COMPUTER VIRUSES AND CIVIL LIABILITY: A CONCEPTUAL FRAMEWORK

Meiring de Villiers

This article analyzes a negligence cause of action for inadvertent transmission of a computer virus. It provides an introduction to the principles of operation and detection of viruses and analyzes the elements of negligence liability in the context of virus infection. A final section discusses and analyzes litigation complications that are a direct result of the dynamic and unique nature of virus and virus detection technology.

I. INTRODUCTION

The Internet and modern communications technology have stimulated unprecedented advances in electronic communication, commerce, and information access. These technologies also have dramatically increased the vulnerability of computer networks to hazards, such as malevolent software and rogue programs that are capable of spreading rapidly and causing widespread and substantial damage to electronic data and programs.[1] The most

1. KEN DUNHAM, BIGELOW'S VIRUS TROUBLESHOOTING POCKET REFERENCE xix–xxiii (2000) ("Current Threat of Viruses" and "Interpreting the Threat."); Jeffrey O. Kephart et al., *Blueprint for a Computer Immune System*, IBM Thomas J. Watson Research Center Report, at 1 (originally presented at Virus Bulletin International Conference in San Francisco, California (Oct. 1–3, 1997), *available at* http://www.research.ibm.com/antivirus/SciPapers/Kephart/VB97 ("There is legitimate concern that, within the next few years, the Internet will provide a fertile medium for new breeds of computer viruses capable of spreading orders of magnitude faster than today's viruses . . . [T]he explosive growth of the Internet and the rapid emergence of applications that disregard the traditional boundaries between computers threaten to increase the global spread rate of computer viruses by several orders of magnitude."); *How Fast a Virus Can Spread*, *in* PHILIP FITES ET AL., THE COMPUTER VIRUS CRISIS 21 (2d ed. 1992); Carey Nachenberg, *Future Imperfect*, VIRUS BULL. (Aug. 1997) ("With the ubiquitous nature of the Internet, new viruses can be made widely accessible within minutes."); BIZREPORT NEWS,

Meiring de Villiers (mdv@unsw.edu.au) is John Landerer Faculty Fellow at the University of New South Wales School of Law in Sydney, Australia.

notorious of these rogue programs is the so-called computer virus, a program capable of attaching itself to a host program, cloning itself, and spreading the cloned copies to other host programs, analogously to a biological virus. In addition to replicating and spreading, many viruses are also capable of harm, such as information theft and corruption of electronic data. This article focuses on the computer virus and its legal impact.

A collateral effect of the proliferation of malevolent software is exposure to legal liability, not only for the virus author and the intentional transmitter of a virus, but also for one who inadvertently transmits a virus. An example of the latter would be someone who unwittingly forwards an infected e-mail attachment. A civil action against an inadvertent transmitter would most likely be pursued under a negligence theory, the most widely used theory of liability in the law of torts.[2]

Negligence is a breach of the duty not to impose an unreasonable risk on society. It applies to any risk that can be characterized as unreasonable, including the risks associated with malevolent software.[3] A victim of a virus attack may therefore bring legal action under a negligence theory against anyone who failed to take reasonable care to eliminate or reduce the risk of virus infection.

Potential defendants in a virus case include such individuals as commercial software providers who sell infected products; entities involved in software distribution, such as website operators and participants in shareware arrangements; and individuals who transmit infected e-mail attachments. The system operator in a workplace who becomes aware that an internal network is infected with a virus may have a duty to external e-mail recipients to reduce or eliminate the risk of infection. This can be accomplished by advising internal e-mail users, blocking all external e-mail traffic, or including warnings with outgoing e-mail, until the system has been disinfected with reasonable certainty.[4]

Sept. 12, 2003 (reporting that five to fifteen new viruses are released on the Internet daily), *at* http://www.bizreport.com/print.php?art_id = 4917. For those interested in pursuing the scientific aspect further, IBM's website *at* http://www.research.ibm.com/antivirus/SciPapers.htm provides hyperlinks to numerous papers on viruses, including many cited in this article.

2. *See*, *e.g.*, James A. Henderson, *Why Negligence Law Dominates Tort*, 50 UCLA L. Rev. 377 (2003). *See also* Gary T. Schwartz, *The Vitality of Negligence and the Ethics of Strict Liability*, 15 Ga. L. Rev. 963 (1981); Gary T. Schwartz, *The Beginning and the Possible End of Modern American Tort Law*, 26 Ga. L. Rev. 601 (1992).

3. Prosser and Keeton on the Law of Torts § 31 (5th ed. 1984). Restatement (Second) of Torts, § 282 (1965) (describing negligence as conduct "which falls below the standard established by law for the protection of others against unreasonable risk of harm"); Dan B. Dobbs, The Law of Torts 258 (the plaintiff can assert that *any* conduct counts as negligence).

4. Clive Gringras, The Laws of the Internet 61, 62 (1997). An English court held that a defendant who stored biological viruses had a duty to cattle owners who would be affected by the spread of the virus. *Weller and Co. v. Foot and Mouth Disease Research Institute*, 3 All E.R. 560, 570 (1965) ("[T]he defendant's duty to take care to avoid the escape of the virus was due to the foreseeable fact that the virus might infect cattle in the neighborhood and

To pursue a successful negligence cause of action, a victim of viral infection must prove that (1) the defendant had a duty to the plaintiff to take reasonable care to avoid the infection, (2) there was a breach of that duty, (3) the breach was the actual and legal cause of the plaintiff's loss, and (4) the breach resulted in actual harm.

Technology plays a crucial role in a negligence analysis involving virus infection. Courts require a plaintiff to prove breach of duty in a negligence action by identifying an untaken precaution and showing that the precaution would have yielded greater benefits in accident reduction than its cost. Such a cost–benefit analysis requires a familiarity with the technology as well as economics of viruses and virus detection.

Section II of this article reviews the principles of computer viruses and virus detection technology. Section III presents an analytical framework for the evaluation of a negligence cause of action in a virus context, including an analysis of legal and economic aspects of damages due to computer virus infection.

The dynamic nature of virus technology may complicate proof of negligence liability. The central element of a negligence plaintiff's litigation strategy is the cost-effective untaken precaution. Failure to take a particular precaution may constitute breach, but the claim nevertheless may fail on proximate cause grounds if, for instance, the virus evolved unpredictably and caused an unforeseeable type of harm. An alternative precaution may pass the actual and proximate cause hurdles but would likely not be cost-effective, and therefore fail the breach-of-duty element. Such interaction between the dynamic and volatile nature of virus technology and the legal principles of negligence may create a Catch-22 situation that leaves the virus victim without legal recourse. Section IV analyzes and discusses these and other complications to litigation strategy. A final section discusses and concludes.

II. OPERATION AND STRUCTURE OF COMPUTER VIRUSES

A. *Background*

Malevolent software is intended to cause damage to or disrupt the operation of a computer system. The most common of these rogue programs is the computer virus. Other forms of malicious software include so-called logic bombs, worms, Trojan horses, and trap doors.[5]

cause them to die. The duty is accordingly owed to the owners of cattle in the neighborhood . . ."). Bulletin Boards, which allow downloading and uploading of software, are particularly vulnerable to computer virus infection due to the sheer quantity of transactions performed through Bulletin Board Systems. *See, e.g.*, FITES ET AL., *supra* note 1, at 60.

5. *See, e.g.*, DOROTHY E. DENNING & PETER J. DENNING, INTERNET BESIEGED 75–78 (1998).

The term "virus," Latin for "poison," was first formally defined by Dr. Fred Cohen in 1983,[6] even though the concept goes back to John von Neumann's studies of self-replicating mathematical automata in the 1940s.[7] Dr. Cohen describes a computer virus as a series of instructions (in other words, a program) that (i) infects other computer programs and systems by attaching itself to a host program in the target system, (ii) executes when the host is executed, and (iii) spreads by cloning itself, or part of itself, and attaching the copies to other host programs on the system or network. In addition, many viruses have a so-called payload capable of harmful side-effects, such as data corruption.[8]

A virus may infect a computer or a network through several possible points of entry, including via an infected file downloaded from the Internet, through Web browsing, via an infected e-mail attachment, or even through infected commercial shrinkwrapped software.[9] The recent trend in virus transmission has been a decrease in infected diskettes and an increase in infection through e-mail attachments. In a 1996 national survey, for instance, approximately 9 percent of respondents listed e-mail attachments as the means of infection of their most recent virus incident, while 71 percent put the blame on infected diskettes. In 2003, the corresponding numbers were 88 percent for e-mail attachments and zero for diskettes.[10]

As the definition suggests, computer viruses consist of three basic modules or mechanisms, namely an infection mechanism, a payload trigger, and the payload. The infection mechanism allows the virus to replicate and

6. Fred Cohen, Computer Viruses (1985) (unpublished Ph.D. dissertation, University of Southern California) (on file with the University of Southern California library).

7. Jeffrey O. Kephart et al., *Fighting Computer Viruses*, Sci. Am., Nov. 1997, at 55. Dr. Gregory Benford published the idea of a computer virus as "unwanted code." Benford apparently wrote actual "viral" code, capable of replication. Denning & Denning, *supra* note 5, at 74.

8. John Macafee & Colin Haynes, Computer Viruses, Worms, Data Didlers, Killer Programs, and Other Threats to Your System 26; Frederick B. Cohen, A Short Course on Computer Viruses 1–2 (2d ed. 1994). In his Ph.D. dissertation, Dr. Cohen defined a virus simply as any program capable of self-reproduction. This definition appears overly general. A literal interpretation of the definition would classify even programs such as compilers and editors as viral. Denning & Denning, *supra* note 5, at 75.

9. There are three mechanisms through which a virus can infect a program. A virus may attach itself to its host as a shell, as an add-on, or as intrusive code. A shell virus forms a shell around the host code so that the latter effectively becomes an internal subroutine of the virus. The host program is replaced by a functionally equivalent program that includes the virus. The virus executes first and then allows the host code to begin executing. Boot program viruses are typically shell viruses. Most viruses are of the add-on variety. They become part of the host by appending their code to the host code, without altering the host code. The viral code alters the order of execution, by executing itself first and then the host code. Macro viruses are typically add-on viruses. Intrusive viruses, in contrast, overwrite some or all of the host code, replacing that with its own code. *See, e.g.*, Denning & Denning, *supra* note 5, at 81; Fites et al., *supra* note 1, at 73–75.

10. Inst. for Computer Sec. & Admin., ICSA Labs 9th Annual Computer Virus Prevalence Survey 2003, Table 10, at 14, *available at* http://www.icslabs.com/2003avpsurvey/index.shml.

spread, analogously to a biological virus. This is the most salient property of a computer virus.[11] The infection module first searches for an appropriate executable host program to infect. It then installs a copy of the virus into the host, provided the host has not yet been infected.

When the host program executes, the virus is also executed. Upon execution, the virus typically performs the following sequence of actions. It replicates (clones) by copying itself to other executable programs on the computer.[12] During execution, the virus program also checks whether a triggering condition is satisfied. When the condition is satisfied, the virus executes its harmful component, the so-called payload module. Triggering events come in a variety of forms, such as a certain number of infections, Michelangelo's birthday, or the occurrence of a particular date. The Friday-the-13th virus, for instance, only activates its payload on dates with the cursed designation.[13]

Execution of the payload may produce harmful side effects, such as destruction or corruption of data in spreadsheets, word processing documents, and databases and theft of passwords.[14] Some effects are particularly pernicious because they are subtle and undetectable until substantial harm has been done: transposing numbers, moving decimal places, stealing passwords and other sensitive information.[15] Payloads are not necessarily destructive and may involve no more than displaying a humorous message.[16] Some virus strains do not destroy or corrupt information but consume valuable computing resources.[17]

---

11. Rogue Programs: Viruses, Worms, Trojan Horses 247 (Lance J. Hoffman ed. 1990) ("The ability to propagate is essential to a virus program."); Denning & Denning, *supra* note 5, at 73–75.

12. Potential target hosts include application and system programs and the master boot record of the hard disks or floppy disks in the computer.

13. *See*, *e.g.*, Eric J. Sinrod & William P. Reilly, *Cyber Crimes: A Practical Approach to the Application of Federal Computer Crime Laws*, 16 Santa Clara Computer & High Tech. L.J. 177, 217 n.176 (2000).

14. Jan Hruska, Computer Viruses and Anti-Virus Warfare 17, 17–18 (1990) (In addition to self-replicating code, viruses often also contain a payload. The payload is capable of producing malicious side effects.). *See also* Cohen, *supra* note 8, at 8–15 (examples of malignant viruses and what they do); Macafee & Haynes, *supra* note 8, at 61.

15. Macafee & Haynes, *supra* note 8, at 61.

16. Sinrod & Reilly, *supra* note 13, at 218 (describing the W95.LoveSong.998 virus, designed to trigger a lovesong on a particular date).

