithstanding the fact that not only has the problem grown and changed, and essentially become more problematic, the same solution has been proffered; return to the things that we know used to work and that didn't have this problem. In fairness, there has been little alternative but to do many of these things. Even those who have recognised this in the industry have had to admit that there is no currently available solution (FitzGerald, 2000).

Progress is going to happen. Systems are going to get faster and more powerful and the software we run will get more complex. Worldwide broadband 'net access is going to be a reality. The Internet will never be 'safe' again, if it ever was. These are all facts, and all the fingers in the world aren't going to plug the dyke forever. In fact, we should probably re-write that sentence as:

*“Progress is happening, systems are getting faster and more powerful and the software we run is getting more complex. Worldwide broadband access is almost a reality.”*

It's too late to ban the bow, the bomb, or biological warfare. Malicious software is not going to un-happen.

## Here be Draconians

It is our argument that progress in secure computing will only be made if we turn away from such short-sighted views as are currently held by many working in the field, and look to solving the problem as it exists. This will inevitably require new tools and new methods and, most importantly of all, new ways of thinking. Security through reduced functionality is as undesirable as new functionality that brings new insecurity.

Robert Slade describes in his “Guide to Computer Viruses” (Slade, 1995) how an organisation contacted him for help when they were attacked by a virus, but would not allow him onto the premises to deal with it because it would “breach their security”. Indeed, we can draw on much more recent examples of security overriding functionality. One of the present authors has to perform all development work on one of his own machines because it is against organisational policy to allow him administrative rights on his work machine. In another example, an international training organisation summarises one of its security courses as addressing the risks posed by institutions that are “increasing their exposure to customers, competitors, browsers and hackers on the Internet”. We presume that they do, in fact, have some means of distinguishing between these groups. But then, given that we still see experts claiming that between 60% and 80% of security incidents are attributable to employees, perhaps they don't, or shouldn't. After all, those figures only make sense as long as we remember that most security breaches are due to human error rather than malicious action.

It is not our contention that these breaches should not be addressed. It is to everyone's benefit that all security threats be reduced. The question is whether slips of the keyboard and coding errors should be looked at in the same way as theft of identity and privileged data, or as criminal damage.

## So, whose problem is it?

Today more and more people can access the world of the computer. The technology has reached a state where it can be manufactured for so little cost that even the tightest of budgets can afford to possess greater computing power than NASA used to put men on the moon. The politically correct thing to do is to regard this as a Good Thing. In the world of Computer Security, this has rather been seen as the precursor to the end of the world. Interestingly, the threat to national and international security may be greater now than ever, simply and precisely because of the easy and affordable access to computer systems and global networks, so perhaps we can feel some sympathy with the position.

The main reason that security professionals are so convinced that it's a 'Bad Thing' is that several million completely clueless computer users have invaded their holy sanctuary. Computing is no longer an elitist occupation, and is becoming increasingly commonplace. Clueless 'newbies', so the argument goes, don't know anything about computers, yet they buy new machines, discover the Internet, get infected with viruses and then proceed to spread them to all and sundry. Unfortunately there is some truth in that, but the fault, in our

opinion, does not lie exclusively with the users, but also with the systems that they use, and the organisational systems of which they are part.

It is a frequently mooted argument - often by virus writers or vX (virus eXchangers)\* - that the virus problem is the users' fault since it is they who click on attachments. Sadly, many experienced computer users share this opinion. Yet, if one received an incendiary device via snail mail, surely one would not be at fault if the house burned down upon opening it?

This analogy is perhaps limited, mainly because drawing comparisons between explosive devices and computer malware is something best left to the sensationalist quarter of the media. But it does illustrate the point that if people believe that computers are easy-to-use consumer goods with no drawbacks, they do so because there are whole sectors of manufacturing, software publishing and the media all with a vested interest in persuading them that this is so.

A useful parallel can be drawn with the automobile industry. Sometime ago, the relative safety of a car was less of an issue than the size of the engine, the colour, or the number of seats. Likewise with computers; the (probably poor) security of its operating system is not going to be at the forefront of a purchaser's mind; unless they have some experience. For many people, this may be their first purchase. They may never have used a computer before, so to expect them to understand that they have bought a flawed and insecure product is not only unreasonable, but also faintly ridiculous. After all, it's hardly going to be blazoned on the marketing material: "SALE: 1GZ processor, 40 GB hard Drive, 256MB RAM, DVD, Totally insecure and virus vulnerable Operating System". More recently of course, there is a trend for car manufacturers to sell on safety and security - acceleration speed data has been superseded by braking distance data. It is our belief that the Information & Communication Technology (ICT) industry as a whole will eventually be driven towards a similar model of marketing where security is to the fore. We shall return to this analogy and examine the possible driving factors behind this shift in a later section of this paper.

So, we can dismiss the 'It's the users' fault' theory as unfounded, or at least a red herring. Are users therefore exempt from all responsibility for their own actions? Is it unreasonable to expect them to learn enough about the mechanics of computing to reduce the risks?

It is sometimes convincingly argued that it is no more reasonable for an end-user to be required to be a computer expert, let alone a security expert, than it is to expect a motorist to be a qualified mechanic. However, this is not altogether a fair analogy. While it is not compulsory for a motorist to be able to strip and remount an engine, we expect him or her to know enough to fill up with gasoline without setting themselves on fire. We expect them to know enough about the rules of the road to pass some form of driving test, before being allowed onto the public highway, since an incompetent driver is a danger to themselves and to everyone else using the same streets. Of course, safe drivers do not assume that all other drivers are as careful as they are. Even if all end users were required to prove their practical competence and acquaintance with some sort of Information Superhighway Code, designed by a competent authority, this would not guarantee their subsequent safety. You can lead a horse to water, but you can't make him check for Legionnaire's Disease.

---

\* Sarah Gordon, now Senior Research Fellow with Symantec Corporation, coined the term 'vX' to refer to those who participate in uncontrolled Virus eXchange; it is now often used incorrectly as slang for virus writers or their sympathisers.

However we should provide a corollary: a computer user should know enough (ethically and practically) to 'do no harm' and not to trust others to be equally well informed.

The next theory is that it is the fault of the virus writer. We have much more sympathy with this argument than the first one since it follows that if there were no malware, then there would be no malware problem. However, it's a far more complex problem than that (and certainly more complex than we have scope to fully discuss here). Again, there are hard-line adherents to this theory (some very vocal) who will brook no contradiction. While at a certain level this theory is workable, it is useless in either defining or solving the problem; malware exists, and it's not going to suddenly disappear in a puff of its own illogic. One could as well say that if there were no computers there would be no computer malware. (Clearly an extrapolation of Richard's laws of computer security: 1. Don't buy a computer, 2. If you do buy a computer, don't switch it on.)

Similarly, if there were no virus writers, there would be no malware, or at least no new malware. This is a little more hopeful: if virus writers could be persuaded not to create malware, whether through education or by "termination with extreme prejudice", this would certainly form at least a partial solution. The need for greater punitive measures will be discussed further at a later point in this paper.

