DON'T SEND A LAWYER TO DO AN ENGINEER'S JOB: HOW ADVANCING TECHNOLOGY CHANGES THE SOFTWARE MONOCULTURE DEBATE

JOHN BERGMAYER*

INTRODUCTION AND SUMMARY

Modern computer and telecommunications technologies are particularly susceptible to network effects, where the value of a technology increases the more that people use it. Network effects combine with related phenomena, such as the drive toward technological standardization, to create markets that are often dominated by one

* John Bergmayer is a law student at the University of Colorado. He would like to thank Paul Ohm, Scott Challinor, and Shanelle Kindel for their feedback, and Susie Baker, for her patience.

technology. In personal computer software, and by analogy with the agricultural phenomenon, the dominance of Microsoft technology has been called a "software monoculture." In addition to its software being pervasive, it has been argued that Microsoft's engineering practices result in its software being overly complicated and insecure. Because of the widespread dependence on Microsoft's software, these insecurities are then argued to have widespread negative repercussions for the economy and national security. Various proposals, such as requirements that Microsoft share its technology, an expansion of tort liability principles, or merely isolating high-value computer systems from the Internet, have been advanced to deal with this problem.

This Note neither seeks to defend Microsoft's engineering practices, nor to argue that the dominance of its software is a good thing from an economic or engineering perspective. It only notes that the problem of software monoculture is largely a problem with technology, and that technological developments alone, without any legal or policy impetus, may be sufficient to deal with the problem. It also notes that the experience of a particular company following particular engineering principles at a particular time should not be extrapolated to general policy prescriptions. Because the evidence of the negative consequences of software monocultures is usually related to Microsoft products, the case against "monoculture" is really a case against one company.

The analogy to the dangers of excess homogeneity in biological systems is instructive when thinking about technology and software, and many of the same principles that explain the rise of an agricultural monoculture also explain the rise of a software monoculture. But measures that seek to improve diversity, while perhaps appropriate to agriculture, may not be applicable to the more malleable domain of computer software. Even if there are valid policy justifications for some intervention to increase technological diversity, countering the security effects of software monocultures is not among them. Legal reforms should be approached warily, because the risk of unintended consequences that could follow from an improperly calibrated liability regime is very great. Although the picture may have looked different only a few years ago, recent experience shows that such reforms are not justified as a means of counteracting negative security consequences of software monocultures.

I. BACKGROUND

A. Network Effects and Related Phenomena

Microsoft has undoubtedly attained a dominant, near-monopoly position in some software markets.[1] Unlike accounts that attribute Microsoft's success solely to its business and technological acumen, or that paint it as a company perpetually engaging in abusive, anti-competitive behavior, this Note will argue that its success is at least in part attributable to a number of economic and technological phenomena that have amplified and sustained its successes. In order to understand why a software monoculture might arise, it will be helpful to explain these phenomena and see how they apply to software monocultures and to Microsoft. To that end, this section will discuss network effects, indirect network effects such as the "applications barrier to entry," standardization, and path dependence. Often, the same phenomenon can be viewed as an example of more than one of these concepts. The purpose of this section is not to hold up aspects of Microsoft or of other software monocultures as exemplars of particular concepts. Rather, it is to help demonstrate that the rise of a software monoculture is at least partly the result of a complicated interplay of related economic, technical, and market forces, and not necessarily solely the result of improper behavior or an abuse of a dominant market position. Viewing the dominance of Microsoft in this light, it may be easier to accept that the similar forces can be relied on to counteract social costs that may have been caused by its success.

1. Network Effects

Markets for communications technology are particularly susceptible to network effects. Products that feature network effects become more valuable to a person the more that other people use them. While a hammer is just as valuable to a carpenter no matter how many other carpenters use the same kind of hammer, a fax machine is valuable only if there are other fax machines to send faxes to. The more people who have fax machines, the more valuable all fax machines become. Robert

1. Much has been written on whether Microsoft is a monopoly, and whether markets for software products can be considered natural monopolies. *See, e.g.*, COMPETITION, INNOVATION, AND THE MICROSOFT MONOPOLY: ANTITRUST IN THE DIGITAL MARKETPLACE (Jeffrey A. Eisenach & Thomas M. Lenard, eds., 1999). This discussion is largely irrelevant to this Note as whether or not a software company is a monopoly in the sense that it is able to take monopoly profits, it may still control a monoculture by having a large enough base of installed users. Additionally, monopolies may be distinguished from technology monocultures in that it is possible for one company to support multiple technologies, and for multiple companies to contribute to a software monoculture.

Metcalfe, co-inventor of Ethernet, stated the value of a telecommunications network is proportional to the square of the number of the users of the system.[2] While "Metcalfe's Law" has been criticized as inaccurate and overly simplistic,[3] it helps to keep in mind, as a rule of thumb, that communication networks obtain their value not just from the quality of their technology, but from the number of people who use them. Telephone networks offer a key historical example of network effects in a communication system. Although at first, there were several competing telephone networks that did not interconnect with one another, one network, the Bell System, soon drew enough users to make its offering significantly more valuable than that of its competitors.[4] Network effects are pervasive in newer electronic communications networks, as well. Even social networking sites such as MySpace benefit from network effects.[5]

Many computer technologies, notably operating systems,[6] are subject to network effects. The dominance of Microsoft Windows makes it vital that all PCs, even those running Linux and Macintoshes, be able to communicate with Windows PCs.[7] Therefore, even as Apple has

2. *See* Simeon Simeonov, *Metcalfe's Law: More Misunderstood Than Wrong?*, HIGH CONTRAST, July 26, 2006, http://simeons.wordpress.com/2006/07/26/metcalfes-law-more-misunderstood-than-wrong.

3. *See* ANDREW ODLYZKO & BENJAMIM TILLY, DIGITAL TECH. CTR., UNIV. OF MINN., A REFUTATION OF METCALFE'S LAW AND A BETTER ESTIMATE FOR THE VALUE OF NETWORKS AND NETWORK INTERCONNECTIONS (2005), http://www.dtc.umn.edu/~odlyzko/doc/metcalfe.pdf.

4. JONATHAN E. NUECHTERLEIN & PHILIP J. WEISER, DIGITAL CROSSROADS 5-6 (paperback ed. 2007).

5. Dion Hinchcliffe, *Web 2.0's Real Secret Sauce: Network Effects*, July 15, 2006, SOCIAL COMPUTING MAGAZINE, http://web2.socialcomputingmagazine.com/web_20s_real_secret_sauce_network_effects.htm.

6. Roman Beck has written on this topic, observing that

> Like other computing technologies, competing standards battle acrimoniously on the market to read a critical mass to take over the marker. Once established, a dominant standard becomes even stronger due to positive feedback effects while the "outgunned" standards lose even more market share. In extreme cases, a monopoly can be established, also known as the "winner takes it all" in increasing returns networks. Despite strong positive feedback effects accelerating the diffusion of dominating standards, stable equilibriums with several coexisting standards can also emerge. A prominent example of a stable oligopoly is the operating system software market for computers with Microsoft Windows as the dominant standard and Linux, as well as Mac OS for Apple Macintosh as sturdy clusters. Although Microsoft extended the positive feedback effects of its standards by adding complementary applications (e.g., by integrating Windows Internet Explorer), it was not able to displace its competitors completely. The former example indicates that standards on network effect markets can tend to lead to natural monopolies.

ROMAN BECK, THE NETWORK(ED) ECONOMY: THE NATURE, ADOPTION AND DIFFUSION OF COMMUNICATION STANDARDS 60 (2006).

7. Compatibility in computer systems is analogous to interconnection in

moved away from its proprietary AppleTalk networking technology in favor of the open standard TCP/IP,[8] it has continued to improve its support of SMB,[9] a technology used by Windows. The widespread support for SMB on non-Microsoft platforms is an example of SMB benefitting from a direct network effect.[10] Network effects also account for the rise of the Internet itself.[11] Although most software applications do not benefit from this kind of network effect,[12] applications that engage in any form of communication do. For instance, Microsoft Office benefits from the network effect of large numbers of computers being able to view and edit its documents.[13] In the 1980s, when word

---

communications systems. *See* BECK, *supra* note 6, at 55-56 (describing different forms of "compatibility" and noting that in some instances the dominant platform (Windows) is compatible with the smaller one (Macintosh)); KLAUS W. GREWLICH, GOVERNANCE IN "CYBERSPACE": ACCESS AND PUBLIC INTEREST IN GLOBAL COMMUNICATIONS 148 (1999) ("Without interconnection a small network would be severely disadvantaged relative to a large one").

8. Shelly Brisbin, *All Roads Lead to Rendezvous*, MACWORLD, July 2, 2003, http://www.macworld.com/article/26841/2003/07/allroadstorendevous.html (

> AppleTalk is a Mac-only technology in a cross-platform world. These days, most network hardware, PCs, and printers–as well as other devices don't support AppleTalk. They use TCP/IP, the language of the Internet. Universal TCP/IP support provides both seamless communication with the Internet and a single networking medium that all computer makers, software vendors, and users can agree on. As a result of this, Apple–while continuing to support AppleTalk in OS X–has started to focus on TCP/IP.

).

9. Daniel Drew Turner, *Apple Preps Early Release for Jaguar*, EWEEK, July 3, 2002, http://www.eweek.com/c/a/Past-News/Apple-Preps-Early-Release-for-Jaguar ("Jaguar will include new support for cross-platform standards such as . . . SMB . . . .").

