Note

Deworming the Internet*

I. Introduction

In 1988, graduate student Robert T. Morris released the first "worm"[1] to have a major impact on the internet.[2] The self-replicating program directly and indirectly disabled thousands of time-shared, multi-user computers[3] by exploiting a range of security vulnerabilities, including poor system configuration, easily guessed passwords, and software defects.[4] Fifteen years later, the underlying software and hardware have changed several times over, but the same problems provide opportunities for worms to wreak havoc,[5] and

---

* I am indebted to a wide range of people who commented on earlier drafts and provided useful suggestions. At The University of Texas School of Law, thanks go to my advisor, Professor Oren Bracha, as well as to Professors Mark Gergen, Ronald Mann, and Wendy Wagner. From the world of computer security, Mark Seiden and Adam Shostack provided many helpful comments, and Eric Brewer, Jim Roskind, and Dan Wallach kindly provided suggestions and insight into particular issues. My editors at the Texas Law Review have been extremely patient and helpful as well, particularly Tom O'Brien, who provided key editorial assistance early on. Finally, I thank Jane Stavinoha for helping me stay sane, for her unstinting support of this project, and for being my first reader.

1. This Note will use the term "worm" to mean any kind of self-replicating program that takes advantage of defective or poorly configured software to spread quickly from machine to machine over a network, whether or not it requires inadvertent human intervention. *See* ROGER A. GRIMES, MALICIOUS MOBILE CODE 4–5 (2001) (adopting this definition). The term "virus" is sometimes also used to distinguish programs that require human intervention from worms which do not, but the distinction is not used consistently in the literature and is not helpful for the purposes of this Note. *Compare* Robert A. Clyde, *Guarding Against Network Security Attacks*, J. COUNTERTERRORISM & HOMELAND SECURITY INT'L, Winter 2003, at 39, 39 (citing the ability to self-propagate without human intervention as the essential difference between viruses and worms), *with* Wikipedia.org, *Computer Worm*, *at* http://en.wikipedia.org/wiki/Computer_worm (last visited Oct. 14, 2004) ("A computer worm is a self-replicating computer program, similar to a computer virus. A virus attaches itself to, and becomes part of, another executable program; however, a worm is self-contained and does not need to be part of another program to propagate itself."), *and* DAVID HARLEY ET AL., VIRUSES REVEALED 665 (2001) ("Many researchers regard worms as a special case or subset of viruses."). Where the term "virus" is used in original material or the name of the program, it will be preserved, but this Note will refer to all such self-replicating programs as worms.

2. *See* Eugene H. Spafford, *Crisis and Aftermath: The Internet Worm*, 32 COMM. ACM 678, 678 (1989) (detailing the progression of the Morris worm).

3. Ted Eisenberg et al., *The Cornell Commission: On Morris and the Worm*, 32 COMM. ACM 706, 707 (1989). The computers affected by the worm ran the Berkeley Software Distribution variant of the Unix operating system, at that time the most popular operating system used on the internet. GENERAL ACCOUNTING OFFICE, GAO/IMTEC-89-57, VIRUS HIGHLIGHTS NEED FOR IMPROVED INTERNET MANAGEMENT 13 (1989).

4. Spafford, *supra* note 2, at 678, 680.

5. *See* GRIMES, *supra* note 1, at 9 (showing an exponential increase in the number of worms); *Fighting the Worms of Mass Destruction*, ECONOMIST, Nov. 29, 2003, at 65–66 [hereinafter *Worms of Mass Destruction*].

wave after wave of worms continue to sweep through the internet.[6]

One thing is different: The stability of the internet has become increasingly important as a matter of collective economic security. While predictions of an "electronic Pearl Harbor" are likely overblown,[7] internet worm attacks carry substantial economic costs.[8] Unless something changes, these attacks are likely to continue.

Much as Larry Lessig has famously asked "what things regulate" cyberspace,[9] in this Note I ask a more narrow question: What things can regulate, or should regulate, worms in cyberspace? Worms are written by people and are transmitted over the internet, where they take advantage of latent defects in software installed on other internet-connected computers.[10] Once infected, these computers are used as jumping-off points for the worms to infect other computers.[11] My focus in this paper is on worm authors,[12] vulnerable software, and the people who buy this software.

---

6. *See, e.g.*, Charles E. Ramirez, *Why Computer Viruses Make Businesses Sick*, DETROIT NEWS, Sept. 7, 2003, at 1A (describing widespread outbreaks in August 2003); Internet Security Systems, *BlackICE Witty Worm Propagation* (describing the rapid spread of a highly malignant worm that overwrites key hard disk sectors), *at* http://xforce.iss.net/xforce/alerts/id/167 (Mar. 20, 2004); Computer Economics Inc., *MyDoom Virus Update: Fastest Spreading Virus Yet* (describing latest worm), *at* http://www.computereconomics.com/article.cfm?id=932 (Feb. 2004).

7. *See* Joshua Green, *The Myth of Cyberterrorism*, WASH. MONTHLY, Nov. 2002, at 8 (noting that there has been "no instance of anyone ever having been killed by a terrorist (or anyone else) using a computer"). The phrase "electronic Pearl Harbor" was coined by "an alarmist tech writer named Winn Schwartau to hype a novel." *Id.* at 12.

8. While the methodology used by experts to estimate damage from worm disruption is unclear and may result in exaggerated figures, the dollar value of all estimates is on the rise. Estimates of worm damage in 2003 range from $12.5 billion to over $80 billion. *See* Email from Mark McManus, Vice-President, Computer Economics, Inc., to Douglas Barnes (Mar. 26, 2004, 14:10 CST) (on file with author) (estimating damage at $12.5 billion); Bryan Chaffin, *2003 Damages From Windows Viruses Tops 9/11 Insurance Claims*, MACOBSERVER (citing report in which security group Mi2g estimated that worms caused approximately $80 billion worth of damage year to date), *at* http://www.macobserver.com/article/2003/10/02.12.shtml (Oct. 2, 2003).

9. LAWRENCE LESSIG, CODE 88–89 (1999). This Note uses the terms "regulate" or "regulation" to mean a broad range of activity, including not only traditional regulation by administrative agencies but also criminal sanctions, incentive schemes, taxes, civil penalties, litigation, and even the actions of markets themselves.

10. *See infra* subpart IV(A).

11. *See* GRIMES, *supra* note 1, at 2–3 (describing the basics of "malicious mobile code").

12. In other contexts, there can be an important distinction between a worm "author" and a worm "disseminator." Although they are usually the same person, some advocate the creation of worms or worm techniques for research purposes. There is some debate about whether this behavior by itself should be criminal. More importantly, many individuals make minor modifications to existing worms and then disseminate the new version. For example, the MyDoom worm has been modified several times since being created by a German high school student, spawning enough variants to warrant naming its subsequent iterations. *Web Worms Can Google, Too*, BUSINESS WEEK ONLINE, *at* http://yahoo.businessweek.com/technology/content/jul2004/tc20040727_1171_tc024.htm (July 27, 2004). One might ask a metaphysical question about worm authorship that resembles authorship issues in copyright law—how much material must an individual contribute to an existing worm to be considered the "author" of the resulting creation?

Part II of this Note evaluates two approaches to regulating worm authors. These authors have long been the most popular focus for U.S. regulatory attention,[13] and this attention has generally proceeded according to a broad principle of deterrence that is achieved by punishment and by setting the moral high ground.[14] Because this strategy has accomplished little or nothing, cyberspace pundits, including Lessig, are now calling for bounties and changes to internet architecture that would make it easier to catch worm authors, punish them, and deter future authors. Part II argues that these approaches are both unlikely to work and potentially destructive to the internet.

Part III discusses the role of the market in regulating the worm problem. In technical circles, the role of vulnerable software in creating worm crises is well understood. However, to the extent that this concern appears on the agenda of policymakers, they apparently assume that markets will eventually provide the right incentive for software publishers to produce better software. This has not happened. Part III develops a theory to explain both why the market has not yet produced substantially more secure software in its current configuration and why it is not likely to do so in the future absent changes in incentives or market structure.

Part IV addresses the issue of government intervention in the worm crisis. Market failure alone does not justify government intervention; it must be possible for intervention to make the situation better. Part IV first argues that software publishers have long had the ability to prevent the types of worms that are currently afflicting the internet and that the damages suffered from worms are large, if hard to pin down with any precision. Part IV goes on to argue that these preventable damages make a prima facie case for government intervention, despite the belief in some circles that the internet should be protected from regulation.

Part V looks at regulation of the worm problem through litigation. Virtually no lawsuits have been filed for worm-related damage. Current law does not provide a useful vehicle for those damaged by worms, whether they are the software purchasers themselves, or non-purchasers who are nevertheless affected when the internet either slows down or grinds to a complete halt. Part V examines why current law fails in this regard and argues that even if tort causes of action could be extended to cover worm-related damage, this would not necessarily be a good thing.

Part VI of this Note proposes potential reforms for regulating worms in cyberspace. Subpart VI(A) proposes a mandatory "bugs bounty" program

---

Except in the final Part, this Note avoids both of these distinctions and the corresponding debates and simply refers to worm "authors."

13. The Computer Fraud and Abuse Act criminalizes worm distribution. 18 U.S.C. § 1030 (2000); *see also infra* notes 17–18.

14. *See, e.g.*, Mary M. Calkins, *They Shoot Trojan Horses, Don't They? An Economic Analysis of Anti-Hacking Regulatory Models*, 89 GEO. L.J. 171, 189 (2000) (advocating increased criminal sanctions as an efficient deterrent that would also make a "strong moral statement").

which offers rewards for those who discover and report, but do not exploit, new worm vulnerabilities in software. Subpart VI(B) suggests minimum quality standards for software, a measure that penalizes software publishers who could have prevented worms by choosing methodologies and technologies that can achieve a high level of worm resistance. Specifically, subpart VI(B) proposes a "lemon law" that provides for refunds and disclosure of information, both of which are necessary in order for competitors to create compatible products. Finally, in light of the need to direct users toward more secure software, subpart VI(C) explains the benefits of at least minimal penalties for users of infected software.

II. Crime and Punishment

Efficient crime reduction requires the right tradeoff between spending on prevention, where the goal is to increase the difficulty of committing crimes, and law enforcement, where the goal is to reduce crime by catching and punishing criminals.[15] Although the latter approach is traditionally a public sector venture, the private sector can voluntarily assist by offering rewards, or it can be enlisted to help indirectly through regulations that make it easier to identify and capture criminals post hoc.[16] This Part argues that law enforcement targeted at worm authors, even with strong private sector cooperation, is particularly ill-suited for preventing harm from worm attacks, and thus the solution for preventing worms lies primarily in prevention.

A. Direct Law Enforcement Approaches

Authoring worms is a crime in the United States,[17] and some authors have been caught and prosecuted.[18] Nonetheless, tens of thousands of worms

---

15. *See, e.g.*, Thomas J. Philipson & Richard A. Posner, *The Economic Epidemiology of Crime*, 39 J.L. & ECON. 405, 408 (1996) (describing tradeoffs between public deterrence and private prevention expenditures on preventing crime). *See generally* Steven Shavell, *The Optimal Structure of Law Enforcement*, 36 J.L. & ECON. 255 (1993) (discussing the theoretically optimal structure of law enforcement and various methods actually employed in light of social welfare considerations).

16. One related example of this approach would be the extensive reporting requirements imposed on financial institutions in order to detect money laundering. *See generally* Charles Thelen Plombeck, *Confidentiality and Disclosure: The Money Laundering Control Act of 1986 and Banking Secrecy*, INT'L LAW., Spring 1988, at 69.

17. *See* 18 U.S.C. § 1030(a)(5)(A)(i) (1996 & Supp. 2004) (imposing liability on any person who knowingly transmits a "program, information, code, or command, and as a result of such conduct, intentionally causes damage without authorization, to a protected computer"). This code section has steadily evolved since its inception in 1984 from a measure narrowly focused on national security to a catch-all for a wide variety of computer crimes with interstate effects. *See* Robert Ditzion et al., *Computer Crime*, 40 AM. CRIM. L. REV. 285, 291–92 n.34 (2003) (reviewing the history of 18 U.S.C. § 1030).

18. Robert T. Morris was convicted under 18 U.S.C. § 1030, and the conviction was upheld by the Second Circuit. United States v. Morris, 928 F.2d 504 (2d Cir. 1991); Michael Alexander, *Three-Year Probation for Morris; Internet Worm Author Won't Pay Restitution*, COMPUTERWORLD, May 7, 1990, at 1. In 1999, David Smith, the author of the "Melissa" virus, pled guilty to violating 18 U.S.C. § 1030 and was subsequently sentenced to 20 months in prison. Press Release, U.S.

and worm variants have been authored, and only a handful of worm authors have been caught and convicted.[19] There is, then, a justifiable perception among worm authors that only exceptionally careless authors get caught, and this causes authors to deeply discount the occasional law enforcement success.[20] For instance, the author of a variant of the "Blaster" worm was recently caught because Romanian-language text in the worm led police to a webpage that listed his address and phone number.[21] David Smith, the author of the "Melissa" virus, was caught when a trace placed on the phone line he used to transmit the virus led authorities to his apartment.[22]

In 1997, an investigator for the Los Angeles Police Department explained why law enforcement agencies have problems addressing computer crime by citing higher priorities, the cop "tough-guy" self-image, jurisdictional complications, and lack of resources.[23] Although there are some signs of improvement and many law enforcement agencies have established special computer crime squads,[24] enforcement problems will likely persist for a number of reasons.

First, worms are particularly difficult to trace. They can be quickly uploaded from any internet-connected computer and are typically not identi-

---

Dep't of Justice, Creator of "Melissa" Computer Virus Pleads Guilty to State and Federal Charges, *at* http://www.cybercrime.gov/melissa.htm (Dec. 9, 1999); *Melissa Virus Creator Jailed*, BBC NEWS, *at* http://news.bbc.co.uk/1/hi/world/americas/1963371.stm (May 2, 2002).

19. Although federal prosecutions for violations of 18 U.S.C. § 1030 have been steadily increasing, in 2001 there were still only 106 prosecutions of any sort, resulting from 53 distinct cases. *See Query the Federal Justice Statistics Database*, *at* http://fjsrc.urban.org/noframe/wqs/q_intro.cfm (providing searchable statistics for the number of prosecutions of any section of 18 U.S.C. in any year from 1984–2002). A "representative sample" of cybercrime cases on the DOJ website shows only three virus or worm cases out of approximately seventy listed. U.S. Department of Justice, Computer Crime & Intellectual Property Section, Computer Intrusion Cases, *at* http://www.cybercrime.gov/cccases.html (last visited Dec. 4, 2003); *see also* Martha Mendoza, *Web Virus Writers, Senders Rarely Jailed*, *at* http://www.siliconvalley.com/mld/siliconvalley/news/6657374.htm (Aug. 30, 2003) (citing authorities calling for stiffer, tougher laws but also admitting the difficulty of catching virus writers). "'Finding the creator of a virus is a rarity,' says Matt Yarbrough, former head of the Cybercrimes Task Force in the Justice Department. 'It's easier to profile a terrorist from the Middle East.'" Jon Swartz, *Cops Take a Bite, or Maybe a Nibble, Out of Cybercrime*, USA TODAY, Sept. 2, 2003, at 1B.

20. In her article, *Virus Writers: The End of Innocence?*, *at* http://www.research.ibm.com/antivirus/SciPapers/VB2000SG.htm (last visited Sept. 1, 2004), virus expert Sarah Gordon surveyed virus authors and IT security professionals following "Melissa" virus author David Smith's arrest. The eleven virus authors unanimously agreed that the arrest would have no effect, while the sixteen security professionals were evenly split. *Id.*

21. *2nd Blaster Suspect Arrested*, CHI. TRIB., Sept. 4, 2003, at 8.

22. *Officials: AOL Info Cracked Virus Case*, *at* http://zdnet.com.com/2100-11-514222.html (Apr. 1, 1999).

23. Marc D. Goodman, *Why the Police Don't Care About Computer Crime*, 10 HARV. J.L. & TECH. 465, 477–87 (1997).

24. *See Computer Crime Task Forces—USA*, *at* http://www.ccmostwanted.com/CP/LEccuUS.htm (last visited Sept. 1, 2004) (collecting links to computer crime units in the U.S.).

fied and diagnosed for hours or days.[25] Describing the problem of tracking worm authors, one expert was quoted as saying:

> The worm was spread by sending out a single packet of data using a type of technology known as the user datagram protocol, or UDP. The initial packets could have had any source address that the attacker wanted. Given that, the best hope that security experts and authorities may have is that the author could do something dumb such as brag.[26]

Identifying worm authors resembles an epidemiological inquiry, looking for a "patient zero,"[27] more than a criminal investigation.[28] But unlike more focused computer crimes such as computer intrusion, there is no sustained contact between the perpetrator and the target, making tracing all the more difficult.[29]

Another frequently cited problem is that worms are an international phenomenon; worms have originated from a wide range of countries including Brazil, China, Israel, Romania, and Russia.[30] Finally, even if it could be effective, heightened enforcement would require time, training, money, and reduced attention to other law enforcement priorities.

B. Bounties and Worm Author Psychology

A series of particularly acute worm attacks in the fall of 2003 prompted Microsoft to offer bounties of up to $250,000 for information leading to the

---

25. Brian Livingston, *How Long Must You Wait for an Anti-Virus Fix?*, *at* http://itmanagement.earthweb.com/columns/executive_tech/article.php/3316511 (Feb. 23, 2004) (reporting on a study showing lag times of seven to thirty hours). *But see* Diane Frank, *DHS Names Federal Worm Hunter*, *at* http://www.fcw.com/fcw/articles/2003/0922/news-dhs-09-22-03.asp (Sept. 22, 2003) (noting a goal of improving worm alert times to under thirty minutes).

26. Robert Lemos, *Setbacks in Search for Worm Author*, *at* http://news.com.com/2100-1001-982284.html (Jan. 27, 2003) (quoting Marc Maiffret, chief hacking officer for security software firm eEye Digital Security).

27. *See* Patient Zero, *at* http://en.wikipedia.org/wiki/Patient_Zero (last modified Sept. 1, 2004) ("Patient Zero refers to the central or initial patient in the population sample of an epidemiological investigation.").

28. *See, e.g.*, *Internet Worm Keeps Striking*, *at* http://www.cbsnews.com/stories/2003/01/28/tech/main538200.shtml (Jan. 27, 2003) (using the epidemiological analogy when pointing out that one worm's rapid spread made it "nearly impossible for researchers to find the electronic equivalent of 'patient zero,' the earliest-infected computers").

