Cyber*In*security: The Cost of Monopoly

How the Dominance of Microsoft's Products Poses a Risk to Security

Authored by:

Dan Geer
Rebecca Bace
Peter Gutmann
Perry Metzger
Charles P. Pfleeger
John S. Quarterman
Bruce Schneier

Table of Contents

Authors of the report

Daniel Geer, Sc.D – Chief Technical Officer, @Stake

Charles P. Pfleeger, Ph.D – Master Security Architect, Exodus Communications, Inc.

Bruce Schneier – Founder, Chief Technical Officer, Counterpane Internet Security

John S. Quarterman – Founder, InternetPerils, Matrix NetSystems, Inc.

Perry Metzger – Independent Consultant

Rebecca Bace – CEO, Infidel

Peter Gutmann – Researcher, Department of Computer Science, University of Auckland

Cyber*In*security Report

Introduction by

Computer & Communications Industry Association

No software is perfect. This much is known from academia and every-day experience. Yet our industry knows how to design and deploy software so as to minimize security risks. However, when other goals are deemed more important than security, the consequences can be dangerous for software users and society at large.

Microsoft's efforts to design its software in evermore complex ways so as to illegally shut out efforts by others to interoperate or compete with their products has succeeded. The monopoly product we all now rely on is thus both used by nearly everyone and riddled with flaws. A special burden rests upon Microsoft because of this ubiquity of its product, and we all need to be aware of the dangers that result from reliance upon such a widely used and essential product.

CCIA warned of the security dangers posed by software monopolies during the US antitrust proceeding against Microsoft in the mid and late 1990's. We later urged the European Union to take measures to avoid a software "monoculture" that each day becomes more susceptible to computer viruses, Trojan Horses and other digital pathogens.

Our conclusions have now been confirmed and amplified by the appearance of this important report by leading authorities in the field of cybersecurity: Dan Geer, Rebecca Bace, Peter Gutmann, Perry Metzger, John S. Quarterman, Charles Pfleeger, and Bruce Schneier.

CCIA and the report's authors have arrived at their conclusions independently. Indeed, the views of the authors are their views and theirs alone. However, the growing consensus within the computer security community and industry at large is striking, and had become obvious: The presence of this single, dominant operating system in the hands of nearly all end users is inherently dangerous. The increased migration of that same operating system into the server world increases the danger even more. CCIA is pleased to have served as a catalyst and a publisher of the ideas of these distinguished authorities.

Over the years, Microsoft has deliberately added more and more features into its operating system in such a way that no end user could easily remove them. Yet, in so doing, the world's PC operating system monopoly has created unacceptable levels of complexity to its software, in direct contradiction of the most basic tenets of computer security.

Microsoft, as the US trial record and experience has shown, has added these complex chunks of code to its operating system not because such programming complexity is necessary, but because it all but guarantees that computer makers, users and consumers will use Microsoft products rather than a competitor's.

These competition related security problems have been with us, and getting worse, for years. The recent spate of virus attacks on the Internet is one more sign that we must realize the danger we are in. The report Cyber*In*security – The Cost of Monopoly is a wake up call that government and industry need to hear.

September 24, 2003

CYBER*IN*SECURITY: THE COST OF MONOPOLY
HOW THE DOMINANCE OF MICROSOFT'S PRODUCTS POSES A RISK TO SECURITY

Executive Summary

Computing is crucial to the infrastructure of advanced countries. Yet, as fast as the world's computing infrastructure is growing, security vulnerabilities within it are growing faster still. The security situation is deteriorating, and that deterioration compounds when nearly all computers in the hands of end users rely on a single operating system subject to the same vulnerabilities the world over.

Most of the world's computers run Microsoft's operating systems, thus most of the world's computers are vulnerable to the same viruses and worms at the same time. The only way to stop this is to avoid monoculture in computer operating systems, and for reasons just as reasonable and obvious as avoiding monoculture in farming. Microsoft exacerbates this problem via a wide range of practices that lock users to its platform. The impact on security of this lock-in is real and endangers society.

Because Microsoft's near-monopoly status itself magnifies security risk, it is essential that society become less dependent on a single operating system from a single vendor if our critical infrastructure is not to be disrupted in a single blow. The goal must be to break the monoculture. Efforts by Microsoft to improve security will fail if their side effect is to increase user-level lock-in. Microsoft must not be allowed to impose new restrictions on its customers – imposed in the way only a monopoly can do – and then claim that such exercise of monopoly power is somehow a solution to the security problems inherent in its products. The prevalence of security flaw in Microsoft's products is an *effect* of monopoly power; it must not be allowed to become a *reinforcer*.