Then, there is the virus writer's contention that it's the fault of the virus eXchange types: distribution is the key, rather than creation. This of course is ridiculous. On the whole, it is the virus writers who distribute their creations, either to their peers (however limited a set) or more generally. This theory is simply an attempt to shirk the responsibility for an essentially irresponsible act. The issue of whether virus writing and virus exchange are totally discrete is straightforward. The answer is "Yes, if the viral code isn't intentionally made available to anyone who wants it, irrespective of whether they may misuse it." (Harley, Slade, Gattiker, 2001)

There is certainly more discussion to be had on this, but it lies outside the scope of our point here, suffice to say culturally, people groups view "responsibility" very differently. The online community with its relative anonymity and anarchic structure is an area in which personal ethics are to the fore, and the clash and mix of ideas could easily form the basis for many sociological and anthropological studies in the area. Notable for the seminal work in this area is Sarah Gordon, whose writing has provided the AV research community with many rich insights into the workings of the virus writer's mind. [<http://www.badguys.org/>]

Possibly the most appealing theory of all is that it is the fault of the operating system manufacturer. After all, would it be your fault if someone stole your car because the manufacturer did not install it with an effective lock?

While this is as true as the theory that it is the fault of the writer of malware, it is far from satisfactory, however tempting, as a working hypothesis. It is indisputable that there are some operating systems that are built in a deliberately secure manner, and have little contribution to the malware problem; it is equally true that there are others that have security features akin to that of a strongbox made of tissue paper. The difficulty arises from the fact that this was almost inevitable. Very secure operating systems, such as OpenBSD [<http://www.openbsd.org>] were developed as a direct result of the insecurities of other systems. This is a reactive scenario, whereas popular but insecure systems were built and adopted in a time where the security of the system was far less important than its ease of use and low cost. The simple reason why security initially was ignored is that at that time of development the threats we see today did not exist, or were far less prevalent. The fact that such systems became the *de facto* standard for business desktop use, even after the

security problems were beginning to be realised, is in part a testament to the triumph of marketing over common sense.

The most pertinent problem perhaps is that malware is able to function at all in spite of its detrimental effects on the target system. This, however, is inevitable. Any given piece of malware is simply a collection of programmatic instructions that a computer executes or interprets, the problem being that, so is every other piece of software, it's rather the whole point of the computer system. Yet the problem stands. Code will run on a computer regardless of intent, and despite best efforts to identify malicious software automatically. (Cohen, 1994)

An oft touted reason (again usually by vX or virus writers) is that malware is written as a proof that it can exploit that which it was created to exploit. This urge to demonstrate 'proof of concept' is analogous to burning your neighbour's house to prove that it is flammable. We concede that there are a number of such demonstrations that have been instrumental in improving the security of systems, but these have usually been submitted in a responsible manner to the manufacturer of the vulnerable product. (Many people would probably argue that "responsible" in this context probably includes a suggestion, whether implicit or explicit, that (irresponsible) non-action on the part of the vendor will sooner or later result in secondary action on the part of the demonstrator, such as a BugTraq posting. We are as fascinated by this debate as anyone else, but will not pursue it further on this occasion.) Malware writing is unlikely to be a fast or effective means of achieving such improved security, since it is usually distributed in an untargeted and uncontrolled manner (Bontchev, 1994): it tends to rely on reaching the appropriate vendor by way of incident victims and potential copycat malware authors. We therefore conclude that this is at best an excuse for generally irresponsible behaviour.

Another, more subtle difficulty with malware is that it often resists simple definition. While it's a reasonably safe assumption that if a program turns out to be self-replicative, it's probably some sort of virus, identification of malicious software in general can be altogether more difficult, hinging as it does upon the assumption that malicious intent is easily ascertainable. A remote access program installed for administrative use on a network can be an amazingly useful tool, but installed on a system without the knowledge of that system's owner can be a disaster. A program for formatting hard disks may be highly desirable as long as you aren't under the impression that you're running a utility to optimise video performance (Harley, Slade, Gattiker, c 2001). To this end it is often argued that the major issue is consent. This in itself is problematic, in that, it could be said a user clicking on attachment in their email can be said to be consenting to it's installation on their system, and having dismissed the user fault theory as untenable, we must also concede the flaws in this theory.

While a deeper discussion of the problems of malware definition is beyond the scope of this paper, the issues are discussed in some detail in the "Malware Defined" section of *Viruses Revealed*. (Harley, Slade, Gattiker), which examines many of the previously posited definitions. Further discussion of malware definition can be found in Chapters 17 and 18 of *Maximum Security 3e*, (Anonymous, SAMS 2001), Carey Nachenberg's "Computer Parasitology" (Nachenberg, 1999) and Ian Whalley's "Testing Times for Trojans" (Whalley, 1999)

It used to be the case that viruses were relatively slow spreading file infectors, Boot Sector Infectors (BSIs) or Partition Sector Infectors (PSIs). Fast spreading massive infectors are a more recent phenomenon. There were exceptions, 'Stoned' and a few others certainly spread widely, but nothing existed that could infect worldwide within an hour (Staniford, Grim & Jonkman, 2001).

This made the reactive update Anti-virus model reasonably effective for that situation. It would usually be weeks, months or possibly years before a virus crossed your path, and there was often plenty of time to make sure that your AV product was up to date. Unless, of course, accepting that someone had to be the first victim, that victim was you. Unless you worked in AV, by the time you even heard of a virus, it was probably months old, and it was perfectly feasible for many corporations to wait for their next quarterly update. However, even in that era, a fast file infector that got loose on a network could create almost as many local difficulties as a 21<sup>st</sup> Century network-aware virus spreading through local email and network shares.

Today, fast spreading global infectors (fast-burners) are not only possible, but commonplace. Some of them will make headline news within hours or days of their release. Everyone and their dog will have heard about them within a few days. Not only that, many hundreds of large corporations and many thousands of home users will have been infected, sometimes in minutes rather than hours. It has been suggested that home users, increasingly, may be the vectors for mass-mailers, perhaps with an overall movement away from fast-burners to worms like SirCam that start fairly slowly and build up (Harley 2001). At the time of writing, however, informal observation of the initial spread of W32/Badtrans.B suggests a nearly 50/50 split between instances reported inside large corporate networks and instances received from personal accounts. This may reflect a transitional stage. It seems at least possible that fast-burners will decline in terms of throughput, especially as compared to the currently epidemic worms using transport mechanisms other than or as well as email (Code Red II, Nimda).

This is a paradigm shift from the previous situation where the so-called 'Boeing Effect' was accepted as a given: that is, that fast burners picked up their impetus from reverberation around large companies and secondary distribution to other sites. The decline in this trend derives, at least in part, from the increasing tendency for large corporations to apply generic filters, and to introduce email-specific filters (characteristic Subject, message text, or filename) which may be available through early-warning mailing lists (EWS/AVIEN, F-Secure Radar and so on) well in advance of the availability of virus-specific definitions updates and interim drivers.

The world has woken up to the virus threat, it has turned to AV vendors for a solution, only to discover that often the vendors are found wanting. Despite advances in heuristics and automated analysis systems which have certainly made some difference; the classic attack-response cycle (virus released: vendors get samples; vendors analyse; vendors generate means of detection and disinfection; vendors distribute/make available fix) is still simply too slow for a fast-burning mass-mailer. So far, the effect of an 'Anna' or 'Lovebug' has been comparatively inconsequential. What happens when a fast-burner with a destructive payload on a short fuse gets lucky?