29. *See* Mendoza, *supra* note 19 ("'It's one thing to trace a hacker one or two steps back, but in these cases it could be 20 or 30 or 40 steps back, . . . through multiple servers, and with each step it's not twice as hard, it's logarithmically more difficult.'") (quoting federal prosecutor Ross Nadel).

30. *See, e.g.*, Michael France, *A Bear of a Virus in Hibernation*, BUS. WEEK, Feb. 19, 2001, at 14 (Brazil); *Information Security: Code Red, Code Red II, and SirCam Attacks Highlight Need for Proactive Measures: Testimony Before the House Comm. on Gov't Reform, Subcomm. on Gov't Efficiency, Financial Mgmt., and Intergovernmental Relations*, 107th Cong. (2001) (statement of Keith A. Rhodes), *available at* http://www.gao.gov/new.items/d011073t.pdf (last visited Oct. 28, 2004) (China); Bob Woods, *Melissa Variant Takes on Microsoft 'Monopoly'*, COMPUTING CANADA, Aug. 20, 1999, at 13 (Israel); Jim Krane, *Another Blaster Suspect*, INFO. WEEK, Sept. 8, 2003, at 12 (Romania); *MyDoom Worm Linked to Russian Sources*, ABC News Online, *at* http://www.abc.net.au/news/newsitems/s1035350.htm (Jan. 31, 2004) (Russia).

arrest of certain worm authors.[31] Larry Lessig liked the idea of using bounties to encourage hackers to catch hackers so much that he was willing to "bet [his] job that it works."[32] Professor Lessig can rest easy now because the bounty has apparently worked, at least in a narrow sense: The first such bounty will go to tipsters who led authorities to the alleged author of the "Sasser" and "Netsky" worms.[33] But whether such bounties will work in the sense of reducing the number or severity of worms remains to be seen.

Bounties and other forms of victim enforcement are attractive complements to public law enforcement because the resources expended are more likely to flexibly mirror the harm done and because enforcers and (sometimes) informants have clear-cut incentives to succeed.[34] Moreover, for many of the reasons discussed above,[35] public law enforcement agencies are particularly disadvantaged when it comes to identifying worm authors, and it is reasonable to expect properly compensated private actors to perform somewhat better. In particular, entities like Microsoft can offer bounties proportional to the global harm they experience, while there is no equivalent well-funded transnational law-enforcement agency with a similar perspective.

Unfortunately, the bounty approach only works if hackers can discover the identity of virus authors. The evidence, however, is that only parts of the virus subculture are both social and motivated by fame,[36] and therefore likely to brag to others. Other frequently cited motivations for worm authors are that they get a buzz from vandalism, believe they are fighting authority, and like matching wits with others.[37] Worm-compromised computers can, in turn, be used to engage in further vandalism or as servers for illicit content.[38]

---

31. Dennis Fisher, *Microsoft Puts Bounty on Virus Writers*, *at* http://www.eweek.com/article2/0,4149,1373578,00.asp (Nov. 5, 2003). In addition, the SCO Group, the target of a "denial of service attack" launched by worms hosted on Microsoft software, offered a similar $250,000 bounty. The SCO Group, *SCO Offers Reward for Arrest and Conviction of Mydoom Virus Author*, *at* http://ir.sco.com/ReleaseDetail.cfm?ReleaseID=127545 (Jan. 27, 2004).

32. *See Worms of Mass Destruction*, *supra* note 5, at 67.

33. Martyn Williams, *Microsoft Bounty Helps Nail Sasser Suspect*, PCWORLD.COM, *at* http://www.pcworld.com/news/article/0,aid,116064,00.asp (May 10, 2004) ("These were individuals who were aware of who the perpetrator was. . . . They did not stumble upon this simply through technical analysis. They were aware of who this individual was.") (quoting Microsoft vice president and general counsel Brad Smith).

34. *See* Gary S. Becker & George J. Stigler, *Law Enforcement, Malfeasance, and Compensation of Enforcers*, 3 J. LEGAL STUD. 1, 13–16 (1974) (analyzing the economics of victim enforcement).

35. *See supra* notes 25–30 and accompanying text.

36. DAVID HARLEY ET AL., VIRUSES REVEALED 451–53 (2001).

37. *Id.*

38. *See* Edward Skoudis, *The Worm Turns*, INFORMATION SECURITY, July 2002, at 43 (noting that worms are increasingly used as transport vehicles for spreading other attack tools, "such as DDoS zombies, sniffers, distributed scanners, and distributed password crackers").

In addition to these applications, worm-compromised computers are increasingly used for a pecuniary purpose—sending spam.[39]

Given the variety of actors engaged in worm authorship and distribution, the circumstances of the first bounty-related arrest are not encouraging. The alleged perpetrator was a somewhat hapless eighteen-year-old German high-school student who received the equivalent of a "B" in his informatics class.[40] The worms did not carry a damaging payload and in fact were "programmed to clean PCs of MyDoom and Beagle virus infections."[41] While the worms were still disruptive, the available evidence places the perpetrators at the opposite end of the spectrum from, say, those who might calculatedly author and distribute worms for pecuniary purposes. As long as there are opportunities for worm authors with these other motivations, bounties may help thin their ranks, but the attacks will continue.[42]

C. Regulating Worm Authors by Regulating Architecture

Some commentators have proposed changes to the internet's architecture in order to facilitate capture of computer criminals, including worm authors.[43] These measures would attempt to eliminate the de facto anonymity of the internet so that worm authors could be reliably identified when releasing worms on the internet.[44]

Imposing strong user authentication would entail enormous technology-switching costs. To the extent that the authentication measures would be "bolted onto" existing standard software, they would tend to increase the

---

39. Tiernan Ray, *E-mail Viruses Blamed as Spam Rises Sharply*, SEATTLE TIMES, Feb. 18, 2004, at E6. In a related development, the Federal Trade Commission (FTC) recently issued a report weighing the question of whether or not to issue bounties for improving enforcement of the laws regulating unsolicited commercial email, also known as "spam." *See generally* FEDERAL TRADE COMMISSION, A CAN-SPAM INFORMANT REWARD SYSTEM: A REPORT TO CONGRESS (2004). The report is most skeptical of the ability of non-law-enforcement "cybersleuths" to develop admissible evidence in proceedings against spammers, while holding out only limited hope that bounties could provide an effective incentive for whistleblowers or other insiders who are aware of the spammer's activities. *Id.* at 23–28.

40. Ben Aris, *Sasser Boy Wonder Was Helping Mum*, GUARDIAN (London), May 10, 2004, at 11.

41. Kevin Murphy, *Microsoft Reward Leads to Arrest of 'Skynet'*, COMPUTER BUSINESS REVIEW ONLINE, *at* http://www.chrononline.com/news.asp (May 11, 2004).

42. Note that others have examined preference shaping as an approach to regulating worm authors and reached a range of conclusions. *Compare* Calkins, *supra* note 14, at 208–10 (rejecting preference shaping as a viable solution for illicit hacking), *with* Michael P. Dierks, *Computer Network Abuse*, 6 HARV. J.L. & TECH. 307, 309 (1993) (analyzing a number of problems with ex post criminalization and concluding that ex ante prevention is a more practical alternative).

43. *See Worms of Mass Destruction*, *supra* note 5, at 67 ("A parallel approach to the problem of internet insecurity is . . . to focus on the internet's users, discouraging bad behaviour and ensuring that criminals can be traced."). The article also cites Larry Lessig for the proposition that "anonymity could be replaced by pseudonymity," given appropriate procedural safeguards. *Id.*

44. *Cf.* Joel Michael Schwarz, *"A Case of Identity": A Gaping Hole in the Chain of Evidence of Cyber-Crime*, 9 B.U. J. SCI. & TECH. L. 92, 115–26 (2003) (considering various approaches to achieve systematic "credentialing" at public internet terminals).

complexity of the software and introduce new defects.[45] Moreover, worms are rarely traced to a particular point of origin, so it matters little whether these points of origin are robustly associated with individuals.[46] Even when worms are traced to a particular computer, this is no guarantee that the owner of the computer is responsible, since the same types of flaws that enable worms also enable worm authors to compromise the security of particular computers for the purpose of releasing worms.[47] In other words, if companies are undermotivated or unable to produce software that is worm-invulnerable, it is unlikely that they can produce user authentication add-ons or patches that are worm-invulnerable.[48]

Any law enforcement approach, whether or not backed by supporting architectural changes, also has to contend with the fact that worm authorship is a highly international phenomenon.[49] Any solution based on law enforcement or legislatively mandated authentication would likely leave pockets of unrestricted worm authorship in some countries, which would leave the entire internet vulnerable to worm attacks.[50] This will be addressed in more detail in the next subpart.

D. National Laws and International Worm Authors

Although there is both a popular and academic view that the internet is borderless and homogenous,[51] a more accurate view is that it is logically borderless and homogenous, while the physical equipment and allocation of

---

45. *Information Technology—Essential But Vulnerable: Internet Security Trends: Testimony Before the House Comm. on Gov't Reform, Subcomm. on Gov't Efficiency, Financial Mgmt., and Intergovernmental Relations*, 107th Cong. (2002) (statement of Richard D. Pethia) ("With increased complexity comes the introduction of more vulnerabilities . . . ."), *available at* http://www.cert.org/congressional_testimony/pethia-11-02/Pethia_testimony_11-19-02.html (last visited Oct. 28, 2004).

46. *See supra* note 19 (explaining the rarity of locating virus creators).

47. Many recent worms create a "backdoor" when they infect a host computer, allowing the computer to be operated remotely. *See, e.g.*, CERT Coordination Center, *CERT Incident Note IN-2004-01: W32/Novarg.A Virus*, *at* http://www.cert.org/incident_notes/IN-2004-01.html (Jan. 27, 2004) ("W32/Novarg.A . . . has been reported to open a backdoor to the compromised system and possibly launch a denial-of-service attack . . . .").

48. *See* Brian Krebs, *'Witty' Worm Wrecks Computers*, WASHINGTONPOST.COM, *at* http://www.washingtonpost.com/ac2/wp-dyn/A11310-2004Mar20?language=printer (Mar. 21, 2004) (describing a worm that infected a security add-on product).

49. *See supra* note 30.

50. This could be avoided to a certain extent by walling off portions of the internet that refused to comply. This would still leave the internet substantially vulnerable to digital identity thieves and would also reduce the value of the internet. *See* Neal Kumar Katyal, *Criminal Law in Cyberspace*, 149 U. PA. L. REV. 1003, 1007 (2001) ("If potential victims and third parties like ISPs are forced to take precautionary measures—from building strong firewalls to forgoing communication with risky computer systems—these measures may diminish the value of the internet.").

51. *See, e.g.*, David R. Johnson & David Post, *Law and Borders—The Rise of Law in Cyberspace*, 48 STAN. L. REV. 1367, 1370 (1996) ("[T]he cost and speed of message transmission on the Net is almost entirely independent of physical location. Messages can be transmitted from one physical location to any other location without degradation, decay, or substantial delay . . . .").

bandwidth are quite nonhomogenous and subject to bottlenecks.[52] Although internet service providers (ISPs) are starting to block the transmission of known worms and even to anticipate future worms,[53] the introduction of new worms is effectively unconstrained by borders and operates at this borderless and homogenous logical layer.

At the same time, improved resistance to worms can have disproportionate regional benefits. Worm-related disruption is best evaluated with respect to the overlapping groups of machines that communicate using a given subset of the infrastructure, and internet traffic is strongly biased toward domestic communication, especially in the United States.[54] Disruption is higher when a high percentage of machines in such a group is infested, and lower when no machines, or only a very small percentage of communicating machines, are infested. Consequently, a single worm release can quickly infect all vulnerable computers worldwide, but a worm's ability to disrupt network traffic from simple replication attempts alone is roughly proportional to the number of machines in each communicating group engaging in a segregable body of traffic.[55] Because internet traffic patterns are highly regional, regional measures to increase worm resistance can prevent or reduce explosive traffic congestion within a particular region (because links to other regions create natural bottlenecks).

Ultimately, it seems unlikely that law enforcement, whether regional or global, can significantly mitigate the worm crisis by itself. Even if a global identification and enforcement regime is possible, it makes far more sense to focus on what can be directly effected through legislation at a national level, rather than through the slower and more uncertain process of global regulation.

III. Flaws in the Software Market

There is no great technological secret to producing software without the vulnerabilities exploited by internet worms.[56] But worm-free software costs

52. Maps of international internet bandwidth provide one view of this. *See, e.g.*, Telegeography Research, *Global Internet Map 2002*, *at* http://www.telegeography.com/resources/index.php (last visited Sept. 1, 2004).

53. *See generally* David Moore et al., *Internet Quarantine: Requirements for Containing Self-Propagating Code*, 22 CONF. IEEE COMPUTER & COMM. SOCIETIES 1901 (2003) (investigating the potential of widespread containment mechanisms to mitigate network-borne epidemics), *available at* http://www.ieee-infocom.org/2003/papers/46_04.pdf.

54. Maps of the international physical structure of the internet illustrate the physical centrality of the United States. *See supra* note 52. Consequently, locations in the United States host a disproportionately large percentage of key international resources.

55. Worms can also disrupt selected targets. Variants of the "Mydoom" worm targeted the Microsoft Windows operating system and the SCO Group, knocking the SCO website off the internet. John Schwartz, *Virus Plagues Computers and SCO Site*, N.Y. TIMES, Feb. 2, 2004, at C10.

56. *See infra* subpart IV(A).

more, takes longer, and has fewer features.[57] Markets are widely considered to be the best way of handling tradeoffs between quality and cost, and some policymaking circles have expressed considerable bias against regulating the internet.[58] At least one commentator has argued that given a free market in software, software publishers must be spending the right amount on this aspect of security.[59] If one believes this argument, then the continued presence of worms tells us that the cost of solving the worm problem exceeds the value (to software purchasers) of worm-free software.

This Part proposes an alternate explanation—that the market for standardized internet-connected software is deeply flawed. These flaws occur for three reasons. First, when companies are competing to establish a standard or to displace an existing standard, users do not have a reliable way to get information about the internal quality of the software until it has become the standard. Second, because of various perverse incentives and because of the need to resist challengers to the standard, security improvements are a poor investment for the standard-bearer. Finally, even if these problems could be addressed, the tendency of users to undervalue and unevenly value security would lead to suboptimal investment in security.

This Part develops this picture by first discussing the effects of standardization. Subpart A describes the flaws that occur in the process of users collectively settling on de facto software standards. Subpart B turns to flaws that emerge after software becomes standard. When software either implements or embodies a proprietary standard and defects begin to become apparent, the software publisher is insulated from market punishment by the high cost of switching. Subpart C addresses the flaws that result from the perverse incentives that software publishers enjoy as a result of latent software defects. Subpart D argues that users themselves are flawed because they both undervalue security and unevenly value security while over-valuing the network effects of the software they have loaded on their computers.

---

57. ANNELIESE VON MAYRHAUSER, SOFTWARE ENGINEERING 583 (1990) ("Once the software exists, complexity is a major determinant of maintainability, testability, and reliability."); Bruce Schneier, *Liability Changes Everything*, *at* http://www.schneier.com/essay-liability.html (Nov. 2003) (explaining that software publishers have not spent much money on security because of significant costs in terms of time, expense, reduced functionality, and frustrated end users).

58. *See* WILLIAM J. CLINTON & ALBERT GORE, JR., A FRAMEWORK FOR GLOBAL ELECTRONIC COMMERCE 4 (1997) ("For electronic commerce to flourish, the private sector must continue to lead. Innovation, expanded services, broader participation, and lower prices will arise in a market driven arena, not in an environment that operates as a regulated industry.").

59. *See, e.g.*, Dierks, *supra* note 42, at 309 ("Unless there is market failure in the market for computer security equipment, an efficient level of spending occurs on preventive measures with a corresponding efficient level of computer abuse."). *But cf.* STEVEN SHAVELL, INDIVIDUAL PRECAUTIONS TO PREVENT THEFT: PRIVATE VS. SOCIALLY OPTIMAL BEHAVIOR 5 (Nat'l Bureau of Econ. Research, Working Paper No. 3560, 1990) ("There is, though, no necessary relationship between the socially optimal level of precautions and the levels chosen by individuals."), *available at* http://www.nber.org/papers/w3560.pdf.

Finally, subpart E considers the argument that the market might correct these flaws on its own without government intervention.

A. Flaws in the Software Standardization Process

Publishers of internet-connected software know that there are essentially three possible outcomes when they develop a new product.[60] The first is that the software will become wildly successful and its file formats, interfaces, and network protocols will become an open standard, generating positive network externalities.[61] The second possible outcome is similar to the first, but instead the publisher will effectively own the standard and will constrain interoperability through some combination of patent, copyright, and trade secret.[62] The final—and most likely—possible outcome is that the product will fail. This can happen either because another publisher gets to market sooner or with a better product, or simply because the market was not ready for a product with those features. Software displaced or out-competed by a victorious standard has little value. Software that does win such competitions can be quite valuable to software publishers, at least until a new standard comes along with sufficiently attractive features to displace the previous standard.[63]

The standardization process interacts with the unfortunate fact that latent software security defects tend to remain hidden until after software has become popular, and consequently, such defects play no role in the competition to set standards.[64] Defenders of Microsoft's poor track record with security are quick to suggest that its competitors would have as many

---

60. This Note uses the term "software" throughout to refer to a fairly narrow segment of the software industry—mass-market personal software that communicates with other software using the internet. Viewed more broadly, the software industry clearly includes large segments that do not cleanly fall under this analysis—including segments like business enterprise software, internal-use custom software, hosted software services, and software embedded in automobiles and appliances. For descriptions of some of these segments, see generally MARTIN CAMPBELL-KELLY, FROM AIRLINE RESERVATIONS TO SONIC THE HEDGEHOG: A HISTORY OF THE SOFTWARE INDUSTRY (2003).

61. Mark A. Lemley, *Antitrust and the Internet Standardization Problem*, 28 CONN. L. REV. 1041, 1052 (1996) ("The nature of the Internet, and indeed of most computer software markets, is such that a single standard is likely to emerge as the dominant one . . . .").

62. *Id.* at 1054 ("Specifically, while companies owning intellectual property in a potential standard can if they wish license it freely, they will not do so if they believe that the result of their refusal will be a standards competition which they will win."); John E. Lopatka & William H. Page, *Antitrust on Internet Time: Microsoft and the Law and Economics of Exclusion*, 7 SUP. CT. ECON. REV. 157, 169 (1999) ("Producers in network markets receive continuously increasing returns to scale, reinforcing early successes and aggravating early defeats. This process may lead to 'tipping' of the market to a single producer, or a single standard or kind of product.").