10. *See generally* Chris Hertel, *Samba: An Introduction*, Nov. 27, 2001, http://us3.samba.org/samba/docs/SambaIntro.html.

11. B.G. KUTAIS, INTERNET POLICIES AND ISSUES 224-25 (2002). The Internet itself has also been described as a monoculture. *See* WILLIAM R. CHESWICK, STEVEN M. BELLOVIN & AVIEL D. RUBIN, FIREWALLS AND INTERNET SECURITY: REPELLING THE WILY HACKER 112 (2003).

12. For instance, a non-networked computer game does not derive its primary value from a large installed base. By contrast, game consoles themselves (as platforms) are subject to various kinds of network effects. *See* David S. Evans, *The Antitrust Economics of Multi-Sided Platform Markets*, 20 YALE J. ON REG. 325, 364 (2003).

13. STAN J. LIEBOWITZ & STEPHEN E. MARGOLIS, WINNERS, LOSERS, AND MICROSOFT: COMPETITION AND ANTITRUST IN HIGH TECHNOLOGY 181 (Independence Institute 2001); Knowledge@Wharton, *Rivals Set Their Sights on Microsoft Office: Can They Topple the Giant?*, http://knowledge.wharton.upenn.edu/article.cfm?articleid=1795 (quoting Kendall Whitehouse as saying

> [Y]ou bought Word because people send you Word files and you need to edit them… The one thing that's critical [in a competing product] is the ability to read and write those files. If you have a Mac [using iWork] that can read and write Word and PowerPoint files, then your ability to switch [away from Office] becomes a lot easier. The differentiator becomes user interface, speed and stability.

(bracketed text in the original.)) To be sure, Microsoft probably gained an initial edge due to its having a superior product. LIEBOWITZ & MARGOLIS, *supra*, at 180-200 (arguing that

processors were used primarily to create printed documents, there was much more competition among different technologies than exists today. Today, it is just as important that people be able to email each other documents and be sure that they were able to be viewed.[14]

2. Indirect Network Effects

Although technologies such as Java that have sought to lessen the importance of desktop applications have not lived up to expectations,[15] the recent rise of web-based applications has been very rapid. For some users, web-based applications offer advantages over desktop software in categories such as email.[16] In the near term, however, desktop software is likely to remain important.[17] The overwhelmingly popular choice for desktop operating systems is Microsoft's Windows, and the majority of new applications are written for that platform.[18] This remains a significant advantage for the Windows platform, which continues to benefit from indirect network effects that reinforce its popularity.

Microsoft's dominance in word processors came about through the quality of their products, and containing charts showing the reduction in the number of major players in the word processing market). *But see* Ed Foster, *The Gripelog: How Did Word Perfect Go Wrong?*, Dec. 27, 2007, INFOWORLD.COM, http://weblog.infoworld.com/gripeline/archives/2007/12/how_did_wordper.html (quoting one reader as writing that "MS Office crushed its competition for one reason and one reason ONLY—undocumented application programming interfaces".).

14. LIEBOWITZ & MARGOLIS, *supra* note 13, at 181.

15. Andy Johnson-Laird, *Looking Forward, Legislating Backward?*, 4 J. SMALL & EMERGING BUS. L. 95, 115 (2000) (

> Created by Sun Microsystems . . . Java is an object-oriented programming language with goals that include the ability to write programs that will run on many different computers. This goal was dubbed "Write Once, Run Anywhere," and while being an admirable goal, it has not been attained yet--the epithet "Write Once, Debug Everywhere" is unfortunately more appropriate.

); RUBICON CONSULTING, GROWTH OF WEB APPLICATIONS IN THE US: RAPID ADOPTION, BUT ONLY WHEN THERE'S A REAL BENEFIT (2007), http://www.rubiconconsulting.com/downloads/whitepapers/Rubicon_-_Rising_adoption_of_web_applications.pdf (adoption of web applications is very rapid, although use varies among kind of application).

16. Brad Stone, *Firms Fret as Office E-Mail Jumps Security Walls*, N.Y. TIMES, Jan. 11, 2007, § A, at 1.

17. BOB BAXLEY, MAKING THE WEB WORK: DESIGNING EFFECTIVE WEB APPLICATIONS 28 (2002); Martin Lamonica, *Ray Ozzie's Quiet Revolution at Microsoft*, ZDNET, May 1, 2007, http://news.zdnet.com/2100-3513_22-6180539.html (Microsoft's Ray Ozzie on the continuing importance of the desktop in an era of web applications).

18. The total market share of operating systems for personal computers of various kinds of Windows has been estimated to be around 90%, with the Macintosh at around 7.3%, and Linux and "other" rounding things out. NEOWIN.NET, OPERATING SYSTEM MARKET SHARE (2007), http://staff.neowin.net/slimy/dec2007.pdf. On February 23, 2008, the website Versiontracker.com registered 52 updates its directory of Windows applications, and only 18 updates for its directory of Mac OS X applications. *See* Versiontracker, http://www.versiontracker.com (snapshot of site from Feb. 23, 2008 on file with author).

Indirect network effects are advantages popular platforms enjoy other than those directly related to interconnection. A computer platform becomes more valuable if many third-party applications are written for it, but developers will only create applications for platforms that have many users. A self-reinforcing cycle can develop as users gravitate towards platforms with the most applications, and application developers gravitate towards platforms with the most users. As this cycle makes it difficult for a new entrant to create a software platform, it has been called "the applications barrier to entry."[19] The past success of a software platform therefore contributes to its future success, in an example of "path dependence."[20]

Switching costs also contribute to the continued popularity of a platform.[21] Once a user has invested time and money in a particular platform, the costs of switching to a new platform may outweigh any gain to be had from adopting a new platform.[22]

A popular platform enjoys a few other advantages besides software availability and direct switching costs. For example, it is easier to obtain support and assistance for technologies that are widely used,[23] and new employees may need less training.[24] Companies that provide popular

19. Kenneth G. Elzinga, David S. Evans, & Albert L. Nichols, *U.S. v. Microsoft Corp.: Remedy or Malady?*, *in* MICROSOFT, ANTITRUST, AND THE NEW ECONOMY 154 (David S. Evans, ed., 2002). Note that the current prevalence of three video game consoles (the Wii, XBox 360, and Playstation 3), each of which has a software library incompatible with the others, demonstrates that the desire for software compatibility is not sufficient by itself enough to create a platform monopoly. *Cf.* BECK, *supra* note 6.

20. *See* PAUL J. EDWARDS ET. AL, UNDERSTANDING INFRASTRUCTURE: DYNAMICS, TENSIONS, AND DESIGN 17 ("Path dependence refers to the 'lock-in' effects of choices among competing technologies. It is possible, following widespread adoption, for inferior technologies to become so dominant that superior technologies cannot unseat them in the marketplace.").

21. *See generally* Michael L. Katz & Carl Shapiro, *Antitrust in Software Markets*, *in* COMPETITION, INNOVATION AND THE MICROSOFT MONOPOLY: ANTITRUST IN THE DIGITAL MARKETPLACE 29, 31-34 (Jeffrey A. Eisenach & Thomas M. Lenard eds., 1999).

22. Daryl Lim, *Copyright Under Siege: An Economic Analysis of the Essential Facilities Doctrine and the Compulsory Licensing of Copyrighted Works*, 17 ALB. L.J. SCI. & TECH. 481, 506 fig.2 (2007). It has recently been suggested that part of Apple's strategy with the iTunes App Store (which sells applications that can run on the iPhone and the iPod Touch) is to create switching costs for its users. Sean Devine, *Inconsequential Apps Used by Many People Increase Stickiness*, DEAL RANGE, Jan. 4, 2009, http://dealrange.typepad.com/deal_range/2009/01/inconsequential-apps-increase-stickiness-if-everyone-uses-them.html; Sean Devine, *The App Store: First Comes Power*, DEAL RANGE, Jan. 3, 2009, http://dealrange.typepad.com/deal_range/2009/01/the-apple-app-store-and-pricing-power.html.

23. For example, of the 48 businesses listed in the 2008 Yellow Pages for Boulder, Colorado, under "Computers–Service & Repair," only seven advertise expertise in Macintosh computers, and none in Linux. DEX: OFFICIAL DIRECTORY–BOULDER 226-31 (2007).

24. ICT Hub Knowledgebase, Software Standardization, http://www.icthubknowledgebase.org.uk/softwarestandardisation ("If you standardise [sic] on software which is widely used in the outside world, it will make sharing information easier and

technologies are unlikely to go out of business, leaving their customers "orphaned."[25] Furthermore, it is easier for IT purchasers to justify investments in widely-used platforms.[26]

3. Standardization

Communication technologies and computer software are also subject to pressures of standardization. Among other things, a "standard" is a technology or method that is selected because there is an advantage to picking just one.[27] There may not be any advantage at all to a standard, other than the fact that it *is* a standard. The metric system is a standard. So is the gauge of railroad tracks,[28] household electric voltage, the custom of driving on the right-hand side of the street in most countries, the use of certain formulations of gasoline, and AM radio. Standards may exist because of law, a dominant marketplace position, or habit. By virtue of its dominant marketplace penetration, Microsoft's software has become a de facto standard for home and business use. Standards have wide-ranging benefits. Railcars can move easily between different railroad tracks that share a standard gauge.[29] Different manufacturers create AK-47 rifles, and different manufacturers produce the 7.62 mm ammunition they use. Because people in a country all drive on the same side of the street, accidents are reduced.[30] There are human

---

reduce training needs of new staff. Many organisations [sic] have standardised [sic] on Microsoft Office for this reason.").