17. Viruses can cause economic losses, e.g., by filling up available memory space, slowing down the execution of important programs, locking keyboards, adding messages to printer output, and effectively disabling a computer system by altering its boot sector. The Melissa virus, for instance, mailed copies of itself to everyone in the victim's e-mail address book, resulting in clogged e-mail servers and even system crashes. *See*, *e.g.*, Fites et al., *supra* note 1, at 23–24 ("The Christmas card [virus] stopped a major international mail system just by filling up all available storage capacity."); Sinrod & Reilly, *supra* note 13, at 218 (describing the Melissa virus).

See Section III(D), *infra*, for an analysis of damages from computer virus infection. For examples of benign viruses and how they operate, *see*, *e.g.*, Cohen, *supra* note 8, at 15–21.

It was once believed that viruses could not be transmitted by data files such as e-mail attachments. Viruses such as the infamous Melissa taught us otherwise. Melissa typically arrived in the e-mail inbox of its victim disguised as an e-mail message with a Microsoft Word attachment. When the recipient opened the attachment, Melissa executed. First, it checked whether the recipient had the Microsoft Outlook e-mail program on its computer. If so, Melissa would mail a copy of itself to the first fifty names in Outlook's address book, creating the appearance to the fifty new recipients that the infected person had sent them a personal e-mail message. Melissa would then repeat the process with each of the fifty recipients of the infected e-mail message (provided they had Outlook) by automatically transmitting clones of itself to fifty more people.[18] A Melissa attack frequently escalated and resulted in clogged e-mail servers and system crashes.

B. *Technical Antivirus Defenses*

Antivirus technology comes in two broad categories: virus-specific and generic. Virus-specific technology, such as signature scanners, detect known viruses by identifying patterns that are unique to each virus strain. These "identifying patterns" are analogous to human fingerprints. Generic technology detects the presence of a virus by recognizing generic viruslike behavior, usually without identifying the particular strain.

A virus-specific scanner typically makes a specific announcement, such as that "the operating system is infected with (say) the Cascade virus," while its generic counterpart may simply say, "the operating system is (or may be) infected with an (unidentified) virus." Virus-specific technology is more accurate and produces fewer false positives, but generic technology is better at detecting unknown viruses. Heuristic techniques combine virus-specific scanning with generic detection, providing a significantly broadened range of detection.

Technical antivirus defenses come in four varieties, namely scanners, activity monitors, integrity checkers, and heuristic techniques.[19] Scanners are virus-specific, while activity monitors and integrity checkers are generic. Activity monitors look out for suspicious, viruslike activity in the computer. Integrity checkers sound an alarm when they detect suspicious modifications to computer files.

1. Scanners

Scanners are the most widely used antivirus defense. A scanner reads executable files and searches for known virus patterns. These patterns, or "sig-

18. David Harley et al., Viruses Revealed: Understand and Counter Malicious Software 406–10 (2001).

19. *See, e.g.*, Denning & Denning, *supra* note 5, at 90–93; Dunham, *supra* note 1, at 78–83, 102–08.

natures," are the most reliable technical indicator of the presence of a virus in a computer system. A virus signature consists of patterns of hexadecimal digits embedded in the viral code that are unique to the strain.[20] These signatures are created by human experts, such as researchers at IBM's High Integrity Computing Laboratory, who scrutinize viral code and extract sections of code with unusual patterns. The selected byte patterns then constitute the signature of the virus.[21] The scanner announces a match with its database of known viral signatures as a possible virus.

The virus signature pattern is selected to be a reliable indicator of the presence of a virus. An ideal virus signature gives neither false negatives nor false positives.[22] In other words, it should ideally always identify the virus when present and never give a false alarm when it is not.[23] The IBM High Integrity Computing Laboratory has developed an optimal statistical signature extraction technique that examines all sections of code in a virus and selects the byte strings that minimize the incidence of false positives and negatives.[24]

Scanners are easy to use, but they are limited to detecting known virus signatures. A scanner's signature database has to be continually updated, a burdensome requirement in an environment where new viruses appear rapidly. Use of scanners is further complicated by the occurrence of false positives. This occurs when a viral pattern in the database matches code that is in reality a harmless component of otherwise legitimate data. A short and simple signature pattern will be found too often in innocent software and produce many false positives. Viruses with longer and more complex patterns will less often give a false positive, but at the expense of more false negatives.[25] Finally, as the number of known viruses grows, the scanning process will inevitably slow down as a larger set of possibilities has to be evaluated.[26]

20. Hruska, *supra* note 14, at 42.

21. Jeffrey O. Kephart et al., *Automatic Extraction of Computer Virus Signatures*, *in* Proceedings of the 4th Virus Bulletin International Conference (R. Ford ed., 1994), *available at* http://www.research.ibm.com/antivirus/SciPapers/Kephart/VB94/vb94.html/179–94, at 2.

22. A false positive is an erroneous report of the activity or presence of a virus where there is none. A false negative is the failure to report the presence of a virus when a virus is in fact present.

23. Hruska, *supra* note 14, at 42. For short descriptions and hexadecimal patterns of selected known viruses, *see id.* at 43–52; Kephart et al., *supra* note 1, at 11 ("[A] signature extractor must select a virus signature carefully to avoid both false negatives and false positives. That is, the signature must be found in every instance of the virus, and must almost never occur in uninfected programs."). False positives have reportedly triggered a lawsuit by a software vendor, who felt falsely accused, against an antivirus software vendor. *Id.*

24. Kephart et al., *supra* note 21, at 179–94.

25. Dunham, *supra* note 1, at 78–83; Kephart et al., *supra* note 7. *See also* Sandeep Kumar & Eugene H. Spafford, *A Generic Virus Scanner in C++*, Technical Report CSD-TR-92–062, Dep't of Computer Science, Indiana University, at 6–8, *available at* ftp://Ftp.cerias.purdue.edu/pub/papers/sandeep-kumar/kumar-spaf-scanner.pdf.

26. *See, e.g.*, Pete Lindstrom, *The Hidden Costs of Virus Protection*, Spire Res. Rep. 5 (June

2. Activity Monitors

Activity monitors are resident programs that monitor activities in the computer for behavior commonly associated with viruses. Suspicious activities include operations such as attempts to rewrite the boot sector, format a disk, or modify parts of main memory. When suspicious activity is detected, the monitor may simply halt execution and issue a warning to alert the user, or take definite action to neutralize the activity.[27] Activity monitors, unlike scanners, do not need to know the signature of a virus to detect it. It works for all viruses, known as well as unknown. Its function is to recognize suspicious behavior, regardless of the identity of the culprit.

The greatest strength of activity monitors is their ability to detect unknown virus strains, but they also have significant weaknesses. They can only detect viruses that are actually being executed, possibly after substantial harm has been done. A virus, furthermore, may become activated before the monitor code and thus escape detection until well after execution. A virus also may be programmed to alter monitor code on machines that do not have protection against such modification. A further disadvantage of activity monitors is the lack of unambiguous and foolproof rules governing what constitutes suspicious activity. This may result in false alarms when legitimate activities resemble viruslike behavior. Recurrent false alarms ultimately may lead users to ignore warnings from the monitor. Conversely, not all illegitimate activity may be recognized as such, leading to false negatives.[28]

3. Integrity Checkers

Integrity checkers look for unauthorized changes in system areas and files. The typical integrity checker is a program that generates a code, known as a checksum, for files that are to be protected from viral infection. A file checksum, for instance, may be some arithmetic calculation based on the total number of bytes in the file, the numerical value of the file size, and the creation date. The checksum effectively operates as a signature of the file. These checksums are periodically recomputed and compared to the original checksum. Tampering with a file will change its checksum. Hence, if the recomputed values do not match the original checksum, the file has presumably been modified since the previous check and a warning is issued.

2003) ("In this day of 80,000+ known viruses and frequent discovery of new ones, the size of the signature file can be large, particularly if the updates are sent out as cumulative ones. Large updates can clog the network pipelines . . . and reduce the frequency that an administrator will push them out to the end users.").

27. Kumar & Spafford, *supra* note 25, at 3–4.

28. Hruska, *supra* note 14, at 75.

Since viruses modify and change the contents of the files they infect, a change in the checksum may be a sign of viral infection.[29]

The advantage of integrity checking is that it detects most instances of viral infection, as infection must alter the file being infected. The main drawback is that it tends to generate many false alarms, as a file can change for legitimate reasons unrelated to virus infection.[30] On some systems, for instance, files change whenever they are executed. A relatively large number of false alarms may trigger compliance lapses, as users may ignore warnings or simply not use the utility. Integrity checking works best on static files, such as system utilities, but is, of course, inadequate for files that naturally change frequently, such as Word documents.

4. Heuristic Detection Methods

A fourth category of virus detectors uses heuristic detection methods. Heuristic rules are rules that solve complex problems fairly well and fairly quickly, but less than perfectly. Virus detection is an example of a complex problem that is amenable to heuristic solution. It has been proven mathematically that it is impossible to write a program that is capable of determining with 100 percent accuracy whether a particular program is infected with a virus, from the set of all possible viruses, known as well as unknown.[31] Heuristic virus detection methods accept such limitations and attempt to achieve a solution, namely a detection rate that is acceptable, albeit below the (unachievable) perfect rate.

Heuristic virus detection methods examine executable code and scrutinize its structure, logic, and instructions for evidence of viruslike behavior. Based on this examination, the program makes an assessment of the likelihood that the scrutinized program is a virus, by tallying up a score. Instructions to send an e-mail message with an attachment to everyone in an address book, for instance, would add significantly to the score. Other high-scoring routines include capabilities to replicate, hide from detection, and execute some kind of payload. When a certain threshold score is reached, the code is classified as malevolent and the user so notified.

The assessment is necessarily less than perfect and occasionally provides false positives and negatives. Many legitimate programs, including even

29. Fites et al., *supra* note 1, at 69–76 (Figures 5.2–5.5); Dunham, *supra* note 1, at 79. *See also* Kumar & Spafford, *supra* note 25, at 5–6.

30. Fites et al., *supra* note 1, at 125.

31. Diomidis Spinellis, *Reliable Identification of Bounded-Length Viruses Is NP-Complete*, 49:1 IEEE Transactions on Information Theory 280, 282 (Jan. 2003) (stating that theoretically perfect detection is in the general case undecidable, and for known viruses, NP-complete.); Nachenberg, *supra* note 1. *See also* Francisco Fernandez, *Heuristic Engines*, *in* Proceedings of the 11th International Virus Bulletin Conference 407–44 (Sept. 2001); David M. Chess & Steve R. White, *An Undetectable Computer Virus*, *at* http://www.research.ibm.com/antivirus/SciPapers/VB2000DC.htm.

some antivirus programs, perform operations that resemble viruslike behavior.[32] Nevertheless, state-of-the-art heuristic scanners typically achieve a 70 percent to 80 percent success rate at detecting unknown viruses.[33]

A heuristic scanner typically operates in two phases. The scanning algorithm first narrows the search by, for instance, identifying the location most likely to contain a virus. It then analyzes the code from that location to determine its likely behavior upon execution. A static heuristic scanner, for instance, compares the code from the most likely location to a database of byte sequences commonly associated with viruslike behavior.[34] The algorithm then decides whether to classify the code as viral.[35]

A dynamic heuristic scanner uses central processing unit (CPU)[36] emulation. It typically loads suspect code into a virtual computer, emulates its execution, and observes its behavior. Because it is only a virtual computer, viruslike behavior can safely be observed in what is essentially a laboratory setting, with no need to be concerned about real damage. The program is monitored for suspicious behavior while it runs.[37]

Although dynamic heuristics can be time-consuming due to the relatively slow CPU emulation process, they are sometimes superior to static heuristics. This will be the case when the suspect code (i) is obscure and not easily recognizable as viral in its static state but (ii) clearly reveals its viral nature in a dynamic state.

A major advantage of heuristic scanning is its ability to detect viruses, including unknown strains, before they execute and cause damage. Other generic antivirus technologies, such as behavior monitoring and integrity checking, can only detect and eliminate a virus after exhibition of suspicious behavior, usually after execution. Heuristic scanning is also capable of detecting novel and unknown virus strains, the signatures of which have not yet been catalogued. Such strains cannot be detected by conventional scanners, which only recognize known signatures. Heuristic scanners are capable of detecting even polymorphic viruses, a complex virus family that complicates detection by changing their signatures from infection to infection.[38]

32. Fernandez, *supra* note 31, at 409 ("Many genuine programs use sequences of instructions that resemble those used by viruses. Programs that use low-level disk access methods, TSRs, encryption utilities, and even anti-virus packages can all, at times, carry out tasks that are performed by viruses.").

33. Nachenberg, *supra* note 1, at 7.

34. Certain byte sequences, for instance, are associated with decryption loops to unscramble a polymorphic virus when an infected routine is executed. If it finds a match, *e.g.*, the scanner detects the presence of a decryption loop typical of a polymorphic virus, it catalogues this behavior.