63. *See* CARL SHAPIRO & HAL R. VARIAN, INFORMATION RULES 235–36 (1999) (describing possible reactions of incumbent standard-bearers to new potential standards).

64. *See* Lemley, *supra* note 61, at 1055 (describing the ability of consumers to discipline the market prior to standardization).

problems if they were as successful as Microsoft.[65] As Larry Lessig put it, "A small upstart company making a small operating system would not present much of a target to hackers . . . ."[66]

This is not to suggest that software publishers get a free ride when these defects are discovered. But because these defects are revealed post-standardization, their costs can be discounted for both time and the chance that the product will fail for unrelated reasons. In addition, lost profits from latent defects carry with them some perverse benefits, discussed at length below.[67]

Standardization can occur at a number of different levels when users in effect "buy" a collection of standards by converging on one software package and not its competitors. These standards are embodied in the particular ways in which a given software package stores information, talks to other software and hardware, and interacts with users; they create positive network externalities to the extent that users want to share files with others, communicate with others' software, and carry skills from job to job.[68] Because publishers of proprietary software can protect themselves from simple imitation with a combination of patents, copyrights, trade secrets, and simple obscurity, potential competitors either have to overcome these barriers or trigger a new standards competition.[69] Consequently, although individual software packages are sold on the open market, there is a parallel market in the standards themselves, with very slow, very infrequent transactions that are difficult to reverse.

From the perspective of potential users, many aspects of the software-embodied standard are readily apparent—such as price, features, and availability. Yet because software publishers rely heavily on trade secret

---

65. *See Worms of Mass Destruction*, *supra* note 5, at 66 ("Mr. Nash also denies that Windows' code is less secure than other operating systems', such as Linux or Apple's Mac OS X. Scott Charney, another Microsoft executive, goes further and defends the monoculture. If one operating system is dominant, he says, companies can save costs by training IT staff only once, and security updates are easier since there is only one source of the patches that mend flaws.").

66. *Id.* at 67.

67. *See* discussion *infra* subpart III(C).

68. For a more in-depth discussion of standards choice, see STAN J. LEIBOWITZ & STEPHEN E. MARGOLIS, WINNERS, LOSERS & MICROSOFT: COMPETITION AND ANTITRUST IN HIGH TECHNOLOGY 87–112 (rev. ed. 2001) (proposing a model of standards adoption in light of positive network externalities). Although much of their model seems correct, Leibowitz and Margolis argue that positive network externalities are unlikely to result in adoption of the "wrong," standard, but in doing so fail to take into account aspects of quality that are not apparent during the standardization process.

69. Even when potential users seek out "open" standards, the long delays in processing patent applications can result in standards adoption before users learn that the standard practices a patent. *See, e.g.*, Rambus Inc. v. Infineon Techs. AG, 318 F.3d 1081, 1085–86 (Fed. Cir. 2003) (describing counterclaims against patentee for failing to disclose patents and patent applications to an association that sets standards for memory technology).

protection, they must limit disclosure of the source code.[70] More importantly, it would likely be very costly to even imperfectly communicate the level of care taken to address security concerns in the development of the software. As a result, security concerns play little role in the software standards adoption process.[71]

Absent some other constraint, this market in standards is well positioned to suffer from a problem similar to the "lemons equilibrium."[72] Generally stated, a lemons equilibrium tends to emerge when one feature, such as price, is readily apparent, but the seller has private knowledge of other features, such as quality, which are not readily apparent.[73] If sellers can produce low-quality goods that buyers cannot distinguish from high-quality goods, then it will not be possible for sellers of high-quality goods to compete with sellers of low-quality goods. As a consequence, lower-quality goods will tend to drive higher-quality goods from the market.[74]

This problem becomes acute in the market for software standards. To the extent that preventing latent software security defects is merely expensive, the situation facing high-quality software vendors is the standard lemons equilibrium. In the more likely situation where preventing software defects is not only expensive but also increases the time to market or demands a smaller set of software features, the cost to would-be high-quality publishers is increased, and they run the risk of missing out on the standards competition altogether. As long as software is maintained as a trade secret, and development occurs behind closed doors, buyers have nothing more to go on than vague, unprovable assertions about quality and security (which are cheap to make).

---

70. *See* Bradford L. Smith & Susan O. Mann, *Innovation and Intellectual Property Protection in the Software Industry: An Emerging Role for Patents?*, 71 U. CHI. L. REV. 241, 243 (2004) (describing the software industry's historical reliance upon trade secret law for protecting software against misappropriation).

71. The Security Across the Software Development Lifecycle Task Force, co-chaired by security experts from Microsoft and Computer Associates, issued a report in April 2004 noting that software publishers are focused on "other goals that are viewed as more important [for product success] such as user convenience, additional functionality, lower cost, and speedy time to market plus evidence that these are what have sold products in the past." IMPROVING SECURITY ACROSS THE SOFTWARE DEVELOPMENT LIFECYCLE, TASK FORCE REPORT app. B-8, *at* http://www.cyberpartnership.org/SDLCFULL.pdf (Apr. 1, 2004) [hereinafter SOFTWARE DEVELOPMENT LIFECYCLE].

72. *See generally* George A. Akerlof, *The Market for "Lemons": Quality Uncertainty and the Market Mechanism*, 84 Q.J. ECON. 488 (1970) (discussing how asymmetrical information as between buyers and sellers of goods can affect markets). In the United States, the term "lemon" is slang for a chronically defective automobile or other piece of machinery.

73. *Id.* at 489.

74. *See* FRANK H. EASTERBROOK & DANIEL R. FISCHEL, THE ECONOMIC STRUCTURE OF CORPORATE LAW 280–83 (1991) (summarizing the lemons equilibrium and describing its application in the context of securities disclosures).

A number of institutions and practices have been suggested as potential solutions for lemons equilibria. In his original paper on lemons equilibria, George Akerlof suggested that "name brands" serve this function, through the mechanism of repeat play.[75] But as Akerlof also pointed out, the brand must "not only indicate quality but also give the consumer a means of retaliation if the quality does not meet expectations."[76] The problem for software is that *individual* purchases of software are not the fundamental transactions that need to be subject to repeat play and retaliation by consumers. Rather, the more fundamental transaction is the collective adoption of that software as a standard.

The next subpart explains how network effect lock-in can eliminate the ability of software purchasers to retaliate effectively, because opportunities for retaliation come infrequently and are only effective if a critical mass of customers act together. Consequently, establishing even one successful standard embodied in proprietary software is so lucrative that in some ways it resembles a one-time grab more than a repeated game.[77]

B. Flaws in the Software Market Post-Standardization

When the software standardization process results in open standards, disgruntled users are much more capable of delivering effective market punishments to software publishers. They can do this by gradually switching to competing software that also embodies the standard.[78] However, when consumers settle on a proprietary standard, a large percentage of the software

---

75. Akerlof, *supra* note 72, at 499.

76. *Id.* at 500.

77. As of March 31, 2004, Microsoft had over $50 billion in cash reserves and short-term investments. *Microsoft Third Quarter FY 2004 Earnings Release*, *at* http://www.microsoft.com/msft/earnings/FY04/earn_rel_q3_04.mspx (Apr. 22, 2004). Most of Microsoft's wealth can be attributed, either directly or indirectly, to the Windows operating system. *See* William M. Bulkeley, *Can Linux Take Over the Desktop? Open-source Software Is Ready to do Battle on a New Front; Here's a Look at its Chances*, WALL ST. J., May 24, 2004, at R1 (noting Microsoft's Windows and Office software products are "Microsoft's biggest revenue and profit engines"). This is not to suggest that Microsoft itself has engaged in a one-time grab, but simply that the stakes are certainly large enough to provide this type of motivation.

78. This has happened with respect to Sendmail, the open source email server software exploited by the Morris worm and subsequently plagued by numerous additional security problems. *See* Spafford, *supra* note 2, at 679 (describing how the Morris worm exploited a security flaw in Sendmail); CERT Coordination Center, *CERT Advisory CA-2003-07: Remote Buffer Overflow in Sendmail*, *at* http://www.cert.org/advisories/CA-2003-07.html (Mar. 3, 2003) (describing the latest buffer overflow vulnerability in Sendmail). Market share for Sendmail has been steadily declining since the early 1990s despite commercialization of the technology and the investment of substantial resources. *See* D.J. Bernstein, *Bogus Popularity Claims for Sendmail*, *at* http://cr.yp.to/surveys/sendmail.html (last visited Sept. 1, 2004) (showing a steady decrease from 80% in the mid nineties to 42% in 2001); SMTP Survey for 2003-05-14, *at* http://www.tty1.net/smtp-survey/survey-2003-05-14_en.html (last visited Sept. 1, 2004) (showing a decline to 35.6% by May of 2003); Sendmail, *Company Overview*, *at* http://www.sendmail.com/company/overview/index.shtml (last visited Sept. 1, 2004) (demonstrating the commercialization of technology).

users must be motivated to abandon both the software and the standard in order to deliver a market punishment to the publisher.[79] This in turn means that three factors must come together more or less simultaneously: (1) a critical mass of disgruntled users; (2) one or more alternative standards or embodiments of that standard; and (3) a process in which the disgruntled users successfully settle on a new standard.

This type of locked-in standard operates as a de facto monopoly, albeit a temporary one.[80] However, monopoly over a locked-in standard does not enable a software publisher to charge any price it wants, nor does it enable it to ignore freshly revealed software defects or other quality problems. There is a profit-maximizing price for a monopolist, but this price is typically a higher price at a lower output than for a firm operating in a competitive market.[81] A monopolist cannot, however, ignore the underlying demand curve for a product. Consequently, a monopolist's profit-maximizing price is still sensitive to changes in quality that make a product more or less desirable.

However, once software becomes a standard, it is tremendously more expensive for the software publisher to thoroughly purge latent software defects.[82] The original programmers have gone on to other projects, memories have grown dim, and the hurriedly written software is poorly documented. The raw costs entailed by this sort of endeavor are compounded by the fact that a total rewrite of the software runs the risk of changing the behavior of the software to such an extent that it could trigger a new standard-setting process if compatibility with previous versions is lost.

On the other hand, other kinds of user-perceived quality improvements, such as adding features, are less expensive and also serve to minimize opportunities for competitors to gain a foothold in the market.[83] Capturing a software standard does not result in a stable monopoly because rivals can

---

79. The related difficulty of shifting away from insecure standards, even when open, shows that defects in the standards themselves can be almost as difficult to address as defects in the software.

80. For a strong view of the consequences of network effect lock-in, see Dennis S. Karjala, *Copyright, Computer Software, and the New Protectionism*, 28 JURIMETRICS J. 33, 45–46 (1987) ("All it takes is widespread public acceptance of one particular manufacturer's product, for whatever reason. Once lock-in begins, it can become self-sustaining.").

81. *See* HERBERT HOVENKAMP, FEDERAL ANTITRUST POLICY: THE LAW OF COMPETITION AND ITS PRACTICE, § 1.3b, at 19–20 (2d ed., 1999) (describing the fundamental economics of monopoly).

82. *See* G. GORDON SCHULMEYER, ZERO DEFECT SOFTWARE 177–78 (1990) (describing the costs associated with fixing bugs later, rather than avoiding them in the first place).

83. Neal Stephenson dramatically describes the feature treadmill:

> By continuing to develop new technologies and add features onto their products they can keep one step ahead of the fossilization process, but on certain days they must feel like mammoths caught at La Brea, using all their energies to pull their feet, over and over again, out of the sucking hot tar that wants to cover and envelop them.

NEAL STEPHENSON, IN THE BEGINNING WAS THE COMMAND LINE 16 (1999).

leapfrog the standard-bearer by offering substantially improved technology.[84] Winners of standards competitions must therefore continually leverage their status as de facto standards to prevent the entry of competitors seeking the opportunity to displace them. Moreover, even if the standard-bearer's customers are unhappy about worm vulnerabilities, there is no way to evaluate claims of superior worm resistance with any certainty.

The expense of security retrofitting, combined with the greater attractiveness of adding features, predictably redirects potential investment away from security improvements and toward additional features. Ironically, few practices are more likely to introduce security flaws than the rapid addition of features to an existing product.[85]

Most importantly, unlike other costs of participating in the standards competition, the potential future downside of latent security defects is a cost that is only paid if the publisher wins. Any net loss in later user-perceived value, as long as it is less than the software publisher's monopoly surplus, can be paid back from that surplus.

C. Perverse Incentives for Software Publishers

Within the bounds that prevent the formation of a critical mass of disgruntled users, a steady stream of later-revealed defects actually provides a range of benefits for software publishers.[86] Software publishers suffer from highly variable sales volume; once a given market niche is saturated, new revenue from that niche can only come from upgrades. Upgrades are primarily marketed on the basis of new features (which tend to exacerbate the problem of latent defects),[87] but a percentage of users always remain who

---

84. *See* JOSEPH A. SCHUMPETER, CAPITALISM, SOCIALISM AND DEMOCRACY 87 (3d ed., Harper & Row 1962) (1942) (describing how new technologies reduce the impact of monopolistic practices through "creative destruction"); Ronald A. Cass & Keith N. Hylton, *Preserving Competition: Economic Analysis, Legal Standards and Microsoft*, 8 GEO. MASON L. REV. 1, 38 (1999) (describing the pressure to innovate in network markets). This Note does not take a position as to whether actions taken to preserve a potentially transitory de facto monopoly provide grounds for antitrust remedies. *See* United States v. Microsoft Corp., 253 F.3d 34, 49–50 (D.C. Cir. 2001) (discussing the potential effect of Schumpeterian competition on de facto monopolies based on technology).

85. *See* Bruce Schneier & Adam Shostak, *Results, Not Resolutions*, *at* http://www.securityfocus.com/news/315 (Jan. 24, 2002) ("Complexity is the worst enemy of security, and systems that are loaded with features, capabilities, and options are much less secure than simple systems that do a few things reliably.").

86. Somewhat ironically, on-the-fly patching of later-revealed software defects will periodically introduce new flaws. *See* Don Clark, *Cigital Says Microsoft Program Isn't Secure*, WALL ST. J., Feb. 14, 2002, at B6 (criticizing Microsoft for releasing a security patch that introduced a new security flaw).

87. *See* Lynn Greiner, *Forced Upgrades Are a Nuisance*, COMPUTING CANADA, June 1, 2001, at 19 ("[D]evelopers seem compelled to stuff more and more into their systems. And with each addition, bugs creep in. Convoluted code may be easy to write, but it's hard to debug and maintain.").

cannot be easily motivated to upgrade.[88] Latent security defects provide a way of "mopping up" these residual customers and forcing them to upgrade.[89] To the extent that software publishers sell software that generates an indefinitely long stream of defects, software publishers can retire older versions and refuse to provide additional software patches.[90] This puts increasing pressure on the owners of these versions to upgrade, despite their indifference to new features.[91] Whether deliberate or not, in effect this phenomenon operates analogously to planned obsolescence for durable goods.[92] This has become such an established phenomenon that it has almost certainly contributed to software publishers' increasing ability to successfully sell software on a subscription basis.[93]

Software that must be updated frequently also enables software vendors to better police piracy because users can be compelled to authenticate themselves in conjunction with obtaining or applying a security patch.[94] "Cracked" software often contains telltale identifying features that can be detected by patch programs, or it simply breaks when patches are applied (because of changes made to the binaries to circumvent copy authentication mechanisms).

Finally, because patch programs provide an opportunity to obtain user agreements, software publishers can modify or amend the terms of license agreements on a take-it-or-leave-it basis, in connection with security upgrades.[95] The law governing disclaimers, forum selection, arbitration, and

---

88. For example, Intuit, a personal finance software company, announced that it would terminate support for certain products because "older products like Quicken 98 essentially represent depreciating assets on the company's balance sheet." Mike Musgrove, *Panned Obsolescence*, WASH. POST, Apr. 4, 2004, at F7.

89. For a related analysis, see generally Glenn Ellison & Drew Fudenberg, *The Neo-Luddite's Lament: Excessive Upgrades in the Software Industry*, 31 RAND J. ECON. 253 (explaining why the practice of selling non-forward-compatible software upgrades can be welfare-reducing).

90. *See, e.g.*, Peter Galli, *Microsoft Bars Office 11 from Windows 9x*, EWEEK, Nov. 4, 2002, at 1 ("[T]his latest attempt by Microsoft to force them into upgrading follows the company's past moves to phase out support for older products and to push users to upgrade to its new licensing agreements."). One of the reasons cited by Microsoft for discontinuing support for earlier versions was "Windows 9x is inherently insecure." *Id.* (internal quotations omitted).

91. *Id.*

92. *See* Ellison & Fudenberg, *supra* note 89, at 256 n.5 (observing how forced product upgrades parallel the planned obsolescence of durable goods).

93. For an initial reaction to Microsoft's plans to offer software by subscription, see Heather Wright, *Microsoft Defends Licensing Changes*, INFOTECH WKLY., Sept. 24, 2001, § 2, at 5 ("The changes are most likely to hurt companies that are less frequent upgraders, upgrading perhaps once every three or four years.").

94. One game company even promises to "[d]elete[] all zip, rar and exe files if it detects a cracked version." Larian Studios, *Divinity*, *at* http://www.larian.com/Site/english/divinity/german.html (last visited Sept. 1, 2004).

95. *See* Brian Livingston, *Sneaky Service Packs*, INFOWORLD, *at* http://reviews.infoworld.com/article/02/08/23/020826opwinman_1.html (Aug. 23, 2002) (describing license changes made in connection with service packs for Microsoft operating software); Thomas C. Greene, *MS Security Patch EULA Gives Billg* Admin Privileges *on Your Box*, REGISTER, *at* http://www.theregister.co.uk/

other contract clauses changes over time, creating an incentive for publishers to utilize security updates as a vehicle for "renegotiating" license agreements in order to obtain terms optimized for the current state of the law.

D. Flaws in Users

Users play an important role in the failure of the software market to reduce latent security defects for two reasons: users generally undervalue security, and users unevenly value security.[96]

Certain types of internet participants, such as large e-commerce websites, are substantially better at taking security precautions than others.[97] Though small in number, they obtain concentrated benefits from connection to the internet. By adopting specialized server software and purchasing expensive third-party security products, these "concentrated-benefit users" can greatly reduce their chances of worm infection. It might be cheaper for them if the standardized software was free of latent security defects, but their benefits from avoiding infection are easy enough to calculate and pay for.[98] Despite these precautions, concentrated-benefit users are helpless to prevent the internet-wide disruptions that come with worm infections.