25. *See* About.com: Desktop Publishing, Where Are They Now? Finding Software Orphans, http://desktoppub.about.com/library/weekly/aa033199.htm.

26. *See* LOIS KELLY, BEYOND BUZZ: THE NEXT GENERATION OF WORD-OF-MOUTH MARKETING 115 (2007) ("The classic anecdote, 'You'll never get fired for buying IBM,' was based on anxieties. If I buy a little-known technology and it bombs, I'll be fired for it. If I hire IBM and the technology fails, IBM will be blamed, not me.")

27. *See generally* Yesha Y. Sivan, *Knowledge Age Standards: A Brief Introduction to Their Dimensions*, *in* INFORMATION TECHNOLOGY STANDARDS AND STANDARDIZATION 1-18 (2000).

28. For an informed discussion of the forces at work in settling on a standard gauge for railroads, see GEORGE HILTON, AMERICAN NARROW GAUGE RAILROADS (1990).

29. In order to move freight from Russia into Germany, Hitler's Germany had to offload freight from cars using one railroad gauge onto cars of another gauge. ALBERT L. WEEKS, RUSSIA'S LIFE-SAVER: LEND-LEASE AID TO THE U.S.S.R. IN WORLD WAR II 91 (2004).

30. Sometimes, countries switch their traffic directionality from one side of the street to the other. *See* Paul Friedlander, *H-Day is Coming to Sweden*, N.Y. TIMES, Aug. 20, 1967, § 10, at 1 (describing the transition from left side of the road driving to right side of the road driving that was to take place in Sweden on September 3, 1967); *see also* Scott Berinato, When Voice Becomes Data, CSO ONLINE, Sept. 21, 2006, http://www.csoonline.com.au/index.php/id;924061898;fp;16;fpid;0 (switch from driving on one side of the road to another in Sweden had no measurable effect on the accident rate in the long term). The arbitrariness of the choice of which side of the road to drive on can be seen by the importance placed on it by the "xenophobic, capricious, [and] superstitious" General Ne Win, former president of Burma (now Myanmar). *General Ne Win*, DAILY TELEGRAPH, Dec. 6, 2002, at 31. In addition to having been observed "in the middle of the night, dressed as a

benefits, as well. When a standard exists and is widely adopted, there is a greater pool of human knowledge to draw on regarding that standard. A Microsoft-certified engineer has better employment prospects than a computer specialist who knows only IBM System z servers.[31]

There are many reasons why markets for computer operating systems are subject to pressures of standardization. Purchasers' lives are made easier, because they don't have to worry about picking the "right" system. The old saying that "you never get fired for buying an IBM" today applies to Microsoft.[32] Users only have to be trained on one kind of system, and there are fewer worries about compatibility. Nevertheless, as discussed below, the homogenizing pressures of technology standards can have negative consequences.

B. "Monocultures" in Agriculture and Technology

The analogy between agriculture (and biology generally) and software is pervasive in discussions of computer security. The term "monoculture" itself has an agricultural origin, and many computer security threats have biological names, like "worms" or "viruses." The perceived threat from excess homogeneity in software is likened to the threat to crops and species from insufficient genetic diversity.[33] Therefore, it will be helpful in the understanding of the above-described economic effects to understand how they might apply to agriculture, as well as to complex technologies and computer networks.

The economic and cultural pressures on agricultural tend to create

---

king, walking backwards over a bridge in Rangoon, apparently on the advice of his soothsayers[,]" he directed his nation to begin driving on the right hand side of the road, instead of the left. *Id.*

31. On February 23, 2008, there were eight job postings in the "Computer/Software" category with "System z" as a keyword, but 152 job postings with "MSCE" as a keyword. Monster.com, Job Search, http://www.monster.com (Feb. 23, 2008) (on file with author).

32. BETH FOSS ET AL., IS CORPORATE AMERICA READY FOR OPEN SOURCE SOFTWARE? (2002), http://www.danmccreary.com/Open_Source_Report.pdf.

33. Jim Chen, *Webs of Life: Biodiversity Conservation as a Species of Information Policy*, 89 IOWA L. REV. 495, 505 (2004) ("

> Though the biosphere and the world of human-generated information teem with diversity, both are slouching toward uniformity. Driven by the value that inheres in networks and in the cost-reducing benefits of uniform operating standards, the quest for univeral [sic] interoperability in electronic communication and commerce has come close to realization. This quest has come dangerously close, in fact, for uniformity carries a cost of its own, in the natural realm as well as the electronic. "Never before in human history have there been comparable monocultures … of billions of genetically similar plants covering millions of acres across whole continents."

) (citing H. Garrison Wilkes, *Plant Genetic Resources over Ten Thousand Years: From a Handful of Seed to the Crop-Specific Mega-Gene Banks*, *in* SEEDS AND SOVEREIGNTY: THE USE AND CONTROL OF PLANT GENETIC RESOURCES 67, 73 (Jack R. Kloppenburg, Jr. ed., 1988)).

food economies heavily dependent on particular crops.[34] In the first instance, only certain plants are suitable for domestication.[35] However, among the possible candidates, species that were domesticated in one place were not domesticated in another.[36] In many instances, one plant is domesticated while its equally suitable near relatives are not.[37] The choice of exactly which plant to domesticate may therefore be seen as somewhat arbitrary—a standard, like what side of the road to drive on or the length of a meter.

While the reasons one plant species may be domesticated and another not may be complex, it is easier to continue to grow already-domesticated crops than to domesticate new ones. This demonstrates path dependence. Crops can also be viewed as being subject to indirect network effects, because many agricultural "applications," from particular formulations of pesticide to planting cycles, run on top of agricultural "platforms." Technological advances have greatly increased the pressure to rely on only a few crops.[38] While historically, farmers had to grow a wider variety of crops in order to effectively exploit their soil, modern fertilizers have limited the need for that kind of crop rotation.[39] Furthermore, while genetic diversity within a species was once the norm, commercial seed distribution has homogenized crops to an unprecedented degree. As a result of these pressures, currently, "[a] mere dozen species account for over 80 percent of the modern world's annual tonnage of all crops."[40] Michael Pollan has described how commercial pressures create incentives for farmers to rely heavily on monocultures,[41]

---

34. The same pressures apply to the agricultural use of animals. However, I will limit my discussion to plant-based agriculture.

35. JARED DIAMOND, GUNS, GERMS, AND STEEL 132-133 (2nd ed. 1999).

36. *Id.* at 133.

37. *Id.* at 134.

38. Kyuma, *infra* note 50, at 68 (

> Three technological factors pushed farmers toward monoculture. The first is mechanization, which enabled farmers to expand their farms. ... With a heavy investment in large, specialized machinery, the farmer has a strong incentive to grow only the crop for which the machinery was designed.
>
> The improvement of crop varieties is the second force pushing farmers toward monoculture…. By concentrating on a single, improved crop, the farmer can exploit its traits to the utmost.
>
> The third technological factor underlying the shift toward monoculture is the development of chemicals, i.e., fertilizers and pesticides, which have made it possible to grow a single crop year after year….

).

39. GUNS, GERMS, AND STEEL, supra note 35, at 134.

40. *Id.* at 132.

41. MICHAEL POLLAN, THE BOTANY OF DESIRE: A PLANT'S-EYE VIEW OF THE WORLD 231 (2002) ("Monoculture is where the logic of nature collides with the logic of

and his thinking on agricultural issues generally has been extremely influential.[42]

C. Negative Side Effects

Above, this Note briefly covered how different economic and technological phenomena work together to create a standard technology or product. In most cases, this standardization leads to great economic efficiency. However, the homogenizing results of these phenomena cause negative "externalities."

An externality is a cost or benefit to a party not involved in a transaction that is caused by the transaction.[43] Economic regulation is often focused on limiting externalities that have a negative social impact. For instance, A and B may enter into a transaction that is mutually beneficial. But that transaction may impose costs on C that exceed the benefit to A and B together. In those instances, the transaction is a net loss to society, and government regulation may seek to modify or prevent it. Alternatively, the transaction may be a net benefit to society, but equity concerns may motivate the government to limit the costs borne by third parties such as C. Markets that prominently feature negative externalities justify regulatory intervention.[44] There may be compelling reasons that lead to the creation of technology standards and software monocultures, but there may also be negative externalities and costs associated with those processes must be acknowledged.

The creation or maintenance of a monopoly may be an example of a negative externality caused by transactions that create technology standards. Monopolies and monocultures are different creatures, but they may contribute to one another. A firm that manages to have one of its technologies become a standard may have monopoly control of that technology, which it can maintain through patent, copyright, or otherwise. A monopoly is a firm that has little or no competitive pressure, and is able to charge high prices for a product it has no incentive to improve.[45] Public utilities and communications regulation

---

economics . . . .").

42. Pollan's thinking has started to shape the national debate over food policy. For one prominent example, then-candidate Obama at one point remarked that he had just read an article by Pollan, and went on to say that our agriculture system is "creating monocultures that are vulnerable to national security threats, are now vulnerable to sky-high food prices or crashes in food prices, huge swings in commodity prices, and are partly responsible [for various health care problems]." *The Full Obama Interview*, TIME, Oct. 23, 2008, http://swampland.blogs.time.com/2008/10/23/the_full_obama_interview.