Governments must set an example with their own internal policies and with the regulations they impose on industries critical to their societies. They must confront the security effects of monopoly and acknowledge that competition policy is entangled with security policy from this point forward.

=====================

The threats to international security posed by Windows are significant, and must be addressed quickly. We discuss here in turn the problem in principle, Microsoft and its actions in relation to those principles, and the social and economic implications for risk management and policy. The points to be made are enumerated at the outset of each section, and then discussed.

1. THE PROBLEM IN PRINCIPLE

To sum up this section:

- Our society's infrastructure can no longer function without computers and networks.
- The sum of the world's networked computers is a rapidly increasing force multiplier.
- A monoculture of networked computers is a convenient and susceptible reservoir of platforms from which to launch attacks; these attacks can and do cascade.
- This susceptibility cannot be mitigated without addressing the issue of that monoculture.
- Risk diversification is a primary defense against aggregated risk when that risk cannot otherwise be addressed; monocultures create aggregated risk like nothing else.
- The growth in risk is chiefly amongst unsophisticated users and is accelerating.
- Uncorrected market failures can create and perpetuate societal threat; the existence of societal threat may indicate the need for corrective intervention.

Discussion

Computing is essential to industrialized societies. As time passes, all societal functions become more deeply dependent on it: power infrastructure, food distribution, air traffic control, emergency services, banking, telecommunications, and virtually every other large scale endeavor is today coordinated and controlled by networked computers. Attacking national infrastructures is also done with computers – often hijacked computers. Thus, threats to computing infrastructures are explicitly and inherently risk harm to those very societies in proportion to those society's dependence on them. A prior history of catastrophe is not required to make such a finding. You should not have to wait until people die to address risks of the scale and scope discussed here.

Regardless of where or how it is used, computing increases the capabilities and the power of those who use it. Using strategic or military terminology that means what it sounds like, computing is a “force multiplier” to those who use them – it magnifies their power, for good or ill. The best estimates of the number of network connected computers show an increase of 50% per year on a worldwide basis. By most general measures what you can buy for the same amount of money doubles every eighteen months (“Moore's Law”). With a conservative estimate of a four year lifetime for a computer – in other words, consumers replace computers every four years on average – the total computing power on the Internet therefore increases by a factor of 2.7 per annum (or doubles every 10 months). If a constant fraction of computers are under threat of misuse, then the force available to misusers will thus double every 10 months. In other words, the power available to misusers – computer hackers, in popular parlance – is rising both because what they can buy grows in power per dollar spent and because the total number of networked computers grows, too. Note also that this analysis does not even include attacks enabled by storage capacity, which doubles in price-performance twice as fast as CPU (doubles every nine months rather than eighteen).

Internetworked computing power makes communication feasible. Communication is of such high value that it has been the focus of much study and much conjecture and not just recently. For one-way broadcast communication, the value of the network itself rises proportionally to N, the potential number of listeners (“Sarnoff's Law”). By way of example, advertisers pay for television time in rough proportion to the number of people viewing a given program.

For two-way interactive communications – such as between fax machines or personal e-mail – the value of the network rises proportionally to N^2, the square of the potential number of users (“Metcalfe's Law”). Thus, if the number of people on email doubles in a given year, the number of possible communications rises by a factor of four.

Growth in communications rises even more when people can organize in groups, so that any random group of people can communicate with another. Web pages, electronic mailing lists and online newsgroups are good examples of such communications. In these cases, the value of the network rises proportionally to 2^N, the potential number of groups being an exponential growth in N (“Reed's Law”).

Assume for now that the Internet is somewhere between the Metcalfe model, where communications vary according to the square of the number of participants (N^2), and the Reed model, where communications vary according to two raised to the Nth power (2^N).

If we make this assumption, then the potential value of communications that the Internet enables will rise somewhere between $1.5^2 = 2.3$ and $2^{1.5} = 2.8$ times per annum. These laws are likely not precisely accurate. Nonetheless, their wide acceptance and historic record show that they are good indicators of the importance of communication technology.

To extend this simple mathematical model one final step, we have assumed so far that all communications are good, and assigned to the value of the network a positive number. Nonetheless, it is obvious that not all communications (over computer networks, at least) are positive. Hackers, crackers, terrorists and garden-variety criminals use the network to defraud, spy and generally wreak havoc on a continual basis. To these communications we assign a negative value.

The fraction of communications that has positive value is one crucial measure, and the absolute number of negative communications is another. Both are dependent on the number of networked devices in total. This growth in the number of networked devices, however, is almost entirely at the "edges" of networked computing – the desktop, the workstation, the home, the embedded system, the automated apparatus. In other words, the growth in "N" is not in the core infrastructure of the Internet where highly trained specialists watch over costly equipment with an eye towards preventing and responding to attacks. Growth, rather, is occurring mostly among ordinary consumers and non-technical personnel who are the most vulnerable to illegal intrusions, viruses, Trojan horse programs and the like. This growth at the periphery, furthermore, is accelerating as mobile, wireless devices come into their own and bring with them still more vulnerabilities.