35. Kumar & Spafford, *supra* note 25, at 4–5 ("Detection by static analysis/policy adherence.").

36. The CPU, or central processing unit, of a computer is responsible for data processing and computation. *See, e.g.*, Hruska, *supra* note 14, at 115; D. Bender, Computer Law: Evidence and Procedure § 2.02, at 2–7, 9 (1982).

37. Kumar & Spafford, *supra* note 25, at 4.

38. Polymorphic viruses have the ability to "mutate" by varying the code sequences written

The explosive growth in new virus strains has made reliable detection and identification of individual strains very costly, making heuristics more important and increasingly prevalent.[39] Commercial heuristic scanners include IBM's AntiVirus boot scanner and Symantec's Bloodhound technology.

We now turn to a formal analysis of negligence in a virus context.

III. VIRUS INFECTION AS NEGLIGENCE CAUSE OF ACTION

A product, a service, or conduct cannot and does not have to be perfectly safe to avoid liability. Society does not benefit from products that are excessively safe, such as bugfree software and automobiles built like armored cars and limited to top speeds of twenty miles per hour. Even if bugfree software were feasible, the resources consumed in achieving it would make the product prohibitively expensive when it is finally released, and also likely obsolete.

Society does not benefit from products that are too risky either. Society benefits most from an optimal level of safety.[40] In this section, we explore the legal meaning of these concepts and the closely related question: how safe does a product, including an intangible such as a computer program, have to be to avoid liability?

Any risk in principle can be reduced or eliminated, at a cost. For many risks, this cost exceeds the benefit of the risk reduction. We call such risks "unavoidable." Risks that, on the other hand, can be reduced at a cost less than the benefit of the reduction are called "avoidable." Unavoidable risks provide a net benefit to society and, as a matter of public policy, should not be eliminated. The converse is true in the case of avoidable risks.

The law of negligence recognizes this distinction and limits liability to harm caused by avoidable risks. The primary legal meaning of the term *negligence* is conduct that is unreasonably risky; in other words, conduct that imposes an avoidable risk on society.[41]

---

to target files. To detect such viruses requires a more complex algorithm than simple pattern matching. *See, e.g.*, Denning & Denning, *supra* note 5, at 89.

39. Nachenberg, *supra* note 1, at 9.

40. Bender, *supra* note 36, at 8–41 to 8–42 n.108; C. Cho, An Introduction to Software Quality Control 4, at 12–13 (1980) (a software provider is under a duty to invest resources in program debugging only up to the point where the cost of additional debugging would outweigh the benefits of further error reduction); Thomas G. Wolpert, *Product Liability and Software Implicated in Personal Injury*, Def. Coun. J. 519, 523 (Oct. 1993) ("By the time a product is completely debugged, or nearly so, most likely it is obsolete.") *See also* Ivars Peterson, Fatal Defect 166 (1995) ("We live in an imperfect world . . . Absolute safety, if attainable, would . . . cost more than it's worth.").

41. Prosser and Keeton on the Law of Torts, *supra* note 3, § 31; Dobbs, *supra* note 3, at 275 ("Negligence is conduct that creates or fails to avoid unreasonable risks of foreseeable harm to others."). The term also refers to the cause of action, namely the legal rules and procedures that govern a negligence lawsuit. *Id.* at 269.

The remainder of this section discusses and analyzes the legal principles that define the dividing line between avoidable and unavoidable risks, and applies the principles in the context of computer virus infection.[42]

The plaintiff in a negligence action has to prove the following elements to establish his or her claim.

1. A legal *duty* on the part of the defendant not to expose the plaintiff to unreasonable risks.
2. A *breach* of the duty; namely, a failure on the part of the defendant to conform to the norm of reasonableness.
3. A *causal* connection between defendant's conduct and plaintiff's harm. This element includes actual as well as proximate cause. Defendant's negligence is the actual cause of the plaintiff's harm if, but for the negligence, the harm would not have occurred. Proximate causation means that the defendant's conduct must be reasonably closely related to the plaintiff's harm.
4. Actual *damage* resulting from the defendant's negligence.

We now turn to an analysis of these elements in a computer virus context.

A. *Duty*

The first step in a negligence analysis considers whether the defendant had a duty to the plaintiff to act with due care or, conversely, whether the plaintiff is entitled to protection against the defendant's conduct.[43] But how and where do we draw the line that divides the plaintiffs who are entitled to such protection from those who are not? Professor Richard Epstein phrases the rhetorical question, "[w]ho, then, in law, is my neighbor?" He finds an answer in *Donoghue v. Stevenson*: My neighbors are "persons who are so closely and directly affected by my act that I ought reasonably to have them in contemplation as being so affected when I am directing my mind to the acts or omissions which are called in question."[44]

The courts frequently analyze the duty issue as a matter of public policy. A defendant has a duty to the plaintiff if a balancing of policy considerations dictates that the plaintiff is entitled to legal protection against the defendant's conduct.[45] The policy benchmark is based on fairness under the

42. Liability fo*r intentional* transmission of a virus is governed by criminal law. A software provider who intentionally transmits a computer virus with the purpose of stealing, destroying, or corrupting data in the computer of his competitor may be prosecuted under criminal statutes such as the Computer Fraud and Abuse Act, 18 U.S.C. § 1030. This act is the principal federal statute governing computer-related abuses, such as transmission of harmful code.

43. Prosser and Keeton on the Law of Torts, *supra* note 3, at 357 n.14.

44. Donoghue v. Stevenson, [1932] App. Cas. 562, 580 (H.L. Scot. 1932) (cited in Richard A. Epstein, Simple Rules for a Complex World 196 (1995)).

45. Brennen v. City of Eugene, 591 P.2d 719 (Or. 1979); Bigbee v. Pac. Tel. & Tel. Co., 183 Cal. Rptr. 535 (Ct. App. 1982); Prosser and Keeton on the Law of Torts, *supra* note 3, at 358 ("[D]uty is not sacrosanct in itself, but is only an expression of the sum total of those considerations of policy which lead the law to say that the defendant is entitled to protection.").

contemporary standards of a reasonable person.[46] Prosser succinctly summarizes, "[n]o better general statement can be made than that the courts will find a duty where, in general, reasonable persons would recognize it and agree that it exists."[47]

In fleshing out the reasonable person policy benchmark of duty, courts consider factors such as the relationship between the parties, the nature of the risk, the opportunity and ability to take care, the public interest,[48] and whether the defendant created the risk that caused the loss.[49]

Courts are more likely to recognize a duty in cases where the defendant possesses a "special relationship" with the plaintiff.[50] A common carrier, for instance, has a duty to aid a passenger in trouble, an innkeeper to aid a guest, and an employer to aid an employee injured or endangered in the course of his employment.[51] The law does not, however, impose a general duty to aid another human being who is in grave, even mortal, danger. A champion swimmer, for instance, is not required to help a child drowning before his eyes, nor is anyone required to warn someone about to stick his hand into a milling machine.[52]

Given the high level of awareness and publicity surrounding virus attacks and computer security, courts are likely to find that software providers and distributors generally do have a duty not to impose an unreasonable risk of viral infection on those foreseeably affected.[53] A software provider, for instance, who invites customers to download a software product from a commercial website creates a risk that the software may contain a virus.

46. Casebolt v. Cowan, 829 P.2d 352, 356 (Colo. 1992) ("The question whether a duty should be imposed in a particular case is essentially one of fairness under contemporary standards—whether reasonable persons would recognize a duty and agree that it exists."). *See also* Hopkins v. Fox & Lazo Realtors, 625 A.2d 1110 (N.J. 1993) ("Whether a person owes a duty of reasonable care toward another turns on whether the imposition of such a duty satisfies an abiding sense of basic fairness under all of the circumstances in light of considerations of public policy.").

47. Prosser and Keeton on the Law of Torts, *supra* note 3, at 359.

48. *Hopkins*, 625 A.2d at 1110.

49. Weirum v. RKO Gen., Inc., 15 Cal. 3d 40, 46 (1975).

50. Lopez v. S. Cal. Rapid Transit Dist., 710 P.2d 907, 911 (Cal. 1985); *see also* Tarasoff v. Regents of Univ. of Cal., 551 P.2d 334, 342 (Cal. 1976).

51. Prosser and Keeton on the Law of Torts, *supra* note 3, at 376, 377 nn.32–42.

52. Handiboe v. McCarthy, 151 S.E.2d 905 (Ga. Ct. App. 1966) (no duty to rescue child drowning in swimming pool); Chastain v. Fuqua Indust., Inc., 275 S.E.2d 679 (Ga. Ct. App. 1980) (no duty to warn child about dangerous defect in lawn mower).

53. Fites et al., *supra* note 1 at 141, 142 (Bulletin Board System operators provide a forum for exchange of information, data, and software. Hence, a BBS operator may have a duty to screen uploaded software for malicious components or, at least, warn users to use caution in using downloaded software.); Palsgraf v. Long Island R.R. Co., 162 N.E. 99 (N.Y. 1928) (establishing the precedent that a duty is extended only to those foreseeably affected). *See also* David L. Gripman, *The Doors Are Locked but the Thieves and Vandals Are Still Getting In: A Proposal in Tort to Alleviate Corporate America's Cyber-Crime Problem*, 16 J. Marshall J. Computer & Info. L. 167, 170 (1997).

Everyone who downloads the software is within the scope of the risk of virus infection and may have a cause of action if harmed by a virus.

B. *Breach*

"Breach of duty" refers to a violation of the duty to avoid unreasonable risks of harm to others. The legal standard of reasonableness against which the defendant's conduct is to be measured is known as the "reasonable person" standard. The reasonable person standard imposes on all people the duty to "exercise the care that would be exercised by a reasonable and prudent person under the same or similar circumstances to avoid or minimize reasonably foreseeable risks of harms to others."[54]

Courts have interpreted the reasonable person standard in three broad ways.[55] First, the reasonable person is endowed with characteristics, such as a certain level of knowledge and ability. The reasonable person has shortcomings that the community would tolerate but is otherwise a model of propriety and personifies the community ideal of appropriate behavior. He is allowed to forget occasionally, for instance, but is presumed never to do something "unreasonable" such as crossing the street on a red light at a busy intersection.[56] The defendant's conduct is then compared to that which can be expected from this hypothetical reasonable person. The defendant is considered to be in breach of her duty of due care if her conduct does not measure up to this standard.

Under a second interpretation of the reasonable person standard, a court may adopt rules of conduct, the violation of which is considered prima facie negligence. Violation of a statute, such as a speed limit, is an example of prima facie negligence.

Finally, courts define the reasonableness of a risk in terms of a balance of its costs and benefits.[57] Under the cost–benefit approach, avoidable risks that can be eliminated cost-effectively are considered unreasonable. Failure to eliminate or reduce such risks constitutes a breach of duty. When harm results from an unavoidable risk, on the other hand, the defendant escapes liability.[58]

Professor Henry Terry appears to have been the first to define reasonableness of conduct in terms of a cost–benefit balancing.[59] This approach is an analytical embodiment of the reasonable person standard, and has

54. O.W. Holmes, The Common Law (1881) (the negligence standard is objective, "based on the abilities of a reasonable person, and not the actual abilities of individuals").

55. *See generally* Dobbs, *supra* note 3, at 279.

56. Prosser and Keeton on the Law of Torts, *supra* note 3, § 32.

57. Dobbs, *supra* note 3, at 279.

58. Prosser and Keeton on the Law of Torts, *supra* note 3, § 29 ("[A]n accident is considered unavoidable or inevitable at law if it was not proximately caused by the negligence of any party to the action, or to the accident.").

59. Henry Terry, *Negligence*, 29 Harv. L. Rev. 40 (1915).

become part of mainstream negligence analysis.[60] In fact, this is how courts actually decide negligence cases.[61] Cost–benefit balancing applies naturally in a virus context, and the availability of cost–benefit models of viruses and antivirus defenses in the computer security literature makes it logical and feasible.[62]

Courts apply the cost–benefit approach in a negligence case by focusing on precautions the defendant could have taken but did not.[63] The courts impose on the negligence plaintiff the burden to specify an *untaken precaution* that would have prevented the accident, if taken. The defendant will then be considered negligent if the benefits of risk reduction provided by the pleaded precaution exceed its cost.[64]

The role of the untaken precaution in negligence law is well illustrated in *Cooley v. Public Service Co.*[65] In *Cooley*, the plaintiff suffered harm from a loud noise over a telephone wire. She suggested two untaken precautions that would have prevented the harm, namely (i) a strategically positioned wire mesh basket and (ii) insulating the wires. The court ruled that neither untaken precaution constituted a breach of duty. Both precautions would have increased the risk of electrocution to passersby sufficiently to outweigh the benefits in harm reduction.