Perhaps here we should briefly discuss the advances made in detection over the last few years, as to simply dismiss them would be slightly remiss. While at the AV administrator and user level the perception, and to some extent the reality, is still that there is a flurry of downloading new updates every time a new virus breaks the horizon, behind the scenes things have changed somewhat. Heuristics have advanced to a point where variants of known viruses can often be caught without update, or at least can be flagged as suspicious. At least one vendor has implemented automated analysis systems, where in many cases suspicious files can be flagged and can be analysed and detection provided back to the customer very quickly without any human intervention (at either end), though this is only true of certain classes of virus. This approach has been extremely successful with Macro

Viruses, and fairly so with some types of file virus, it is less so (for good reason) with others such as classic worms (White, 1998). Several vendors have had measurable success with generically detecting viruses generated by kits. This is not necessarily a triumph of heuristic analysis as it is implemented in known virus scanners, though it *is* an application of heuristics in a more general sense. We regard drivers that detect possible future variations of a current virus as an extension of virus-specific technology, rather than purely generic. Possibly *more* so for a “heuristic” that identifies a possible virus because it has characteristics that suggest a particular generator.

While these systems and methods are far from perfect, any degree of success is surely a good thing.

Detection updates when they are needed can often be provided to a system automatically and certainly more swiftly in most cases than used to be normal. Automatic update systems are wonderful as far as they go, but do raise issues of quality control and evaluation of new detection updates. We know of few competent system administrators who would roll out updates to their entire network without some form of QA.

Having acknowledged these advances though, the fact remains that we are often left in a position of vulnerability for some hours where something new can get in. This is especially true if you are the first site to be hit, if any heuristics have not flagged a file as suspicious, and many systems become infected. There is always a time factor in rolling out updates to an entire network, a time factor that increases the larger a network is.

To date, the fact that ‘successful’ malware generally favours maximisation of spread over speed of delivery of destructive payloads has mitigated the degree of destruction associated with worm spread. However, it would be breathtakingly naïve to assume that it will never occur to a virus author that there is little advantage in waiting hours, let alone months, to deliver a destructive payload when you can mail everyone in the Global Address List within minutes.

In an ‘always on’ broadband world the length of time that it will take a mass mailing worm/virus to spread is equal to the time it will take for each email to be delivered and read. Now factor in how long it will take an AV company to be able to detect that malware. Even presuming that they get it at the same time as the first person to receive it, or be infected by it, the time between receipt and an update being available from that vendor's site (let alone installed on the customer's machines) will determine the time in which the malware can spread entirely unchecked.

Even supposing that this space of time was, at minimum, an hour, the number of infected machines would be astronomical.

The technology is clearly available; there have been more than a few VBS Worm Generator (VBSWG) created mass mailers and no reason to suppose that the trend will stop. For instance the ‘Homepage’ and ‘Anna’ mass mailers are only two of many, perhaps with more (and other kits) to come. Fortunately, there has been some indication of a slowdown in fastburner effectiveness, possibly due to education and corporate filtering, and as noted, an improvement in at least some vendors’ generic detection of such objects.

## **System Security and Fad Diets**

Traditional system security can be compared to a low fat diet. Cut out all the 'fun to eat' stuff and only eat what is necessary to stay healthy. So with security the idea is only to allow what is absolutely necessary. The problem with this is that it has neither been, nor is enforceable, nor even necessarily desirable. As technology develops, there will be more and more people wanting to use it to its full potential, and no real reason why they shouldn't. To state otherwise demonstrates a fundamentally flawed logic.

If we can only achieve security by going back to the last generation of technology (or remaining at a static level) then we will never achieve security. True computer security can never be achieved in a system that requires the disabling of some part of its functionality to render it secure, since such measures involve the acceptance of a degree of non-availability – in effect, an always-on denial of service attack.

For instance we often hear it asked, "Why does the GrottyScan site make you use ActiveX, or whatever, as we all know it's insecure?" Actually, the real question should be "Why does the GrottyScan make you use ActiveX when there are alternatives? If there aren't any alternatives in this particular instance, why doesn't it enable you to check that your particular installation is reasonably safe?" There are very good reasons for some people to use xyzscript, the WSH, Outlook etc. These reasons include:

- Increased personal control over their computing environment
- Enabler of interactive computing/data processing
- Roll-your-own programs without having to learn a whole esoteric programming architecture.

The trick is to retain the functionality while increasing the security. Most pundits (the safe hex advocates, Microsoft, et. al.) do not attempt to address this. They give you stark choices.

- All macros or no macros.
- All .EXE or none
- IIS or Apache
- Scripting or no scripting

## **Summary of technical alternatives**

There are many things that we can currently do, some things we can't always do – which may include things that we are likely to be able to do (or would like to be able to do) in the future.

### **Things we can currently do**

When one of the authors is invited to speak intra-organizationally about virus management, it's usually with the expectation that he'll reinforce the need to read all the alerts and apply all the patches and updates. To which his answer is usually "Yes, but..." What we can do currently is what we've always had to do, but it was never quite enough, and is now totally inadequate without the application of other controls.

## Virus-specific scanning

As previously discussed, reactive known-virus scanning is the most prevalent solution currently, and, despite its clear limitations, is unlikely to ever be entirely dispensed with. To quote Martin Overton " Virus scanners have their place, but (and it's a big one) virus scanners are no longer enough." (Overton, 2000)

Anti-virus at the Desktop is currently considered the de-facto defense. Indeed, a frightening number of organizations still use the model of 'Give the problem to the end-user'. However, most organizations of any size and reasonable experience are at least aware that other options exist, including use of AV scanning (sometimes using alternative products) at other access points, to wit:

- On file servers – This often raises issues of performance, but is widely used, and also allows for the use of the server as a means of reinforcing, validating and updating the desktop scanner.
- On mail gateways – Gradually becoming more prevalent. In terms of mass-mailers this may not be effective unless in conjunction with some sort of filtering. Often trying to scan massive amounts of incoming or outgoing mail can cause failure of the scanning software.
- At the firewall – A far less preferable alternative to mail gateway scanning, due to the increased insecurity of running other software on a firewall. The authors are generally of the opinion that a firewall should be just that, with no additions.

At the time of writing we can identify around 75,000 miscellaneous items of software (viruses, worms, Trojans, jokes, junk), using 'Known Virus Scanning' techniques, of which a few hundred actually turn up on people's desktops. Most products get this right most of the time, but at a cost. Costs are often more far reaching than the casual observer might presume. True costs include administration, maintenance, processing overheads, misdiagnoses and false alarms, system instabilities and augmenting the AV vendor revenue streams. Although AV software isn't often solely responsible for conflicts and general flakiness, it often accentuates the negative aspects of other less well-behaved software including, all too often, the host operating system itself.

Security professionals usually expect us to say that the answer is to keep anti-virus software up-to-date. It is an answer. It may be the best answer we have, but it's a rotten answer.

Conceptually, at least from a user/administrator point of view, we're still locked in the late 1980's with Known Virus Scanning (KVS). We are well aware that KVS is not the only game in town. However, it is the main focus of much of the anti-virus industry. Some vendors who at one time included optional change detection in their workstation software suites have dropped it in recent years, along with a number of other generic techniques (behavior blocking/monitoring for instance), though one product has incorporated change detection into its virus-specific implementation, to some advantage. The addition of heuristic analysis to some virus-specific scanners as an optional switch has extended their detection capabilities to some extent (an extent somewhat overstated by many marketing departments), but often at a high administrative cost in terms of processing overhead, increased false positives, or examination and processing of objects flagged as "suspicious". This approach works best in organizations with enough faith in their favored product to discard any such object, or with the resources to analyse flagged problem objects manually and to continually update and refine the heuristics accordingly. To date, the latter approach has been more successful at the messaging level than at the desktop.