Viruses, worms, Trojan horses and the like permit malicious attackers to seize control of large numbers of computers at the edge of the network. Malicious attackers do not, in other words, have to invest in these computers themselves – they have only to exploit the vulnerabilities in other people's investments.

Barring such physical events as 9/11, an attack on computing is a set of communications that take advantage of latent flaws already then present in those computers' software. Given enough knowledge of how a piece of software works, an attacker can force it to do things for which it was never designed. Such abuse can take many forms; a naturalist would say that attacks are a broad genus with many species. Within this genus of attacks, species include everything from denial of service, to escalation of authority, to diversion of funds or data, and on. As in nature, some species are more common than others.

Similarly, not all attacks are created equal. An annoying message that pops up once a year on screen to tell a computer user that he has been infected by Virus XYZ is no more than that; an annoyance. Other exploitations cost society many, many dollars in lost data, lost productivity and projects destroyed from data crashes. Examples are many and familiar including the well known ILOVE YOU, NIMDA, and Slammer attacks not to mention taking over users' machines for spamming, porn distribution, and so forth. Still other vulnerabilities, though exploited every day and costing society substantial sums of time and money, seldom appear in the popular press. According to London-based computer security firm, mi2g Ltd., global damage from malicious software inflicted as much as $107 billion in global economic damage this year. It estimates that the SoBig worm, which helped make August the costliest month in terms of economic damage, was responsible for nearly $30 billion in damage alone.[1]

For an attack to be a genuine societal-scale threat, either the target must be unique and indispensable – a military or government computer, authoritative time lookup, the computer handling emergency response (911) calls, airport flight control, say – or the attack must be one which once triggered uncontrollably cascades from one machine to the next. The NIMDA and Slammer worms that attacked millions of Windows-based computers were examples of such "cascade failure" – they spread from one to another computer at high rates. Why? Because these worms did not have to guess much about the target computers because nearly all computers have the same vulnerabilities.

Unique, valuable targets are identifiable so we, as a society, can concentrate force around them. Given enough people and training (a tall order to be sure), it is possible to protect the unique and core assets. Advanced societies have largely made these investments, and unmitigated failures do not generally occur in these systems.

Not so outside this core: As a practical and perhaps obvious fact, the risk of cascade failure rises at the edges of the network where end users are far more likely to be deceived by a clever virus writer or a random intruder. To put the problem in military terms, we are the most vulnerable when the ratio of available operational skill to available force multiplication is minimized and thus effective control is weakest. Low available skill coupled to high potential force multiplication is a fair description of what is today accumulating on the periphery of the computing infrastructures of every advanced nation. In plainer terms, the power on the average desktop goes up very fast while the spread of computers to new places ensures the average skill of the user goes down. The average user is not, does not want to be, and should not need to be a computer security expert any more than an airplane passenger wants to or should need

1 "Government Issue," David Zeiler, *The Baltimore Sun/SunSpot.net.* September 18, 2003

to be an expert in aerodynamics or piloting. This very lack of sophisticated end users renders our society at risk to a threat that is becoming more prevalent and more sophisticated.

Regardless of the topic – computing versus electric power generation versus air defense – survivability is all about preparing for failure so as to survive it. Survivability, whether as a concept or as a measure, is built on two pillars: replicated provisioning and diversified risk. Replicated ("redundant") provisioning ensures that any entity's activities can be duplicated by some other activity; high availability database systems are such an example in computing just as backup generators are in electric power. The ability of redundant systems to protect against random faults is cost effective and well documented.

By contrast, redundancy has little ability to protect against cascade failure; having more computers with the same vulnerabilities cannot help if an attack can reach them all. Protection from cascade failure is instead the province of risk diversification – that is, using more than one kind of computer or device, more than one brand of operating system, which in turns assures that attacks will be limited in their effectiveness. This fundamental principle assures that, like farmers who grow more than one crop, those of us who depend on computers will not see them all fail when the next blight hits. This sort of diversification is widely accepted in almost every sector of society from finance to agriculture to telecommunications. In the broadest sense, economic diversification is as much the hallmark of free societies as monopoly is the hallmark of central planning.