In a negligence case, more than one untaken precaution may have greater benefits than costs, and the plaintiff may allege several precautions in the alternative. The court may base a finding of negligence on one or more of the pleaded untaken precautions.[66] The *Cooley* court noted that there may

60. Dobbs, *supra* note 3, at 267.

61. Mark F. Grady, *Untaken Precautions*, 18 J. Legal Stud. 139 (1989) (courts actually decide negligence cases by balancing the costs and benefits of the untaken precaution).

62. *See, e.g.*, Fred Cohen, *A Cost Analysis of Typical Computer Viruses and Defenses*, *in* Computers & Sec. 10 (1991).

63. Grady, *supra* note 61, at 139. The "untaken precautions" approach is how courts actually decide negligence cases. The positive economic theory of breach of duty posits that negligence law aims to minimize social cost. Under this theory, a software provider would escape liability by taking the cost-minimizing amount of precaution. The global social cost–minimization approach is a theoretical idealization, while the untaken precautions approach is a more realistic description of how courts actually determine negligence.

The seminal articles on the positive economic theory of negligence include John Brown, *Toward an Economic Theory of Liability*, 2 J. Legal Stud. 323 (1973); W. Landes & R. Posner, *A Theory of Negligence*, 1 J. Legal Stud. 29 (1972); S. Shavell, *Strict Liability versus Negligence*, 9 J. Legal Stud. 1 (1980).

64. Grady, *supra* note 61, at 139, 143 (1989) (the courts "take the plaintiff's allegations of the untaken precautions of the defendant and ask, in light of the precautions that had been taken, whether some particular precaution promised benefits (in accident reduction) greater than its associated costs"); Delisi v. St. Luke's Episcopal-Presbyterian Hosp., Inc., 701 S.W.2d 170 (Mo. Ct. App. 1985) (plaintiff had to prove physician's breach of duty by specifying the antibiotic he should have been given).

65. 10 A.2d 673 (N.H. 1940).

66. In Bolton v. Stone, [1951] App. Cas. 850 H.L., the plaintiff was hit by a cricket ball and pleaded three untaken precautions, namely failure to erect a sufficient fence, failure to place the cricket pitch further from the road, and failure to prevent cricket balls from falling into the road.

exist a cost-effective precaution, other than the ones actually pleaded, that would have satisfied the breach requirement. It is, however, the plaintiff's burden to identify and plead such a precaution, if indeed it exists.

The cost–benefit approach was first formally adopted by the courts in a decision by Judge Learned Hand, in *United States v. Carroll Towing Co.*[67] In *Carroll Towing*, a barge broke loose and caused an accident. The accident could have been avoided if, for instance, the owner of the barge had had an employee on board who could have prevented the barge from breaking away. According to Judge Hand, "the owner's duty . . . to provide against resulting injuries is a function of three variables: (1) The probability that [the barge] will break away; (2) the gravity of the resulting injury, if she does; [and] (3) the burden of adequate precautions."[68]

Denoting the burden of precaution by *B*, amount of harm by *L*, and the probability of harm by *P*, Judge Hand provided his celebrated formula: Liability would be imposed if *B* is less than the product of *L* and *P*; in other words, when the burden of precaution is less than the expected damages avoided.[69]

The negligence calculus weighs the cost of an untaken precaution against the value of the reduction in *all* foreseeable risks that the precaution would have achieved, not just the risk that actually materialized.[70] In Judge Hand's assessment, the benefit of the reduction in *all foreseeable risks* that would have resulted from having a bargee on board exceeded the cost of the

67. 159 F.2d 169 (2d Cir. 1947).

68. Judge Hand summarized the principles of negligence in *Carroll Towing*: "Since there are occasions when every vessel will break away . . . and . . . become a menace to those about her, the owner's duty . . . to provide against resulting injuries is a function of three variables: (1) The probability that she will break away; (2) the gravity of the resulting injury if she does; (3) the burden of adequate precautions." Denoting the probability by *P*, the injury by *L*, and the burden by *B*, liability depends on whether *B* is less than *P* times *L*. *Id.* at 173.

69. *See also* Indiana Consol. Ins. Co. v. Mathew, 402 N.E.2d 1000 (Ind. Ct. App. 1980) (court discussed the factors involved in negligence analysis, without formally quantifying them, to reach decision that defendant's action was reasonable).

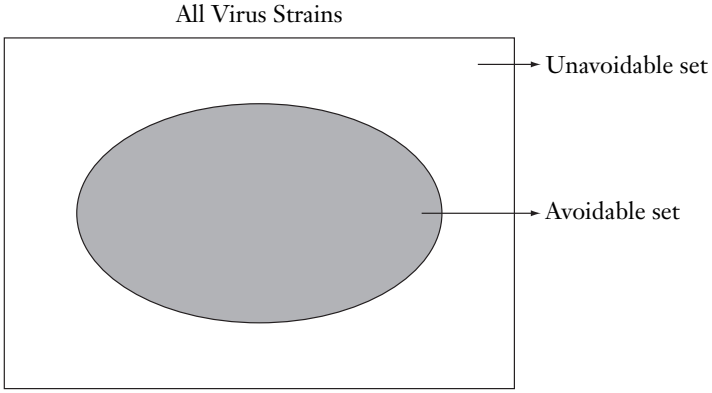
70. *See, e.g.*, Restatement (Second) of Torts § 281(b), cmt. e (1965): "Conduct is negligent because it tends to subject the interests of another to an unreasonable risk of harm. Such a risk may be made up of a number of different hazards, which frequently are of a more or less definite character. The actor's negligence lies in subjecting the other to the aggregate of such hazards."

See also In re Polemis & Furness, Withy & Co., [1921] 3 K.B. 560 (C.A.). In *Polemis*, the defendant's workman dropped a plank into the hold of a ship, causing a spark that caused an explosion of gasoline vapor. The resultant fire destroyed the ship and its cargo. The arbitrators found that the fire was an unforeseeable consequence of the workman's act but that there was nevertheless a breach of duty. The key to the finding of negligence is the fact that courts base their analysis of untaken precautions on a balancing of all foreseeable risks (not just the risk that materialized) against the cost of the untaken precaution. In finding for the plaintiff in *Polemis*, Lord Justice Scrutton stated, "[i]n the present case it was negligent in discharging cargo to knock down the planks of the temporary staging, for they might easily cause some damage either to workmen, or cargo, or the ship [by denting it]." *Id.* at 577.

bargee. The barge owner therefore breached his duty of due care by failing to have a bargee on board.

Like general errors, virus strains can be classified as avoidable or unavoidable. The transmission of a virus strain that a reasonably careful provider would detect and eliminate is an avoidable strain; an unavoidable strain is one that even due care would not have prevented. An example of an unavoidable virus is an unknown, complex strain that could only be detected and eliminated at unreasonably high cost, by, for instance, implementing expensive and sophisticated scanning techniques based on artificial intelligence technology. If the computing environment is such that the stakes are not particularly high, it may not be cost-effective to acquire and implement the expensive technology required to detect such a complex virus.

The universe of all virus strains therefore can be divided into an avoidable and an unavoidable subset, as illustrated in the following diagram.



The following numerical example illustrates application of the cost–benefit principle to prove breach of duty in a virus context. A hypothetical commercial software provider uses a signature scanner[71] to scan for viruses in her software products. A virus escapes detection and finds its way into a product sold to a customer. The virus causes harm in the computer system of the customer. The culprit virus is a novel strain that has been documented fairly recently for the first time. It was not detected because its signature was not included in the database of the software provider's scanner.

The customer contemplates a negligence lawsuit. She must prove the defendant software provider's breach of duty by showing that the defendant

71. See Section II.B, Technical Antivirus Defenses, *supra*, for a discussion of technologies such as signature scanners.

could have used an alternative cost-effective precaution that would have avoided the virus.

The plaintiff has several pleading options. Potential untaken precautions include more frequent updating of the signature database, or perhaps use of a generic scanner that does not depend on an updated database. Each option has its own set of costs and benefits that have to be tallied to evaluate its cost-effectiveness in order to establish liability.

Consider, for instance, the plaintiff's pleading that the software provider should have updated the signature database of her scanner more frequently. This incremental precaution (based on the numbers in this stylized example) is efficient, because doing so would add three cents to the firm's average cost of production but would reduce the expected accident loss by eight cents. The numerical data for the example are summarized in Table 1, below.[72]

Table 1

Behavior of firm	Firm's cost of production per unit	Probability of infection	Loss if infection	Expected loss	Full cost per unit
Current	40 cents	1/100,000	$10,000	10 cents	50 cents
Proposed	43 cents	1/500,000	$10,000	2 cents	45 cents

The first column lists the defendant's alternative precautions, namely scanning at the current rate and scanning at the proposed increased rate, respectively. The second column lists the total production cost per unit of software for each precaution option. The third column lists the probabilities of virus transmission corresponding to the respective precautions; the fifth, the expected losses from a virus attack; and the final column, the full cost per unit of software product, namely production plus expected accident costs. We assume that a virus attack will result in expected damages of $10,000.

With the software provider's current level of precaution, the production cost per unit is forty cents, the chance of an infection is 1/100,000, and the loss if an infection occurs is $10,000. The expected accident loss per unit therefore is ten cents (1/100,000 $\times$ $10,000), and the total cost per unit of software is fifty cents. If, on the other hand, the software provider implemented the proposed precaution pleaded by the plaintiff, the production cost would be forty-three cents, the probability of infection would decline

72. Based on an example in A.M. Polinsky, Introduction to Law and Economics 98 (Table 11) (1983).

to 1/500,000, and the expected loss would be two cents, giving a total cost per software unit of forty-five cents.

Given this information, it is clear that the untaken precaution is efficient, and the plaintiff would prevail on the issue of breach. Although increasing the frequency of signature database updating to the level suggested by the plaintiff would increase production costs by three cents per unit, it lowers expected accident losses by eight cents.

C. *Cause in Fact*

A plaintiff must show that the defendant's negligence was the cause in fact of the plaintiff's harm. Courts usually employ the "but-for" test to determine cause in fact. Under this test, plaintiff's failure to take a precaution is the cause in fact of the harm if the precaution would have avoided the harm. In other words, but for the precaution, the harm would not have occurred.

A plaintiff may fail the but-for test if she pleads the "wrong" untaken precaution. Suppose, for example, that a product manufacturer negligently fails to put a warning about a product hazard in the owner's manual. A user of the product is subsequently injured because of the hazard. If the injured plaintiff admitted he had never read the manual, the manufacturer's negligent failure to warn would not be a but-for cause of the customer's injury. An unread warning would not have been helpful to the user.[73]

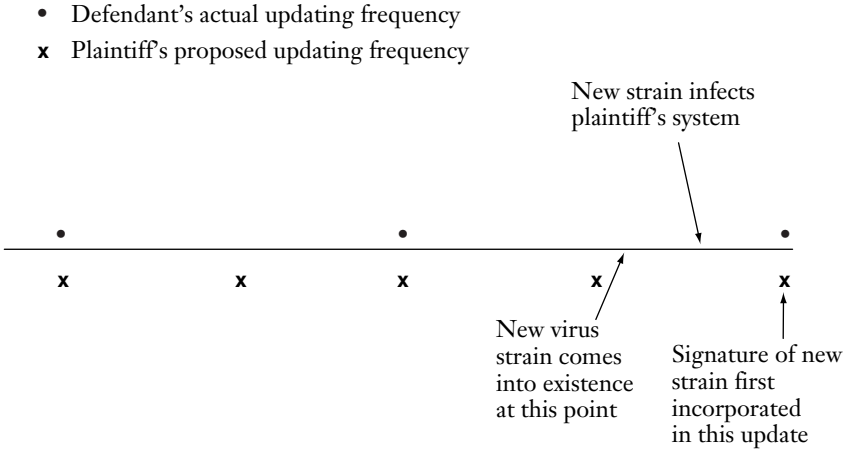
The but-for principle applies similarly in a virus context. Due care may dictate that a virus scanner signature database be updated once a month. If the defendant admits, or discovery shows, that he skipped a month, breach is easily established. If, however, the virus strain is a sufficiently novel variety, its signature would not have been included even in the skipped update. A scanner with a database updated at the due care level would still not have detected the particular strain that caused the harm. Failure to take this precaution constitutes breach of duty but is not an actual cause of the infection.

This hypothetical is illustrated in Figure 2, a timeline of events. The "dot" symbols (•) represent the defendant's actual frequency of signature

73. Dobbs, *supra* note 3, at 410. *See also* McDowall v. Great W. Ry., 1903, 2 K.B. 331 (C.A.), *rev'g* [1902] 1 K.B. 618 (An improperly secured railcar became loose and injured the plaintiffs. The court held that failure to secure the car behind its catchpoint constituted negligence but that the precaution would not have prevented the plaintiff's injuries, as evidence suggested that they were determined to set the car free. The cause-in-fact requirement was therefore not met and the negligence action failed. Failure to take the pleaded untaken precaution constitutes negligence but was not the cause in fact of the accident. Hence, plaintiff's negligence action properly failed.).

database updating. Each dot represents an update. The "cross" (**x**) symbols represent the plaintiff's proposed frequency, the untaken precaution.