[<http://www.message-labs.com>] Even here, it may work better for the anti-virus vendor or ISP than for the customer (at any rate, the smaller customer) whose needs may be adequately served by a less heavily engineered solution such as the use of content analysis tools or relatively simple attachment filtering. We should note that KVS is actually heuristic unless it uses exact identification. (Nachenberg, 1998; Harley, 1999)

AV software is more sophisticated (most essentially, real-time scanning has largely replaced scheduled scans), and heuristics are certainly improving. But the loop mechanism usually remains thus: identify a virus, add a definition, and wait for the next sample (White, 1998). This worked quite well when viruses were in two or three figures and spread so slowly that you could think in terms of quarterly definitions/utility updates. BSIs spread that slowly. File viruses spread faster, but mostly only if servers were sloppily administered. Macro viruses were the first turning point. Though it never really became widely distributed, WM/Sharefun set the basis for mail-aware macro viruses such as W97M/Melissa, which in turn sparked other types of mail-aware fast-burning malware. More recently we saw pure script worms arrive (though there's no altogether convincing distinction between scripts and macro viruses). Most recently of all we have seen increasing use of assembled code, while much of the more destructive and widespread malware arrives in the form of compiled executables.

## Generic Measures

Corporations are increasingly using generic measures to plug some of the gaps left by reactive update model scanning. Generic measures can allow for currently unknown malware. If implemented well, they can be an important aid, if implemented badly they can be rather like trying to bandage an amputated limb with a Kleenex.

On a basic level, generic measures act based on blanket rules rather than specific items.

What can we test generically (identify seems a little over-positive)?

- Suspect Subject fields – e.g. 'Homepage'
- Suspect attachment types: e.g.
  - .EXE
  - .VBS
  - .PIF
- Suspect filenames and double extensions
  - Readme.exe
  - Midgets.scr
  - Mycreditcardlist.doc.lnk

This approach is examined in more detail below.

On identification of a suspect attachment we can then specify action:

- Detect and notify
- Detect and quarantine
- Detect and discard
- Detect and log only

And at each of these points, we could submit to vendor automatically in some cases. Generic measures can be implemented in various places, including the desktop and file server, but seem to be most successfully applied at the mail server.

## Things we can't always do

Current solutions tend to exhibit similar problems, and leave areas of weakness in the defenses. These areas often fall between the two stools of the security and the anti-virus markets, with neither prepared to go after solutions to them. However, without answers to these problems any sort of cohesive defense policy will be incomplete.

What can't we always do?

- Break encryption
- Catch unknown viruses
- Cope effectively with web-hosted email and other evasions of perimeter protection
- Cope effectively with *Convergent Threats*, i.e. attacks that combine
  - Droppers
  - File-less worms
  - Hybrids
  - Multi-polar
  - Rootkits

Added to these is the trend towards viruses/worms starting to use spam techniques, in which the problems are difficult and multiplicitous:

- You can't notify the senders: you don't know who they are.
- You can't always identify the source if it is internal
- The senders don't know: it's not in their outbox.
- Uses forged addresses. Relaying off unsecured server.
- If you (or your sysadmin) tell them, the users don't believe you because of the above.

As we noted earlier, technology is continuously advancing, and with it virus technology, and as it advances the list of things we simply can't do will only get longer. More often than not, it's not the really clever stuff that hits mailboxes heavily. Magistr, SirCam, MTX, are a steady stream, not a flood. It's mostly simpleminded scripts like 'Homepage', 'Anna', 'Apost', and 'Loveletter' that melt servers, and yet in each of those instances, many current anti-virus programs have not been up to the task of stopping them before they have caused mayhem.

## How do we improve?

It is tempting to try to simply ignore the problems, keep on buying our AV updates, and hope that the next Big Thing in malware will pass by with our systems being unscathed. However, this sort of attitude, coupled with an astonishing lack of understanding in the general IT populace is the major contributory factor to the current state of play. Maintenance of the *status quo* is quite simply not an option.

Before we go any further, let's identify the Real Problems

- The Virus Writer
- The Manager
- The Security Administrator
- The Computer User

Surprising isn't it? - It is usually presumed that the problems are not to do with the humans that use the systems, but with the systems themselves. This is an erroneous presumption. The fact is that system security is reasonably simple to achieve if the competency level of

the user is of a sufficiently high level. Even on a relatively insecure system, effective malware protection can be ensured by a competent user, without anti-virus scanners, without firewalls, and with little more than common sense.

The systems that we use can certainly be made more secure, but there is no way to divorce the user from their role in the process. The most secure facility in the world is still reliant on its users to observe correct procedures.

There are three classic approaches to management of malicious software (Harley, McKay, 2001), but no one is, by itself, a complete solution.

- Technological Solutions
- Educational Solutions
- Political Solutions

## **Technological Solutions**

Much of the ability to protect the users from themselves is already inherent in business systems. Operating systems like Windows NT and 2000 offer a fair degree of user lockdown, meaning that theoretically the access of a user can be restricted to an extent where they are unable to do any great harm to their system. Of course, the home user will still be vulnerable, as will the less clueful system administrator. One of the present authors has experienced several support situations where extensive infestation of networks has occurred due to the elevated privileges of administrator accounts being used by less than careful system administrators. System administrators like to think they are all powerful. But, power without knowledge is a dangerous thing.

## **Saving the users from themselves - Technical Solutions to Social Engineering**

Old time viruses worked because they were attached to legitimate code, whether in the boot sector, a system utility, or a document. These required the user to run the code, but both the infector and the infected believed they were trading 'legitimate' code, in the sense that they presumed it to be virally uncompromised.

By 2000, many viruses had crossed the lines that distinguish worms, viruses and Trojans. The marks had to be conned into cooperating in their own demise. This is true of many forms of fraud and other non-violent crime, of course, but the recent worm syndrome marks a paradigm shift: that is, from infection by legitimate but compromised files, to infection (infestation is sometimes a more accurate term) by illegitimate files given spurious authority by a form of masquerading. More recently, we've seen viruses that cause extra confusion by header spoofing, so that the real identity of the previous sender and victim is hidden.

User-launched worms and viruses are effectively Trojans in all but name (in fact, some vendors do classify such programs as Trojans). They must decoy the user into running them, so they promise something wonderful (or at least interesting) if you do open them. For instance; including pictures of Jennifer Lopez, Anna Kournikova or other nubile young celebrities; a movie of the destruction of the World Trade Center or the execution of a murderer. Others threaten something dire if you don't run them; 'hackers will get you if you don't run this patch', 'this is the latest anti-virus software', and so on. Others attempt to appeal to your sense of altruism or make some political point, rather like sympathy chain letters, hoax virus alerts and so on.

Many old time worms were self-launching. They used a system vulnerability to spread, irrespective of direct action by the victims. There are still some worms/viruses (KAK and its siblings, Code Red and the hybrid W32/Nimda) that maintain this tradition.

True worms have traditionally been, and will continue to be, more difficult to deal with than user launched ones, but they do lend themselves to control via integrity management software. Simple solutions include monitoring important registry keys and system files for modification or replacement. Other threats necessitate measures such as monitoring HTTP traffic and ports associated with Trojan/backdoor activity.