The harm experienced by concentrated-benefit users has a counterpart in the land of ordinary users. To the extent that the concentrated-benefit users are sellers in a competitive market and ordinary users are buyers, it is fair to assume that the aggregate economic harm experienced by ordinary users as a group is roughly the same as the harm experienced by the concentrated-benefit users.[99] For ordinary users, however, the harm is diffuse and is offset by the information and coordination costs required to evaluate the degree of harm and take appropriate action.[100] Moreover, this assumes that users

---

content/4/25956.html (June 30, 2002) (describing a license change made in connection with security patches to Windows Media Player).

96. User inertia is also a key part of the causal chain for many worm attacks when patches are available. *See, e.g.*, Alex Bakman, *Software Insecurity—Don't Blame Microsoft*, E-COMMERCE TIMES, *at* http://www.ecommercetimes.com/perl/story/32040.html (Nov. 5, 2003). However, the number and frequency of patches that must be applied can be daunting.

97. *See* Anthony Browne, *Virus Coming Soon to a PC Near You*, TIMES (London), Aug. 14, 2003, at 1 ("[S]mall businesses and home users have borne the brunt of the attack. . . . The virus causes computers to crash repeatedly but does not appear to delete files or create major damage.").

98. Global Information, Inc., *Enterprise Security Product Markets*, *at* http://www.gii.co.jp/english/dc15570_enterprise_security.html (Aug. 2003) ("The increased demand for security and the increasing maturity of key individual markets mean that the overall enterprise security products market is set to grow at a CAGR of just over 17% from 2002 to 2006, from a base of around $7.1bn in 2002 to reach over $13.5bn in 2006.").

99. Similar benefit ratios appear likely for relationships between concentrated-benefit users and ordinary users for relationships other than buyer and seller, such as government informational websites and their users.

100. Richard Forno, *Overcoming 'Security By Good Intentions'*, REGISTER, *at* http://www.theregister.co.uk/content/55/31094.html (June 9, 2003) ("Windows users must hedge

behave rationally, when in fact each user must overcome his or her psychological tendencies toward inertia.[101] The simple matter of discovering and applying security patches is often neglected by even fairly sophisticated users.[102]

Even among ordinary users there is a high degree of variation in the value placed on security.[103] For example, some users purchase anti-virus software (which also acts as anti-worm software), while others do not.[104] As Christine Jolls has pointed out, "[a]n amazingly robust finding about human actors—and an important contributor to the phenomenon of risk underestimation—is that people are often unrealistically optimistic about the probability that bad things will happen to them."[105]

If a significant number of users do not buy the extra software or the version with the extra features, worms can still cause widespread disruption, even for those who do make the additional investment.[106] Furthermore, variation among users' security concerns reduces the ability of users to reach the critical mass of dissatisfaction with latent security defects that is necessary to discipline software publishers.[107] This is because users who stand to lose more in the event of worm infection or other security problems would

---

their bets: do they install a patch to fix today's problem now but risk creating newer ones costing additional time and labor to fix tomorrow?").

101. *See* Thomas Gilovich & Victoria Husted Medvec, *The Experience of Regret: What, When, and Why*, 102 PSYCHOL. REV. 379, 380 (1995) (observing that "people experience more regret over negative outcomes that stem from actions taken than from equally negative outcomes that result from actions foregone"); Russell Korobkin, *Inertia and Preference in Contract Negotiation: The Psychological Power of Default Rules and Form Terms*, 51 VAND. L. REV. 1583, 1613–14 (1998) (describing research showing that individuals would predict that others would feel higher levels of regret for action as opposed to non-action).

102. For example, the "SQL Slammer" worm only affected computers running particular versions of a Microsoft database product, for which there was a security patch available. It substantially disrupted traffic throughout the internet for several days. F-Secure, *F-Secure Virus Descriptions: Slammer*, *at* http://www.f-secure.com/v-descs/mssqlm.shtml (Jan. 25, 2003).

103. *See* Calkins, *supra* note 14, at 216 (noting, similarly, that "corporations and ISPs base their security spending on their own needs or the desires of their customer base").

104. Kevin Pinkney makes a similar argument in discussing the problem of externalities created by poor security practices on the part of business users. Kevin R. Pinkney, *Putting Blame Where Blame Is Due: Software Manufacturer and Customer Liability for Security-Related Software Failure*, 13 ALB. L.J. SCI. & TECH. 43, 66 (2002).

105. Christine Jolls, *Behavioral Economics Analysis of Redistributive Legal Rules*, 51 VAND. L. REV. 1653, 1659 (1998).

106. *See Internet Worm Keeps Striking*, *supra* note 28 (describing the extent of the disruption caused by the large-scale attacks).

107. *See supra* subpart III(B). By having a noncompeting third party create compatible add-on products, the standard-bearer can protect competitors from gaining a toehold. *See* S. J. Liebowitz & Stephen E. Margolis, *Should Technology Choice Be a Concern of Antitrust Policy?*, 9 HARV. J.L. & TECH. 283, 306 (1996) (positing that "one would expect entrant firms to try to specialize their products to appeal to particular groups of users" because it is "one simple way for firms to overcome any natural monopoly advantage that might exist in production costs of an incumbent").

ordinarily form the vanguard of disgruntled users.[108] Instead, anti-virus software (to the extent it works) removes these users from the equation, much as the concentrated-benefit users are removed from the equation. The remaining users, weighing information and switching costs against the harm they predict they will experience, fail to take precautions, and consequently, the network environment suffers.

E. Possible Market Self-Corrections

Although the software market has followed the patterns I have described for over two decades, there are some trends that might, over time, cure or partially cure some of these market flaws. Attempts to address the worm problem should be careful to facilitate, rather than thwart, these emerging trends.

The most important such trend is an independent movement toward open source software.[109] Open source or "free software" approaches solve the lemons equilibrium problem by removing the trade secret and copyright barriers to interoperability.[110] Because anyone can look at the source code and development occurs in the open, the information asymmetry between developers and users is dramatically reduced. Competing versions can be created with the current version as a starting point, a process called "forking."[111] Open source software is often developed by, or with substantial participation from particularly security-conscious users.[112] These users have strong incentives to participate in initial development in order to prevent having to rework the product later or create a more secure "fork."[113] Open

---

108. Microsoft is apparently well aware of this phenomenon, and at one point it was preparing to replace third-party suppliers and start charging a premium for add-on security features. *See* Peter Judge, *Microsoft: Users May Have to Pay for Security*, ZDNETUK, *at* http://news.zdnet.co.uk/business/0,39020645,2123526,00.htm (Oct. 8, 2002) ("Microsoft sees security not just as a necessary condition to reassure existing and future customers, but also as a potential source of revenue.").

109. *See, e.g.*, David Kirkpatrick, *How the Open-Source World Plans to Smack Down Microsoft, and Oracle, and . . .* , FORTUNE, Feb. 23, 2004, at 92 ("Open-source software is popping up everywhere, in PC's and cellphones and set-top boxes, in servers that power the world's websites and in giant corporate and government systems."); Malcolm Wheatley, *The Myths of Open Source*, CIO MAGAZINE, Mar. 1, 2004, at 84 ("[Open source] is nevertheless proving attractive enough for a host of CIOs to make the switch.").

110. *See, e.g.*, Free Software Foundation, *GNU General Public License*, *at* http://www.gnu.org/copyleft/gpl.html (last visited Sept. 1, 2004) ("The GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.").

111. *See* Mary Foley, *Open-Source Angst: Fear of Forking*, ZDNET, *at* http://zdnet.com.com/2100-11-524722.html?legacy=zdnn (Oct. 15, 2000) (describing reactions to a potential "fork" of popular Linux file-sharing software).

112. *See* Kurt Dschida, *NetBSD, OpenBSD, and FreeBSD*, *at* http://www.sbei.net/archive/whpapers_articles/bsd_wpaper.pdf (last visited Sept. 1, 2004) (describing the relative strengths, particularly with respect to security, of three variants of "BSD" Unix).

113. *Id.*

source does not directly address the problem of user flaws, and particular projects can be as rushed and buggy as proprietary software.[114] However, because it is open and modifiable by anyone, it is at least *capable* of responding to those users who are concerned.

The open source movement is gathering steam, particularly where existing software standards are least entrenched, such as in the server software market.[115] Outside of the United States, this trend has been accelerated by governments concerned about becoming locked into standards controlled by U.S. companies,[116] and inside the United States, high-security government applications have also proven to be fertile ground for open source.[117]

The open source trend is also driven by unprecedented commercial involvement in creating, improving, and advocating for open source software. Most notable in this regard is IBM's involvement with Linux, an open source operating system that competes with Microsoft Windows.[118] These days IBM sells "solutions," and if they can lower the cost of these solutions while increasing their flexibility, they can sell more solutions and realize larger profits.[119] As particular types of software become more commoditized, it becomes increasingly sensible for competitors to avoid the classic standards competition altogether, cooperatively create shared solutions,[120] and compete in other aspects of the package. These motivations

---

114. *See, e.g.*, CERT Coordination Center, *CERT Advisory CA-2003-24: Buffer Management Vulnerability in OpenSSH*, *at* http://www.cert.org/advisories/CA-2003-24.html (Sept. 16, 2003) (describing one in a long series of security problems with Secure Shell (SSH), an open source project).

115. According to the Netcraft survey, Apache, an open source web server, is used to serve over 67% of the domains on the internet. Netcraft, *September 2004 Web Server Survey*, *at* http://news.netcraft.com/archives/web_server_survey.html (last visited Sept. 1, 2004).

116. *See* Craig S. Smith, *China Moves to Cut Power of Microsoft*, N.Y. TIMES, July 8, 2000, at A1 (citing an official concern "that the country is growing overly dependent on the Windows operating system"); *see also* Kirkpatrick, *supra* note 109, at 97 ("Government leaders, notably in China, are endorsing open source as a way to save money and curb the influence of foreign suppliers, especially Microsoft.").

117. *See, e.g.*, Jonathan Krim, *Open-Source Fight Flares at Pentagon: Microsoft Lobbies Hard Against Free Software*, WASH. POST, May 23, 2002, at E1 ("A May 10 report prepared for the Defense Department concluded that open source often results in more secure, less expensive applications and that, if anything, its use should be expanded.").

118. *See* Tom Foremski & Richard Waters, *Free Software Faces a Rocky Road to Court*, FIN. TIMES (London), Aug. 6, 2003, at 10 ("IBM and Hewlett-Packard, for example, claim that they sell several billion dollars' worth of Linux IT systems every year and that this is their fastest-growing market.").

119. *See id.*; Kirkpatrick, *supra* note 109, at 98 (describing how SAP is increasingly recommending the open source MySQL package instead of Oracle, a move which reduces the total cost for SAP's customers).

120. Cooperative development of open source software reduces the free rider problem by openly and verifiably distributing the effort among those that benefit from the elimination of the current standard-bearer.

combine with other, less obviously profit-driven considerations that also impel participants to develop open source software.[121]

Yet, powerful forces are arrayed against open source. A series of lawsuits by the SCO Group have alleged that the open source operating system, Linux, improperly incorporated derivative works of Unix, an operating system in which SCO claims copyright.[122] Although the lawsuits focus on Linux, they have cast a pall of fear, uncertainty, and doubt over open source generally. Many perceive software patents as a potential problem for free software[123] because there is an obvious inconsistency between free software and either exclusive practice of a patent or licensing in exchange for royalties. However, to the extent that the promotion of open source continues to be a goal of patent-rich organizations such as IBM,[124] the cross-licensing approach taken by these companies could be adapted to include the use of patents by open source software.[125] Other problems hindering widespread acceptance of open source include lack of formal support, lack of a roadmap, missing features, and lack of support from vendors of proprietary software.[126]

In addition to the possibility of open source displacing the proprietary software model, it is certainly possible that proprietary software companies may someday find a cheap, easy, technical fix to their problems. A number of incremental technical advances on the horizon already promise to reduce the likelihood and severity of worm attacks. For instance, Microsoft has recently announced a new plan for enabling rapid online patching of vulnerabilities,[127] although some experts are skeptical that this will improve

---

121. *See* Yochai Benkler, *Coase's Penguin, or, Linux and* The Nature of the Firm, 112 YALE L.J. 369, 423–34 (explaining the interrelated nature of motivations for participating in open source projects for social or monetary rewards).

122. *See, e.g.*, Plaintiff's Complaint, Caldera Systems, Inc. v. IBM, *at* http://www.caldera.com/scosource/complaint3.06.03.html (last visited Sept. 29, 2004) (alleging misappropriation of trade secrets, tortious interference, unfair competition, and breach of contract).

123. Richard Stallman, *The GNU Operating System and the Free Software Movement*, *in* OPEN SOURCES: VOICES FROM THE OPEN SOURCE REVOLUTION 53, 67 (Chris DiBona et al. eds., 1999).

124. *See* RONALD J. MANN, THE MYTH OF THE SOFTWARE PATENT THICKET: AN EMPIRICAL INVESTIGATION OF THE RELATIONSHIP BETWEEN INTELLECTUAL PROPERTY AND INNOVATION IN SOFTWARE FIRMS 43 (Univ. of Tex. Sch. of Law, Law & Econ. Working Paper No. 022, 2004) (noting IBM's large patent portfolio and its use of that portfolio in negotiating cross-license agreements), *available at* http://papers.ssrn.com/sol3/papers.cfm?abstract_id=510103.

125. For a discussion of why IBM has strong motivations for supporting open source software, see generally Robert P. Merges, *A New Dynamism in the Public Domain*, 71 U. CHI. L. REV. 183 (2004) (explaining the motivations that for-profit companies have for putting intellectual property into the public domain).

126. Dan Farber, *Six Barriers to Open Source Adoption*, *at* http://techupdate.zdnet.com/techupdate/stories/main/Six_barriers_to_open_source_adoption.html (Mar. 20, 2004).

127. Aaron Ricadela, *Microsoft To Broaden Security-Patch Software*, INFO. WK., *at* http://www.informationweek.com/story/showArticle.jhtml?articleID=18400479 (Mar. 16, 2004) (discussing Microsoft's plans to release Windows Update Service which aims to replace current installation techniques used by small and midsize companies to apply Microsoft-issued patches).

matters.[128] Security technology is on the horizon that ISPs could use to mitigate the huge traffic surges associated with both worms and worm-related denial of service attacks.[129] While these solutions may eventually address some of the traditional ways in which worms cause damage, technological progress likely means more complex software. More complex software means more opportunities for worm authors, particularly if security is not taken seriously in the initial design and development of new software and new features.

These nascent trends may, over time, change the underlying dynamics of the market for standards. Although they are too speculative to rely on, a carefully crafted government response to the worm problem should strive not to impede these generally positive trends.

IV. Policy Considerations in Regulating Worm-Vulnerable Software

Given the flaws in software markets and the flaws in software users discussed in the previous Part, it seems unlikely that the worm problem will go away through pure market action.[130] This Part builds an argument for why, given this market failure, worm-vulnerable software should be regulated, either directly or indirectly. The subsequent two Parts will examine some specific regulatory approaches, ranging from litigation to minimum technical standards as an incentive scheme; the purpose of this Part is to consider the question of whether to regulate in the abstract. Subparts A and B look, respectively, at the preventable causes of worm-vulnerable software and the resulting damage from this software. Subpart C considers generally the social welfare justifications for regulating worm-vulnerable software, and subpart D discusses the problems of over- and under- deterrence in the context of standards competitions.

A. Publishers' Ability to Prevent Worms

Since the Morris worm in 1988, software publishers have been on notice that network-connected software is uniquely vulnerable to attack.[131] The problem became more acute in the early 1990s, and by 1996 detailed tutorials

---

128. *See* Posting from Dave Farber, Distinguished Career Professor of Computer Science and Public Policy, Carnegie Mellon University, to Interesting People Mailing List, *at* http://www.interesting-people.org/archives/interesting-people/200403/msg00046.html (Mar. 5, 2004 05:47 MST) ("Auto updating has definite dangers from hackers and from errors in updates and interference with other programs. (This has happened in the past). Auto updates are dangerous if not protected and I don't think we know how to do this yet.").

129. *See generally* Moore et al., *supra* note 53.

130. An industry-led task force acknowledged that "it is unclear how much more [users] will be willing to pay for [security] or what the payoff for producers will be." SOFTWARE DEVELOPMENT LIFECYCLE, *supra* note 71, at app. B-8.

131. *See supra* notes 3–4 and accompanying text.

began to emerge, teaching a new generation of young hackers how to exploit common vulnerabilities.[132] Yet the same mistakes and design flaws continue to be introduced in new generations of software. As Richard Pethia, director of CERT[133] at Carnegie Mellon put it: "There is nothing intrinsic about digital computers or software that makes them vulnerable to virus attack or infestation. Viruses propagate and infect systems because of design choices that have been made by computer and software designers."[134] This subpart briefly describes the history of the two most important types of vulnerability, the opportunities vendors have had to eliminate these vulnerabilities, and how these vulnerabilities continue to provide fertile ground for worms.

The Morris worm used, in part, a "stack overflow" attack, a common result of careless programming.[135] As one programmer put it, "Most of these result from using a routine that reads into an internal buffer without checking for buffer overflow. In general, the rule of thumb is simple: never use such routines."[136] Despite this long history, fifteen years later, this same type of mistake still provides footholds for worm attacks on the internet.[137] While by the early 1990s it was well understood that improvements in the development process could have dramatic effects on careless programming errors,[138] there is no indication that such improvements were adopted by mass-market software vendors. By the late 1990s, research into automated tools for detecting or preventing these errors altogether began to bear fruit,[139] yet it was not until 2002 that Microsoft, for instance, released its first product leveraging these approaches.[140]

---

132. *See, e.g.*, Aleph One, *Smashing the Stack for Fun and Profit*, *at* http://www.phrack.org/show.php?p=49&a=14 (Nov. 8, 1996) (describing in detail how hackers can exploit stack overflow vulnerabilities).

133. CERT was formerly known as the Computer Emergency Response Team Centers.

134. *Information Technology—Essential But Vulnerable: How Prepared Are We for Attacks?: Testimony Before the House Comm. on Gov't Reform, Subcomm. on Gov't Efficiency, Financial Mgmt., and Intergovernmental Relations*, 107th Cong. (2001) (statement of Richard D. Pethia), *available at* http://www.cert.org/congressional_testimony/Pethia_testimony_Sep26.html (last visited Oct. 28, 2004)

135. *See id.* (listing common Unix services that had been compromised through stack overflow vulnerabilities).