Figure 2



In this illustration, failure to undertake the plaintiff's pleaded untaken precaution is not the actual cause of the harm. As illustrated, the new virus strain appeared after an update, infected the plaintiff's system, and caused harm before the next proposed update. The update prior to the virus's appearance would not have contained its signature, and the subsequent update was too late. The culprit virus therefore could not have been detected, even with plaintiff's proposed superior precaution, just as the unread manual, in the previous example, would not have prevented the plaintiff's harm. The pleaded untaken precaution therefore fails on actual cause grounds, even though failing to take it does constitute a breach of duty.

D. *Proximate Cause*

The plaintiff in a negligence action has to prove that the defendant's breach was not only the cause in fact but also the proximate, or legal, cause of the plaintiff's harm. The proximate cause requirement limits liability to cases where the defendant's conduct is "reasonably related" to the plaintiff's harm.[74] Proximate cause may be absent, for instance, if the accident was

74. Proximate cause limitations on liability are imposed where, as a matter of principle, policy, and practicality, the court believes liability is inappropriate. *See*, *e.g.*, the dissenting opinion of Judge Andrews, in Palsgraf v. Long Island R.R., 248 N.Y. 339, 352, 162 N.E. 99, 103 (1928): "What we do mean by the word 'proximate' is that, because of convenience, of public policy, of a rough sense of justice, the law arbitrarily declines to trace a series of events beyond a certain point. This is not logic. It is practical politics."

due to the unforeseeable and independent intervention of a second tortfeasor. Absent proximate cause, the first tortfeasor would escape liability even if his breach and actual causation have been clearly demonstrated.

A crisp formulation of the proximate cause requirement is that the realized harm must be within the scope of risk foreseeably created by the defendant, and the plaintiff must belong to the class of persons foreseeably put at risk by the defendant's conduct.[75]

Proximate cause applies to two broad categories of cases, namely those involving (i) multiple risks and (ii) concurrent efficient causes.[76] A multiple-risks case typically involves two risks, both of which would have been reduced by the defendant's untaken precaution. The first is the primary risk, which was clearly foreseeable to a reasonable person, and the second an ancillary risk, which would not have been reasonably foreseeable. Suppose, for instance, a surgeon performs a vasectomy negligently and a child is born. The child grows up and sets fire to a house. The owner of the house sues the doctor for negligence. This is clearly a multiple-risks case. The primary risk consists of foreseeable medical complications due to the incompetent vasectomy, including an unwanted pregnancy. The ancillary risk is the (unforeseeable) risk that the conceived child may grow up to be a criminal.[77] The proximate cause issue is whether the defendant should be held liable for the harm due to the ancillary risk.

A concurrent-efficient-causes case involves multiple causes, all of which are actual causes of the same harm.[78] In a typical concurrent-efficient-causes case, an original wrongdoer and a subsequent intervening party are both responsible for the plaintiff's harm. Suppose, for instance, a technician negligently fails to fasten the wheels of plaintiff's car properly. A wheel comes off, leaving the plaintiff stranded on a busy highway. The stranded plaintiff is subsequently struck by a passing driver who failed to pay attention. The technician and the inattentive driver were both negligent and are both concurrent efficient causes of the plaintiff's harm. The proximate cause issue is whether the second tortfeasor's act should cut off the liability of the first.

Proximate cause is a dualism consisting of two separate doctrines or tests. One doctrine applies to multiple-risks cases and the other to concurrent-efficient-causes cases. When both situations, multiple risks as well as concurrent efficient causes, are present in the same case, both proximate cause

75. Dobbs, *supra* note 3, at 444. *See also* Sinram v. Pennsylvania R.R. Co., 61 F.2d 767, 771 (2d Cir. 1932) (L. Hand, J.) ("[T]he usual test is . . . whether the damage could be foreseen by the actor when he acted; not indeed the precise train of events, but similar damage to the same class of persons.").

76. Grady, *supra* note 61, at 296 ("Proximate cause is a dualism.").

77. Based on a hypothetical in Dobbs, *supra* note 3, at 444.

78. Grady, *supra* note 61, at 299.

doctrines apply and the requirements for both have to be satisfied for proximate cause to exist.[79]

The reasonable foresight doctrine applies to cases of multiple risks, where a primary and ancillary risk both caused the plaintiff's harm. This doctrine establishes the conditions under which the tortfeasor who created the primary risk will be held liable for actual harm that has resulted from the ancillary risk. The bungled vasectomy is a typical reasonable foresight case. The reasonable foresight doctrine determines whether the surgeon would be held liable for damage caused by the ancillary risk, namely the risk that an unwanted pregnancy may produce a future criminal.

The direct consequences doctrine of proximate cause applies to cases involving multiple efficient causes. The doctrine examines concurrent causes to determine whether the person responsible for the second cause has cut off the liability of the person responsible for the first cause. The "loose wheel" case is a typical direct consequences case. The direct consequences doctrine would determine whether the intervening tortfeasor (the inattentive driver who struck the stranded plaintiff) would cut off the liability of the original tortfeasor (the negligent automobile technician). Some accidents involve purely multiple risks, while others involve purely concurrent causes. In some cases, however, both doctrines apply.

Application of the two proximate cause doctrines is greatly simplified and clarified when we divide the cases to which they apply into distinct paradigms. We now turn to an analysis of the paradigms within each doctrine.

1. Paradigms in Direct Consequences Doctrine

The direct consequences doctrine is divided into five paradigms, namely (i) no intervening tort, (ii) encourage free radicals, (iii) dependent compliance error, (iv) no corrective precaution, and (v) independent intervening tort.[80]

The no intervening tort paradigm is the default paradigm. It preserves proximate cause if no tort by anyone else has intervened between the original defendant's negligence and the plaintiff's harm, as long as the type of harm was foreseeable. In this paradigm, the original tortfeasor is not only the direct cause of the harm but also the only wrongdoer. A speeding and unobservant driver who strikes a pedestrian walking carefully in a crosswalk is a clear example of a case within the no intervening tort paradigm.

Under the encourage free radicals paradigm, proximate cause is preserved if the defendant's wrongdoing created a tempting opportunity for judgment-proof people. Proximate cause is preserved under the dependent

79. *Id.* at 298.
80. *Id.* at 301–21.

compliance error paradigm if the defendant's wrongdoing has increased the likelihood that the victim will be harmed by someone else's inadvertent negligence. Proximate cause is broken under the no corrective precaution paradigm if a third party with an opportunity and duty to prevent the plaintiff's harm intentionally fails to do so. Paradigm (v) cuts off the original tortfeasor's liability if an independent intervening tort caused the plaintiff's harm.

Encourage free radicals and dependent compliance error are the most interesting and relevant paradigms in a computer virus context. We now turn to a detailed analysis of these paradigms.

a. Encourage Free Radicals

Negligence law is the most basic form of safety regulation, but it is an ineffective deterrent against defendants who are shielded from liability by anonymity, insufficient assets, lack of mental capacity, or lack of good judgment. Such trouble-prone individuals are termed "free radicals" because of their tendency to bond with trouble. Examples of free radicals include children, anonymous crowds, criminals, mentally incompetent individuals, hackers, and computer virus authors.[81] The deterrence rationale of negligence law would be defeated if responsible people who foreseeably encourage free radicals to be negligent were allowed to escape judgment by shifting liability to the latter. Common law negligence rules therefore preserve the liability of the responsible individuals.[82]

Satcher v. James H. Drew Shows, Inc.[83] illustrates the free radicals paradigm. In *Satcher*, the plaintiff bought a ticket for a ride on the bumper cars in an amusement park. A group of mental patients on an excursion joined the plaintiff's group. When the ride started, the patients converged on the defendant and repeatedly crashed into her from all angles, injuring her neck permanently. The plaintiff filed suit, alleging that the defendant owner and operator of the ride had been negligent in allowing the patients to target and injure her. The appellate court reversed the trial court's decision for the defendant on the grounds that the defendant had encouraged free radicals.

Another free radicals case is presented by *Weirum v. RKO General, Inc.*[84] The defendant radio station broadcast a contest in which a disk jockey would drive throughout Los Angeles. He would stop occasionally and announce his location on the radio. Teenagers would race to meet the disk jockey and he would give a prize to the first one who reached him. Even-

81. *Id.* at 306–12.
82. *Id.* at 308.
83. 177 S.E.2d 846 (Ga. Ct. App. 1970).
84. 539 P.2d 36 (Cal. 1975).

tually, two racing teenagers were involved in a road accident, killing the plaintiff's deceased. There were two concurrent efficient causes of the accident, namely the organizers of the contest and the reckless teenage drivers. The radio station negligently encouraged the free radical teenagers to drive recklessly. The wrongdoing of the teenagers therefore did not cut off the defendant radio station's liability. The defendant radio station was held jointly liable with the teens and, as the deeper pocket, likely paid most of the damages.

(i) Limitations on Liability for Encouraging Free Radicals—The defendant will not be liable for encouraging free radicals unless she did so negligently.[85] This implies that the behavior of the free radicals must have been ex ante foreseeable, the actions of the free radicals must not have gone far beyond the encouragement, and the opportunity created for them must have been relatively scarce, to hold the defendant liable.

The defendant's act of encouragement would not amount to negligence unless the behavior of the free radicals was ex ante reasonably foreseeable. The defendant would not be liable for the actions of the free radicals if either they acted independently of the defendant's actions or their behavior went far beyond the defendant's encouragement. In *Weirum*, for instance, it must have appeared reasonably probable to the radio station that its contest would induce the kind of behavior that ultimately led to the accident, in order to hold the station liable. If one of the contestants had shot another in order to gain an advantage, the radio station would probably have escaped liability.[86]

If, besides the opportunity created by the defendant, several alternative opportunities were available to the free radical to cause the same or similar harm, the defendant's encouragement likely did not significantly increase the probability of the harm. The defendant therefore may escape liability if the opportunity created for the free radicals is not particularly scarce. A person flashing a wad of $100 bills would probably not be liable for the harm caused by a fleeing thief who runs into and injures someone. Because of the availability to the thief of many other similar opportunities, the flash of money did not increase the likelihood of the type of harm that occurred. If the person had not flashed the money, a determined thief would have found another opportunity.[87]

The person encouraged by the defendant may be a responsible citizen and not a free radical at all. In such a case, the defendant would escape

85. Grady, *supra* note 61, at 309 ("The pattern of EFR cases indicates that a defendant will not be liable for free radical depredations unless it negligently encouraged them.").

86. *Id* at 308.

87. *Id.* at 310 ("The defendant, in order to be liable, must negligently provide some special encouragement of wrongdoing that does not exist in the normal background of incitements and opportunities.").

liability. If Bill Gates had responded to the Weirum radio broadcast by racing to collect the prize, his intervening conduct would have almost certainly cut off the defendant's liability. Likewise, in the unlikely event that Bill Gates would use a virus kit to create a virus that exploits a weakness in Windows, the creator of the kit would escape liability. If, however, a free radical, such as a judgment-proof hacker, did the same, proximate causality would likely not be broken.

(ii) Encouragement of Virus Authors—Virus authors, as the originators of dangerous malevolent software, are, directly or indirectly, responsible for the harm caused by their creations. As such, they are always potential targets of lawsuits related to the harm. Virus authors often receive technical assistance, such as access to virus kits on the Internet that allow creation of custom viruses. Such virus tool kits, which enable people who have no knowledge of viruses to create their own, are commonly available on the Internet. Some of these kits are very user-friendly, with pull-down menus and online help available. Such a kit was used, for instance, to create the infamous Kournikova virus.[88]

Although a virus kit is useful to someone who lacks technical proficiency, it is not particularly helpful to a technically skilled person. A skilled and determined virus author would not wait for a kit to appear on the Internet, just as a determined thief would not wait for someone to flash a wad of $100 bills before acting. The creator of a virus kit may escape liability if a technically competent person downloaded and used the kit to create a virus. Even if the technically competent virus author were a judgment-proof free radical, the fact that the kit did not provide a means or encouragement beyond resources already available to the author cuts off liability of the original creator of the kit.

Virus authors also get assistance and inspiration from existing viruses that can be easily copied and modified. Once an original virus is created, altered versions are usually much easier to create than the original. Such altered versions may have capabilities that make them more pernicious than the original.[89] A virus named NewLove, for instance, was a more destruc-

88. *See, e.g.*, http://www.cknow.com/vtutor/vtpolymorphic.htm; Sarah Gordon, *Virus Writers: The End of Innocence*, IBM White Paper, http://www.research.ibm.com/antivirus/SciPapers/VB2000SG.htm (reporting the existence on the Internet of several sites with viruses in executable or source code form, available for download).