In general, though, technological solutions continue to fall into three classes:

Hi-tech Solutions:

- Ever-more sophisticated but essentially traditional ‘known virus scanning’
- Ever-more frequent definitions updates
- More and more automation
- Remote administration

Low-tech Solutions such as generic gateway blocking of attachments according to inappropriate match of file attribute and MIME type, or suspicious filename extension extensions such as the following:

- \*.VBS \*.VBE, \*.JS etc. indicating a potentially malicious script
- \*.SCR, \*.EXE, \*.COM, indicating a potentially malicious binary executable
- \*.PIF, \*.SHS, \*.LNK indicating a possible binary executable with a filetype that may not show up on the target system before the victim sees it and tries to execute it
- \*.DOC, \*.DOT, \*.XLS, indicating a data file that can contain executable programs (macros)
- \*.\* possibly indicating an attempt to exploit the “double extension” trick, which relies on the possibility that the victim will see the first (harmless) extension such as .JPG or .TXT but not the second (e.g. .EXE, .VBS). A better targeted heuristic is to compare the last extension to a blacklist comprising suspicious filename extensions and block accordingly, especially if the previous pseudo-extension matches a whitelist of “safe” extensions such as .RTF or .GIF. This improved targeting is preferred because Unix users, for instance, have been known to use the period character as a simple delimiter between components of a filename. This is quite common in operating environments allowing long filenames, flexible use of punctuation characters within filenames, and not requiring the allocation of fixed filename extensions to executables.
- Filenames containing an unusually high proportion of consecutive space characters, suggesting a possible attempt to crowd a suspicious filename extension off the screen.

Appendix A includes an expanded but by no means all-inclusive list of common, possibly suspicious filename extensions

Some even Lower Tech Solutions include:

- Blocking by worm-specific Subject line, such as
  - Iloveyou
  - Here you have ;-)
- Blocking by characteristic message content, usually a fixed message or set of messages (Kournikova, LoveLetter, SirCam, Homepage, Apost). Other viruses such as Badtrans or Magistr may also be associated with characteristic message content such as an empty

message or random, meaningless text, but these characteristics may be harder to filter generically.

- Blocking by attachment filename: MTX, Apost, PrettyPark, Nimda, and Hybris may be susceptible to this approach, but some may also be associated with random or pseudo-random attachment filenames (Hybris is an example of a virus that covers bestrides both categories, partly because of its ability to evolve by download extra components).

Now we should consider what Cohen (Cohen, 1995) calls non-technical defenses, pointing out "History has shown that no technical method alone is effective for information protection."

## Educational Solutions

Educational solutions are probably the least explored by many corporations, yet may be the most effective. If users had higher competency, rather than being dumped in front of a PC and left to fend for themselves, many of the malware issues would be avoided, or at least significantly reduced. Inevitably, this would have a knock-on effect to home users. Indeed, a more integrated and rational view of computer training would benefit many organisations in terms of productivity, in areas not primarily security-focused. But we can't change the whole world at once.

One of the attractions of education as a security tool is that it deals with a social problem by social means, whereas even the most hardcore draconians will usually admit that social problems cannot usually be satisfactorily fixed by technological solutions. However, education and training comprise an expensive and very high maintenance solution. Extensive security training attempts to make a security expert out of the everyday user or systems administrator. This often results in the propagation of fallacious, unsafe assumptions. Unfortunately, we can't solve a global educational problem and instill a real ethical sense into every spotty nerd who has access to a computer. Boys will be boys (we use the term advisedly: as far as we can tell, there is an amazing preponderance of males writing viruses and cracking systems: it's probably not coincidental that most security administrators are male, too). (Dunham, 2000)

However, if ethics and security issues were integral parts of ICT curricula, and taught as a matter of course in any lesson that included working with PC's (as increasingly happens today, even at a very junior level), we would expect to see improvements in the levels of ethical and practical awareness, and consequent reductions in the numbers of infections if only because it's unlikely that sound teaching of practical measures for handling possible sources of infection and placing computing activities in an ethical context would have an adverse long-term effect on infection incidence. (We would expect a short-term increase in the number of reported incidents, as is common when sound anti-virus measures are implemented or enhanced.). Sadly, we don't think ethical training *is* accepted as the "way to go" by many corporates (Harley, 1998). We aren't aware of any quantitative research that measures the potential or actual improvement in virus-related behaviour from such training, though there are studies that suggest that the existence of a code of conduct without a corresponding educational initiative has no significant impact on behaviour. A great deal of attention is now being given to the teaching of ethics, particularly in the US (Schwartau, [www.nicekids.net](http://www.nicekids.net)), but also in other places such as Germany, albeit, mainly at the University level.

The educational approach works surprisingly well, given the will. Our experience is that it usually works with lower-grade staff (secretaries, office administrators etc.) and often fails

with IT engineers, departmental managers and higher-grade executives, as well as the more obvious categories of employee such as peripatetic workers, home-workers, contractors and consultants.

It may also be true that truly useful definitions of malware can only emerge from the point of view of an accepted ethical framework. It is increasingly difficult to accurately define malware, and the point at which a piece of code can be classed as malicious. [Ford, <http://www.malware.org>]

## Political Solutions

It is not within the scope of this paper to deal too deeply with the subject of political solutions, but for the sake of completeness we will mention briefly some areas that we feel are relevant.

## Legislation

Unfortunately the history of technology legislation is fairly ignominious, consisting mainly of poorly thought out and breathtakingly draconian laws that have been ineffective in hitting the target of the legislation whilst greatly reducing security and privacy for the rest of us. That said, there is certainly some need for governments to take a greater interest in the threats that malware offers to national and international infrastructure. Also, the wider implications of true educational solutions to the problem will necessarily involve legislation to alter and improve curricula, and enact legislation to allow punitive measures against writers of deliberately malicious software. Then of course there are the logistical problems of determining jurisdiction and establishing international standards. So far, in several high profile cases, problems of jurisdiction have made prosecution impossible (Gordon, 1994).

## Punitive measures

The fairly recent high profile viruses VBS/Anna and W97M/Melissa are notable for the fact that in both cases a suspect was arrested and charged with their creation. As well as highlighting how ineffective forensic measures have proved to date in the field of virus management, they also show the parlous state of anti-malware legislation. Jan de Wit, who was responsible for pressing 'Go' on the Anna worm he created, using the VBSWG, was sentenced to a few hours community service, while David Smith, the alleged writer of the Melissa virus, faces up to 40 years in prison.

While it is quite reasonable to expect some punitive measure to be taken against those whose actions cause disruption and damage to computer systems, it seems there is no middle ground in the application of such penalties between woefully inadequate wrist-slapping and laughably extreme (Dunham, 2000).

Governments the world over are beginning to wake up to the fact of malware (it's only taken them 20 years), and unless there is a voice of reason, it is likely that they will continue to legislate without forethought. Some would judge this to already be the case with previous computer-related legislation such as the Digital Millennium Copyright Act (DMCA), the Communications Decency Act (CDA) and UCITA - Uniform Computer Information Transactions Act in the US, the Regulation of Investigatory Powers Act (RIPA) in the UK, and practically any spam-control legislation, which have seemed to make the situation worse rather than better. *(N.B. UCITA isn't directly concerned with malware: it's concerned with making vendors immune to accountability. Actually, none of these are closely related to*

*malware as such: the point is that they're poorly conceived and express total misunderstanding of the technological issues.*) However, it is clear that something needs to be done. Whether it will be effective or not is a matter for future generations to judge. It is to be hoped that governments will seek credible advice on the subject, but since there are so many readily available self-proclaimed malware "experts", it seems unlikely that they will get it.