43. John A. Rothchild, *The Social Cost of Technological Protection Measures*, 34 FLA. ST. U. L. REV. 1181, 1198 (2007).

44. *Id.* at 1204-05.

45. Some markets, such as utilities or telecommunications, can be described as "natural

has traditionally been premised on the presumed negative consequences of allowing a monopolist to control a network that many depend on without check.[46] Despite some similarities, however, monopolies are a different concept than monocultures. A monopoly is a single firm that has excess market power—but it may provide many different, robust technologies. It is at its core an argument about a lack of diversity among firms. By contrast, a monoculture may be supported by a variety of firms—it is an argument about a lack of technological diversity. Nevertheless, technology monocultures may *tend* to produce business monopolies.[47] Therefore, to the extent that monopolies are undesirable for their own set of reasons, it may be desirable to limit the technology monocultures that may contribute to them.

The heavy reliance on any single commodity can have wide-ranging economic repercussions, as it creates a "bottleneck" and a single point of failure.[48] One key example is the 1970s oil crises, where the world's dependence on a single commodity for much of its energy needs showed that even a modern industrial economy could be surprisingly fragile.[49] A farmer's reliance on a single crop can also cause him problems.[50] While agricultural monocultures carry certain economic benefits, there are also attendant risks. Large-scale agricultural monocultures, though efficient,

---

monopolies" that exhibit traits that limit competition. *See generally* NUECHTERLEIN & WEISER, *supra* note 4, at 12-15. Once a physical network of wires or pipes is built, it may be uneconomical for a new entrant to build a duplicate network to compete with the first one—even though the new entrant may have a more efficient technology. In the case of operating systems or other software, the high cost of building a user base may be such a high initial, fixed cost that the current dominant player can be seen as having a natural monopoly. As mentioned *supra*, note 1, consideration of monopoly issues is beyond the scope of this Note.

46. *See* ALFRED E. KAHN, THE ECONOMICS OF REGULATION 28 (1988).

47. For example, business monopolies can be created if the vendor behind the technology monoculture is insulated from competition through patent or copyright law.

48. The fear that our complex economy can be brought down by a single weak point is widespread. *See* Frank J. Cilluffo et al., *Bad Guys and Good Stuff: When and Where Will the Cyber Threats Converge?*, 12 DEPAUL BUS. L.J. 131, 141-42 (1999/2000) (

> Modern societies are dependent upon critical infrastructures, such as telecommunications, electric power, health services, banking and finance, transportation, and defense systems, as they provide a comfortable standard of living. These systems are increasingly interdependent on one another and damage to one can potentially cascade and impact others - with single point failures being of great concern.

).

49. MICHAEL CARR, NEW PATTERNS: PROCESS AND CHANGE IN HUMAN GEOGRAPHY 367 (1997).

50. K. Kyuma, *Protection of the Environment: Sustained Agriculture, Sustained Ecosystems*, *in* PHOSPHORUS REQUIREMENTS FOR SUSTAINABLE AGRICULTURE IN ASIA AND OCEANIA 57, 68 (1990) (It is seen as a problem that "monoculture, which is widely practiced in the United States as an efficient means to attain high crop productivity, may not be compatible with the other goal of a good farming system, i.e., sustained production through protection of the environment.")

require more modern agricultural products such as chemical pesticides than do mixed plantings.[51] Not only is the farmer more economically vulnerable to swings in the price of the crop, but everything he grows becomes susceptible to the same pests and diseases. Because 19th century Ireland depended on the potato for much of its nutrition, when the potato blight struck Ireland in 1845, mass starvation resulted.[52] Indeed, Pollan writes that "it was not the potato so much as potato monoculture that sowed the seeds of Ireland's disaster."[53] Genetic homogeneity carries risks outside of agriculture, as well. A genetically homogenous human population is more susceptible to endemic disease, and an ecosystem with many different species is considered more robust than a simpler ecosystem.[54]

Monocultures in crops, commodities, or technologies create economic fragility. They may contribute to the creation of a monopoly that is able to charge higher prices for its products, and when an economy depends heavily on a monocultural bottleneck for an important activity, threats to that single item can bring an entire economy to its knees.

D. Software Monoculture

While it is hard to argue against the virtues of standardization when it comes to light bulbs or soda can sizes, as with agriculture, the standardization of desktop operating systems has had certain negative side effects. By analogy with agriculture, the prevalence of Microsoft's products has been called a "software monoculture."[55] Just as large-scale plantings of single crops are susceptible to being wiped out by a single disease, a software monoculture can lead to a majority of the world's computers simultaneously becoming susceptible to the same security vulnerability; and just as biodiversity contributes to an ecosystem's robustness,[56] a more diverse software "ecosystem" may be less susceptible to security flaws.

It is important to bear in mind that Microsoft is neither the only company that has had a dominant position in a software market, nor the only example of such dominance leading to widespread security incident.

---

51. *See* POLLAN, *supra* note 41, at 225-26.

52. CORMAC Ó GRÁDA, BLACK '47 AND BEYOND: THE GREAT IRISH POTATO FAMINE IN HISTORY, ECONOMY, AND MEMORY 13, 203 (1999).

53. *See* POLLAN, *supra* note 41, at 230.

54. Charles C. Mann, 1491: NEW REVELATIONS OF THE AMERICAS BEFORE COLUMBUS 112-18 (Vintage 2006) (2005).

55. The term is certainly loaded. Describing Microsoft Windows as a "standard" has neutral or even positive connotations, while the term "monoculture" is extraordinarily negative. Nevertheless, it is the usual term used when discussing this issue.

56. Chen, *supra* note 33, at 549-50.

The Morris worm of 1988, the first computer worm to propagate itself over the Internet, took advantage of security vulnerabilities in sendmail and other programs to hobble the Internet.[57] It infected 6,000 Unix computers, crashed 10% of the Internet, and caused $100 million in damage.[58] The Morris Worm incident led to the creation of the Computer Emergency Response Team Coordination Center.[59] The widespread use of MoveableTypes's blogging software has made it easier for spammers to take advantage of weakness and post spam comments on blogs.[60] The Internet's dominant web serving software,[61] the open source program Apache, has occasionally been subject to a security vulnerability that put the majority of the world's web sites in danger simultaneously.[62] As John Quarterman writes, "[m]onoculture is not limited to operating systems or application software, nor even to application servers. Monoculture can exist in network routers as well. And if an exploit becomes widely known for a widely used router, big problems could result."[63] These examples show that the issue of monoculture is pervasive in computer technology. Nevertheless, most of the attention given to software monocultures has focused on Microsoft,[64] and the majority of the most widespread and severe security incidents have affected Microsoft products.[65]

One of the first Microsoft vulnerabilities to receive widespread attention was the "I love you" virus, which in late May 2000 spread rapidly throughout the world by taking advantage of flaws in VBScript, a simple programming language included in all versions of Microsoft Windows since 1998.[66] The "I love you" virus destroyed data on millions

57. MICHAEL ERBSCHLOE, TROJANS, WORMS, AND SPYWARE: A COMPUTER SECURITY PROFESSIONAL'S GUIDE TO MALICIOUS CODE 35, 36 (2005).

58. Thomas M. Chen & Jean-Marc Robert, *Worm Epidemics in High-Speed Networks*, 37 COMPUTER 48, 49 (2004).

59. *Id.*

60. Posting of Jacques Distler, to Musings, Software Monoculture, http://golem.ph.utexas.edu/~distler/blog/archives/2003_10.shtml#s000236 (Oct. 15, 2003).

61. For current statistics on Apache's market share, see Netcraft, Web Server Survey Archives, http://news.netcraft.com/archives/web_server_survey.html.

62. LWN.net, The Apache Vulnerability, Full Disclosure, and Monocultures, http://lwn.net/Articles/2756/ (June 18, 2002).

63. John Quarterman, *Managing Internet Risk in a Scale-Free World*, *in* SCIENCE AND SECURITY: INFORMING NEW ZEALAND 79, 81 (2005).

64. Amit Singh, A Taste of Computer Security, Unix vs. Microsoft Windows, http://www.kernelthread.com/publications/security/uw.html.

65. *See, e.g,* MARK F. GRADY & FRANCESCO PAIRISI, THE LAW AND ECONOMICS OF CYBERSECURITY 119 (2006) (a discussion of monoculture immediately raising the issue of Windows dominance).