136. Eric Allman, *Worming My Way. . .*, UNIX REV., Jan. 1989, at 74, 76.

137. In fact the highly destructive "Witty" worm, which struck on March 20, 2004, used a buffer overflow attack. Internet Security Systems, *supra* note 6.

138. *See* SCHULMEYER, *supra* note 82, at 7 (stating that The Software Development Integrity Program was presented as a way to minimize software errors through careful design and development).

139. *See, e.g.*, Crispin Cowan et al., *StackGuard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks*, PROC. 7TH USENIX SECURITY SYMP. (Jan. 26–29, 1998) (describing a working approach to avoid the most frequent types of buffer overflow problems), *available at* http://www.usenix.org/publications/library/proceedings/sec98/full_papers/cowan/cowan.pdf.

140. *See* MICHAEL HOWARD & DAVID LEBLANC, WRITING SECURE CODE 70 (2002) (describing the use of StackGuard in Visual C++ .NET); Tom Yager, *Visualizing .Net*, INFOWORLD, Feb. 11, 2002, at 17 (describing a mid-February 2002 release date for Visual C++ .NET).

In parallel with the security problems provided by stack overflows, another category of security flaws began to appear: scripting vulnerabilities. These vulnerabilities arise when programs such as web browsers and email programs have built-in programming languages that allow strangers (web page authors or email senders) to run software on the recipient's machine. Although these languages enable a range of useful features, such as more interactive web pages, running programs written by strangers carries obvious risks and can enable worms to propagate. Even before Netscape released its first web browser with the embedded language "Javascript," security researchers started to express concerns.[141] By early 1996, laundry lists of security flaws associated with embedded languages in web browsers began circulating in the technical community.[142] Towards the end of the year, related problems also began to emerge with the VBScript language,[143] which could be triggered by incoming emails, leading to the "Love Letter" email worm in 2000.[144] Scripting vulnerabilities have been such a common source of security problems that many security experts strongly suggest disabling such features altogether.[145] Yet, scripting-based email worms continue to create widespread disruption,[146] although experts outlined approaches for making these features secure as early as 1997.[147]

Computer security expert Bruce Schneier summarized the situation:

---

141. *See* Usenet post from Jim Smithson, LiveScript(JavaScript) Built in Functions?, *at* http://groups.google.com (Dec. 7, 1995) ("I know the spec is under development but can someone supply me a pointer to documentation on the built in functions that ARE NOW in Netscape 2.0 beta 3. I'm trying to assess the security implications of LiveScript.").

142. A single issue of *The Risks Digest* in March 1996 contained postings about bugs in both Java and Javascript embedded browser languages. Jack Decker, *Java/JavaScript Security Breaches*, THE RISKS DIGEST, *at* http://catless.ncl.ac.uk/Risks/17.83.html#subj9 (Mar. 4, 1996); David Hopwood, *Java Security Bug (Applets Can Load Native Methods)*, THE RISKS DIGEST, *at* http://catless.ncl.ac.uk/Risks/17.83.html#subj13 (Mar. 4, 1996).

143. Richard M. Smith, *Making Good ActiveX Controls Do Bad Things*, THE RISKS DIGEST, *at* http://cutless.nel.ac.uk/Risks/18.61.html#subj4 (Nov. 15, 1996).

144. *See* CERT Coordination Center, *CERT Advisory CA-2000-04: Love Letter Worm*, *at* http://www.cert.org/advisories/CA-2000-04.html (May 4, 2000) ("The 'Love Letter' worm is a malicious UBScript program which spreads in a variety of ways . . . including electronic mail . . . .").

145. *E.g.*, CERT Coordination Center, *Configure the Web Browser to Minimize the Functionality of Programs, Scripts, and Plug-ins*, *at* http://www.cert.org/security-improvement/practices/p079.html (last updated Apr. 30, 2001) (suggesting such a practice).

146. *See, e.g.*, CERT Coordination Center, *CERT Incident Note IN-2003-02: W32/Mimail Virus*, *at* http://www.cert.org/incident_notes/IN-2003-02.html (Aug. 2, 2003) (describing an email worm).

147. *See* Vinod Anupam & Alain Mayer, *Security of Web Browser Scripting Languages: Vulnerabilities, Attacks, and Remedies*, *at* http://www.usenix.org/publications/library/proceedings/sec98/full_papers/anupam/anupam.pdf (Jan. 1998) (proposing more secure embedded scripting languages); Dan S. Wallach et al., *Extensible Security Architectures for Java*, *at* http://www.cs.princeton.edu/sip/pub/sosp97.pdf (Oct. 1997) (describing the security risks of embedded scripting languages and approaches to addressing these risks).

> As scientists, we are awash in security technologies. We know how to build much more secure operating systems. We know how to build much more secure access control systems. We know how to build much more secure networks. To be sure, there are still technological problems, and research continues. But in the real world, network security is a business problem.[148]

Technology, then, is not the barrier to a worm-free internet. The unresolved questions are whether government intervention is justified and, if it is justified, whether it can improve the situation.

B. Worm Damage

Although worms seldom cause physical damage,[149] they invariably create a cloud of diffuse economic harms—a kind of internet-wide malaise.[150] Internet users can be harmed by worms in two principal ways: when the software they purchase becomes infected, and when others' worm-infected computers are used to swamp particular sites with traffic or to send huge volumes of spam.[151]

Quantifying this damage can be difficult. Home users who purchase worm-vulnerable software are often the least capable of dealing with such problems,[152] and the harm they experience, compared to other types of users, is probably the most difficult to translate into dollar figures. The small monetary sum spent on a worm or virus removal or prevention utility will usually be dwarfed by overall inconvenience. For instance, as a result of a worm attack, these users may lose data, experience erratic computer crashes,

---

148. Schneier, *supra* note 57.

149. *See* Krebs, *supra* note 48 (describing the "Witty" worm, which rendered hard drives useless and required reinstallation of the operating system, but recognizing that worms usually only allow hackers to access and control computers). There was speculation that the "Blaster" worm caused the massive power outage in the northeastern United States in the fall of 2003, but this was never substantiated. *See* Michele Dyson, *The Power Structure and the Power Struggle*, WASH. POST, Oct. 19, 2003, at B8 ("Some officials advanced the notion that the cause of the blackout was the 'Blaster' computer worm, which had been making its way around the internet that week.").

150. *See, e.g.*, Schwartz, *supra* note 55 ("The virus affects computers running the Microsoft Windows operating system, though users of all operating systems have been annoyed by the flood of e-mail messages generated by [the virus]."); Katie Hafner & Kirk Semple, *Fearing PC Havoc, Gumshoes Hunt Down a Virus*, N.Y. TIMES, Aug. 23, 2003, at A1 ("[T]he SoBig virus proved an enormous nuisance. Like gum on a shoe, it stuck around. By the end of the week, the virus had sent out tens of millions of unsolicited messages.").

151. *See* Brian Krebs, *New Hacker Program Prompts Alert: Security Experts Scramble to Get Control of 'Phatbot'*, WASH. POST, Mar. 18, 2004, at E5 (describing "PhatBot," a tool installed on unwitting computers via worms that "allows its authors to gain control over computers and link them into [P2P] networks that can be used to send large amounts of spam e-mail messages or to flood [web sites] with data in an attempt to knock them offline").

152. *See* Brian Krebs & Jonathan Krim, *Internet Worm Targets Windows: Maryland MVA Hit, Forced to Shut Down*, WASH. POST, Aug. 13, 2003, at A1 ("[M]ost home users never download the patches when prompted, and even fewer keep their anti-virus subscriptions current after the trial subscriptions expire.").

get knocked offline for a while, and spend hours on hold trying to get technical support.[153] Furthermore, these harms vary greatly from user to user and do not come with handy receipts.[154] Business users, in contrast, can theoretically track or estimate employee time spent restoring computers or listening to the cheerful on-hold music, as well as employee time wasted while computers are down.[155] Businesses also lose profits and, in severe cases, data.[156]

Calculating damages for affected nonpurchasers can be even more difficult. In some cases the third-party damage is purely parasitic on the damages experienced by purchasers, such as in the case of e-commerce sites whose customers are knocked offline by worm infestations. Some of these damages are highly diffuse, for instance when worm-related traffic impairs internet traffic generally, or when worm-infested computers are used as proxies to increase the already high level of unsolicited commercial email.[157] In other instances, worm-captured computers are used to launch "denial of service" attacks that shut down particular websites.[158] In these cases the damages would be as highly focused and as quantifiable as any other business interruption damage.

ISPs also suffer harm from worms because the product they are offering—internet service—is directly degraded by worm activity. Worm activity can generate massive quantities of network traffic; even when worms launch targeted denial-of-service attacks, this can degrade service for a whole range of customers in the vicinity of the attack.[159] In some instances, intermediate ISPs have to pay directly for these increases in bandwidth; in addition, ISPs are increasingly finding themselves forced to invest in expen-

153. *See* Browne, *supra* note 97 ("[S]mall businesses and home users have borne the brunt of the attack . . . . The virus causes computers to crash repeatedly but does not appear to delete files or create major damage.").

154. *Id.*

155. *See, e.g.*, Richard A. Elnicki, *Virus, Worm & Spam Costs 1: An Episode at the University of Florida*, *at* http://nersp.nerdc.ufl.edu/~dicke/vwsc.html (last visited Sept. 29, 2004) (studying lost time of employees at the University of Florida due to computer viruses and worms).

156. *See, e.g.*, David R. Cohen & Roberta D. Anderson, *Insurance Coverage for "Cyber-Losses"*, 35 TORT & INS. L.J. 891, 895 (2000) (describing business interruption losses).

157. *See Hackers' Computer 'Worm' Burrows Deep, Afflicts Net*, CHI. TRIB., Aug. 13, 2003, at C16 (describing user frustration at nonfunctional internet connections because of worm attacks).

158. *See* Kirk Semple, *New Worm Is Spreading Rapidly Via E-Mail*, N.Y. TIMES, Jan. 28, 2004, at C3 (describing a worm that used hosts to engage in a denial-of-service attack on the website of the SCO Group).

159. *See, e.g.*, *id.* (describing the adverse effects of the Mydoom worm on corporate and personal computer users in the vicinity); Clive Thompson, *The Virus Underground*, N.Y. TIMES, Feb. 8, 2004, § 6 (Magazine), at 28, 30 (discussing how "the [SQL] Slammer worm infected nearly 75,000 servers in 10 minutes, clogging Bank of America's A.T.M. network and causing sporadic flight delays").

sive network equipment upgrades to detect and throttle worm-related traffic surges.[160]

Estimates of worm damage vary widely, with reports in 2003 ranging from $12.5 billion to over $80 billion.[161] Mi2g, the organization that promulgated the higher figure, has been widely criticized for inflating its estimates of worm damage.[162] Mark McManus, vice president of Computer Economics, Inc., the author of the lower figure, explained that their company uses labor costs, tool acquisition costs, outside consultant costs, and loss of revenue (both to those infected and those not infected) to generate their estimates.[163] However, information regarding how this figure breaks down into each category is not available.[164] This estimate does not even attempt to assess the inconvenience to home users or the entire causal chain of economic effects that result when customers are knocked off the net.[165]

One thing is clear: worm damage affects both those who are parties to individual market transactions in worm-vulnerable software and those who have carefully avoided such software. Even conservative estimates of the damages caused are significant enough to warrant regulatory attention. These two types of damage potentially create different bases for regulation, as will be discussed in the next subpart.

---

160. *See, e.g.*, Steve Makris, *Internet Service Providers Swallow Worms*, EDMONTON TIMES, Mar. 15, 2004 (explaining that Shaw, a North American internet service provider, "just spent $3 million on spam and virus filter technology"), *available at* http://www.sandvine.com/solutions/pdfs/ISPs_Swallow_Worms.pdf; Dinesh C. Sharma, *Worms Nibble Away at ISP Profits*, ZDNet News, *at* http://news.zdnet.com/2100-1009_22-5169232.html (Mar. 3, 2004) (reporting that worms "exact a massive toll by forcing [internet] service providers to mobilize premium resources").

161. *See supra* note 8.

162. *See* Thompson, *supra* note 159, at 28 (noting criticisms); Rob Rosenberger, *British Fearmonger Calculates Viruses in U.S. Dollars*, *at* http://www.vmyths.com/rant.cfm?id=45&page=4 (July 29, 1999) (providing a less moderate critique of Mi2g's approach). Rosenberger is generally critical of all virus damage estimates, including those from Computer Economics. *See* Rob Rosenberger, *Mathematical Atrocity*, *at* http://vmyths.com/rant.cfm?id=136&page=4 (May 22, 2000) (highlighting the wide variety of damage estimates from the "ILoveYou" virus reported in the media and mocking the "absurdly accurate" figure of $2.61 billion from Computer Economics).

163. Email from Mark McManus, Vice President, Computer Economics, Inc., to Douglas Barnes (Mar. 26, 2004, 18:41 CST) (on file with author); *see also* DKS & Eric Hayes, *How the FBI Investigates Computer Crime*, INFOSEC OUTLOOK, Aug. 2000, at 3 (describing categories of losses including staff hours, temporary help, damaged equipment, data lost, customer credits, loss of revenue, and value of trade secrets compromised), *at* http://www.cert.org/infosec-outlook/infosec_1-5.pdf.

164. *See* McManus, *supra* note 163.

165. The reverse effect can occur when servers are disabled and users lose the value of the site. *See, e.g.*, Christian Davenport & Hamil R. Harris, *At the MVA, Waiting, Venting: Computer Woes, Staff Shortage, High Demand Plague State*, WASH. POST, Aug. 24, 2003, at C6 (describing negative effects on users when the Motor Vehicle Administration website was disabled by a worm).

C. Justifying Regulation

In the mid-to-late 1990s, debates over regulating the internet, e-commerce, and internet-connected software were dominated by a general hostility towards regulation by prominent internet users and an official hands-off policy on the part of the government.[166] In 1998, with the passage of the Internet Tax Freedom Act, Congress prohibited state and local taxation of internet access and prohibited discriminatory taxes on e-commerce.[167] Yet it would be difficult to argue that currently the internet and its enabling tools are particularly immune to regulation. The past six years have seen efforts by the states and the federal government to regulate unsolicited commercial email,[168] attempts to regulate obscenity and indecency,[169] federal restrictions on software that can defeat copyright protection,[170] and state laws criminalizing the ownership or manufacture of unauthorized software or hardware connected to communication services (including the internet).[171] When the Internet Tax Freedom Act came up for renewal in the winter of 2003, it languished in committee and was not renewed, to a certain extent marking the end of the hands-off approach.[172]

These regulations are relevant not because they are wise or unwise, but because they show a growing tendency toward regulating the internet in accordance with the same policy calculus that would be applied to any other critical element of the national economy and national infrastructure. Given the growing levels of economic damage caused by worms, one element is certainly present—pressure for the government to do something. Combined with the technical ability of software publishers to prevent most (if not all) worms, the next question is whether regulation of software and software publishers is capable of making the situation better.

As described in the previous Part, cutting corners on security in the initial stages of software development can be highly profitable for the software publisher if it enables that publisher to win a standards competition.[173]

---

166. *See* CLINTON & GORE, *supra* note 58, at 30 (noting that the U.S. government "will encourage the creation of private fora to take the lead in areas requiring self-regulation such as privacy, content ratings, and consumer protection and in areas such as standards development, commercial code, and . . . interoperability").

167. Internet Tax Freedom Act, Pub. L. No. 105-277, 112 Stat. 2681-719 (1998) (imposing a three-year moratorium on a variety of internet-related taxes).

168. *See, e.g.*, 15 U.S.C.A. § 7704 (2004).

169. *See* Communications Decency Act, 47 U.S.C. § 223 (2000); Child Online Protection Act, 47 U.S.C. § 231 (2000).

170. *See* 17 U.S.C. § 1201 (a)–(i) (2000).

171. *See, e.g.*, ARK. CODE ANN. § 5-37-402 (Michie 2003).

172. *See* Jim Geraghty, *No Net Tax Ban Seen in Spending Bill*, MULTICHANNEL NEWS, Dec. 1, 2003, at 20 ("[T]he massive appropriations package apparently won't include any deal to extend a lapsed moratorium on Internet taxes.").

173. *See supra* subpart III(A).

While this is nice for the software publisher, this private profit does not represent a correspondingly large increase in social welfare, for two reasons. First, and most obviously, the worm-vulnerable software generates negative externalities by allowing disruption of the internet for everyone. Second, users of worm-vulnerable software are harmed but, given the flaws discussed above,[174] do not or cannot retaliate sufficiently through the market to force the software publisher to fully internalize these harms.

Causing economic actors to internalize negative externalities is a basic goal of a wide range of government policies.[175] When worms leverage worm-vulnerable software to clog the internet or send massive quantities of unsolicited email, the problem takes on some of the characteristics of environmental pollution. When the same vulnerabilities are exploited to launch a targeted attack, the problem begins to resemble that of drivers leaving their keys in their car or handgun owners leaving their guns lying around, with tragic consequences. These analogies will be explored in more detail in the subsequent Part.

The justification for regulation is less clear with respect to the harm done to the purchasers of the software. Even without regulation, the company will be forced to internalize some of the harm that its users experience, through damage to the company's reputation and by a somewhat lower perceived value of the product. The company will be under strong pressure to repair the vulnerability and issue an update of some sort.[176] These steps will quite possibly cost more in the long run than if the software had been developed with fewer flaws in the first place.[177]

Yet despite the harms to themselves, to users, and to others, companies persist in producing worm-vulnerable software because time to market and features can be essential to winning standards competitions.[178] A single company that tried to break the pattern would leave itself vulnerable to opportunistic competitors that could get more feature-laden products to market faster. Because of this market failure, we collectively end up locked into standards that are embodied in software replete with worm vulnerabilities. This is not to say that software that lacks lock-in effects or is otherwise sub-

---

174. *See supra* subpart III(B).

175. *See* WILLIAM M. LANDES & RICHARD A. POSNER, THE ECONOMIC STRUCTURE OF TORT LAW 6 (1987) (describing the role of externality in tort policy).

176. If users become too disgruntled, they could achieve the critical mass needed in order to defect from the standard. *See supra* subpart III(B).

177. *See* SCHULMEYER, *supra* note 82, at 177 (describing cost savings from preventing bugs rather than fixing them later).