It is clear that any political solution will require the backup of technology and education, entailing a need for reasonably universal guidelines and Codes of Conduct. Whilst many in the industry subscribe to their own codes of ethics, it would be naive to assume that this will always be the case, particularly as boundaries of morality blur, and the problems mutate and perhaps become less well defined. Furthermore, there is need for corporations to develop strong security policies, preferably before they are forced to do so by legislation. Certain areas of the world are more prone than others to the threat of litigation, but it is clearly undesirable for a company to infect its customers. As potential customer bases expand from local to global, this is likely to be more of an issue. The recent case of a "PowerPuff Girls" DVD being released infected with the W32/Funlove virus is an extraordinary admission of failure to implement adequate security policies by a major company, and is a good illustration of this point.

To return to the earlier analogy drawn between the automotive industry and the nascent AV industry, it is interesting to consider the driving (no pun intended) factors behind the paradigm shift in that older industry. It could be cogently argued that the automotive industry changed its focus away from speed and gadgetry towards security due to the influence not so much of the end consumer, but another industry altogether, namely that of the insurers. Clearly insurance companies have a vested interest in having the cars that they insure be more safe and secure. If there are certain models of car that are considered uninsurable then it is hardly a controversial statement to say that the manufacturer of that car will either modify the design or will drop the model.

Perhaps only once the insurance industry works out its approach to charging and meeting claims related to computer based attacks, will the general software industry, AV industry, and the end user be held accountable for their respective roles and responsibilities. For example, it is the car owners' responsibility to ensure that they don't leave their keys in the car ignition. Yes, if it's stolen, that is certainly a crime, but the insurance is unlikely to pay up more than once for a new car, if indeed they pay up at all. Added to that, repeat offenders will become a bad risk, and the benefits of ensuring sensible security precautions (possibly as laid down by the insurers) become increasingly clear. In the same way, a business faced with the prospect of possible liability and no insurance fallback is likely to want to make sure that they address the problem suitably quickly. This would be equally true for all involved from software manufacturers, through the AV industry and the system administrator, to the end user at every desktop.

### **Safe Hex – The Prophylactic Solution**

What follows is a summary of best practice ideas, working within the framework of what is currently available. As such technology as we have speculated about earlier does not yet exist (or at least not in an entirely useful form), we recognise the need to work within current limitations to provide an adequate framework for the effective management of malware. We recognise that these are most likely to be applied by corporate institutions and, as noted earlier, the problem of home users will go largely unanswered. Having said that, we believe that increasing awareness and competence through education in the workplace will have

clear benefits in the home use arena. Safe-hex is a combination of technological and educational solutions, and while it has the drawback of often recommending security over functionality, it may be the nearest to a 'one-fits-all' solution available. Indeed, with experience, the more draconian measures can be safely discarded, and a fully functional system be used. It is to be hoped that these guidelines will ameliorate the currently poor state of security in many corporations. However, our intention here is less to present yet another "Safe Hex" guidelines document than to question the assumptions behind such guidelines.

Table 1.

<b>Assertion</b>	<b>Rationale and Assumptions</b>	<b>Discussion</b>
<p>Make sure your anti-virus is kept up-to-date</p>	<p>Virus-specific anti-virus software is a Good Thing.</p>	<p>This catches most current malware. Up-to-date is somewhat vague. In the context of a new fastburner, refreshing every hour or less may not be enough to prevent encroachment. "Anti-virus" is also vague, but the options are rarely explored, and they are not: the choices are:</p> <ul style="list-style-type: none"> <li>• Rely on on-access scanning as much as possible. This bypasses the need for the individual user to show due diligence, which is good, but may be associated with processing overhead and software conflicts.</li> <li>• Schedule on-demand scans, which is technically unsatisfactory.</li> <li>• Rely on the end-user's due diligence, in the hope that they'll scan whenever necessary. Not a safe bet.</li> </ul> <p>The first option is usually the safest. Do it, but don't rely exclusively on it, if only because no virus-specific scanner can catch everything. And don't underestimate the administration costs.</p>
<p>Use heuristic analysis and other generic measures such as change detection and other forms of integrity management.</p>	<p>This increases the chances of catching previously unknown viruses.</p>	<p>It also increases the risk of false positives. It's not coincidental that heuristics are rarely enabled by default. The choice is between discarding everything that could be infected or accepting the administration costs of discriminating between false positives (FPs) and real viruses. If extended to serious integrity management, heavy administration costs are unavoidable.</p>
<p>Use multilayering: more than one scanner, or cover more than one entry point, or more than one type of anti-virus</p>	<p>Increases protection.</p>	<p>Also increases administration costs quite drastically. As Cohen has observed, "History shows that the cost of incremental protection increases as perfection is approached." (Cohen, 1995)</p>

Assertion	Rationale and Assumptions	Discussion
scanner.		Cohen probably had in mind the law of diminishing returns, but it's also true in the sense that multiple solutions can introduce conflicts and instabilities, necessitating significant development and reactive costs.
Be cautious.	Be afraid. Be very afraid. Just because you're paranoid, it doesn't mean they're not out to get you. There are no safe environments.	True, all true. But the more you worry about this, and the more time and money you spend on defence, the more you risk damaging your business by <i>replacing</i> functionality with security.
Don't install programs like games, joke programs, cute screen-savers, and unauthorised utility programs and so on. Be particularly cautious about programs found in unsafe environments such as Internet chat-rooms, newsgroups and so on.	These can cause difficulties even if they're genuinely non-malicious. Sometimes it is forbidden to install them, as a matter of local policy	This is here expressed educationally. Many corporations are choosing to impose this more forcefully in terms of locking down the desktop and severely restricting some forms of Internet access. (Telnet, instant messaging.)
Beware attachments.	If they come from someone you know, don't assume the attachment must be OK because you think they are trustworthy. Worms generally spread by sending themselves without the knowledge of the person whose account they spread from.	"Beware" is a little vague: but that's often as far as guidelines like this go. Finer-grained issues are explored further below. More generally, it's worth checking with the sender that they intended to send any attachment. If you were expecting an attachment from them, this may not apply. It needs to be borne in mind that even a legitimate, expected attachment can be virus infected: worms and viruses are related, but slightly different problems. And of course, not all mail-borne viruses rely on being carried by file attachments
Beware of particular kinds of message.	This approach comprises an attempt to teach recognition of the kind of social engineering trickery	This is usually done with an informal rule-based approach like the examples below, which come from an in-house guidance document by one of the authors.