66. An earlier program called "Melissa" was also very fast-spreading, but was relatively benign compared with "I love you." John Markoff, *April 30-May 6; An "I Love You" Virus Becomes Anything But*, N.Y. TIMES, May 7, 2000, § 4, at 2; John Markoff, *A Disruptive Virus Invades Computers Around the World*, N.Y. TIMES, May 5, 2000, § A, at 1.

of computers, and was the first volley in what was to be several years of fast-spreading and damaging computer viruses and worms. Another incident leading to greater consciousness of the problem of a security vulnerability being discovered and exploited on a large number of computers simultaneously was the "Code Red" worm (named in part because the researchers who identified it drank Code Red Mountain Dew to "fuel[] their efforts").[67] The Code Red worm demonstrated "the speed at which a malicious exploit of a ubiquitous software bug can incapacitate host machines."[68] Drawing a biological analogy, the authors also noted that "[a]s is the case with biologically active pathogens, vulnerable hosts can and do put everyone at risk."[69] High-profile computer worms demonstrated the risks and costs of a software monoculture could be very high. Insecure software was widely deployed to end users who may not have had much computer expertise, but "machines operated by home users or small businesses (hosts less likely to be maintained by a [sic] professional systems administrators) [were] integral to the robustness of the global Internet."[70] This widespread deployment of insecure software operated by nonexpert users led to several years of high-profile security exploits, as names like "Nimba," "Blaster," and "Slammer" joined the rogue's gallery with "I love you" and "Code Red."[71]

A more exotic phenomenon facilitated by the Microsoft software monoculture is the "zombie botnet." A "botnet" is a network of computers that have been infected by some computer worm that then connects them, unbeknownst to their owners, to a network of other computers that have also been so infected.[72] Each infected computer is known as a "zombie," and the botnet is then indirectly controlled by its "owner" to perform some nefarious act.[73] Common uses of botnets are Distributed Denial of Service attacks (DDoS), whereby many thousands of computers simultaneously try to access some resource on a target computer, overloading and perhaps damaging the target.[74] Another use is

67. David Moore, Colleen Shannon, & K. Claffy, *Code-Red: A Case Study on the Spread and Victims of an Internet Worm*, PROCEEDINGS OF THE 2ND ACM SIGCOMM WORKSHOP ON INTERNET MEASUREMENT 2002, at 273-74 (2002).

68. *Id.* at 282.

69. *Id.*

70. *Id.*

71. *See generally* Evan Cooke, Z. Morley Mao, & Farnam Jahanian, *Hotspots: The Root Causes of Non-Uniformity*, *in* SELF-PROPAGATING MALWARE, INT'L CONFERENCE ON DEPENDABLE SYS. & NETWORKS 179-80 (2006) (listing some of the "most significant worms to strike the Internet").

72. Lilian Edwards, *Dawn of the Death of Distributed Denial of Service: How to Kill Zombies*, 24 CARDOZO ARTS & ENT. L.J. 23, 26-27. (2006).

73. *Id.*

74. *Id.*

to send spam–it is difficult to block unsolicited email based on its source, when its source is thousands of seemingly unrelated computers located around the world.[75] The magnitude of botnets is hard to overstate. In 2005, Dutch police discovered and managed to shut down a botnet comprised of 1.5 million infected zombies,[76] and Vint Cerf has estimated that one in four computers connected to the Internet are part of one or more botnets.[77] Peter Gutmann, a computer scientist at the University of Auckland, recently stated that the "Storm" botnet could be viewed as the most powerful supercomputer in the world.[78]

II. RESPONSES TO MONOCULTURE

Concerns about the negative consequences of monocultures and dependence on bottleneck technologies or commodities are very domain-specific, although certain common features can be noted. The responses either seek to do away with the monoculture by increasing diversity in some way, or give the government a regulatory role in limiting the harms caused by the monoculture. In crops, different planting and crop rotation techniques can limit the bad effects of monocultures.[79] To counter the heavy reliance on oil, energy independence has become a watchword not only among environmentalists, but among those concerned with national security.[80] For most of the twentieth century, telecommunications regulation was premised on the assumption that telecommunications networks are natural monopolies.[81] More recently, the net neutrality

75. Jacqui Cheng, *Botnets Cause Significant Surge in Spam*, ARS TECHNICA, Oct. 30, 2006, http://arstechnica.com/news.ars/post/20061030-8111.html.

76. Gregg Keizer, *Dutch Botnet Bigger Than Expected*, INFORMATION WEEK, Oct. 21, 2005, http://www.informationweek.com/news/security/government/showArticle.jhtml?articleID=172303265.

77. Tim Weber, *Criminals "May Overwhelm the Web*," BBC NEWS, Jan. 25, 2007, http://news.bbc.co.uk/1/hi/business/6298641.stm.

78. *See* Insecure.org, World's Most Powerful Supercomputer Goes Online, http://seclists.org/fulldisclosure/2007/Aug/0520.html (archiving a message of computer science professor Peter Gutmann); Sharon Gaudin, *Storm Worm Botnet More Powerful than Top Supercomputers*, INFORMATIONWEEK, Sept. 6, 2007, http://www.informationweek.com/news/internet/showArticle.jhtml?articleID=201804528.

79. *See* James F. Power, *Legumes and Crop Rotations*, *in* SUSTAINABLE AGRICULTURE IN TEMPERATE ZONES 178, 198 (Charles A. Francis et al. eds., 1990).

80. *See* Stephen D. Solomon, *For National Security, Get Off Oil*, SCIENTIFIC AM., Oct. 2008, http://www.sciam.com/article.cfm?id=is-oil-a-threat (former CIA director R. James Woolsey sees oil dependence as a national security threat); Set America Free Coalition, An Open Letter to the American People, http://www.setamericafree.org/openletter.htm ("our present dependency creates unacceptable vulnerabilities. In Iraq and Saudi Arabia, America's enemies have demonstrated that they can advance their strategic objective of inflicting damage on the United States, its interests and economy simply by attacking critical overseas oil infrastructures and personnel.").

81. *See* NUECHTERLEIN & WEISER, *supra* note 4, at 55.

movement has advocated the regulation of ISPs in order to prevent them from becoming Internet gatekeepers.[82] The responses to perceived excess homogeneity in software, and its attendant negative consequences, have been similarly varied, ranging from regulation of the dominant software provider designed to increase interoperability, to a modification of tort and contract law principles, to more pragmatic approaches designed to better protect important computer systems from security vulnerabilities.

A. Geer

In 2003, a report titled "CyberInsecurity: The Cost of Monopoly–How the Dominance of Microsoft's Products Poses a Risk to Security" was published.[83] The report's principal author, Daniel Geer, was shortly thereafter fired from his position at @Stake, a computer security firm with ties to Microsoft.[84] This widely-publicized firing helped make the report (whose thesis was attention-grabbing in itself) famous. More than any other document, Geer's report kicked off the monoculture debate.[85] Although primarily focused on perceived problems with Microsoft's engineering practices, the report explored the risks of software monocultures generally. For instance, it notes that "[a] monoculture of networked computers is a convenient and susceptible reservoir of platforms from which to launch attacks . . . [t]his susceptibility cannot be mitigated without addressing the issue of that monoculture."[86] It further notes that "[t]he NIMDA and Slammer worms that attacked millions of Windows-based computers . . . 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."[87]

However, the bulk of Geer's argument is focused on problems specific to Microsoft.[88] He argues that certain engineering practices of Microsoft exacerbate network effects and create a level of consumer lock-in that would not otherwise exist.[89] He explains that Microsoft tightly

82. SaveTheInternet.com, Frequently Asked Questions, http://savetheinternet.com/=faq ("The nation's largest telephone and cable companies—including AT&T, Verizon, Comcast and Time Warner—want to be Internet gatekeepers, deciding which Web sites go fast or slow and which won't load at all.") (last visited May 5, 2009).

83. DAN GEER ET. AL., CYBER *IN*SECURITY: THE COST OF MONOPOLY (2003), http://www.ccianet.org/papers/cyberinsecurity.pdf (emphasis in original).

84. *See* Ellen Messmer, *Oh Dan Geer, Where Art Thou?*, NETWORKWORLD, Dec. 22, 2003, http://www.networkworld.com/weblogs/security/003879.html.

85. *See Warning: Microsoft "Monoculture"*, WIRED, Feb. 15, 2004, http://www.wired.com/politics/security/news/2004/02/62307.

86. GEER ET AL., *supra* note 83, at 7.

87. *Id.* at 10.

88. *See id.*

89. *Id.* at 13.

integrates its operating system and its application software in ways that give its own applications significant advantages, arguing that it uses "inter-module interfaces so complex, undocumented and inaccessible"[90] that no one outside Microsoft can effectively exploit them. He also argues that Microsoft integrates certain components, such as its Internet Explorer software, more deeply into the operating system than necessary from an engineering perspective, thereby making it difficult to replace or replicate Microsoft software or any of its components.[91] Geer argues that "[t]ight integration of applications and operating system achieves user lock-in by way of application lock-in. It works."[92] In other words, from a business perspective, Microsoft's engineering strategy has been a very successful means of holding on to customers.

An unintended side effect of Microsoft's engineering strategy, however, has been to increase the complexity of its products. As Geer points out, "[t]he central enemy of reliability is complexity."[93] By achieving user lock-in through creating a high level of dependence between different pieces of software, Microsoft has created a software ecosystem that is both dominant (because difficult to switch away from or replace) and highly unreliable (because overly complex). He argues that "[a]bove some threshold level of code complexity, fixing a known flaw is likely to introduce a new, unknown flaw"[94] and that the Microsoft's code base is unlikely to ever become secure.[95]

Finally, while Geer notes an increased awareness of security issues in Microsoft at the time of his article's publication, he worries that certain solutions then pushed for by Microsoft, such as those known under the rubric of "Trusted Computing," could serve to increase Microsoft's dominance.[96] His solution to these problems is quite broad. For instance, he proposes requiring that Microsoft release comparable versions of its application software such as Office for Mac OS X and Linux before it is allowed to release Windows versions. He would also support requiring thorough and open documentation of Microsoft APIs, to allow better competition with its products.[97]

In the "Coda" section of the report, Geer writes that "[t]hese 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

90. *Id.* at 13.
91. *See id.*
92. GEER ET AL., *supra* note 83, at 13.
93. *Id.* at 14.
94. *Id.* at 15.
95. *Id.*
96. *See id.* at 16-17.
97. *See id.* at 18-19.

or AT&T."[98] It is true that dominant firms are often accused of tying their products together, and attempting to unfairly leverage a powerful position in one market into a powerful position in another market. Dominance by particular firms has long been a phenomenon of high technology and telecommunications markets. However, no company besides Microsoft has been accused using bad engineering practices to accomplish a kind of tying that has such severe negative security consequences. There are no other firms with "similar dominance under current circumstances,"[99] and new ones are unlikely to arise. In its specifics, Geer's monoculture argument is applicable to Microsoft alone.