Governments in free market societies have intervened in market failures – preemptively where failure was be intolerable and responsively when failure had become self-evident. In free market economies as in life, some failure is essential; the "creative destruction" of markets builds more than it breaks. Wise governments are those able to distinguish that which must be tolerated as it cannot be changed from that which must be changed as it cannot be tolerated. The reapportionment of risk and responsibility through regulatory intervention embodies that wisdom in action. If governments are going to be responsible for the survivability of our technological infrastructure, then whatever governments do will have to take Microsoft's dominance into consideration.

2. MICROSOFT

To sum up this section:

- Microsoft is a near-monopoly controlling the overwhelming majority of systems.
- Microsoft has a high level of user-level lock-in; there are strong disincentives to switching operating systems.
- This inability of consumers to find alternatives to Microsoft products is exacerbated by tight integration between applications and operating systems, and that integration is a long-standing practice.
- Microsoft's operating systems are notable for their incredible complexity and complexity is the first enemy of security.
- The near universal deployment of Microsoft operating systems is highly conducive to cascade failure; these cascades have already been shown to disable critical infrastructure.
- After a threshold of complexity is exceeded, fixing one flaw will tend to create new flaws; Microsoft has crossed that threshold.
- Even non-Microsoft systems can and do suffer when Microsoft systems are infected.
- Security has become a strategic concern at Microsoft but security must not be permitted to become a tool of further monopolization.

Discussion:

Near-monopoly dominance of computing by Microsoft is obvious beyond the findings of any court. That percentage dominance is at peak in the periphery of the computing infrastructure of all industrial societies. According to IDC, Microsoft Windows represented 94 percent of the consumer client software sold in the United States in 2002.[2] Online researcher OneStat.com estimates Microsoft Windows' market share exceeds 97 percent.[3] Its Internet Explorer and Office Suite applications share similar

[2] "Wal-Mart sells more Linux wares online," Matt Hines, News.com. August 21, 2003.
[3] "Microsoft's Windows OS global market share is more than 97% according to OneStat.com," OneStat.com press release. September 10, 2002.

control of their respective markets. The tight integration of Microsoft application programs with Microsoft operating system services is a principal driver of that dominance and is at the same time a principal driver of insecurity. The "tight integration" is this: inter-module interfaces so complex, undocumented, and inaccessible as to (1) permit Microsoft to change them at will, and thus to (2) preclude others from using them such as to compete.

Tight integration of applications and operating system achieves user lock-in by way of application lock-in. It works. The absence of published, stable exchange interfaces necessary to enable exchange of data, documents, structures, etc., enlists such data, documents, or structures as enforcers of application lock-in. Add in the "network effects," such as the need to communicate with others running Microsoft Office, and you dissuade even those who wish to leave from doing so. If everyone else can only use Office then so must you.

Tight integration, whether of applications with operating systems or just applications with each other, violates the core teaching of software engineering, namely that loosely-coupled interfaces make maintenance easier and life-cycle costs lower. Academic and commercial studies supporting this principle are numerous and long-standing. Microsoft well knows this; Microsoft was an early and aggressive promoter of modular programming practices within its own development efforts. What it does, however, is to expressly curtail modular programming and loose-coupling in the interfaces it offers to others. For whatever reason, Microsoft has put aside its otherwise good practices wherever doing so makes individual modules hard to replace. This explains the rancor over Prof. Ed Felten's Internet Explorer removal gadget just as it explains Microsoft's recent decision to embed the IE browser so far into their operating system that they are dropping support for IE on the Macintosh platform. Integration of this sort is about lock-ins through integration too tight to easily reverse buttressed by network effects that effectively discourage even trying to resist.

This integration is not the norm and it is not essential. Just limiting the discussion to the ubiquitous browser, it is clear that Mozilla on Linux or Safari on Macintosh are counter-examples: tight integration has no technical necessity. Apple's use of Safari is particularly interesting because it gets them all the same benefits that Microsoft gets from IE (including component reuse of the HTML rendering widget), but it's just a generic library, easy to replace.[4] The point is that Microsoft has performed additional, unnecessary engineering on their products with the result of making components hard to pull out, and thus raising the barrier to entry for competition. Examples of clean

[4] "Apple Releases its own browser," Joe Wilcox, News.com, January 7, 2003.

interfaces are much older than Microsoft: the original UNIX was very clean and before that Multics or Dijkstra's 1968 "THE" system showed what could be done. In other words, even when Microsoft was very much smaller and very much easier to change these ideas were known and proven, therefore what we have before us today is not inadvertent, it is on plan.

This tight-integration is a core component of Microsoft's monopoly power. It feeds that power, and its effectiveness is a measure of that power. This integration strategy also creates risk if for no other reason that modules that must interoperate with other modules naturally receive a greater share of security design attention than those that expect to speak only to friends. As proof by demonstration, Microsoft's design-level commitment to identical library structures for clients and servers, running on protocols made explicitly difficult for others to speak (such as Microsoft Exchange), creates insecurity as that is precisely the characteristic raw material of cascade failure: a universal and identical platform asserted to be safe rather than shown in practice to be safe. That Microsoft is a monopoly makes such an outcome the *default* outcome.