178. For a related discussion justifying apparently paternalistic regulation, see Cass R. Sunstein, *Legal Interference With Private Preferences*, 53 U. CHI. L. REV. 1129, 1138 (1986) ("[A] majority may have a collective preference; the public, acting through government, may attempt to bind itself against the satisfaction of its own misguided choices.").

ject to ordinary market forces is invulnerable to worms.[179] But there is simply a much better chance that the security of such software (or replacements offered by competitors) can improve rapidly through the operation of market forces, while software that embodies a locked-in proprietary standard will be more likely to contain security flaws initially and is much less likely to be displaced by a competitor offering superior security attributes.[180]

One analogy to traditional regulatory activities is safety features in automobiles. While reserving judgment on the issue of whether seatbelt-style regulation would make sense for worm vulnerabilities, the underlying regulatory justification is similar. Automobile purchasers' low appreciation for risk has led to mandates for not only seatbelts but also passive restraints such as airbags.[181] The core motivation for these measures is public health, which includes the burden on society from losing the productivity of dead and disabled non-seatbelt-wearers.[182] At first brush, this analogy may seem inappropriate in the context of internet worm damage. But the diffuse damage caused by worms to the reliability and integrity of the internet forms the basis for a similar rationale. Even if worms did not create more direct negative externalities such as congestion and denial of service attacks, worms that knock large numbers of users offline still reduce the collective benefits of the internet for everyone.[183]

For some, the problem is obvious. Computer security expert Bruce Schneier has opined that "[i]f we expect software vendors to reduce features, lengthen development cycles, and invest in secure software development processes, they must be liable for security vulnerabilities in their products."[184] Richard Pethia, director of CERT at Carnegie Mellon, also called for liability in his testimony before Congress:

---

179. *See* Krebs, *supra* note 48 (describing the "Witty" worm, which infected a security add-on product).

180. It will be interesting to see what the market does to Atlanta-based Internet Security Systems, publisher of two security add-on products that are vulnerable to the "Witty" worm, which destroys the victim's hard drive. *See id.* Unlike operating systems and office productivity software, firewalls are highly standardized products and lack strong positive network externalities.

181. *See* Russell B. Korobkin & Thomas S. Ulen, *Law and Behavioral Science: Removing the Rationality Assumption from Law and Economics*, 88 CAL. L. REV. 1051, 1107 (2000) (advocating "removing choices from the realm of individual decision making" when consumers irrationally fail to wear seatbelts or demand airbags). *But see* Fred Mannering, *Automobile Air Bags in the 1990s: Market Failure or Market Efficiency?*, 38 J.L. & ECON. 265, 278 (1995) (suggesting that regulations requiring air bags were unnecessary).

182. *See* Richard J. Arnould & Henry Grabowski, *Auto Safety Regulation: An Analysis of Market Failure*, 12 BELL J. ECON. 27, 29 (1981) (describing the externalities associated with injuries due to failure to wear seatbelts).

183. *See Internet Worm Keeps Striking*, *supra* note 28 (describing the extent of the disruption caused by the large-scale attacks).

184. Schneier, *supra* note 57.

> Technology evolves so rapidly that vendors concentrate on time to market, often minimizing that time by placing a low priority on the security of their products. Until customers demand products that are more secure or there are changes in the way legal and liability issues are handled, the situation is unlikely to change.[185]

Although liability may not be the answer, it is clear that some sort of externally imposed mechanism is necessary. This, however, is just the beginning. The mechanism must also be properly tailored to avoid doing more harm than good, which will be discussed further in the next two Parts.

V. Regulation Through Litigation

When computer security experts talk about the problem of worm vulnerabilities, the conversation often turns to lawsuits.[186] Yet, existing law provides very little latitude for either purchasers of worm-vulnerable software or affected third parties to succeed in court. Despite the enormous damage caused by worms, exactly zero worm-related cases have been successfully brought under common law tort, nuisance, or warranty causes of action.[187] Only one putative class action has been brought, and it rests entirely on idiosyncratic California consumer protection statutes.[188] One explanation for the lack of cases may be that the claims are diffuse and, because of the particularized nature of the damages, lend themselves poorly to class action status.[189] Or perhaps the plaintiffs' bar is simply waiting for someone else to bankroll a test case.[190] Nevertheless, the most plausible

---

185. *Information Technology*, *supra* note 45.

186. *See, e.g.*, Declan McCullagh, *A Legal Fix for Software Flaws?*, *at* http://news.com.com/2100-1002_3-5067873.html (Aug. 26, 2003) (quoting a variety of computer experts calling for software publisher liability); Schneier, *supra* note 57 (advocating liability for security flaws in software products).

187. *See* KEVIN P. CRONIN & RONALD N. WEIKERS, DATA SECURITY AND PRIVACY LAW: COMBATING CYBERTHREATS § 10:25, at 10–52 (2004) ("[T]he author is not aware of cases where a party has been held liable for negligence in forwarding or spreading a computer virus or other harmful code to third party users.").

188. *See* Complaint, Hamilton v. Microsoft Corp., 2003 WL 23698922 (Cal. Super. Ct., filed Sept. 30, 2003) (No. BC303321) (alleging violations of California common law, the California Business and Professions Code, the California Consumers' Legal Remedies Act, and the California Civil Code). In addition, there is a loosely related effort to start litigation against ISPs and network equipment providers for their failure to prevent distributed denial of service attacks. *See Distributed Denial of Service—Class Action Lawsuit—FAQ*, *at* http://www.ddos-ca.org/faq.php (last visited Sept. 1, 2004).

189. *See* FED. R. CIV. P. 23(b)(3) (requiring that "questions of law or fact common to the members of the class predominate over any questions affecting only individual members"). *But see* Chang v. United States, 217 F.R.D. 262, 270 (D.D.C. 2003) ("The existence of factual distinctions between the claims of putative class members will not preclude a finding of commonality.").

190. *See* David Rosenberg, *Of End Games and Openings in Mass Tort Cases: Lessons from a Special Master*, 69 B.U. L. REV. 695, 711 (1989) ("[S]ince early claims are effectively test cases, supplying the basis on which subsequent comprehensive settlement will be patterned, a free-rider problem arises.").

explanation is that such claims are unlikely to succeed. Subpart A briefly examines the most significant infirmities of claims for worm damage under existing causes of action. Subpart B argues that even if courts or legislatures saw fit to evolve one or more of these causes of action to embrace worm-vulnerability liability, it would be a mistake. The overall impact of unrestricted ordinary litigation on software publisher behavior would be highly uncertain because of the difficulty of calculating damages caused by worms, the highly technical proof required at trial, and the possibility of novel opportunities for worm attack. As discussed in the previous Part, poorly tailored penalties could fail to deter software publishers from taking short cuts, while simultaneously deterring investment in, and competition for, setting new standards. Moreover, litigation could end up undermining positive dynamics, such as the growth of open source software.

A. Worms Under Existing Law

Actions for worm damage do not fit neatly into existing law.[191] There are, however, three colorable claims that could be brought against software publishers by worm-damage victims: negligence, product liability, and warranty.[192] All three present serious challenges for the potential plaintiff. The first hurdle facing the two tort claims is the issue of proximate cause—that is, whether the intervening act by the worm author should absolve the software publisher of liability.[193] Assuming this hurdle can be surmounted, the economic loss rule would bar tort claims where there is no physical damage to people or other property.[194] With respect to negligence claims, the question of whether software publishers have a duty of ordinary care creates another opportunity for claims to be barred.[195]

---

191. One commentator describes existing law as "a veritable maze through which few victims of software malfunction emerge with compensation for their losses." Donald R. Ballman, *Software Tort: Evaluating Software Harm by Duty of Function and Form*, 3 CONN. INS. L.J. 417, 475 (1997); *see also* CRONIN & WEIKERS, *supra* note 187, § 10:25 (acknowledging hypothetical claims for negligence while describing the many obstacles to succeeding with such a claim).

192. Public nuisance might also provide an additional cause of action because worms could be construed as "an unreasonable interference with a right common to the general public." RESTATEMENT (SECOND) OF TORTS § 821B(1) (1977). However, the viability of public nuisance claims would vary from state to state and would depend on sufficient political motivation for public officials to take action. *See* David Kairys, *The Governmental Handgun Cases and the Elements and Underlying Policies of Public Nuisance Law*, 32 CONN. L. REV. 1175, 1175 (2000) (noting that the usual plaintiffs are executive branch officials). For a related idea, see Dan L. Burk, *The Trouble With Trespass*, 4 J. SMALL & EMERGING BUS. L., 27, 53 (suggesting that nuisance, rather than trespass to chattels, might be the preferable common law approach to mitigating unsolicited commercial email).

193. *See* discussion *infra* section V(A)(1).

194. *See* discussion *infra* section V(A)(2).

195. *See* discussion *infra* section V(A)(3).

1. Proximate Cause.—In any claim for product liability, negligence, or nuisance, the plaintiff must prove two types of causation: cause-in-fact and proximate cause.[196] In worm litigation, cause-in-fact, or "but-for" causation, would not present a serious problem because by definition a worm relies on vulnerable software to propagate. Proximate cause is another matter.

Defendant software publishers could argue that worm-vulnerable software does not disrupt the internet, worm distributors do.[197] That is, while the software publisher merely made the worm-vulnerable software available, the worm author's intervening act was the proximate cause of the disruption. Historically, this type of argument quickly disposed of the case because intervening criminal acts were considered to be superseding causes as a matter of law.[198] The strict version of this historical doctrine "has [now] been rejected everywhere."[199] Nonetheless, intervening criminal conduct still plays a role in courts' case-by-case findings of "remoteness" or lack of proximate cause.[200]

In contemplating how this issue might unfold, it may be instructive to look at the splits between courts in recent handgun manufacturer liability cases. The analogy may not be immediately obvious, but much like the worm-related claims we are considering, the handgun litigation claims involve attenuated chains of causation and intervening criminal acts. The software story of causation is that the software is sold to innocent users, who then install it on internet-connected computers, which then provide a pool of hosts for worms, enabling worm authors to write worms, which when distributed disrupt or slow the internet, causing a wide variety of economic harms. The handgun story of causation is that the guns are sold to distributors and retailers, who deliberately or negligently sell them to criminals, who then commit more crimes, straining municipal resources required to combat the rise in crime. Both stories depend on attenuated chains of causation combined with foreseeable intervening criminal acts.

In the handgun cases, courts have split based on their willingness to view the chain of causation as "natural and foreseeable,"[201] as opposed to

---

196. *See* 1 DAN B. DOBBS, THE LAW OF TORTS § 180, at 443 n.2 (2001) ("[P]roximate cause limitations are fundamental and can apply in any kind of case in which damages must be proven.").

197. In media stories surrounding the *Hamilton v. Microsoft* class action, Microsoft has consistently raised this point, stating that "[t]he problems caused by viruses and other security attacks are the result of criminal acts by the people who write viruses." Steve Lohr, *Product Liability Lawsuits Are New Threat to Microsoft*, N.Y. TIMES, Oct. 6, 2003, at C2.

198. 1 DOBBS, *supra* note 196, § 190, at 471 ("In an earlier era, courts tended to hold that intervening criminal acts were unforeseeable as a matter of law.").

199. *Id.* at 472 (citing Britton v. Wooten, 814 S.W.2d 443, 449 (Ky. 1991)).

200. *Id.* at 473.

201. City of Gary v. Smith & Wesson Corp., 801 N.E.2d 1222, 1244 (Ind. 2003); *see also* 1 DOBBS, *supra* note 196, § 190, at 472 (explaining that courts look to see if a criminal act was foreseeable to determine if the act was a superseding cause); RESTATEMENT (THIRD) OF TORTS § 34

"attenuated."[202] Several courts have imported proximate cause tests from antitrust cases, but have reached opposing conclusions.[203] However, a recent trend reveals higher courts finding proximate cause in handgun manufacturer cases, overturning dismissals, and allowing cases to proceed.[204]

Although the claims against handgun manufacturers are starting to produce some favorable results for plaintiffs, they most likely do not represent a general principle upon which worm damage plaintiffs could anchor their claims. While handguns are legal, their social utility is a matter of hot debate. There is no corresponding debate over the utility of software development.[205] Consequently, deterring legal acquisition of handguns is not nearly as troubling to many judges as the possibility of deterring software development.

2. The Economic Loss Rule.—When a plaintiff with an otherwise valid negligence or product liability claim suffers economic loss without property or personal damage, the "economic loss" rule comes into play.[206] The rule operates in two principal contexts.[207] First, it operates to establish a cutoff point for liability by preventing plaintiffs from being held liable for the attenuated economic consequences of their actions. Second, it operates to police the boundary between contract and tort, holding parties to their contracts as an exclusive remedy except when there is personal injury or physical damage to other goods.[208]

The economic loss rule only applies when there is no physical harm to people or "other property."[209] One source of analogies here would be a number of cases applying the intentional tort of trespass to chattels to unsolicited

---

cmt. d, at 104 (Tentative Draft No. 3, 2003) (stating that "an actor [can] be found negligent when there is a foreseeable risk of improper conduct").

202. Camden County Bd. of Chosen Freeholders v. Beretta U.S.A. Corp., 123 F. Supp. 2d 245, 257–59 (D.N.J. 2000) (using the word "attenuated" several times).

203. *Compare* James v. Arms Tech., Inc., 820 A.2d 27, 43 (N.J. Super. Ct. App. Div. 2003) (applying the antitrust measure of proximate cause but rejecting a motion to dismiss), *with Camden County Bd. of Chosen Freeholders*, 123 F. Supp. 2d at 258 (applying the antitrust measure of proximate cause but sustaining a motion to dismiss).

204. *See James*, 820 A.2d at 12 (holding that a gun manufacturer who knowingly or negligently floods the gun market, and in doing so knowingly increases the flow of guns into black-market sales, may be liable as a proximate cause of any public nuisance caused by those guns); *see also* NAACP v. AcuSport, Inc., 271 F. Supp. 2d 435, 449–51 (E.D.N.Y. 2003) (noting that gun manufacturers could easily and voluntarily mitigate harm caused by the secondary gun market).

205. All software development, whether in the context of winner-take-all standards competitions or through the mechanism of open source community development, would be impacted by general liability for worm vulnerability. *See infra* section V(B)(3).

206. William Powers, Jr. & Margaret Niver, *Negligence, Breach of Contract, and the "Economic Loss" Rule*, 23 TEX. TECH. L. REV. 477, 480 & n.13 (1992).

207. *Id.* at 481.

208. *Id.* at 482.

209. RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 21 cmt. e. (1998).

email, where network and server congestion has been held to be a sufficient level of harm.[210] This approach has been heavily criticized and was recently limited by the California Supreme Court.[211] In addition, it seems entirely plausible that courts would be more likely to take a flexible view of harm in the context of an intentional act than in the context of unintentionally producing worm-vulnerable software. Better analogies can be found in cases concerning bad software that has damaged data.

There are two ways in which courts have found that bad software that destroys data or causes other disruption does not cause physical damage to other property. The first approach is to deny that the damage is physical. For instance, in *Rockport Pharmacy v. Digital Simplistics*, the Eighth Circuit, applying Missouri law, determined that a loss of data was not physical harm.[212] The second approach is to deny that the computer as a whole is "other property" once the software is installed on it. This is the approach taken in *Transport Corp. of America, Inc. v. IBM, Inc.*, where the court found that when a hard drive is incorporated into a computer, any harm to the computer or its data was not harm to "other property."[213] In other software defect cases, the harm is even more plainly economic, such as harm arising from drilling oil wells in the wrong place or disruption of restaurant business.[214]

Once it is determined that the damage is purely economic, how the economic loss rule operates depends on the status of the plaintiff. Purchasers of the software will be prevented from suing in tort because the provisions of the U.C.C. are intended to be exclusive in instances of economic harm.[215] Nonpurchasers may also find the rule applied to them because the economic loss rule is also used to truncate attenuated chains of economic harm.[216]

---

210. *See, e.g.*, eBay, Inc. v. Bidder's Edge, Inc., 100 F. Supp. 2d 1058, 1072 (N.D. Cal. 2000) (concluding that eBay would likely prevail on a claim for trespass to chattels where less than 2% of eBay's server capacity was used by the defendant); CompuServe Inc. v. Cyber Promotions, Inc., 962 F. Supp. 1015, 1027 (S.D. Ohio 1997) (concluding that plaintiff would likely prevail on a claim for trespass to chattels where the defendant sent large volumes of spam).

211. Intel Corp. v. Hamidi, 71 P.3d 296, 308 (Cal. 2003) (finding that a loss of employee time due to unsolicited email from a disgruntled former employee was not a deprivation of property that could support a claim of trespass to chattels).

212. 53 F.3d 195, 198 (8th Cir. 1995). *Cf.* AOL, Inc. v. St. Paul Mercury Ins. Co., 207 F. Supp. 2d 459, 471 (E.D. Va. 2002) (determining that system crashes and freezes caused by AOL 5.0 were not physical damage, and therefore were not covered by the insurance policy issued by the defendant).

213. 30 F.3d 953, 957 (8th Cir. 1994).

214. *See, e.g.*, Hou-Tex, Inc. v. Landmark Graphics, 26 S.W.3d 103, 105–07 (Tex. App. 2000) (finding no recovery in tort for a company that drilled an oil well in the wrong place because of faulty software); Gus' Catering, Inc. v. Menusoft Sys., 762 A.2d 804, 808 (Vt. 2000) (denying recovery for lost restaurant business due to a faulty computer system).

215. *See* Powers & Niver, *supra* note 206, at 483 (contending that permitting plaintiffs to recover under strict products liability for pure economic loss caused by poor performance of the product would impinge on the domain of the U.C.C. warranty provisions).

216. *Id.* at 481 & n.20.

By foreclosing tort liability when products cause economic harm, the economic loss rule prevents contract law from "drown[ing] in a sea of tort."[217] The rule has been applied several times in this context to deny recovery for economic damage caused by faulty software.[218] Mass market software is generally treated as a good,[219] and the overwhelming trend of the cases is to force purchasers who are economically harmed by software to rely on their warranty rights.[220] Consequently, the economic loss rule will effectively eliminate tort claims by product purchasers.