B. Picker

Randal Picker agrees that insecure software can be a real problem, but he argues for what he sees as a more cost-effective response.[100] In the first instance, he does not disagree that the rise of the networked economy has been accompanied by regrettable side effects. He argues that just as networking computers together has given rise to positive externalities in the form of what Yochai Benkler has described as "shareable goods," it has also given rise to negative externalities.[101] He cites spam and phishing as prominent examples, and goes on to a broader discussion of problems of computer security.[102]

Picker does not address the arguments made by Geer that network effects and negative security consequences are exacerbated by specific engineering choices made by Microsoft. Neither does he deny that homogenous networks can have negative consequences. He writes that "there is a real downside to all of this connectivity: problems percolate quickly throughout an interconnected system, and problems that might have been just local disturbances end up everywhere."[103] He later continues that

> [t]he monoculture is another name for a homogenous, connected system. In the monoculture framework, heterogeneity is used as a barrier to the spread of a virus throughout a connected computer system. The anti-monoculture idea also taps into our sense of

98. GEER ET AL., *supra* note 83, at 20.

99. *Id.*

100. Randal C. Picker, *Cyber Security: Of Heterogeneity and Autarky*, (Univ. of Chicago Law Sch. John M. Olin Law & Economics Working Paper No. 22, 2004), *available at* http://ssrn.com/abstract=590927).

101. *Id.* at 1-5.

102. *Id.*

103. *Id.* at 12.

necessary biodiversity.[104]

Although he has many quibbles with the specifics of the monoculture argument as laid out by Geer and others, the thrust of his argument is quite simple: Even if Geer is right about the security consequences of software monoculture, a simpler and more cost-effective solution than forced heterogeneity is what he calls "autarky"—simply isolating important computer systems from the Internet, so that they are immune to the negative externalities associated with networking computers together.[105] Picker's argument does little, however, to address the concerns that remain for those machines that, for various reasons, must stay connected to the Internet.

C. Government Support of Open Source

Some governments have sought to counteract problems in the software market by adjusting their policies in ways that benefit alternatives—primarily open source software.[106] For instance, in 2001, several Brazilian municipalities began giving open source software preference.[107] There are many similar initiatives throughout the world.[108] These actions are undertaken for a variety of reasons, not all of which are specifically aimed at reducing software monoculture. But some actions, such as the Japanese government's recently-announced policy to promote open source software, are expressly designed to lessen their dependence on Microsoft software.[109] Government policies favoring software diversity, even if they are limited to shaping the government's own purchasing decisions, have the potential to reduce monoculture by sustaining alternative products that otherwise would not thrive in the marketplace.

These policies, however, are not without their critics. David S. Evans and Bernard J. Reddy argue that government preferences for open source software will likely cause more problems than they solve.[110] They

104. *Id.*

105. *Id.* at 6.

106. *See* ROBERT WILLIAM HAHN, GOVERNMENT POLICY TOWARD OPEN SOURCE SOFTWARE 4-5 (2002) ("Open source" software has many interesting characteristics and has license terms that prevent vendor lock-in. From the perspective of governments, however, sometimes its chief selling point is that it is not from Microsoft.)

107. *Id.* at 5.

108. For a frequently updated list of resources concerning the use of open source software in government, see California Environmental Protection Agency Air Resources Board, Government-Related Open Standards/Open Source Software Articles, http://www.arb.ca.gov/oss/articles/articles.htm.

109. Phil Hochmuth, *The Japanese Government Looks to Go Open Source*, LINUXWORLD, May 9, 2007, http://www.linuxworld.com/newsletters/linux/2007/0507linux2.html.

110. David S. Evans & Bernard J. Reddy, *Government Preferences for Promoting Open-*

note that they are

> aware of no compelling evidence that governments have special expertise in analyzing the software industry to effect solutions . . . . Whether 'open code' in any given situation is actually 'as powerful' as 'closed code' is an everyday business judgment that should be made by businesses, governments, and private users; it does not strike us as a policy issue that should be decided by bureaucrats or legislators, or even by lawyers and economists.[111]

If open source software (or any software alternative) has advantages, then government IT professionals and purchasers would be well-advised to carefully consider those products when making their purchasing decisions, as part of their business judgment.[112] Even skeptics of the monoculture argument acknowledge that it can be rational to take into account the effects of buying into or supporting a software monoculture when making a technology choice.[113]

But, since government decisions can have negative unintended consequences and can distort markets, governments should hesitate before fixing software preferences as law or policy.

D. Extension of Law

Several commentators argue that legal reform increasing liability for software vendors who ship insecure products may alleviate negative consequences of technology monoculture. As Robert W. Hahn and Anne Layne-Farrar note, "the liability rules governing the distribution and use of software remain unclear, even after some thirty years of debate."[114] A few scholars have introduced proposals to clarify those rules. They have generally noted that the current legal climate does not assign liability to those parties best able to bear it, and certain behaviors in the marketplace create negative externalities for third parties. Through various means, they propose to realign the economics of software security by internalizing negative externalities. By causing costs to be borne by those who create them, they attempt to define a legal and economic

---

Source Software: A Solution in Search of a Problem, 9 MICH. TELECOMM. & TECH. L. REV. 313, 394 (2003) ("The net effect is likely to be a reduction in the total 'externality' benefits of software.").

111. *Id.* at 393-94.

112. *See id.* at 394.

113. Greg Goth, *Addressing the Monoculture*, IEEE SECURITY AND PRIVACY, Nov/Dec 2003, at 8-10 (quoting John Carroll as saying "[i]t is good that consumers factor monoculture costs into their calculations when choosing a particular platform. It is not good to treat those costs as more important than any others.").

114. Robert W. Hahn & Anne Layne-Farrar, *The Law & Economics of Software Security*, 30 HARV. J.L. & PUB. POL'Y 283, 327 (2006).

environment more likely to result in secure software. After briefly touching on ideas of Pamela Samuelson and Jennifer A. Chandler, this section will focus in detail on Emily Kuwahara's approach.

1. Chandler, Samuelson

Jennifer A. Chandler has argued that given the unique nature of Distributed Denial of Service attacks,[115] only holding software vendors liable would properly internalize risks to those most able to prevent them. Unlike many other security problems, DDOS attacks can cause harm to computers that do not themselves have any security vulnerabilities—and the users whose computers *are* compromised may not suffer any economic loss. Increasing the liability for the creators of insecure software is therefore the only way to create incentives to prevent the harm. Chandler therefore proposes to create a new tort of "negligently creating an unreasonable risk of harm from third parties."[116]

In an article from the early days of the online revolution, Pamela Samuelson notes that the policy reasons explaining why information vendors are generally not held liable in the same way that products vendors are for defects or errors do not necessarily apply to software or electronic information.[117] Concerns about free expression have led courts to limit liability for defective or erroneous information to defamatory statements and situations where a person claims to have specialized knowledge (for instance, through malpractice actions against doctors or lawyers).[118] But some kinds of "information" seem more like products than like communications, and Samuelson observes that, in a case involving aeronautical charts, there is precedent for treating "information" as a product governed by standard liability rules.[119] She notes that in cases where an information product "behaves like a machine," courts are likely to apply products liability principles,

115. Distributed Denial of Service ("DDOS") attacks occur when a large number of computers simultaneously attempt to access resources on a remote computer. In one common scenario, a large number of computers are compromised by software that allows them to be remotely controlled by a malicious hacker. Those compromised computers then simultaneously send common network requests to a target computer system, overtaxing its ability to deal with them. A DDOS attack therefore allows a computer that is not itself subject to any particular security vulnerabilities to be brought down by a large network of computers that are. *See* Jennifer A. Chandler, *Security in Cyberspace: Combatting Distributed Denial of Service Attacks*, 1 U. OTTAWA L. & TECH. J. 231, 236 (2004).

116. *Id.* at 261. Much of her discussion concerns Canadian cases, although the principles discussed are applicable in American law.

117. Pamela Samuelson, *Liability for Defective Electronic Information*, COMMC'N OF THE ACM, Jan. 1993, at 21.

118. *Id.* at 21-22.