89. *See, e.g.*, Jay Lyman, *Authorities Investigate Romanian Virus Writer*, *at* http://www.linuxinsider.com/perl/story/31500.html ("The amazing side of this peculiar situation is that two people are to stand trial for having modified original code of MSBlast.A (the first blaster worm), but the creator of the worm is still out there . . . Antivirus specialists concur in saying that such altered versions are not as difficult to create as the original."). The possibility of variants of well-known viruses has caused concern. *Id.* ("A senior official at the [FBI] told TechNewsWorld that there is concern about variants and the implications of additional virus writers.").

tive variant of the LoveLetter virus. NewLove was polymorphic, which made its detection more difficult than LoveLetter's. It also overwrote files on the hard disk that were not in use at the time of infection. Due to a (fortunate) programming error, NewLove could not spread as widely as LoveLetter, but it was much more destructive in computers to which it did spread.[90]

Virus authors are also encouraged and helped by a variety of network security flaws that allow and facilitate the transmission of viruses. The Blaster worm, for instance, exploited a security flaw in Microsoft's Windows operating system to invade and crash computers.[91]

In practice, it is often easier to track down individuals who created opportunities for virus authors than the authors themselves. Virus kits are often posted on an identifiable Web page on the Internet and security flaws can be traced to the manufacturer, as in the case of the Microsoft Windows flaw. If virus authors are free radicals, individuals who create these opportunities for them would likely be the proximate cause of the harm. If they are not free radicals, their wrongdoing may be considered an independent intervening tort and, as such, will cut off liability of the encouragers.

(iii) Are Virus Authors Free Radicals?—Virus authors have properties commonly associated with free radicals. They are often judgment-proof and shielded by the anonymity of cyberspace. Virus authors are also increasingly turning to organized crime. Furthermore, virus attacks are underreported and underprosecuted, and the probability of catching a hacker or virus author is comparatively low. Virus authors appear undeterred by the threat of legal liability and often seem unconcerned about the problems caused by their creations. All these factors are consistent with a free radical profile.

The anonymity of the Internet is often exploited by cybercriminals. This complicates the task of detecting computer crimes and tracking down offenders. It also makes it harder to obtain evidence against a wrongdoer such as a virus author.[92] Cyberspace provides the technology and opportunity to a skilled operator to assume different identities, erase digital footprints, and transfer incriminating evidence electronically to innocent com-

90. K. Zetter, *When Love Came to Town: A Virus Investigation*, PC World, Apr. 18, 2004, *available at* http://www.pcworld.com/news/article/0,aid,33392,00.asp.

91. Danny Penman, *Microsoft Monoculture Allows Virus Spread*, Newscientist Online News, Sept. 25, 2003 ("[V]irus writers exploit human vulnerabilities as much as security flaws.").

92. Gordon, *supra* note 88 ("[T]racing a virus author is extremely difficult if the virus writer takes adequate precautions against a possible investigation."); Ian C. Ballon, *Alternative Corporate Responses to Internet Data Theft*, 471 PLI/Pat. 737, 739 (1997); M. Calkins, *They Shoot Trojan Horses, Don't They? An Economic Analysis of Anti-Hacking Regulatory Models*, 89 Geo. L.J. 171 (2000).

puters, often without leaving a trace.[93] Suppose, for instance, a virus were transmitted from the e-mail account of someone named Jill Smith and a copy of an identical virus were tracked down in the same account. This may look like a smoking gun but would likely not prove by a preponderance of the evidence that Jill is the actual culprit. Someone may have hacked into the Smith account, used it to launch a virus, and stored incriminating files in the account.[94]

In several cases, cyber rogues were apprehended because of their recklessness or vanity. In May 2000, a virus named LoveLetter was released into cyberspace. The virus first appeared in computers in Europe and Asia, hitting the European offices of Lucent Technologies, Credit Suisse, and the German subsidiary of Microsoft.[95]

When recipients clicked on the attachment in which it arrived, the virus sent copies of itself, via Microsoft Outlook, to everyone in the user's address book. It would then contact one of four Web pages hosted on Sky Internet, an Internet service provider (ISP) located in the Philippines, from which the virus downloaded a Trojan horse. The Trojan horse then collected valuable usernames and passwords stored on the user's system and sent them to a rogue e-mail address in the Philippines.[96]

Investigators tracked the origin of the LoveLetter virus by examining the log files of the ISP that hosted the Web pages from where the Trojan horse was auto-downloaded. Investigators were able to pierce the anonymity of cyberspace, in part because of clues revealed by the perpetrator, perhaps out of vanity, such as a signature in the virus code.[97]

93. *See*, *e.g.*, Ted Bridis, *Microsoft Offers Huge Cash Rewards for Catching Virus Writers*, *at* http://www.securityfocus.com/news/7371 ("Police around the world have been frustrated in their efforts to trace some of the most damaging attacks across the Internet. Hackers easily can erase their digital footprints, crisscross electronic borders and falsify trails to point at innocent computers.").

94. M.D. Rasch, *Criminal Law and the Internet*, *in* The Internet and Business: A Lawyer's Guide to the Emerging Legal Issues (Computer Law Ass'n). Online version is *available at* http://www.cla.org/RuhBook/chp11.htm. *See also* BizReport News, Sept. 12, 2003 ("There are many ways for virus writers to disguise themselves, including spreading the programs through unwittingly infected e-mail accounts. The anonymity of the Internet allows you to use any vulnerable machine to launder your identity.").

95. The virus was written in Visual Basic code, the most common language for virus code, characterized by a "dot.vbs" extension. Many users did not observe the dot.vbs extension because the Windows default setting hides file extensions.

96. Zetter, *supra* note 90.

97. Investigators traced the origin of the posting to a prepaid account at Supernet, another ISP in the Philippines. The LoveLetter virus was launched from two e-mail accounts, but the prepaid account would have allowed the virus author to remain anonymous if he had not provided additional incriminating evidence to investigators. The perpetrator was eventually tracked down, in part because, perhaps out of vanity, he left a signature in the virus code. The signature consisted of his name, e-mail address, membership in an identifiable small programmer's group, and hometown (Manila). The perpetrator also used his own home computer to launch the virus and dialed the ISP using his home telephone. This allowed the ISP to determine the telephone number from its call-in log files.

The anonymity of cyberspace has enabled virus authors to graduate from cyber-vandalism to organized crime. Virus writers are increasingly cooperating with spammers and hackers to create viruses to hack into computers to steal confidential information, often hiding their identity by spoofing the identity of the legitimate owner. Spammers are using viruses, for instance, to mass-distribute junk mail, by sending out viruses to take over computers and e-mail accounts and using them to mass-distribute spam messages.[98] The owner of the hijacked computer usually does not know it has been hijacked, although there are often subtle indications, such as slower Internet connection.[99]

To further enhance his anonymity, the spammer may use a remailer, i.e., a server that forwards electronic mail to network addresses on behalf of an original sender, who remains unknown. A remailer delivers the e-mail message without its original header, thus hiding the identity of the original sender from the recipient. This ensures almost total anonymity for the spammer.[100]

Virus authors appear to be undeterred by the threat of legal action. In a leading study on the subject, Dr. Sarah Gordon examined the correlation between the number of new viruses in the wild and high-profile prosecutions of virus authors as a measure of the deterrence value of prosecution. Dr. Gordon reports that high-profile prosecutions have had limited deterrent effect.[101]

98. The virus named "Sobig F," for instance, is programmed to turn a computer into a host that sends out spam e-mail messages, often without the knowledge of the owner. It is widely believed that half a million copies of the virus named AVF were sent by a spammer. Unlike Melissa, the AVF virus does not mail copies of itself out to everyone in the infected computer's address book. Instead, AVF makes the infected computer an intermediary by opening a backdoor in the infected machine through which spammers can distribute their junk mail.

99. *Spam Virus Hijacks Computers*, BBC News, *at* http://news.bbc.co.uk/1/hi/technology/3172967.stm; Jo Twist, *Why People Write Computer Viruses*, BBC News, *at* http://news.bbc.co.uk/1/hi/technology/3172967.stm.

100. *Spammers and Viruses Unite*, BBC News, *at* http://news.bbc.co.uk/1/hi/technology/2988209.stm (describing the hijacking program called Proxy-Guzu, which would typically arrive as a spam message with an attachment. Opening the attachment triggers it to forward information about the hijacked account to a Hotmail account. This information then enables a would-be spammer to route mail through the hijacked computer. The source of this spam would be very hard if not impossible to trace, especially if the spammer and the sender of the hijacking program employed anonymity-preserving techniques, such as a remailer.). *See also* Lyman, *supra* note 89 (referring to "the difficulty of tracking down virus writers, particularly when they are skilled enough to cover their digital tracks, [so that] few offenders are ever caught").

101. Gordon, *supra* note 88 (finding no evidence that such prosecutions have alleviated the virus problem, as measured by the rate of creation of new viruses in the wild subsequent to high-profile prosecutions). *See also* R. Lemos, *'Tis the Season for Computer Viruses* (1999), *at* http://www.zdnet.co.uk/news/1999/49/ns-12098.html. It is well known that even after the author of the Melissa virus had been apprehended (and expected to be sentenced to a multiyear prison term), the appearance of new viruses on the Internet continued to proliferate and at an increasing rate.

Dr. Gordon's conclusions were corroborated by another survey she undertook, in which virus authors and antivirus researchers were asked whether the arrest and prospective sentencing of the Melissa author would have any impact on the virus-writing community. All virus authors interviewed stated that there would be no impact, immediate or long-term, while the antivirus researchers were evenly split on the question. These results are consistent with those of comparable surveys by other researchers.[102]

For example, a subsequent survey suggests that new laws will result in more viruses than before. According to the survey results, a majority of virus authors would either be unaffected or actually encouraged by antivirus legislation. A number of them claimed that criminalization of virus writing would actually encourage them to create viruses, perhaps as a form of protest or civil disobedience.[103]

Laws against virus creation cannot be effective unless virus incidents are reported and perpetrators prosecuted. There is evidence that virus crimes are seriously underreported and, as a consequence, underprosecuted.[104] Commenting on the ineffectiveness of the law to combat computer viruses, Grable writes, "[b]oth the federal and New York state criminal statutes aimed at virus terror are ineffective because . . . [t]he combination of the lack of reporting plus the inherent difficulties in apprehending virus creators leads to the present situation: unseen and unpunished virus originators doing their damages unencumbered and unafraid."[105]

b. Dependent Compliance Error

The dependent compliance error paradigm applies where a defendant has exposed the plaintiff to the compliance error—relatively innocent, inadvertent negligence—of a third party. It preserves the liability of the original defendant when the compliance error results in injury to the plaintiff.

102. Gordon, *supra* note 88 (reference to a survey by A. Briney).

103. *Id.*(reference to DefCon survey).

104. *Id.* ("Minnesota statute §§ 609.87 to .89 presents an amendment which clearly defines a destructive computer program, and which designates a maximum (prison term of) ten years; however, no cases have been reported. Should we conclude there are no virus problems in Minnesota?"). *See also* Michael K. Block & Joseph G. Sidak, *The Cost of Antitrust Deterrence: Why Not Hang a Price-Fixer Now and Then?* 68 Geo. L.J. 1131, 1131–32 (1980); Stevan D. Mitchell & Elizabeth A. Banker, *Private Intrusion Response*, 11 Harv. J.L. & Tech. 699, 704 (1998).

105. Gordon, *supra* note 88 (quoting J. Grable, *Treating Smallpox with Leeches: Criminal Culpability of Virus Writers and Better Ways to Beat Them at Their Own Game*, 24 Computers & Law (Spring 1996)). *See also id.* ("[G]iven the small number of virus writers who have been arrested and tried . . . this lack of arrests is one of the primary indicators used by some to argue that laws are not a good deterrent."); *Virus Writers Difficult to Find in Cyberspace*, BizReport News (Sept. 2003) (reporting that it took eighteen days to track down the author of the Blaster worm, even though the author left a clear trail behind, including his alias stitched into the virus code, and references to a website registered in his name), *available at* http://www.bizreport.com/print.php?art_id = 4917.

In *Hairston v. Alexander Tank and Equipment Co.*,[106] a technician negligently failed to fasten the wheels of plaintiff's car properly. A wheel came off, leaving the plaintiff stranded on a busy highway. The stranded plaintiff was subsequently struck by a passing driver whose attention had inadvertently lapsed. Liability of the original tortfeasor, the auto technician, was preserved, because he had put the plaintiff in a situation where he was exposed to a high likelihood of harm due to the compliance error of the inattentive driver.

This principle is particularly applicable to computer security. Consider, for instance, a computer security breach where a flaw, such as a buffer overflow, allowed a virus to penetrate a network.[107] The security apparatus of the network fails to detect and eliminate the virus and it causes considerable harm to one or more computers in the network.