As the damaged parties get further away from the malfunctioning of the product, the rule is applied in a different sense to set limits on claims for indirect economic harms. An excellent illustration of how the economic loss rule is applied in this manner can be found in *Louisiana ex rel. Guste v. M/V Testbank*, a case concerning a chemical spill that devastated fishing and tourism in the Mississippi Gulf region.[221] The court reasoned that "[d]enying recovery for pure economic losses is a pragmatic limitation on the doctrine of foreseeability" that is "both workable and useful."[222] Much like an oil spill, a worm attack can have widespread economic effects. For instance, e-commerce sites lose business when users are knocked offline or are unable to connect through worm-related congestion, as well as when remote business operations are interrupted. Under this aspect of the economic loss rule, recovery for such claims would be extremely unlikely. In other loosely analogous cases, claims stemming from power cable damage, ruptured water mains, and damaged bridges were not successful due to the economic loss rule.[223] Occasionally courts have made exceptions, but these cases are uncommon.[224]

---

217. E. River S.S. Corp. v. Transamerica Delaval Inc., 476 U.S. 858, 866 (1986).

218. *See, e.g.*, Rockport Pharmacy, Inc. v. Digital Simplistics, Inc., 53 F.3d 195, 198 (8th Cir. 1995) (applying Missouri law and refusing to allow recovery for economic loss resulting from data destroyed by faulty software); *Hou-Tex*, 26 S.W.3d at 105–07 (finding no recovery in tort for a company that drilled an oil well in the wrong place because of faulty software); *Gus' Catering*, 762 A.2d at 807–08 (denying recovery for lost restaurant business due to a faulty computer system).

219. *See* Stephen J. Sand, Annotation, *Validity, Construction, and Application of Computer Software Licensing Agreements*, 38 A.L.R.5th § 9 (1996) (cataloging cases that apply the U.C.C. to software licenses).

220. *E.g.*, Peerless Wall & Window Coverings, Inc. v. Synchronics, Inc., 85 F. Supp. 2d 519, 534 (W.D. Pa. 2000); Budgetel Inns, Inc. v. Micros Sys., Inc., 8 F. Supp. 2d 1137, 1149 (E.D. Wis. 1998); NMP Corp. v. Parametric Tech. Corp., 958 F. Supp. 1536, 1546–47 (N.D. Okla. 1997); Prent Corp. v. Martek Holdings, Inc., 618 N.W.2d 201, 205 (Wis. Ct. App. 2000); Huron Tool & Eng'g Co. v. Precision Consulting Servs., Inc., 532 N.W.2d 541, 546 (Mich. Ct. App. 1995).

221. 752 F.2d 1019 (5th Cir. 1985).

222. *Id.* at 1032.

223. *See, e.g.*, Moore v. Pavex, 514 A.2d 137, 138 (Pa. 1986) (finding that economic loss from the rupture of a water main is not recoverable because "liability cannot flow beyond the persons or property injured, for economic losses only, without creating interminable chains of remote consequences"); Neb. Innkeepers, Inc. v. Pittsburgh-Des Moines Corp., 345 N.W.2d 124, 128–29 (Iowa 1984) (applying the economic loss rule when a damaged bridge decreased motel and restaurant business); Byrd v. English, 43 S.E. 419, 420 (Ga. 1903) (finding that an excavator who

To the extent that potential claimants are forced to rely on U.C.C remedies, they are unlikely to prevail in an action for damages unless the remedy offered by the software publisher fails of its essential purpose.[225] Software publishers systematically disclaim warranties and limit their liability through "clickwrap" agreements that require an act of assent from the user when the software is installed. Consequently, software purchasers who click "I agree" have entered into binding agreements with the software publisher that greatly restrict their ability to prevail in any contractual claims.[226] Software warranty disclaimers have been upheld in a wide array of cases. In *M.A. Mortenson Co., Inc. v. Timberline Software Corp.*, the plaintiff used a construction estimating package to generate a bid that was $1.95 million less than it should have been.[227] After finding that the sale of the software was covered by the UCC,[228] the court refused to hold that the limitation of the remedy was unconscionable, despite the plaintiff's enormous losses.[229] Similarly, warranty disclaimers were also upheld in a number of Y2K cases.[230]

---

broke a power cable was not liable for interrupting a printing business because there was no damage alleged to the person or property of the business).

224. *See, e.g.*, People Express Airlines, Inc. v. Consol. Rail Corp., 495 A.2d 107, 116 (N.J. 1985) (holding that the economic loss rule does not apply if a defendant had reason to know the plaintiffs would suffer damage); Stop & Shop Cos. v. Fisher, 444 N.E.2d 368, 372 (Mass. 1983) (holding that "an established business may state a claim in nuisance for severe economic harm resulting from loss of access to its premises by its customers," but recognizing that most federal courts take the opposite approach); Conley v. Amalgamated Sugar Co., 263 P.2d 705, 710 (Idaho 1953) (holding a sugar beet company liable to a grocer for discharging beet pulp into a stream, the odor of which repelled customers from the store for several months).

225. *See* U.C.C § 2-719(2) (1998) ("Where circumstances cause an exclusive or limited remedy to fail of its essential purpose, remedy may be had as provided in this Act."); RRX Indus., Inc. v. Lab-Con, Inc., 772 F.2d 543, 547 (9th Cir. 1985) (upholding a trial court's award of consequential damages against a limitation clause in a software system contract after holding that the clause was unenforceable because it failed of its essential purpose).

226. Unlike shrinkwrap licenses, which are simply put somewhere inside the box containing the software, clickwrap licenses have consistently been held to be binding agreements. *See* i.Lan Systems, Inc. v. Netscout Service Level Corp., 183 F. Supp. 2d 328, 337 (D. Mass. 2002) ("The only issue before the Court is whether clickwrap license agreements are an appropriate way to form contracts, and the Court holds they are."); Hughes v. McMenamon, 204 F. Supp. 2d 178, 181 (D. Mass. 2002) (upholding a clickwrap forum selection clause); Comb v. PayPal, Inc., 218 F. Supp. 2d 1165, 1169, 1173 (N.D. Cal. 2002) (accepting that the clickwrap process created an agreement, but finding the agreement unconscionable as a contract of adhesion); *see also* Specht v. Netscape Communications Corp., 150 F. Supp. 2d 585, 594–95 (N.D. Cal. 2002) (commenting favorably on typical clickwrap licenses and distinguishing those where one can "download and use the software without taking any action that plainly manifests assent to the terms").

227. 998 P.2d 305, 307 (Wash. 2000).

228. *Id.* at 310.

229. *Id.* at 316 ("Unconscionability 'was never intended as a vortex for elements of fairness specifically embodied in other Code provisions.'" (citation omitted)).

230. *See, e.g.*, Against Gravity Apparel, Inc. v. Quarterdeck Corp., 267 A.D.2d 44, 44 (N.Y. App. Div. 1 Dept. 1999) ("The causes of actions for breach of warranty . . . were properly dismissed in view of defendant's disclaimer of all implied warranties, and plaintiff's use of the software without any problems during the 90-day warranty period.").

3. No Duty.—Courts are deeply divided about whether a duty exists to prevent criminal use of a legal product that is legally sold, even when that criminal use is entirely foreseeable.[231] Firearms cases again provide a useful analogy. For instance, in a negligent firearms marketing case, the New York Court of Appeals held that "'[a] defendant generally has no duty to control the conduct of third persons so as to prevent them from harming others, even where as a practical matter defendant can exercise such control.'"[232] Other courts have viewed the act of creating the opportunity for criminal conduct as an affirmative act, rather than a failure to control, and correspondingly found a duty to exercise reasonable care.[233]

The analogy to worm-vulnerable software is not perfect, but the underlying metaphysical question is the same: Is selling a product that could be criminally misused an affirmative action (and therefore subject to an ordinary duty of reasonable care), or is it merely a failure to protect or prevent (which should impose a duty only under special circumstances)? This metaphysical approach is unhelpful in all but the most clear cut cases, and it would appear that, for most courts, it is resolved as a policy question. Some courts are quite explicit about this. For instance, in a case involving a horse startled by a garbage truck, the California Supreme Court found that there was no duty not to startle the horse, holding that "'duty is not an immutable fact of nature but only an expression of the sum total of those *considerations of policy* which lead the law to say that the particular plaintiff is entitled to protection.'"[234] Other courts similarly rely on policy considerations in creating no-duty safe harbors for socially useful conduct.[235]

As applied to worms, the policy analysis could go either way. It is entirely plausible that, just like the noise from garbage trucks, a court would

---

231. *Compare* Kitchen v. K-Mart Corp., 697 So.2d 1200, 1201 (Fla. 1997) (holding that a store that sold a .22 rifle to a "patently drunk" customer could be liable for negligent entrustment), *with* Buczkowski v. McKay, 490 N.W.2d 330, 331 (Mich. 1992) (holding that selling shotgun ammunition to an intoxicated customer did not create a duty of care "[b]ecause the product sold was neither defective nor inherently dangerous, and because the Legislature has not defined a class of purchasers who we may deem legally incompetent to buy ammunition").

232. Hamilton v. Beretta U.S.A. Corp., 750 N.E.2d 1055, 1061 (N.Y. 2001) (quoting D'Amico v. Christie, 518 N.E.2d 896, 901 (N.Y. 1987)).

233. *See* City of Cincinnati v. Beretta U.S.A. Corp., 768 N.E.2d 1136, 1144 (Ohio 2002) ("'Defendants have engaged in affirmative acts (i.e., creating an illegal, secondary firearms market) by failing to exercise adequate control over the distribution of their firearms.'" (quoting City of Boston v. Smith & Wesson Corp., 12 Mass. L. Rptr. 225 (2000))).

234. Parsons v. Crown Disposal Co., 936 P.2d 70, 80 (Cal. 1997) (quoting Ballard v. Uribe, 715 P.2d 624, 628 (Cal. 1986) (internal quotations omitted)).

235. *See, e.g.*, Jansen v. Fid. & Cas. Co., 556 N.Y.S.2d 962, 965 (N.Y. App. Div. 1991) (determining that the defendant insurance company owed no duty to injured workers and noting that "[i]t is the responsibility of the courts to fix the orbit of duty . . . and in exercising this responsibility not only logic and science, but policy play an important role"); RK Constructors v. Fusco Corp., 650 A.2d 153, 156 (Conn. 1994) (reasoning that "[t]he final step in the duty inquiry . . . is to make a determination of the fundamental policy of the law").

find that worms are a natural part of having useful things like feature-filled software and a (mostly) usable internet and that internet users need to deal with them on their own. It is also possible that courts would recognize some of the policy concerns raised in the previous section, and in doing so find that software publishers need to be responsible for the consequences of worm vulnerabilities. Either way, taken as a whole, these factors create a large zone of indeterminacy for a potential plaintiff.

B. Pitfalls of Regulating Software Through Litigation

Extending tort law to the context of worm damage would require courts or state legislatures to determine as a matter of policy that software publishers are a proximate cause of worm attacks, that some subset of economic damages by worm attacks are cognizable, and that software publishers have a duty of ordinary care when releasing these products to the public, or are strictly liable for all damages flowing from worm attacks. Given the diffuse nature of the harm, these claims would somehow need to be aggregated, despite the wide variety of ways in which victims could be damaged, in order to provide a meaningful remedy.

Laying aside generic objections to tort liability and class actions, there are three ways in which ad hoc extensions of tort law to cover worm damage claims could backfire. First, the winner-take-all standards competition discussed earlier could distort the deterrent effect of potential tort liability. Second, because of rapid changes in technology, the degree of uncertainty about the standard of care and the potential amount of damages could lead to inefficient levels of avoidance. Third, tort liability of this sort could have a dramatically negative impact on the open source software movement, which would eliminate a promising potential cure for some of the market flaws that give rise to the problem in the first place.

1. Distorting Effect of the Standards Competition on Deterrence.— Because of the effect of the standards competition, software vulnerability to deterrence differs before and after the standards competition.[236] This section argues that an ad hoc approach to extending tort liability to software publishers for worm vulnerabilities would have a difficult time taking into account this nuance.

Because of the lemons equilibrium problem discussed above,[237] sacrificing time to market or reducing features for the sake of security will not be rewarded in the standards competition. If an anticipated deterrent would simply act to reduce (but not eliminate) the anticipated surplus profit

236. *See generally supra* subparts III(A) and III(B).
237. *See supra* notes 72–74 and accompanying text.

from winning the standards competition,[238] then its impact will be severely diluted. Surplus profits lowered by penalties are still better than no surplus profits at all, such that a publisher attempting to earn these larger profits by taking longer to get to market or reducing features could be undercut in the standards competition by a competitor satisfied with lower surplus profits. Software publishers already build into their calculations the considerable profits lost due to the emergence of latent defects. Rather than a gradual deterrent effect, an insufficient deterrent will have no impact on the existing race to the bottom between software publishers competing to capture a standard.

By and large this effect would not hold for software publishers defending a standard.[239] At this point in the product's lifecycle, the goal of the publisher is to maximize profits, and penalties associated with worm vulnerabilities would detract from this goal. Therefore, moderate penalties could be expected to have a more gradual, linear effect in creating incentives to eliminate vulnerabilities injected during the standards competition.

These distorting effects highlight the need for a fundamental policy decision: Should the government attempt to prevent vulnerabilities from being introduced in candidate standards (with large penalties), or should it attempt to accelerate the repair of these vulnerabilities post standardization (with more gradual penalties)? The first approach would potentially deter more worms, but at the cost of fewer standards candidates and fewer standards competitions, because each candidate would require a higher level of investment to be worm-invulnerable *before* there was a clear winner in the standards competition.[240]

2. Uncertainty and Efficient Deterrence.—There will be a point where preventing an additional worm is not worth the additional expenditure needed to do so.[241] Creating the right set of deterrents to encourage software publishers to reach, but not exceed, this point will be complicated by the large zone of uncertainty about the standard of care for avoiding worm vul-

238. The calculation of anticipated surplus profits would include profits less the cost of capital, less the cost of fixing the security problems, plus the value of perverse benefits such as forcing upgrades, deterring piracy, or cramming down new contract terms.

239. To the extent that existing standard-bearers are leveraging their existing standard to capture new standards or aggressively expand features, the defending standard-bearers could be expected to act like any other publisher attempting to capture the new territory.

240. This Note assumes that standards competitions are efficient and that diverting resources from them would be undesirable.

241. *Cf.* RICHARD A. POSNER, ECONOMIC ANALYSIS OF LAW 224 (1998) ("[A] greater emphasis on criminal punishment of negligent participants in automobile accidents would not only increase the costs of the criminal justice system, but also undermine the compensatory purpose of no-fault plans."). For a sample of the rhetoric surrounding this issue, see Lohr, *supra* note 197 ("Opening the industry up to product liability lawsuits, [software executives] say, would chill innovation and undermine the competitiveness of American companies.").

nerabilities and the large zone of uncertainty about the size of damages. Applying a strict liability approach might reduce some of the uncertainty about the standard of care, but would still be susceptible to wide variations in the calculation of damages. In either scenario, the degree of uncertainty would then lead to inefficient levels of avoidance.[242]

While a single, specific legislated standard of care for software publishers could be effective,[243] ad hoc elaboration by courts or state-by-state approaches would create a good deal of uncertainty. This, of course, is true of any area of tort law where defendants operate on a national scale. However, when rapid technological change is combined with slow legislative cycles and the intermittent, case-driven nature of appellate decisions, this zone of uncertainty would be greatly expanded. The standard formula for ordinary care compares the burden of the precaution with the amount of harm times the likelihood of harm.[244] Rapidly changing technology exacerbates uncertainly about the likelihood and degree of harm, which then creates uncertainty about the optimal level of precaution.

Certainly there would be easy cases, at least on the side of failing to take precautions. Virtually all worms today are caused by well-known, well-understood, and preventable vulnerabilities.[245] The problem would arise with new combinations of technology, where publishers would face uncertainty about whether speculative types of worm attack should be taken into consideration.

Limits on damages, or more importantly the lack thereof, would present an even greater zone of uncertainty for software publishers. As discussed above, placing a value on worm damage is a slippery problem,[246] and provides a wide menu of options.[247] The need to draw finite, foreseeable boundaries around damages caused by a particular act is one of the driving forces behind both the economic loss rule and limitations on proximate cause.[248] No matter how courts or legislatures circumvent these limits on damages to allow claims for worm damage, they will be forging new,

---

242. *See* Jim Lietzel, *Comment*, *Litigation as Regulation: Firearms*, *in* REGULATION THROUGH LITIGATION 99 (W. Kip Viscusi ed., 2002) ("[A] manufacturer (even if risk neutral) facing an uncertain negligence standard will tend to undertake a socially excessive amount of care . . . ."); Robert Cooter, *Prices and Sanctions*, 84 COLUM. L. REV. 1523, 1539 (1984) ("[S]trict liability makes the manufacturer very responsive to imperfections in assessing and computing damages.").

243. *See* Leitzel, *supra* note 242, at 98–99 ("An advantage of legislative standards of care is that such uncertainty is reduced.").

244. *See* United States v. Carroll Towing Co., 159 F.2d 169, 173 (2d Cir. 1947) (enunciating the modern formulation of tort liability).

245. *See supra* subpart IV(A).

246. *See supra* subpart IV(B).

247. *See supra* subpart V(A).

248. *See supra* sections V(A)(1) & (2).

untested, and probably inconsistent rules for truncating damages beyond a certain point.

The potential for an inefficient level of avoidance would be entirely independent of the potential for underdeterrence discussed in the previous section. In fact, both could occur together. This could achieve the remarkable double of both failing to deter much of the undesirable conduct and generally making standards competitions less attractive.

3. Harms to the Open Source Model.—The movement of some of the software industry toward the open source model is a positive trend with respect to reducing worm vulnerabilities and associated worm attacks.[249] Any approach that initially extended liability to software publishers would, absent a safe harbor, also apply to volunteer contributors to open source software. While there are many intangible and indirect benefits to making such contributions, open source volunteers are not compensated and do not receive a stream of income with which to purchase liability insurance. While one might hope that courts would see the intrinsic unfairness of applying the same standard to commercial software publishers and volunteers who create software and give it away "as is" to others, this could easily fall by the wayside.

VI. Proposed Approaches

This Part proposes a series of short-term and long-term measures to address the worm problem, while attempting to avoid the pitfalls of ad hoc litigation described in the previous Part. The first measure, mandatory "bug bounties," is a low-cost program that could be quickly implemented to redirect the energy of worm authors while software quality begins to improve. The second measure, minimum quality standards for software, recognizes the existence of methodologies and technologies that can achieve high levels of worm resistance, and penalizes software publishers for worms that could have been prevented with these approaches. The third measure, a "lemon law" for standards, reduces barriers to competition when standardized software is chronically prone to worm attacks. Finally, at least minimal penalties for users are necessary to push them toward software that is more secure.