119. *Id.* at 23-24.

including, in some cases, strict liability.[120]

2. Kuwahara

Emily Kuwahara argues that product liability law could be extended to hold software vendors liable for defective products, provided the current liability disclaimers are invalidated and an exception is created to the economic loss rule.[121] Kuwahara observes that the "prevalence of viruses and worms on the Internet is astounding",[122] noting that an unprotected computer on the Internet has a 94% chance of being infected within an hour.[123] She writes, though, that the current state of case law suggest that recovery is not available against a software vendor such as Microsoft either in cases of extensive damage caused by a widespread security incident, such as the Slammer worm, or in situations where an individual brings an action after her personal computer has been hacked.[124] She offers a survey of the thinking about increased tort liability for software vendors, from Howard Schmidt, "who oppose[d] liability for software companies because it will raise costs and prices, stifle innovation, and lead to job cuts,"[125] to Bruce Schneier, "who strongly believes that the cost of insecure software is an externality that should not be borne by users, but by software companies."[126] She also discusses more exotic proposals, such as the creation of a new tort of "negligent enablement of cybercrime,"[127] or the creation of a code of professional practice for software engineers, which would open the door to malpractice actions.[128] She also notes the argument that Microsoft's dominant market places special burdens on it that wouldn't necessarily be shared by other software vendors. For instance, the Computer and Communications Industry Association issued a report that placed a "special burden . . . upon Microsoft because of [the] ubiquity of its product."[129]

Kuwahara goes on to detail a number of policy rationales for allocating risk to Microsoft particularly, including: compensation of victims; a lack of competition that reduces its incentives to increase its software's security; its superior ability to bear financial risk; the beneficial

120. *Id.* at 21.
121. *See* Emily Kuwahara, *Torts v. Contracts: Can Microsoft Be Held Liable to Home Consumers For Its Security Flaws?*, 80 S. CAL. L. REV. 997, 1030 (2007).
122. *Id*. at 1000.
123. *See id*.
124. *Id*. at 998-99.
125. *Id*. at 1001-02.
126. *Id*. at 1002.
127. Kuwahara, *supra* note 121, at 1003.
128. *Id*. at 1002-03.
129. *Id*. at 1007.

establishment of a standard of care in software design; the reluctance to allow Microsoft to use contract law to evade liability when consumers often have little choice but to use its products; and little actual bargaining ever occurs; the fact that liability insurance would likely remain available and affordable for all software companies; the fact that increased liability hasn't deterred innovation in other fields; and the fact that mere disclosure of software flaws does not offer consumers a sufficient remedy.[130]

At its core, her argument is that, because products liability has been successful in other areas of commerce, it is likely to be successful in software, as well. It is rooted in an assumption that software is best understood as a "product" or "good"[131] more similar to an automobile than a service.[132] Her alternative approach of a non-disclaimable statutory warranty offers a reasoned compromise to tort liability and addresses the imbalance in bargaining power between large software vendors and consumers. Her argument, however, depends on a number of historically-bound circumstances. It may make sense to be skeptical of adhesion contracts in the context of Microsoft, given that most consumers see no choice but to run its software, and must accept the terms of its licenses. It is also true, however, that in recent years Microsoft competitors, such as Apple, have met with increasing success,[133] and the web is increasingly becoming an important platform for software development. At the margins, at least, these developments may have an effect on how Microsoft does business. Because Kuwahara's argument depends heavily on facts that are specific to Microsoft, and because the market may already be acting to curb some of Microsoft's perceived defects, it is probably premature to adopt her proposed reforms. Additionally, if the software security problem remains primarily a Microsoft problem, as opposed to a problem that is endemic to an industry, it may be more prudent to enact regulations that target Microsoft particularly. Introducing principles of general application based on the behavior of a single company may have unintended consequences on non-culpable parties.

E. Policy Should Not Be Based on Contingent Circumstances

An analysis that proposes to introduce changes to the legal

130. *Id.* at 1012-15.

131. *Id.* at 1019-20; *see* Samuelson, *supra* note 117.

132. Kuwahara, *supra* note 121, at 1014; Samuelson, *supra* note 117.

133. 2007 saw Macintosh hardware sales jump by as much as 40% over the previous year, which is a growth rate between two and three times higher than the computer industry average. Charles Jade, *Apple 2007: Best Year Ever*, ARS TECHNICA, Dec. 24, 2007, http://arstechnica.com/journals/apple.ars/2007/12/24/apple-2007-best-year-ever.

environment is best made without too much reliance on the historically specific (and likely transitory) circumstance of a dominant software firm also having extremely vulnerable products. This circumstance was brought about by a number of specific businesses, technological, and historical reasons and is unlikely to be reproduced again. For example, Windows was initially developed for computers that had rare and transient network connectivity. Today's always-on broadband environment changes that, and increases the exposure of the computer to the outside world for attacks.[134] Computer systems designed for the older world have shown themselves to be not very well suited for the new world, and security incidents proliferated. However, as the risks of always-on network connections have become internalized by software developers, it is likely that the number of vulnerabilities will decrease. For example, in its first year of deployment, Vista had fewer security vulnerabilities than either Windows XP or Mac OS X.[135]

The risk of unintended consequences is too great to justify a change to the law unless there is a real, concrete problem to be addressed. A poorly calibrated liability regime could result, for instance, in more money being spent to prevent security vulnerabilities than the vulnerabilities themselves are likely to cause, resulting in a net social cost. As Steven Pinker suggests, it may sometimes be better to look for practical, engineering solutions to social problems, than to immediately think of redesigning the legal environment. He writes,

> [t]here are many other issues for which we are too quick to hit the moralization button and look for villains rather than bug fixes. What should we do when a hospital patient is killed by a nurse who administers the wrong drug in a patient's intravenous line? Should we make it easier to sue the hospital for damages? Or should we redesign the IV fittings so that it's physically impossible to connect the wrong bottle to the line?[136]

While economic incentives may cause software providers to develop new technologies and improve their products' security in ways they would not have done absent those incentives, without a *technological* solution to the underlying problems that cause software insecurity,

134. Pratyusa K. Manadhata & Jeannette M. Wing, Attack Surface Measurement, http://www.cs.cmu.edu/~pratyus/as.html ("Intuitively, a system's attack surface is the set of ways in which an adversary can enter the system and potentially cause damage. Hence the larger the attack surface, the more insecure the system.").

135. Michael Calore, *Microsoft: Vista Has Fewer Security Flaws in First Year Than XP, Mac OS*, WIRED, Jan. 24, 2008, http://blog.wired.com/monkeybites/2008/01/microsoft-vista.html.

136. Steven Pinker, *The Moral Instinct*, N.Y. TIMES, Jan. 13, 2008, § 6 (Magazine), at 632.

modifications to the liability regime of software markets will amount to little more than a series of transfer payments. Such modifications may be justified as matters of equity, or to harmonize software liability with other areas of products liability. But market and social incentives over the past several years have already increased the focus of the software industry on security issues. Given that the current incentives to create secure software may be adequate, and given that software security has measurably increased in the past few years, changes to the liability environment for software may be premature, and the risk of unintended consequences may be too great, to justify any drastic changes solely on the basis of improving security and counteracting the negative security consequences of a software monoculture.

III. TECHNOLOGY HAS PROVEN SUFFICIENT TO DEAL WITH MOST COMPUTER SECURITY ISSUES

As noted above, discussions of the negative security consequences of software monocultures are generally focused on the problems of a Microsoft monoculture particularly. While any software monoculture can be threatened by the rapid exploitation of a software vulnerability (and, as demonstrated by the Internet Worm, non-Microsoft monocultures have been), in the case of Microsoft, the monoculture effect is seen as a "force multiplier"[137] that greatly increased the effects that are ultimately caused by flawed software in the first place. Therefore, my analysis of the proper policy response to a software monoculture will be based primarily on an analysis of the factors that have led to Microsoft's products being widely viewed as insecure, and on the responses that Microsoft has deployed in order to deal with this problem. It is also informed by an understanding that government interventions in markets often have unintended consequences. As Hahn and Layne-Farrar write,

> From an economist's perspective, before the government decides to intervene to impose software security, it must be reasonably certain that private parties are unable to do so on their own. In other words, it must be clear that the market failed in some way. Otherwise, interventions run the risk of interfering with properly functioning

137. "Force multiplication" is a military concept whereby some factor increases a unit's combat potential. A force multiplier can be favorable weather, decoys, or even sunscreen. About.com: US Military, "Force Multiplier", http://usmilitary.about.com/od/glossarytermsf/g/f2536.htm. Network effects have been analogized to the concept of a force multiplier. *See* LTC Roland Ng Kian Huat, *Force Multiplication Through Network And Networking: A Frame For Discourse*, POINTER: J. OF THE SINGAPORE ARMED FORCES Vol. 30 No. 2 (2004), *available at* http://www.mindef.gov.sg/imindef/publications/pointer/journals/2004/v30n2/features/feature4.html.

> markets and, therefore, of introducing inefficiencies where none existed before–what could be termed a "government failure" as opposed to a market failure.[138]

After a comprehensive review of the marketplace for computer security, those authors remain skeptical that government intervention is needed. They even point out that seemingly benign reforms, such as a "lemon law" for software, could have negative consequences.

Because, as discussed below, technological solutions to many fundamental computer security issues (including the problem of monoculture itself) appear to be making progress, in order to avoid potential negative consequences, the government should not regulate to increase software diversity.

Geer's analysis of the problematic nature of Microsoft's software engineering principles is sound.[139] However, it bears keeping in mind that Microsoft is a software company that became successful in a time before ubiquitous, always-on computer networking. Indeed, broadband adoption is not yet complete: in 2007, 23% of Internet users still used dial-up connections.[140] Microsoft's design strategies may have always been bad from a software engineering standpoint. But most computer worms, virus and trojans today spread over the Internet.[141] In the days where the primary vector of computer malware transmission was the floppy disk or BBS downloads,[142] many computer vulnerabilities would simply not be exploited. The penalty throughout the 1980s and 1990s for insecure software design was not as severe as it is today. It is reasonable to assume that even without any policy action, Microsoft's software engineering strategies will change to reflect the new, networked reality.