The natural strategy for a monopoly is user-level lock-in and Microsoft has adopted this strategy. Even if convenience and automaticity for the low-skill/no-skill user were formally evaluated to be a praiseworthy social benefit, there is no denying the latent costs of that social benefit: lock-in, complexity, and inherent risk.

One must assume that security flaws in Microsoft products are unintentional, that security flaws simply represent a fraction of all quality flaws. On that assumption, the quality control literature yields insight.

The central enemy of reliability is complexity. Complex systems tend to not be entirely understood by anyone. If no one can understand more than a fraction of a complex system, then, no one can predict all the ways that system could be compromised by an attacker. Prevention of insecure operating modes in complex systems is difficult to do well and impossible to do cheaply: The defender has to counter all possible attacks; the attacker only has to find one unblocked means of attack. As complexity grows, it becomes ever more natural to simply assert that a system or a product is secure as it becomes less and less possible to actually provide security in the face of complexity.

Microsoft's corporate drive to maximize an automated, convenient user-level experience is hard to do – some would say un-doable except at the cost of serious internal complexity. That complexity must necessarily peak wherever the ratio of required convenience to available skill peaks, viz., in the massive periphery of the computing infrastructure. Software complexity is difficult to measure but software quality control experts often describe software complexity as proportional to the square of code volume. One need look no further than Microsoft's own figures: On rate of growth, Windows NT code volume rose 35% per year (implying that its complexity rose 80%/year) while Internet Explorer code volume rose 220%/year (implying that its complexity rose 380%/year). Consensus estimates of accumulated code volume peg Microsoft operating systems at 4-6x competitor systems and hence at 15-35x competitor systems in the complexity-based costs in quality. Microsoft's accumulated code volume and rate of code volume growth are indisputably industry outliers that concentrate complexity in the periphery of the computing infrastructure. Because it is the complexity that drives the creation of security flaws, the default assumption must be that Microsoft's products would have 15-35x as many flaws as the other operating systems. [5]

One cannot expect government regulation to cap code size – such a proposal would deserve the derision Microsoft would heap upon it. But regulators would do well to understand that code "bloat" matters most within modules and that Microsoft's strategy of tight integration makes effective module size grow because those tightly integrated components merge into one. It is likely that if module sizes were compared across the industry that the outlier status of Microsoft's code-size-related security problems would be even more evident than the total code volume figures indicate.

Above some threshold level of code complexity, fixing a known flaw is likely to introduce a new, unknown flaw; therefore the law of diminishing returns eventually rules. The general quality control literature teaches this and it has been the received wisdom in software development for a long time (Lehman & Belady at IBM[6] and later in many papers and at least one book). The tight integration of Microsoft operating systems with Microsoft application products and they with each other comes at a cost of complexity and at a cost in code volume. Patches create new flaws as a regular occurrence thus confirming that Microsoft's interdependent product base is above that critical threshold where repairs create problems. Some end-users understand this, and delay deployment of patches until testing can confirm that the criticality of problems fixed are not eclipsed by the criticality of problems created. With mandatory patches arriving at the rate of one every six days (39 through 16 September), it is few users indeed who can keep up.

Two different subsets of users effectively bow out of the patching game: the incapable-many (end-users who have limited understanding of – and limited desire to understand

[5] Microsoft seems at least aware of the problem. See: http://www.wired.com/wired/archive/3.09/myhrvold.html.

[6] L.A. Belady and M.M. Lehman, "A Model of Large Program Development," *IBM Systems Journal* 15(3), p.225–252 (1976).

– the technology even when it is working correctly) and the critical-infrastructure-few (for whom reliability is such a vital requirement that casual patching is unthinkable). Un-patched lethal flaws thus accumulate in the user community. (The Slammer worm fully demonstrated that point – the problem and the patch were six months old when Slammer hit.)[7] Monopoly market dominance is thus only part of the risk story – market dominance coupled with accumulating exploitable flaw density yields a fuller picture. Not only is nearly every networked computer sufficiently alike to imply that what vulnerability one has, so has another, but the absolute number of known-to-be-exploitable vulnerabilities rises over time. Attackers of the most consummate skill already batch together vulnerabilities thus to ensure cascade failure. (The NIMDA virus fully demonstrated that point – it used any of five separate vulnerabilities to propagate itself.)