Assertion	Rationale and Assumptions	Discussion
	<p>employed by virus writers.</p>	<p>Rule 1 was, until recently, the only rule universally acknowledged. It's not a bad heuristic in itself, but it can be counterproductive, in that it gives the impression that mail from a known and trusted person is OK, but it still has some validity. Not all attachments to unsolicited mail are malicious, but few of them are actually useful or wanted. They represent variations on ideas that have been used by virus/worm authors in the past, and seem to communicate reasonably well as examples of a more abstract concept.</p> <ul style="list-style-type: none"> <li>• If they come from someone you don't know, who has no legitimate reason to send them to you.</li> <li>• If an attachment arrives with an empty message.</li> <li>• If there is some text in the message, but it doesn't mention the attachment.</li> <li>• If there is a message, but it doesn't seem to make sense.</li> <li>• If there is a message, but it seems uncharacteristic of the sender (either in its content or in the way it's expressed).</li> <li>• If it concerns unusual material like pornographic web-sites, erotic pictures and so on.</li> <li>• If the message doesn't include any personal references at all, (for instance a short message that just says something like "Hi, take a look at this.").</li> <li>• If the attachment has a filename extension that indicates a program file (such as those listed below (in the Appendix)).</li> <li>• If it has a filename with a "double extension", like FILENAME.JPG.vbs or FILENAME.TXT.scr, that may be extremely suspicious. As far</li> </ul>

Assertion	Rationale and Assumptions	Discussion
		as Windows is concerned, it's the last part of the name that counts.
Avoid Word, Excel, Outlook, and any document format that supports macros. If you must use Office, leave macro protection enabled.	Macros should not be enabled unless they are needed. It may be worth checking with the sender you that it is supposed to contain macros.	For better or worse, Office is disconcertingly close to a <i>de facto</i> standard: avoiding it often isn't practical. Some enterprises are seriously reliant on macros. Avoidance is not an option. However, signed macros may offer an alternative.
Ensure that the PC will not boot from a diskette if you leave one in the drive.	Measure against boot sector viruses.	No significant overheads and has played a large part in suppressing BSI activity in recent years.
Disable WSH if you don't need it.	Stops VBScript malware executing.	VBScript is a highly functional development environment, on the workstation and on web servers. This amounts to a permanent self-inflicted Denial of Service attack.
Ensure that safe versions of Internet Explorer and Outlook /Outlook Express are running, if they are installed on your system.	Keeps vulnerabilities patched and makes it 'safer' to use the product.	Involves an ongoing commitment to applying all patches. Armoured Outlook is too restrictive for many people's tastes.
Ensure that data files are automatically backed up.	This should be done regularly, and include offsite copies and archive points.	High administration costs. However, most administrators would consider it necessary, given the range of security problems it alleviates. Of course if the backup copies are infected...
Clear virus and other security alerts, chain letters and so forth, with the appropriate and authorised person	Even if they're received from your boss, your best friend, or the Queen Mother, it's best to let the Help Desk decide whether it's appropriate to pass it on, and if so, to whom.	High potential administration costs (is there a pattern here?). Requires a knowledgeable person to act as a filter.

## Summary and Conclusions

### Recommendations for Managers and Other Decision-makers

Reactive technical solutions work best for corporations with a comprehensive, enterprise wide system for installation and maintenance, plus an internal or outsourced information resource such as the Anti-Virus Information Exchange Network (<http://www.avien.org>), allowing for serious environmental scanning.

Pre-emptive generic measures work well for organizations with a fairly passive user constituency and the resources to do effective perimeter scanning and content analysis. Sensitivity to false alarms is a contra-indicator. Some sensitivity to customer needs is indicated when deciding what to block, and whether to use quarantining, discard, or pass-through-with-warning. User constituencies with a high proportion of mavericks will be difficult to manage under a Draconian regime.

Policy and education work very well for organizations that take these approaches seriously, especially those with a fairly static workforce, encouraging training at all levels, and with support right the way up the management tree.

### Issues for the 21st Century

*“Computer viruses are the first and only form of artificial life to have had a measurable impact on society. Currently, they are a relatively manageable nuisance. However, two alarming trends are likely to make computer viruses a much greater threat. First, the rate at which new viruses are being written is high, and accelerating. Second, the trend towards increasing interconnectivity and interoperability among computers will enable computer viruses and worms to spread much more rapidly than they do today”*

These prophetic words are not ours, but were written in 1994 by Jeffery Kephart (Kephart, 1994)

There is no doubt at all that commercial interests will continue to drive technology forward. The speed at which such technologies are adopted will be in direct proportion to the success of the company producing the technology. If, as at present, the security and integrity of those systems is not paramount, then we will inevitably see no reduction of the amount of malware developed to exploit such systems. It is also likely that the amount of damage, whether commercial or otherwise, will increase in parallel with this trend.

There are already moves towards wearable computers (Carnegie Mellon University, 2000), and experimentation is underway with computers that function as an extension of the human body. It is conceivably possible that such systems will be the norm, even replacing today's desktop PC's. It seems a logical step to move from palmtop and laptop devices to devices integrated into the human body or mind, simply because it is not unreasonable to state that the slowest part of a computer is its human user. Bearing this in mind, it is interesting to speculate about the nature of malware that could exploit this technology. Perhaps the analogue with biological viruses proposed by Cohen will become an actuality, with no distinction between biological viruses and human viruses.

For instance, Lucent Technologies have been able to create molecular scale transistors, a technological breakthrough that will certainly have application in wearable or symbiotic computers (Schon, Bau: 2001).

Further speculation might lead one to wonder about the effects that such malware could have. Today we talk about business systems and corporate networks being 'taken out' by computer viruses, that is, many workstations in the corporation were infected and caused an overload of the mail systems, or perhaps that those stations were rendered useless by something like W32/CIH. However, were the business systems to be flesh and blood humans, the effects could be far more devastating.

There is a perennial debate in the virus world that deals with the reality or otherwise of malware that can physically damage hardware. This debate is fairly interminable, and the evidence mainly anecdotal or received by methods comparable to the old party game of Chinese whispers. (Harley 1998)

In some future reality though, where computers are an integral part of the human body, whether as enhancements to human function, or a means by which business is transacted, the spectre of malware being able to do physical damage to its host becomes ever more corporeal.

It is our contention that change will necessarily be consumer driven; the vendors of both general software and Anti-Virus have shown little inclination to make more than cursory nods towards change. Perhaps things will start to move once more customers start to tell their vendors what they want, rather than the other way around. Smart vendors will note the sea change and exploit the market. The rest will inevitably fall by the wayside. History has consistently shown that not even the greatest empires have been immune to change, and fall because they do not recognise that change has come.

For business, the way forward may in some senses mean a return to an old idea. Note that we do not say old technology; as we have already discussed, we do not see that as a solution. For business use a central point of control will become almost essential, and a locked application database machine the only way to ensure integrity – this is best achievable by using a central machine, or cluster, and serving the applications over a fast network.

Interestingly, there are many advocates of Application Service Providers (ASPs), and there will almost certainly be moves towards that model, in the area of "productivity" software such as document processing suites, but also in the area of malware management, though the vendor thirst for the latter is based on the urge to sell anti-virus service rather than product, and disregards the potential for reducing the need for such a service, by realistic control of application security. The major concern of almost every Security Professional who understands the issues involved is that these so far have not clearly stated how they are going to achieve the necessary level of security for such applications. The idea however is sound, though currently the network infrastructure is not in place for it to be a viable option for many.

Taking the idea and scaling it to fit within an organisation may be more viable, and could work extremely well. According to this model, the control an administrator can exert on the system is almost absolute, and some infection vectors such as floppy disks can be almost entirely eliminated. The difficulty, of course, lies in determining the drivers for this change. There is also the problem of what we do in the meantime. Our malware free utopia will not be built any time soon, so what about the classically difficult environments? The halls of

academia often ring with the screams of desperate system administrators, required to promise total security in a totally laissez-faire environment. Realistically there are many environments which are not suited to centralised administration, and where such centralisation will not be easily sold.