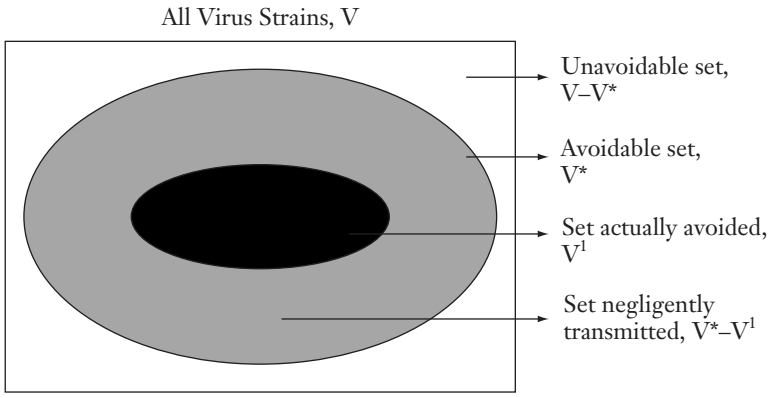
In situations such as this, the security lapse that allowed the virus into the system is foreseeable and likely due to a compliance error. The person responsible for the buffer overflow in the software, however, provided the opportunity, and thus exposed the users of the network to the security compliance error. Under the dependent compliance error paradigm, therefore, the liability of the person responsible for the buffer overflow will not be cut off, in spite of the intervention of the subsequent security lapse.

The schematic diagram, below, summarizes the arguments in this section. It applies to a typical computing environment, such as the computer network in the preceding (buffer overflow) example. The rectangle, V, represents the entire universe of virus strains. The virus universe consists of

106. 311 S.E.2d 559 (N.C. 1984).

107. A buffer is a contiguous piece of memory, usually dedicated to temporary storage of data. A buffer overflow occurs when a program tries to store more data in a buffer than it has the capacity for. The extra information overflows into adjacent buffers, overwriting or corrupting the legitimate data in the adjacent buffers. A buffer overflow has been described as "very much like pouring ten ounces of water in a glass designed to hold eight ounces. Obviously, when this happens, the water overflows the rim of the glass, spilling out somewhere and creating a mess. Here, the glass represents the buffer and the water represents application or user data." Mark E. Donaldson, *Inside the Buffer Overflow Attack: Mechanism, Method and Prevention*, SANS Institute 2002 White Paper, *available at* http://www.sans.org/rr/whitepapers/securecode/386.php. System Administration, Audit, Network and Security (SANS) was founded in 1989 as a cooperative research and education organization, specializing in computer security training and education. Buffer overflow is an increasingly common computer security attack on the integrity of data. The overflowing data, for instance, may contain code designed to trigger specific actions, such as modify data or disclose confidential information. Buffer overflows are often made possible because of poor programming practices. An attacker exploits a buffer overflow by placing executable code in a buffer's overflowing area. The attacker then overwrites the return address to point back to the buffer and execute the planted overflow code. A programming flaw in Microsoft Outlook, for instance, made it vulnerable to a buffer overflow attack. An attacker could invade a target computer and overflow a target area with extraneous data, simply by sending an appropriately coded e-mail message. This allowed the attacker to execute any code he desired on the recipient's computer, including viral code. Microsoft has since created a patch to eliminate the vulnerability.

avoidable viruses (virus strains that could be detected and eliminated at a cost less than its expected harm) and unavoidable viruses. In the diagram, the avoidable set is represented by the larger ellipse inside the rectangle, labeled V*, and the unavoidable set by the white area inside the rectangle but outside the ellipse, labeled V–V*.



The innermost, smaller, and darker ellipse, V^1, represents the possibility that an avoidable virus nevertheless may be transmitted into the computing environment. In the absence of negligence, no strain in V* will be transmitted. In the event of negligence of a party, such as a security flaw in a computer system or failure to use reasonable antivirus precautions, some strains in V* could enter the system, and only a subset of V* will be avoided. V^1 represents the subset that will be avoided, and the rest of V*, the grey area, denoted ($V^*–V^1$), represents the strains in V* that may enter the system due to the negligence. Virus strains in ($V^*–V^1$), as a subset of V*, should be detected if due care were taken. They will not be detected, however, because they are outside of V^1.

The remainder of this section argues that the set of negligently transmitted viruses, represented by ($V^*–V^1$), is large relative to the set of unavoidable viruses, represented by (V–V*). The outer boundary of ($V^*–V^1$) is defined by V*, and the inner boundary by V^1. The larger V* (the "further out" the outer boundary) and the smaller V^1 (the "further in" the inner boundary), the larger ($V^*–V^1$). We show in this subsection, that V* is large relative to V and V^1 is small relative to V*, resulting in a large ($V^*–V^1$). A virus attack therefore likely involves negligence.

Most cases of virus infection governed by the negligence rule involve a compliance error. A defendant who exposes a plaintiff to the negligence of a third party that results in a virus attack is therefore likely the proximate cause of the harm, under the dependent compliance error paradigm.

This explains why, in the previous buffer overflow example, courts would likely preserve the liability of an individual whose negligence was responsible for a buffer overflow in a computer system. The buffer overflow allowed a virus to enter the system and exposed users of the network to a compliance error by the network security administrator. The security person's compliance error, namely failure to detect and eliminate the virus, allowed the virus to remain in the system and wreak havoc.

This conclusion remains valid, by a preponderance of the evidence, even in cases where the culprit virus cannot be reliably identified as avoidable or unavoidable. The reason is that most viruses are avoidable and their presence likely attributable to a compliance error. The key factors that drive this theory are that V* is large and V^1 small.

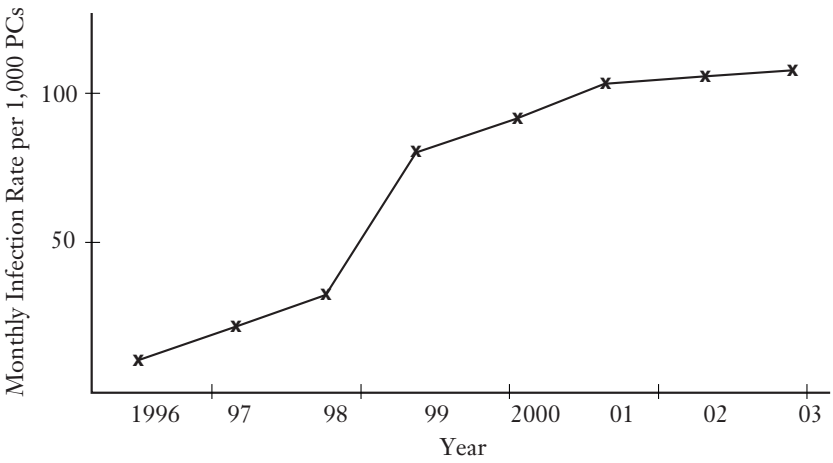
(i) V Is Large*—The Learned Hand formula, $B \geq P \times L$, dictates that, to avoid liability, investment in antivirus precautions (B) should at least equal the expected harm avoided ($P \times L$). In this subsection, we argue that V* is large for the following reasons. $P \times L$ is relatively large, so that the legally mandated precaution level, B, must be large. The efficiency and economy of antivirus technology indicate that a substantial investment in precautions will result in a large avoidable set, V*.

(ii) P × L *Is Large*—Expected harm from infection, $P \times L$, is large, because the probability of virus infection, P, and the harm associated with virus infection, L, are both large. P is large because of the substantial and increasing prevalence of computer viruses on the Internet and in computer networks. L is large because of the unique nature and unusual destructive potential of viruses, both in an absolute sense, as well as compared to general computer security hazards.

(iii) P *Is Large*—Virus prevalence is substantial and increasing.[108] According to the influential 2003 ICSA survey, 88 percent of respondents perceived worsening of the virus problem.[109] Virus prevalence statistics in the survey support the pessimistic response. The following graph, constructed from data in the ICSA Survey, illustrates the trend of an increasing virus infection rate.

108. *See, e.g.*, ICSA Labs 9th Annual Computer Virus Prevalence Survey 2003, *supra* note 10, at 23 ("There is little doubt that the global virus problem is worsening. After a somewhat quiet year in 2002, 2003 arrived with a vengeance. Beginning with the Slammer worm in January, to Mimail and its many variants in December, we have seen one of the most eventful years ever for computer viruses. For the 8th year in a row, virus infections, virus disasters and recovery costs increased.").

109. Qualified respondents to the survey work for companies and government agencies with more than 500 PCs, two or more local area networks (LANs), and at least two remote connections.



The high and increasing infection rate, which is a direct proxy for the probability that any particular network will be hit by a virus attack during a given time interval, suggests a high value for *P* in the Learned Hand formula.

(iv) L *Is Large*—The expected harm associated with virus infection is significant, both in an absolute sense, as well as relative to general computer security hazards and hardware and software errors. The greater inherent danger of viruses is due to the generality, scope of harm, persistence, growing payload severity, and advances in the spreading mechanism of the virus threat.[110]

A typical traditional computer security breach is usually related to a particular identifiable weakness, such as a security flaw that allows unauthorized access to a hacker. Viral infection is a more general and more volatile security threat, which makes it harder to plan a comprehensive preventive strategy. It can enter the system or network in multiple ways, and any and every program or data file is a potential target. It can be programmed to carry virtually any conceivable resource-dissipating or destructive function, and to attach it to any part of a system or network.[111]

110. *See generally* Cohen, *supra* note 8, at 24–27; Inst. for Computer Sec. & Admin., ICSA Labs 6th Annual Computer Virus Prevalence Survey 2000. For a detailed analysis and discussion of the nature and origins of the unusual danger level associated with virus infection, see Meiring de Villiers, *Virus ex Machina Res Ipsa Loquitur*, 1 Stanford Tech. L. Rev., Section V.C (2003).

111. Cohen, *supra* note 8, at 24 ("The virus spreads without violating any typical protection policy, while it carries any desired attack code to the point of attack. You can think of it as a missile, a general purpose delivery system that can have any warhead you want to put on it. So a virus is a very general means for spreading an attack throughout an entire computer system or network.").

The chameleonlike evolution of virus technology poses unique challenges to virus detection and elimination efforts. The shape and form of viral attacks evolve continuously, as evidenced by the appearance of a progression of stealth, polymorphic, macro, and e-mail viruses. Advances in computer technology continuously open up new opportunities for virus writers to exploit. Malevolent software exploiting e-mail technology is a prime example. Conventional wisdom once reassured that it was impossible to become infected by a virus simply by reading an e-mail message. This wisdom was promptly shattered by advances in virus technology designed to exploit the unique characteristics, as well as obscure weaknesses and little-known flaws in new technologies. A family of viruses that exploited a weakness in the JavaScript technology, for instance, was programmed to infect e-mail attachments and, when the e-mail message was read, automatically compromise the computer system, without even having the user actually open the attachment.[112]

The release of a computer virus has been likened to opening a bag of feathers on a tall building on a windy day. The Scores virus, for instance, was created to target a large company, EDS, but ended up attacking several U.S. government agencies, including NASA and the Environmental Protection Agency.[113]

The scope of potential harm caused by computer viruses is unprecedented. In a typical conventional security breach, a hacker may access an account, obtain confidential data, and perhaps corrupt or destroy it. The damage could, of course, be substantial, but it is nevertheless limited to the value of the data and contained within the system or network hacked into. If, instead, a hacker accessed an account by releasing a virus into the system, the virus may spread across computers and networks, even to those not physically connected to the originally infected system.[114] Whereas the

112. Roger A. Grimes, Malicious Mobile Code 394 (2001). JavaScript is a language developed by Netscape in collaboration with Sun Microsystems to increase interactivity and control on Internet Web pages, including the capability to manipulate browser windows. The JavaScript e-mail worm, JS.KAK, which appeared at the end of 1999, exploited an obscure Internet Explorer security flaw to disrupt computer systems and destroy data. It infects e-mail attachments and, when the e-mail message is opened, automatically compromises the computer system, without having the user open the attachment. A related, but less-well-known and shorter-lived e-mail virus, the so-called BubbleBoy, exploited a security hole in the Auto-Preview feature in Microsoft Outlook to send a copy of itself to every listing on the user's address list. BubbleBoy was one of the first attachment-resident viruses that did not require the user to open the attachment in order to do its harm.

113. A. Bissett & G. Shipton, *Some Human Dimensions of Computer Virus Creation and Infection*, 52 Int. J. Human-Computer Studies 899, 903 (2000); E.L. Leiss, Software Under Siege (1990).