Microsoft has had a history of shipping software at the earliest conceivable moment. Given their market dominance, within days if not hours the installed base of any released Microsoft software, however ill thought or implemented, was too large to dislodge or ignore. No more. Of late Microsoft has indeed been willing to delay product shipment for security reasons. While it is too early to tell if and when this will actually result in a healthier installed base, it is an admission that the level of security flaw density was a greater threat to the company than the revenue delay from slipping ship dates. It is also an admission that Microsoft holds monopoly power – they and they alone no longer need to ship on time. That this coincides with Microsoft's recent attempts to switch to annual support contracts to smooth out their revenue streams is, at least, opportunistic if not tactical.

On the horizon, we see the co-called Trusted Computing Platform Association (TCPA)[8] and the "Palladium" or "NGSCB" architecture for "trusted computing." In the long term, the allure of trusted computing can hardly be underestimated and there can be no more critical duty of government and governments than to ensure that a spread of trusted computers does not blithely create yet more opportunities for lock-in. Given Microsoft's tendencies, however, one can foresee a Trusted Outlook that will refuse to talk to anything but a Trusted Exchange Server, with (Palladium's) strong cryptographic mechanisms for enforcement of that limitation. There can be no greater user-level lock-in than that, and it will cover both local applications and distributed applications, and all in the name of keeping the user safe from viruses and junk. In other words, security will be the claimed goal of mechanisms that will achieve

[7] " Slammer worm brings patch mgmt. issues to the fore," Audrey Rasmussen, Network World Fusion, Feb. 5, 2003.

[8] See: http://www.trustedcomputing.org/home

unprecedented user-level lock-in. This verifies the relevance of evaluating the effect of user-level lock-in on security.

3. IMPACT ON PUBLIC PROTECTION

To sum up this section:

- Without change, Microsoft's history predicts its future.
- We must take conscious steps to counter the security threat of Microsoft's monopoly dominance of computing.
- Unless Microsoft's applications and interfaces are available on non-Microsoft platforms it will be impossible to defeat user lock-in.
- Governments by their own example must ensure that nothing they deem important is dependent on a monoculture of IT platforms; the further up the tree you get the more this dictum must be observed.
- Competition policy is tangled with security policy from this point on.

Discussion:

Microsoft and regulators come to this point with a considerable history of flouted regulation behind them, a history which seems unnecessary to recount other than to stipulate that it either bears on the solution or history will repeat itself.

Yes, Microsoft has the power to introduce features unilaterally and one might even say that the current security situation is sufficiently dire that Microsoft as the head of a command structure is therefore somehow desirable. Yet were it not for Microsoft's commanding position economics would certainly be different whether it would be a rise in independent, competitive, mainstream software development industries (because the barriers to entry would be lower), or that today's locked-in Microsoft users would no longer pay prices that only a monopoly can extract. For many organizations the only thing keeping them with Microsoft in the front office is Office. If Microsoft was forced to support Office on, say, Linux, then organizations would save substantial monies better spent on innovation. If Microsoft were forced to interoperate, innovators and innovation could not be locked-out because users could not be locked-in.

Both short-term impact mitigation and long term competition policy must recognize this analysis. In the short term, governments must decide in unambiguous ways whether they are able to meaningfully modify the strategies and tactics of Microsoft's already-in-place monopoly.

If governments do not dismantle the monopoly but choose instead to modify the practices of the monopoly they must concede that that route will, like freedom, require eternal vigilance. Appropriate support for addressing the security-related pathologies of monopoly would doubtless include the introduction of effective, accessible rights of action in a court of law wherever security flaws lead to harm to the end-user. In extreme cases, the consequences of poor security may be broad, diffuse, and directly constitute an imposition of costs on the user community due to the unfitness of the product. Under those circumstances, such failures should surely be deemed "per se" offenses upon their first appearance on the network.

Where risk cannot be mitigated it can be transferred via insurance and similar contracts. As demonstrated in previous sections, the accumulation of risk in critical infrastructure and in government is growing faster than linear, i.e., faster than mere counts of computers or networks. As such, any mandated risk transfer must also grow faster than linear whether those risk transfer payments are a priori, such as for bonding and insurance, or a posteriori, such as for penalties. If risk transfer payments are to be risk sensitive, the price and probability of failure are what matter and thus monopoly status is centrally relevant. For governments and other critical infrastructures, the price of failure determines the size of the risk transfer. Where a software monoculture exists – in other words, a computing environment made up of Windows and almost nothing else – what remains operational in the event of wholesale failure of that monoculture determines the size of the risk transfer. Where that monoculture is maintained and enforced by lock-in, as it is with Windows today, responsibility for failure lies with the entity doing the locking-in – in other words, with Microsoft. It is important that this cost be made clear now, rather than waiting until after a catastrophe.