Then of course, there is the ever-present threat of the 'Next Big Thing', the ultimate malware showstopper that we haven't thought of yet (or aren't prepared to discuss publicly). Are we confident that we want to throw even more responsibility onto systems administrators? Knowing what we know, trusting the average system administrator with making sure everything is up-to-date with the latest OS patches, NOS patches, TS Firmware/Software patches, application patches, security software patches, and the inevitable, dreary anti-virus patches, is gambling not only on his or her expertise, but the amount of pressure he or she is under to prioritise more "productive" work

## References

- ASIS - "Living with Viruses", David Harley, in "Security Management" Volume 44 Number 8, page 94, August 2000
- Bontchev Dr. Vesselin: (1994). Are 'Good Viruses' Still a Bad Idea?  
[On-Line] Available: <http://www.fprot.com/~bontchev/papers/goodvir.html>
- Bontchev Dr. Vesselin: (1999). NT Bugtraq post  
[On-Line] Available: <http://archives.in denial.com/hypermail/ntbugtraq/1999/March1999/0077.html>
- Carnegie Mellon University (2001). "Wearable Computers"  
[On-Line] Available: <http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/vuman/www/frontpage.html>
- Cohen, Dr. Frederick B. (1994) "A Short Course on Computer Viruses" (2<sup>nd</sup> ed.). John Wiley & Sons Inc.,
- Cohen, Dr. Frederick B. (1995). "Protection and Security on the Information Superhighway" John Wiley & Sons Inc., ISBN 0-471-11389-1
- Dunham, Ken: (2000). "Bigelow's Virus Troubleshooting Pocket Reference", Osborne (pp13, 17, table 2.1).
- FitzGerald, Nick: (2000). "Why You Should Stop Using Scanners", 10<sup>th</sup> Virus Bulletin Conference Proceedings.  
[On-Line] Available: <http://www.virusbtn.com/vb2000/Programme/papers/fitz.pdf>
- Gordon, Sarah (1994). Technologically Enabled Crime: Shifting Paradigms for the Year 2000  
[On-Line] Available: <http://www.research.ibm.com/antivirus/SciPapers/Gordon/Crime.html> IBM
- Grimes, Roger A. (2001). Malicious Mobile Code – Virus Protection for Windows, O' Reilly
- Harley, David: (1998). "Refloating the Titanic". EICAR Conference Proceedings
- Harley, David: (1999). Managing malware: mapping technology to function. EICAR Conference Proceedings.
- Harley, David: (2000). "Childhood's End – Demythologising Anti-Virus". (pp10-11) Virus Bulletin Apr. 2000.
- Harley, David: (2001). "Sysadmins are doing it for themselves" Virus Bulletin Sept. 2001
- Harley, D., Slade, R., Gattiker, U: (2001). "Viruses Revealed". (Chapter 3). Osborne McGraw Hill
- Harley, David, McKay, Bill: (2001) Homepage Report [Restricted circulation]
- Kephart J, White, Dr S., Chess, D.(1993) Computers and Epidemiology (*IEEE SPECTRUM* May 1993)  
[On-line] Available <http://www.research.ibm.com/antivirus/SciPapers/Kephart/Spectrum/Spectrum.html>
- Mansfield, Richard (2000). "Hacker Attack! Shield Your Computer From Internet Crime", (pp255-256) Sybex,
- Muttik, Dr. Igor: (1998). "Trojans – The New Threat?" IVPC Proceedings - Protecting the Workplace of the Future April 1998
- Nachenberg, Carey (1998). Staying Ahead of the Virus Writers: an In-Depth Look at Heuristics. Virus Bulletin Conference Proceedings (pp85-98) Abingdon, UK. Virus Bulletin.
- Nachenberg, Carey (1999) "Computer Parasitology" Proceedings of 9th International Virus Bulletin Conference  
[On-Line] Available: <http://enterprisesecurity.symantec.com/pdf/computerparasitology.pdf>
- Overly, Michael A.,(1999). "E-Policy. How to Develop Computer, E-mail, and Internet Guidelines to Protect Your Company and Its Assets",( pp78-79 ) Amacom.
- Overton Martin, "Safe Hex in the 21<sup>st</sup> Century Part II" (pp 14-15 July issue) Virus Bulletin Magazine
- Santayana, George (1905). "Life of Reason. Reason in Common Sense", (p284) Scribner's
- Schmauder, Phil, (2000). "Virus Proof: the Ultimate Guide to Protecting Your PC" (pp258-259) Prima Tech
- Schon, Bao.(2001) "Molecular-scale Transistors" Nature Magazine 18<sup>th</sup> Oct 2001.  
[On-line] Available: <http://www.lucnet.com/minds/transistor/molecular/>
- Slade, Robert. (1995). Robert Slade's Guide to Computer Viruses. Springer.
- Solomon A., Gryaznov D,(1995). Dr Solomon's Virus Encyclopaedia, (p14) S&S International

**Authors: Lee, Andrew J. & Harley, David A.**

**EICAR Conference Best Paper Proceedings 2002**

Staniford, Grim, Jonkman (2001) "Flash Worms: Thirty Seconds to Infect the Internet"

[On-Line] Available: <http://www.silicondefense.com/flash/>

Vibert, Robert, (2000) The Enterprise Anti-Virus Book, Segura.

White, Dr. S: (1998) Open Problems in Computer Virus Research 8<sup>th</sup> Virus Bulletin Conference Proceedings

[On-Line] Available: <http://www.research.ibm.com/antivirus/SciPapers/White/Problems/Problems.html>

Whalley, Ian: (1999) "A Testing Time for Trojans", 9<sup>th</sup> Virus Bulletin Conference Proceedings

[On-line] Available: <http://www.research.ibm.com/antivirus/SciPapers/Whalley/inwVB99.html>

**Appendix A – Suspicious Filename Extensions**

The following is a list of filename extensions that indicate an executable program, or a data file that can contain executable programs in the form of macros. This list is not by any means all-inclusive. Robert Vibert’s book [“The Enterprise Anti-Virus Book” *Segura Solutions* <http://www.segura.ca>. *Appendix A, Page 1*] contains a list of nearly 200 infectable objects. Furthermore, there are filenames like .RTF that shouldn’t include program content, but sometimes can, while Word documents (for instance) can in principle have any filename extension, or none.

<i>.BAS</i>	<i>.BAT</i>	<i>.CHM</i>	<i>.CMD</i>	<i>.COM</i>	<i>.CPL</i>	<i>.CRT</i>
<i>.DLL</i>	<i>.DOC</i>	<i>.DOT</i>	<i>.EXE</i>	<i>.FON</i>	<i>.HTA</i>	<i>.INT</i>
<i>.INS</i>	<i>.ISP</i>	<i>.JS</i>	<i>.JSE</i>	<i>.LNK</i>	<i>.MSI</i>	<i>.MSP</i>
<i>.MST</i>	<i>.OVL</i>	<i>.PIF</i>	<i>.PIT</i>	<i>.PL</i>	<i>.REG</i>	<i>.SCR</i>
<i>.SCT</i>	<i>.SHB</i>	<i>.SHS</i>	<i>.URL</i>	<i>.VB</i>	<i>.VBA</i>	<i>.VBE</i>
<i>.VBS</i>	<i>.WS</i>	<i>.WSC</i>	<i>.WSH</i>	<i>.WIZ</i>	<i>.XLA</i>	<i>.XLS</i>

## **Appendix B: Additional Resources**

Mentioned earlier, but out of scope, interested persons are recommended to read these papers.

### **Biological Comparisons and Models of Computer Viruses**

<http://www.research.ibm.com/antivirus/SciPapers/Kephart/ALIFE4/alife4.distrib.html>

Kephart and Arnold.

### **Automatic Extraction of Computer Virus Signatures**

<http://www.research.ibm.com/antivirus/SciPapers/Kephart/VB94/vb94.html>