A. Mandatory Bug Bounties

In the mid-1990s, then-dominant browser publisher Netscape offered $1,000 bounties to anyone who discovered a security flaw in its software.[250]

---

249. *See supra* subpart III(E).

250. *See* Email from Jim Roskind, lead technical participant in the "Bugs Bounty" program at Netscape, to Douglas Barnes (Mar. 14, 2004, 02:42 CST) (on file with author) (describing the program in detail).

This type of bounty stands in sharp contrast to the bounties offered more recently by Microsoft and advocated by Lessig.[251] Rather than asking hackers to turn in other hackers, bug bounties reward hackers for doing what they do best. Rather than facing the potential stigma of being a "snitch," bug bounties leverage many of the recognized motivations for worm authors, but redirect them in a more positive way. For those that might otherwise author worms, winning a bug bounty offers money, a degree of fame, a chance to match wits with others, and an expression of disdain for large software companies.[252] Only the love of vandalism would not have an outlet through this approach.[253]

Over the course of its program, the number of awards Netscape issued ranged from zero to eight per year.[254] The decision to pay an award was typically made based largely on whether the company felt compelled to issue a security patch.[255] Cryptography company RSA has also successfully harnessed the efforts of thousands of volunteers to put its algorithms to the test by offering bounties.[256] Privacy firm Anonymizer.com has awarded free service to those who are able to find flaws in its service.[257] Dan Bernstein, author of an email server and a popular internet system utility, offers a $500 reward for the discovery of security flaws in his free software, an award that has not yet been claimed.[258]

Yet, many software companies do not offer such bounties voluntarily, and it is not immediately clear why not.[259] One possible explanation is that such a program could reveal even more vulnerabilities than are currently exposed, which would embarrass the company.[260] Eric Brewer, a computer science professor specializing in computer security at the University of California at Berkeley, suggests that bounties as small as $2,500 for serious vulnerabilities would be sufficient to motivate more work in this area.[261]

---

251. *See supra* notes 31–32 and accompanying text.

252. *See supra* note 36 and surrounding text.

253. *See supra* note 36 and surrounding text.

254. Roskind, *supra* note 250.

255. Roskind, *supra* note 250.

256. *See 1,600 Computers Help Break 129-Digit Code*, CHI. TRIB., Apr. 27, 1994, at N3 (describing the first RSA challenge).

257. Ian Hopper, *Online Privacy Firm Offers 'Bug Bounty'*, TORONTO STAR, May 22, 2002, at E5.

258. *The Djbdns Security Guarantee*, *at* http://cr.yp.to/djbdns/guarantee.html (last visited Sept. 1, 2004) (offering reward and describing criteria for claiming it).

259. *See* Roskind, *supra* note 250 (suggesting that those without such a program "haven't thought through the bigger picture very carefully").

260. During the heyday of the Netscape bug bounty, articles reporting the award of a bounty frequently noted a downward tick in Netscape's stock price in connection with the announcement of the bounty. *E.g.*, Hiawatha Bray, *Netscape Calls Firm that Found Bug a Pest: Refuses To Pay 'Bounty' after Demand for More Money*, BOSTON GLOBE, June 14, 1997, at F1.

261. Email from Eric Brewer to Douglas Barnes (Mar. 19, 2004, 08:07 CST) (on file with author).

Compared to $250,000 for the capture of a worm author, failing to offer a smaller bounty to prevent multiple worm variants from ever being written seems short-sighted. One key difference, however, is that bounties for worm authors emphasize external bad guys, while bug bounties emphasize the responsibility of the software publisher.

Whatever the reason, it is far from apparent that software publishers are prepared to systematically offer such a program. Consequently, this subpart recommends a publicly administered program operated somewhat analogously to workers compensation or unemployment insurance.[262] That is, software publishers would contribute into a fund or obtain insurance to pay bounties to anyone who identified new, exploitable security vulnerabilities in commercial software. The total bounties would be capped as a percentage of revenue, and would be awarded through an expeditious online administrative adjudication. The service should be online to allow worldwide participation and would require payment of a small fee to prevent frivolous submissions.[263]

Creating a mandatory procedure runs into the problem that software publishers are better situated than an administrative agency to answer the question of whether something is a vulnerability. Even the newly established United States Computer Readiness Team (US-CERT), which is tracking and analyzing software vulnerabilities across the entire software industry, lacks the inside knowledge of the software publishers.[264] However, the issue of vulnerability could be removed from the claims process altogether, thereby avoiding the competence problem while reducing the potential for litigation. The administrative entity would instead focus on issues of priority and overlapping claims, while the initial determination about the vulnerability would be entirely in the hands of the software company. To offset the software publisher's obvious incentive to refuse valid claims, in the event that software company subsequently patched the vulnerability within a short time after the claim, or fell prey to a worm attack exploiting that vulnerability, refused claimants would be given the right to sue for an amount sufficient to justify the expense and hassle of litigation.

Such a program could be quickly implemented and could immediately begin to identify vulnerabilities before worm authors could exploit them. However, even a program such as this would not provide a complete solution. Microsoft, for instance, has pointed out that many worms are authored between the time a security patch is made available and the time that users download and apply the patch. While more systematic automated update software may be a partial answer, skeptics question whether this kind of

---

262. I am indebted to Mark Gergen for suggesting this line of inquiry.

263. *See* Brewer, *supra* note 261 (pointing out this potential problem).

264. *See* Frank, *supra* note 25 (describing US-CERT).

automation is itself a good idea.[265] In the long run, the better solution is not to standardize software prone to worm vulnerabilities.

B. Quality Standards for Software

When the environmental movement was in its infancy, the first regulations focused on quickly identifying and prohibiting the most egregious behavior, requiring polluters to reduce pollution to levels achievable with the use of the best available technology.[266] Since then, this approach to regulation has been applied to clean air standards,[267] clean water standards,[268] truck noise,[269] and even drug testing for bus drivers.[270] This subpart proposes that a similar approach be used to regulate software publishers.[271]

Many commentators have criticized technology-based standards as a tool for achieving environmental protection.[272] However, others see them as a key tool for achieving quick, effective results when faced with uncertain (but possibly large) harms and rapidly changing technology.[273] Even critics

---

265. Farber, *supra* note 126.

266. *See* Wendy E. Wagner, *The Triumph of Technology-Based Standards*, 2000 U. ILL. L. REV. 83, 83–84, 90, 113 n.27 (recounting history and explaining that most technology-based regulation does not specify a particular technology, but instead sets numeric goals based on the current menu of available technology).

267. *See* 42 U.S.C. § 7411(a)(1) (2000) (requiring the "best system of emission reduction which (taking into account the cost of achieving such reduction and any nonair quality health and environmental impact and energy requirements) the Administrator determines has been adequately demonstrated").

268. *See* 33 U.S.C. § 1311(b)(1)(A) (2000) (requiring the "best practicable control technology currently available").

269. *See* 42 U.S.C. § 4917(a)(1) (2000) (requiring the "best available technology, taking into account the cost of compliance" to reduce truck noise).

270. *See* 49 U.S.C. § 5331(d)(2)(A) (2000) (calling for the Secretary of Transportation to develop standards requiring the "best available technology to ensure the complete reliability and accuracy of controlled substances tests" for mass transit employees).

271. Although the notion of regulating software security may seem radical, a task force co-led by security experts from Microsoft and Computer Associates has acknowledged the possibility, and the representative from the Department of Homeland Security was given the task of recommending "tailored government action" to "[m]otivate development of more secure software . . . ." SOFTWARE DEVELOPMENT LIFECYCLE, *supra* note 71, at app. B-iii, app. B-8, app. D-2–D-4.

272. *See, e.g.*, Lester B. Lave, *The Strategy of Social Regulation: Decision Frameworks for Policy*, *in* FOUNDATIONS OF ENVIRONMENTAL LAW AND POLICY 94, 97 (Richard L. Revesz ed., 1997) ("At some point additional abatement is unwarranted because social costs exceed social benefits; but even then technology is available that would abate emissions further."); Cass R. Sunstein, *Administrative Substance*, 1991 DUKE L.J. 607, 627 ("A large source of regulatory failure in the United States is the use of rigid, highly bureaucratized 'command-and-control' regulation. . . . In the environmental context, command-and-control approaches usually take the form of regulatory requirements of the 'best available technology.'").

273. *See* Wagner, *supra* note 266, at 95 (asserting that "technology-based standards still significantly outpace—generally by a factor ranging from three to ten times—the promulgation rate of most alternatives"). *See generally* Daniel H. Cole & Peter Z. Grossman, *When is Command and Control Efficient?*, 1999 WIS. L. REV. 887 (arguing that command-and-control environmental regulations are efficient, producing social benefits in excess of their costs, and are more efficient than alternative "economic" approaches to regulation); Sidney A. Shapiro & Thomas O. McGarity,

of the approach grudgingly admit that they "made some sense as a crude first-generation strategy."[274]

Worm-related regulation is in roughly the same position that environmental regulation was forty years ago. Certainly there is a need for quick, effective results. Moreover, many criticisms of technology-based regulation would not seem to apply to the potential application of such regulations to publishers of internet-connected software. For instance, in the environmental context, technology-based regulation is criticized for applying a single standard to varying geographical situations and for only applying to new entrants to the market.[275] Yet, a national (in fact global) standard would be ideal for worm regulation, and the need in the software industry continually to produce new versions to defend (or attack) a dominant position counterbalances any tendency of such regulations to deter new entry.

A more viable criticism of technology-based standards is that they are subject to manipulation by the regulated entities.[276] Yet to some extent, even if the software industry had near-total control over the regulations, the situation would still improve somewhat because even industry-friendly regulations would at least have the effect of removing any incentives to sacrifice later profits by cutting corners on security during the standards competition. The regulatory process would provide an opportunity for industry participants to commit collectively to practices that they could not otherwise adopt unilaterally without fear of being undercut. These regulations would act as a minimum standard, which could then be improved upon over time or through other mechanisms.[277] Given, however, the many well-documented and well-understood sources of vulnerability, as well as the approaches needed to prevent them,[278] the initial regulations could accomplish much more.

In contrast to the ad hoc development of tort law, technology-based regulations could clearly spell out the types of vulnerabilities for which recognized solutions exist. Unlike some approaches, which specify technology

---

Not So Paradoxical: The Rationale for Technology-Based Regulation, 1991 DUKE L.J. 729 (critiquing Professor Sunstein and arguing that even when technology-based regulation gives rise to costs that exceed benefits, regulation may nonetheless be justified on normative and practical grounds).

274. Bruce A. Ackerman & Richard B. Stewart, *Reforming Environmental Law*, 37 STAN. L. REV. 1333, 1364 (1985).

275. *See* Sunstein, *supra* note 272, at 628 (listing criticisms of best available technology regulations).

276. *See id.* at 629–30 (describing the tendency of debates over technology-based regulation to increase the power of industry groups and to foster generalized resistance to all regulation).

277. *See* Wagner, *supra* note 266, at 106 (discussing the ease with which technology-based standards can be combined with other forms of regulation, as well as how they can act as a "circuit breaker" or fallback position for other regulations).

278. *See supra* subpart IV(A).

in detail,[279] the better approach for worm vulnerabilities would be to simply confirm the existence of a family of solutions that eliminate particular results. This would reduce the likelihood of forcing a patented solution,[280] which would have the perverse effect of chilling the positive effect of free and open source software.

To achieve the optimal level of deterrence, penalties for failure to use vulnerability-preventing technology or processes should progressively approach the total profits from the software. This would provide the level of deterrence needed to prevent a race to the bottom during the standards competition, while taking into account the possibility that good-faith application of the mandated technologies or processes might still result in a worm or two.

Some might question the institutional competence of an administrative agency to handle the complex technical questions involved in proposing technology-based regulations for computer software. However, in September 2003, the Department of Homeland Security's National Cyber Security Division created the U.S. Computer Emergency Readiness Team (US-CERT), modeled on Carnegie Mellon's CERT Coordination Center CERT/CC.[281] US-CERT now maintains a common vulnerability database in cooperation with CERT and provides regular analyses of newly discovered worm vulnerabilities.[282] In addition, US-CERT is developing a cyber security early warning system.[283] Developing technology-based standards in cooperation with software publishers would be a natural outgrowth of its current activities and plans.

A more decentralized approach would apply lemon laws to the software industry by giving consumers the right to refunds for software that turns out to be poorly constructed to resist worms. Lemon laws are directed mandatory warranties, designed to protect new car buyers when they were unfortunate enough to get a really bad car. The provisions typically provide for replacement or refund for vehicles that require repeated repairs shortly after purchase.[284] The policy aim of lemon laws is to "provide an incentive

279. *See* Wagner, *supra* note 266, at 90 n.26 (describing Congress's regulation in the Clean Air Act Amendments of 1977 of the specific technology of "scrubbers" as the exception in technology-based regulation).

280. *See* Scott H. Segal, *Fuel for Thought: Clean Gasoline and Dirty Patents*, 51 AM. U. L. REV. 49, 77 (2001) (describing problems with patents concerning reformulated gasoline).

281. U.S. Dep't of Homeland Security, *Welcome to US-CERT*, *at* http://www.us-cert.gov/capabilities.html (last visited Sept. 1, 2004).

282. U.S. Dep't of Homeland Security, *Welcome to the US-CERT Vulnerability Notes Database*, *at* http://www.kb.cert.org/vuls (last visited Sept. 1, 2004).

283. Brian Robinson, *Looking for Trouble*, *at* http://www.fcw.com/supplements/homeland/2004/sup1/hom-programs-02-23-04.asp (Feb. 23, 2004).

284. *See*, *e.g.*, TEX. OCC. CODE ANN. § 2301.604–05 (Vernon 2004) (establishing a rebuttable presumption that a car is a lemon if the same problem has to be fixed more than four times, two of which were within the first 12 months or 12,000 miles).

to that manufacturer to promptly return those unfortunate consumers back to where they thought they were when they first purchased that new automobile."[285] Some commentators have suggested lemon laws for computers, which are also complex devices subject to mysterious failures,[286] but a similar approach has not been suggested to address defective standardized software.

The problem with utilizing this approach with software is that, as discussed above, the fundamental transaction is the adoption of the standard, not the purchase of the individual units of software.[287] Allowing consumers to queue up one at a time to obtain individual refunds may shift the tipping point in an active standards competition, but the original purchase price is only a small part of the investment that a user makes when adopting a software package.[288]

A more radical approach would be to administratively determine that, after a certain number or severity of worm attacks, a particular software standard is a lemon. Two kinds of remedies could then follow. Along the lines of a traditional lemon law, with a single adjudication it would allow refunds for all users for a substantial window of time. An even more radical remedy would be for the software publisher to be required to disclose any compatibility information, including source code.[289] The combination of these two approaches would create a critical mass of users, as well as a clear-cut opportunity for competitors to displace the defective standard.[290]

C. Penalties for Users

Given that users undervalue security,[291] there is a possibility that even if they had the ability to choose more secure software, they would not do so. Decreased worm vulnerabilities may mean fewer features or less convenience;[292] consequently, even if there is no standard-bearer at all, users

---

285. Hughes v. Chrysler Motors Corp., 542 N.W.2d 148, 153 (Wis. 1996) (citation omitted).

286. *See generally* Rebecca Crandall, *Do Computer Purchasers Need Lemon Aid?*, 4 N.C. J.L. & TECH. 307 (2003) (stressing the need for state and federal legislation to safeguard consumers from defective computers); Maurice R. Griffithe, *Computer Lemon Laws: An Evaluation of Existing Defective Computer Remedies and the Proposed Illinois Computer Lemon Act*, 27 S. ILL. U. L.J. 575 (2003) (advocating the enactment of proposed computer lemon legislation).

287. *See supra* subpart III(A).

288. Investment in training, compatible products, and creation of files in a proprietary format (which must be converted) all contribute to switching costs. SHAPIRO & VARIAN, *supra* note 63, at 121–23.

289. This is along the lines of the antitrust penalties recently imposed on Microsoft by the European Union. *See* John Burgess, *Europeans Come Down Hard on Microsoft*, WASH. POST, Mar. 25, 2004, at A1 (relating that the EU gave Microsoft "120 days to disclose 'complete and accurate' data on Windows").

290. *See supra* subpart III(B) (discussing the three factors that must converge in order to deliver a market punishment to software publishers).

291. *See supra* subpart III(D).

292. *See supra* note 57 and accompanying text.

might gravitate toward less secure software versions that provide certain desirable elements despite the attendant risk (or even certainty) of worms. This could be particularly problematic if the software in question was not distributed commercially, and therefore not subject to meaningful sanctions for violating minimum quality standards.

A first step might be to focus on users who fail to use or deactivate worm-preventing features, much like current law prohibits disabling emission control devices on automobiles.[293] Another approach might be to require users to purchase and use software that meets the minimum quality standards. However, any user-focused policy will present daunting enforcement problems.

To a certain extent, the answer may lie with ISPs, rather than users themselves. Some ISPs have taken to cutting off internet access for a period of time when user equipment becomes worm infected.[294] This approach has merit because ISPs are already motivated to reduce worm traffic; requiring a consistent response from ISPs with respect to worm-infected users would eliminate any tendency of ISPs to compete on the basis of turning a blind eye.

VII. Conclusion

This Note began by asking "what things can regulate, or should regulate, worms in cyberspace." Law enforcement is plainly not up to the task. Markets, acting alone, show no propensity for addressing the problem in a meaningful or systematic way. Nor is litigation under current law likely to be effective, and extending tort law to regulate worms faces a number of pitfalls. This Note instead suggests a regulatory mix consisting of mandatory bounties for those who discover new security vulnerabilities that could lead to worms, technology-based standards, a "lemon law" for standards, and minimal penalties for users to encourage them to use more secure software.

–Douglas A. Barnes

---

293. *See* 69 Fed. Reg. 39,265 (June 29, 2004) (to be codified at 40 C.F.R. § 1068.101(b)(1)) ("You may not remove or disable a device or element of design that may affect an engine's emission levels.").

294. For instance, in 2003, The University of Texas adopted a policy of disabling network access for worm-infected computers. As a result of this policy, the author's wireless network access was blocked for several days, an event which led in part to the writing of this Note. *See* The University of Texas at Austin Information Technology Services, Microsoft Windows Vulnerability Now Subject to Worm Attack, *at* http://www.utexas.edu/its/alerts/securewindows.html (last visited Oct. 5, 2004) ("It is possible that your system may be infected. If so, your network access will be blocked or restricted.").