The idea of breaking Microsoft into an operating system company and an applications company is of little value – one would just inherit two monopolies rather than one and the monocultural, locked-in nature of the user base would still nourish risk. Instead, Microsoft should be required to support a long list of applications (Microsoft Office, Internet Explorer, plus their server applications and development tools) on a long list of platforms. Microsoft should either be forbidden to release Office for any one platform, like Windows, until it releases Linux and Mac OS X versions of the same tools that are widely considered to have feature parity, compatibility, and so forth. Alternately, Microsoft should be required to document and standardize its Exchange protocols, among other APIs, such that alternatives to its applications could independently exist. Better still, split Microsoft Office into its components – noticing that each release of Office adds new things to the "bundle": first Access, the Outlook, then Publisher. Even utilities, such as the grammar checker or clip art manager, might pose less risk of compromise and subsequent OS compromise if their interfaces were open (and subject to public scrutiny and analysis and validation). Note that one of the earlier buffer overflow exploits involved the phone dialer program, and ordinarily benign and uninteresting utility that could have been embedded within dial-up networking, Internet Explorer, Outlook and any other program that offered an Internet link.

The rigorous, independent evaluations to which these otherwise tightly integrated interfaces would thus be exposed would go a long way towards security hardening them while permitting meaningful competition to arise. Microsoft will doubtless counter that its ability to "innovate" would be thus compromised, but in the big picture sense everyone else would have a room to innovate that they cannot now enjoy.

Where governments conclude that they are unable to meaningfully modify the strategies and tactics of the already-in-place Microsoft monopoly, they must declare a market failure and take steps to enforce, by regulation and by their own example, risk diversification within those computing plants whose work product they value. Specifically, governments must not permit critical or infrastructural sectors of their economies to implement the monoculture path, and that includes government's own use of computing. Governments, and perhaps only governments, are in leadership positions to affect how infrastructures develop. By enforcing diversity of platform to thereby blunt the monoculture risk, governments will reap a side benefit of increased market reliance on interoperability, which is the only foundation for effective incremental competition and the only weapon against end-user lock-in. A requirement that no operating system be more than 50% of the installed based in a critical industry or in a government would moot monoculture risk. Other branches to the risk diversification tree can be foliated to a considerable degree, but the trunk of that tree on which they hang is a total prohibition of monoculture coupled to a requirement of standards-based interoperability.

CODA

These comments are specific to Microsoft, but would apply to any entity with similar dominance under current circumstances. Indeed, similar moments of truth have occurred, though for different reasons, with IBM or AT&T. The focus on Microsoft is simply that the clear and present danger can be ignored no longer.

While appropriate remedies require significant debate, these three alone would engender substantial, lasting improvement if Microsoft were vigorously forced to:

- Publish interface specifications to major functional components of its code, both Windows and Office.
- Foster development of alternative sources of functionality through an approach comparable to the highly successful 'plug and play' technology for hardware components.
- Work with consortia of hardware and software vendors to define specifications and interfaces for future developments, in a way similar to the Internet Society's RFC process to define new protocols for the Internet.

Biographical Information

Daniel Geer, Sc.D - Dr. Geer is Chief Technical Officer of @Stake, in Cambridge, Mass. Dr. Geer has a long history in network security and distributed computing management as an entrepreneur, author, scientist, consultant, teacher, and architect. He has provided high-level strategy in all manners of digital security and on promising areas of security research to industry leaders including Digital Equipment Corporation, OpenVision Technologies, Open Market, and CertCo. He has written extensively on large-scale security issues such as risk management, applications of cryptography, and Web security for The Digital Commerce Society, the Securities Industry Middleware Council, the Internet Security Conference, and the USENIX Association for whom he founded several conferences.

Dr. Geer has testified before Congress on multiple occasions and has served on various relevant advisory committees to the Federal Trade Commission, the National Science Foundation, the National Research Council, the Commonwealth of Massachusetts, the Department of Defense, the National Institute of Justice, and the Institute for Information Infrastructure Protection.

Dr. Geer holds several security patents, an Sc.D. in Biostatistics from Harvard University's School of Public Health and an S.B. in Electrical Engineering and Computer Science from MIT.

Charles P. Pfleeger, Ph.D - Dr. Pfleeger is a Master Security Architect in the Professional Services group of Exodus Communications, Inc. From 1992 to 1995 he was Director of European Operations for Trusted Information Systems, Inc. (TIS) and head of its European office in London. He was a member of the author group of the U.S. Federal security evaluation criteria and a co-author of the evaluation criteria for trusted virtual machine architectures. He led activities in secure networking, security analysis in hardware design, secure system architecture, and research into assured service. Prior to joining TIS in 1988, he was a professor in the Computer Science Department of the University of Tennessee Dr. Pfleeger has lectured throughout the world and published numerous papers and books. His book *Security in Computing* (the third edtion will be available from Prentice Hall in 2002) is the standard college textbook in computer security. He is the author of other books and articles on technical computer security and computer science topics.