In fact, Microsoft's approach to software engineering *has* changed in the past several years. The year before Geer's paper, Microsoft issued its "Trustworthy Computing" whitepaper. This paper called for a fundamental reengineering of computers, down to the level of the

---

138. Hahn & Layne-Farrar, *supra* note 114, at 299.

139. Indeed, despite the progress Microsoft has made in increasing the security of Windows Vista, Windows is still widely seen as overly complicated and slowed in its development cycle by Microsoft's commitment to retain backwards compatibility with older software, and hardware compatibility with as much of the PC ecosystem as possible. *See* Steve Lohr & John Markoff, *Windows Is So Slow, But Why?; Sheer Size Is Causing Delays for Microsoft*, N.Y. TIMES, Mar. 27, 2006, § C, at 1.

140. JOHN B. HORRIGAN & AARON SMITH, HOME BROADBAND ADOPTION 2007 1 (2007), http://www.pewinternet.org/pdfs/PIP_Broadband%202007.pdf.

141. *But see* Gregg Keizer, *Best Buy Sold Infected Digital Picture Frames*, N.Y. TIMES, Jan. 23, 2008, http://www.nytimes.com/idg/IDG_002570DE00740E18002573D9007CF01E.html.

142. *See* DAVID J. STANG, NETWORK SECURITY 237 (1992) (a contemporary source describing PC malware of the early 1990s).

microprocessors, with the aim of increasing security and preventing unauthorized code from running. Many, including Geer himself, have criticized that paper's proposals, arguing that the proposal for a Next Generation Secure Computing Base, commonly referred to as "Palladium," threatened to put too much control of what software can run on a computer into too few hands and to exacerbate the risk of vendor lock-in.[143] Microsoft has since abandoned the most ambitious of its "trusted computing" plans.[144] Although overly ambitious and perhaps misguided, the Trusted Computing whitepaper did at least demonstrate an increased awareness of security issues.

Several other initiatives have had more of a practical impact. In 2002, Microsoft undertook a two-month hiatus in the development of its software in order to focus on security concerns.[145] It has shown itself to be more nimble in its response to problems as they are uncovered.[146] Its research arm has begun to look for long-term security solutions that, unlike secure computing, do not rely on changes to hardware.[147] However, Microsoft's improved dedication to security issues can most clearly be seen on a practical level by looking at a few of the security-related improvements found in the most recent version of the Windows operating system, Vista.[148]

One longstanding weakness in Windows had been that it possessed a "file permissions system" that did not adequately prevent untrained users or rogue programs from making damaging changes to the operating system. Vista addresses this by introducing a more robust, Unix-style permissions system whereby even computer administrators need to supply a password before certain settings or files can be changed. Under Vista, Internet Explorer now runs in a "sandbox" that makes it so neither it, nor any programs it spawns (such as malware from a web site) can do much damage to the underlying system.[149] Vista also contains security features designed to prevent a user's computer from becoming part of a botnet,[150] and the most notorious current worm, Storm, which makes

143. GEER ET AL., *supra* note 83, at 16.

144. Paula Rooney, *Microsoft Shelves NGCSB Project as NX Moves to Center Stage*, CHANNELWEB, May 5, 2004, http://www.crn.com/security/18841713.

145. Peter Judge, *Microsoft Security Push Cost $100m for .Net Server Alone*, ZDNet UK, Jul.CO. UK, July 2, 2002, http://news.zdnet.co.uk/internet/0,1000000097,2118314,00.htm.

146. Matt Mondok, *Microsoft Sets Company Record with WMF Patch*, ARS TECHNICA, Jan. 7, 2006, http://arstechnica.com/journals/microsoft.ars/2006/01/07/2394.

147. Jeremy Reimer, *Microsoft Hefts A Heavy Mithril BrowserShield*, ARS TECHNICA, Sept. 5, 2006, http://arstechnica.com/news.ars/post/20060905-7668.html.

148. Grant Gross, *Microsoft Talks Up Vista Security in DC*, INFOWORLD, Jan. 30, 2007, http://www.infoworld.com/article/07/01/30/HNdcvistalaunch_1.html.

149. MARK JUSTICE HINTON, PC MAGAZINE WINDOWS VISTA SOLUTIONS 70 (2007).

150. Microsoft, Bots, Botnets, and Zombies,

computers it infects part of the Storm botnet, currently does not infect Windows Vista.[151]

These examples show that there are often technological solutions to problems created by technology–solutions that make a policy response unnecessary. One technological change in particular, however, has the potential to alleviate many of the negative externalities caused by software monocultures. This technology, Address Space Layout Randomization (ASLR), uses software techniques to produce a kind of virtual diversity, limiting the vectors by which malware can spread.[152] Elements of software traditionally load into a particular part of a computer's memory. Malware can take advantage of this fact to more easily spread from one computer to another. ASLR reduces the ability of malware to spread from one computer to another by randomly changing the memory location software loads into.[153] As Ollie Whitehouse writes,

> ASLR is a prophylactic security technology that strengthens system security by increasing the diversity of attack targets. Rather than increasing security by removing vulnerabilities from the system, ASLR makes it more difficult to exploit existing vulnerabilities. . . . By randomizing the memory layout of an executing program, ASLR decreases the predictability of that layout and reduces the probability that an individual exploit attempt will succeed.[154]

Although ASLR is not a new technology, its inclusion in Windows Vista shows technological methods taken by Microsoft can lessen the effects of software monoculture. It is the flexible nature of software that gives it the ability to create virtual diversity of this sort–it is difficult to imagine an analogous solution to the problem of, for example, agricultural monoculture. The impressive number of technological solutions Microsoft has brought to bear in Vista in order to address software security should at least argue in favor of giving technology, rather than law and policy, the chance to solve problems in computer security.

Only time will tell whether Vista's improved security model will indeed lead to a more secure system in the long term. But the early signs

---

http://www.microsoft.com/mscorp/safety/technologies/bots.mspx (Last accessed Mar. 8, 2009).

151. Posting of Jim Thompson to Chron.com TechBlog, This Worm is One Quiet Storm, Houston Chronicle Tech Blog, http://blogs.chron.com/techblog/archives/2007/10/is_this_worm_the_perfect_storm.html (Oct. 14, 2007).

152.. OLLIE WHITEHOUSE, AN ANALYSIS OF ADDRESS SPACE LAYOUT RANDOMIZATION ON WINDOWS VISTA 4 (2007).

153. *Id.*

154. *Id.*

are encouraging. For instance, Peter Bright noted in January 2008 that "[a]fter a year on the market, Vista has had fewer security vulnerabilities discovered than XP did in its first year. According to a post on the Windows Vista Security blog, Vista has had 36 fixed and 30 unfixed security vulnerabilities, compared to 68 fixed and 54 unfixed vulnerabilities in XP. Patches have been issued on 9 occasions so far with Vista, compared to 26 for XP."[155]

CONCLUSION

The fact that Microsoft has improved the security of its flagship product in the absence of the kinds of reforms argued for by Geer, Kuwahara, and others, argues against the need for government action or legal reform as a means to improve computer software security. Nevertheless, some of the reform proposals may have other reasons that would justify their adoption. It may be that increased tort liability against Microsoft and other software vendors for shipping vulnerable products is justified from principles of equity—software companies may be superior risk-bearers, even if the added financial incentives were not necessary to get them to improve their products' security. Geer's proposals for forcing Microsoft to be more "open" may be justified as a means of increasing competition in the software market, or as a means to reduce the risk of monopoly. Certain measured responses may be justified even in the absence of evidence that they are necessary to counteract the problems of a software monoculture. For instance, Picker's autarky proposal is probably a sound prophylactic engineering practice under any circumstance. Governments and companies desirous of avoiding vendor lock-in should consider using open data formats, communications protocols, and software. Education of IT buyers could lead to an increase in the awareness of alternative software, which may have its own merits. Finally, governments should ensure that their actions do not *promote* the creation of a software monoculture unnecessarily.[156]

Extraordinary efforts by governments are not needed to address what appears to be a transient, technology-driven problem. In recent years, Microsoft has undertaken a number of security initiatives and adopted a number of new security technologies, including those like Address Space Layout Randomization that partially undermine the

155. Peter Bright, *Microsoft: Vista's Not as Insecure as XP. Please Buy It!*, ARS TECHNICA, Jan. 26, 2008, http://arstechnica.com/news.ars/post/20080126-microsoft-vistas-not-as-insecure-as-xp-please-buy-it.html.

156. Similarly, it has been argued that governments at the very least ought to end subsidies that increase agricultural monocultures to levels perhaps beyond what the market itself would produce. *See* Michael Pollan, *You Are What You Grow*, N.Y. TIMES, Apr. 22, 2007, § 6 (Magazine), at 615.

monoculture argument. The arguably flawed nature of Microsoft products should be a concern for IT managers and technologists, not policy-makers. If Microsoft's continued dominance is to be challenged by regulators, it should be because of established, economics-based antitrust reasons, and not under the guise of an attempt to improve computer security.