He holds a Ph.D. degree in computer science from The Pennsylvania State University and a B.A. with honors in mathematics from Ohio Wesleyan University.

Bruce Schneier - Internationally renowned security expert Bruce Schneier has authored six books--including BEYOND FEAR and SECRETS AND LIES--as well as the Blowfish and Twofish encryption algorithms. Mr. Schneier has appeared on numerous television and radio programs, has testified before Congress, and is a frequent writer and lecturer on issues surrounding security and privacy.

Mr. Schneier is responsible for maintaining Counterpane's technical lead in world-class information security technology and its practical and effective implementation. Mr. Schneier's security experience makes him uniquely qualified to shape the direction of the company's research endeavors, as well as to act as a spokesperson to the business community on security issues and solutions.

Mr. Schneier holds an MS degree in computer science from American University and a BS degree in physics from the University of Rochester.

John S. Quarterman - John S. Quarterman is founder of InternetPerils, an Internet risk-management company. Previously, he was Founder and Chief Technology Officer of Matrix NetSystems Inc., the first company to map and track global traffic across the Internet. Mr. Quarterman has almost thirty years experience with network issues dating as far back as 1974, when he first used ARPANET, the Internet's predecessor, at Harvard University. He subsequently worked on ARPANET Unix software for Bolt, Beranek and Newman, the original prime contractor for the network.

Mr. Quarterman has consulted for a wide range of companies and organizations, including AT&T, HP, IBM, MCI and Nortel, among others. Twice elected to the board of directors of USENIX, he was instrumental in the board's decision to provide funding for UUNet, one of the first two commercial Internet service providers. A published author, he has written for Communications of the ACM, Forbes, First Monday and Computerworld, among others. He has appeared in articles written by others in the New York Times, the San Jose Mercury News, The Economist, The Washington Post, Wired and others too numerous to mention.

Perry Metzger - Perry Metzger is managing partner of Metzger, Dowdeswell & Co LLC, a New York based computer security and infrastructure consulting firm. Prior to this, Mr. Metzger founded and served as CEO of Wasabi Systems, Inc., a startup specializing in operating system software for embedded platforms. Previously Mr. Metzger served as President of Piermont Information Systems Inc., a New York based computer security consulting firm he founded in 1994. Piermont's clients included prominent international banks and brokerages, money management companies, public relations firms and advertising agencies

Before founding Piermont, Mr. Metzger was involved in a variety of innovative technological projects, including highly parallel computer systems, automated equities trading systems, automated systems management software, and the implementation of one of the world's first firewall systems. Mr. Metzger is highly active in the work of the Internet's standardization body, the IETF. He was instrumental in the design and standardization of several major internet security protocols,including IPSEC, for which he served as co-author of several of the initial standards documents.

Becky Bace - Becky Bace is an internationally recognized expert in network security and intrusion detection. A 2003 recipient of Information Security Magazine's Women of Vision Award, she is recognized as one of the most influential women in Information Security today. Ms. Bace has worked in security since the 1980s, leading the first major intrusion detection research program at the National Security Agency, where she received the Distinguished Leadership Award, serving as the Deputy Security Officer for the Computing Division of the Los Alamos National Laboratory, and, since 1997, working as a strategic consultant.

She is currently President and CEO of Infidel, Inc., a security consulting firm. Ms. Bace's publication credits include the books *Intrusion Detection* (Macmillan, 2000) and *A Guide to Forensic Testimony: The Art and Practice of Presenting Testimony as An Expert Technical Witness,* (Addison-Wesley, October, 2002).

She received a B.S., Engineering/Computer Science from the University of the State of New York, and an M.E.S., Digital Systems Engineering, from Loyola College.

Peter Gutmann - Peter Gutmann is a researcher in the Department of Computer Science at the University of Auckland working on design and analysis of cryptographic security architectures. He helped write the popular PGP encryption package and has authored a number of papers on security and encryption including the X.509 Style Guide for certificates.

Over the years, Mr. Gutmann has uncovered numerous security flaws in various computing products, including problems with the encryption used in an early version of the Netscape browser and, later, Internet Explorer. He has also uncovered flaws in previous versions of Norton's Diskreet disk encryption, the Windows 95 password file system and the smart-card fare system used by Auckland's largest public transportation organization.

Gutmann is the author of the much used, open source cryptlib security toolkit.