114. *See, e.g.*, Robin A. Brooke, *Deterring the Spread of Viruses Online: Can Tort Law Tighten the "Net"?* 17 Rev. Litig. 343, 361 ("The market now provides enough statistics indicating both high risk and potentially widespread damage from virus attacks, while either programming prevention or off-the-shelf capabilities to detect viruses may impose a proportionally

conventional hacker can destroy data worth, say, an amount D, releasing a virus to do the same job can cause this harm several times over by spreading into N systems, causing damage of magnitude $N \times D$, where N can be very large. Although the two types of security breaches do similar damage in a particular computer, the virus's greater inherent danger is that it can multiply and repeat the destruction several times over.[115]

Dr. Fred Cohen provides a dramatic illustration: "Sitting at my Unix-based computer in Hudson, Ohio, I could launch a virus and reasonably expect it to spread through 40% of the Unix-based computers in the world in a matter of days. That's dramatically different from what we were dealing with before viruses."[116]

A worm, the so-called Morris Worm, designed and released by a Cornell University student, effectively shut down the Internet and other networks connected to it.[117] It was not designed to damage any data, but conservative estimates of the loss in computer resources and availability range between $10 million and $25 million.[118]

Dr. Cohen's statement was published more than a decade ago. Today, viruses spread much faster, and there is every indication that virus transmission will continue to accelerate. The 2003 ICSA report remarks, for instance, that whereas it took the early file viruses months to years to spread widely, subsequent macro viruses took weeks to months, mass mailers took

smaller burden."); *id.* at 348 ("Widespread proliferation of a virus originally undetectable becomes compounded very quickly. Independent actors along the transmission chain can be unaware of malevolent software residing in their computer, network, files, or disks, even if they use virus protection software, because the software may not sufficiently detect more sophisticated code."). *See also* ALLAN LUNDELL, VIRUS! vii (1989) ("Most mainframe computers can be successfully subverted within an hour. Huge international networks with thousands of computers can be opened up to an illicit intruder within days." (quoting Dr. Fred Cohen)); HRUSKA, *supra* note 14, at 13 ("[N]ew viruses are highly destructive, programmed to format hard disks, destroy and corrupt data. As viral infections become more and more widespread, the danger of damage to data is increasing at an alarming pace); *id.* at 14 ("The virus danger is here to stay. In the USA, the Far East and Africa it has already reached epidemic proportions . . . In just three months in the Spring of 1989, the number of separately identifiable viruses increased from seven to seventeen.").

115. DUNHAM, *supra* note 1, at xx ("Just one virus infection can erase the contents of a drive, corrupt important files, or shut down a network.").

116. COHEN, *supra* note 8, at 25. *See also* GRINGRAS, *supra* note 4, at 58 ("A computer harboring a virus can, in a matter of hours, spread across continents, damaging data and programs without reprieve."). *See also* Bradley S. Davis, *It's Virus Season Again, Has Your Computer Been Vaccinated? A Survey of Computer Crime Legislation as a Response to Malevolent Software*, 72 WASH. U. L.Q. 379, 437 and accompanying text ("[A] user whose computer was infected could connect to an international network such as the Internet and upload a file onto the network that contained a strain of malevolent software. If the software was not detected by a scanning system . . . on the host computer, infection could spread throughout the Internet through this simple exchange of data."); *How Fast a Virus Can Spread*, *supra* note 1, at 21.

117. For an account of the "Internet Worm Incident," *see*, *e.g.*, ROGUE PROGRAMS, *supra* note 11, at 203.

118. FITES ET AL., *supra* note 1, at 51–52.

days, Code Red took approximately twelve hours, and Klez spread around the world in two and one-half hours.[119]

A third major distinction that makes viruses more dangerous than general security hazards is their persistence. A virus can never really be entirely eliminated from a system. Generally, when a programming error or security flaw is rectified, the specific problem can be considered eliminated from the system. In the case of viruses, however, one can never be sure that a particular virus is gone for good. An infected program may be deleted and restored from a backup, but the backup may have been made after the backed-up program was infected and, hence, contain a copy of the virus. Restoring the program will then also restore the virus. This may happen, for instance, in the case of a virus that lies dormant for a while. During its dormancy, periodic backups also will back up the virus. When the virus becomes active, deleting the infected program and restoring it from the backup will only repeat the cycle.[120] Even if the backup is not contaminated, any user of the system with an infected floppy disk or contaminated e-mail could reintroduce the virus into the disinfected system.[121]

Many virus strains tend to survive progressively new generations of software. Replacing an old, infected spreadsheet program with a new and clean version will temporarily eliminate the virus, but the new version will not be immune to the particular virus. If the virus makes its way back, perhaps via an e-mail attachment, it will eventually reinfect the new program.[122]

119. ICSA Labs 9th Annual Computer Virus Prevalence Survey 2003, *supra* note 10, at 25.

120. Shane Coursen, *How Much Is That Virus in the Window*, Virus Bull. 15 (1996) (describing a common virus named Ripper that slowly modifies data while the data are being archived, resulting in corrupted backups); Dunham, *supra* note 1, at 129–30.

121. Brooke, *supra* note 114, at 362 n.95 ("It is likely impossible to eradicate viruses completely. Simply disinfecting a computer system could cost a staggering amount. In 1990, computer infection in the United States alone was estimated to be one percent, or about 500,000 computers . . . Unfortunately, even having a virus removed provides no guarantee of safety from further virus harm. In the United States, 90 percent of all infected users experience re-infection within 30 days of having the original virus removed."); Coursen, *supra* note 120, at 13 ("[T]he fix must be implemented in such a way that it is all-encompassing and simultaneous across infected sites. Tending to one site and neglecting another will surely allow a persistent virus to work its way back again."); *id.* at 16 ("Cleaning your program of a virus does not guarantee that it will not come by for another visit. Just one leftover diskette or program can have a snowball effect and start another virus outbreak. Within a matter of hours, the entire business could be under siege again. Any time spent cleaning up from the initial infection or outbreak can easily be lost in those few hours. The complete virus recovery process would have to be repeated.").

122. *See, e.g.*, Cohen, *supra* note 8, at 27 ("Eventually you probably change every piece of software in your computer system, but the virus may still persist. When you go from DOS 2.01 to DOS 2.3, to 3.0, to 3.1 to 3.2 to 4.0 to 4.1 to 5.0 to 6.0 to OS/2, the same viruses that worked on DOS 2.01 almost certainly work on each of these updated operating systems. In fact, if you wrote a computer virus for the IBM 360 in 1965, chance[s] are it would run on every IBM-compatible mainframe computer today, because these computers are upwardly compatible."). Some viruses do become extinct over time, however. *See, e.g.*, Dunham, *supra*

Converting infected documents to a later version often also automatically converts the virus to one compatible with the new format.[123]

The latest Internet worms and mass mail viruses have more staying power—they remain virulent longer and spawn more variants. When infections do occur, it takes longer and costs more to disinfect systems and recover from virus attacks.[124]

The 2003 ICSA Survey reports an increase not only in prevalence of virus attacks but also in the severity of disasters. The survey defines a "virus disaster" as "25 or more PCs infected at the same time with the same virus, or a virus incident causing significant damage or monetary loss to an organization."[125] In the 2002 ICSA survey, eighty respondents reported a disaster, while the 2003 survey reported ninety-two disasters. Average disaster recovery time increased slightly in 2003 over 2002. Recovery costs, however, increased significantly, by 23 percent, from a 2002 average of $81,000 to $100,000 in 2003.[126] The survey also reports a growth in severity of virus payloads and consequences of infection, as well as changes in attack vectors (modes of distribution), the latter exacerbating the volatility and unpredictability of the virus threat.[127]

The high danger rate associated with computer viruses makes them a potentially potent and destructive tool for a perpetrator of terrorism, industrial espionage, and white-collar crime.[128] U.S. security agencies are reportedly investigating the use of malicious software seriously as a strategic weapon,[129] and the Pentagon established a SWAT team, administered by the Computer Emergency Response Team Coordination Center, to combat destructive programs, such as the Morris Worm.[130]

note 1, at xxi ("[M]any older Macintosh viruses do not function correctly on System 7.0 or later. On PCs, many DOS file-infecting viruses are no longer as functional or successful in the Windows operating system. Still, older viruses continue to work on older operating systems and remain a threat for users of older systems.").

123. Bissett & Shipton, *supra* note 113, at 899, 902.

124. ICSA Labs 9th Annual Computer Virus Prevalence Survey 2003, *supra* note 10, at 24.

125. *Id.* at 1.

126. "For the eighth year in a row, our survey respondents report that viruses are not only more prevalent in their organizations, but are also more destructive, caused more real damage to data and systems, and are more costly than in past years. This despite increases in their use of antivirus products, improved updating and upgrading, better management of antivirus systems. Corporations are also spending more time, energy, and dollars in purchasing, installing, and maintaining antivirus products without achieving their desired results." *Id.*

127. *Id.* at 6.

128. Fites et al., *supra* note 1, at 50–53 (describing the use of viruses to perpetrate acts of sabotage, terrorism, and industrial espionage); Cohen, *supra* note 8, at 151–52; Clifford Stoll, *Stalking the Wily Hacker*, 31 Comms. ACM 484 (1988).

129. Jay Peterzell, *Spying and Sabotage by Computer*, Time, Mar. 20, 1989, at 25 (cited in Rogue Programs, *supra* note 11, at 92 n.134).

130. Rogue Programs, *supra* note 11, at 92 n.133.

In summary, the expected harm from a virus attack, $P \times L$, is relatively large. Applying the Learned Hand formula, it follows that the legally mandated precaution level, *B*, must be large. We now argue that a large *B* implies a large avoidable set, V*. The essence of the argument is that a large avoidable set, V*, is (i) technologically feasible and (ii) legally mandated.

(v) A Large V Is Technologically Feasible*—Antivirus software became available soon after the first appearance of computer viruses and has become increasingly sophisticated and effective, in response to parallel advances in virus technology. Although it is impossible to identify the presence of a virus with 100 percent reliability,[131] state-of-the-art technology has achieved close to a perfect detection rate of known viruses, and a detection rate of unknown virus strains perhaps as high as 80 percent and growing. State-of-the-art heuristic virus scanners, for instance, are capable of detecting at least 70 to 80 percent of unknown viruses.[132]

Organizations such as Virus Bulletin, West Coast Labs, and others periodically publish evaluations of commercial antivirus products. Virus Bulletin,[133] an industry leader, uses a recently updated database of virus strains to test antivirus software for its so-called 100 Percent Award. Products receive this award if they successfully detect all the strains in the database, suggesting that they are capable of detecting virtually all known strains. Antivirus software that have consistently made this grade include products such as Norton AntiVirus, Sophos Anti-Virus, and VirusScan.[134]

West Coast Labs[135] evaluates antivirus software for their ability to detect as well as eliminate viruses. Products such as Norton AntiVirus, VirusScan, and F-Secure, among others, have recently been certified for their ability to detect and eliminate 100 percent of known virus strains.[136] Other organizations, such as the Virus Test Center at the University of Hamburg, regularly test antivirus software and publish their results, including a list of software with a 100 percent detection rate.[137]

Some of the most effective antivirus programs are available free of charge, at least for private users. Free software includes products such as VirusScan, which made Virus Bulletin's 100 Percent Award list and received similar honors from West Coast Labs. Norton AntiVirus, an antivirus product that has been similarly honored and that offers additional

131. Spinellis, *supra* note 31, at 280, 282 (stating that theoretically perfect detection is in the general case undecidable, and for known viruses, NP-complete).

132. Nachenberg, *supra* note 1, at 7; Fernandez, *supra* note 31; Alex Shipp, *Heuristic Detection of Viruses Within e-Mail*, *in* Proceedings 11th Annual Virus Bulletin Conference, Sept. 2001.

133. *See* http://www.virusbtn.com.

134. Dunham, *supra* note 1, at 150–51 (Table 6.3).

135. *See* http://www.check-mark.com.

136. Dunham, *supra* note 1, at 154 (Table 6.6).

137. *See* http://agn-www.informatik.uni-hamburg.de/vtc/naveng.htm.

features such as a user-friendly interface, powerful scan scheduling options, heuristic technology for the detection of unknown strains, and SafeZone quarantine protection, is available at modest cost at the time of writing.[138]

A high detection rate is not limited to known virus strains. State-of-the-art heuristic scanners, such as Symantec's Bloodhound technology and IBM's AntiVirus boot scanner, are capable of detecting 70 to 80 percent of unknown viruses.[139] Heuristic technology is relatively inexpensive. Symantec's Bloodhound technology, for instance, is incorporated in the Norton AntiVirus product.[140]

The technological trend is towards greater sophistication and effectiveness and an increasing detection rate. IBM, for instance, a major center of virus research, has been awarded a patent for an innovative automatic virus detection system based on neural network technology.[141] The system uses artificial intelligence techniques that mimic the functioning of the human brain to enable it to identify previously unknown virus strains. The neural network is shown examples of infected and uninfected code (e.g., viral and uninfected boot sector samples) and learns to detect suspicious code. Care was taken to minimize the occurrence of false alarms. The system reportedly captured 75 percent of new boot sector viruses that had come out since its release, as well as two reports of false positives. Subsequent updates of the product were designed to eliminate false positives of the kind that occurred.

Ambitious research programs are under way that augur well for an even greater detection rate. The inventors of the IBM neural network technology view it as a precursor to an immune system for cyberspace that operates analogously to the human immune system. This envisioned cyber immune system will operate through the Internet to "inoculate" users globally to a virus within minutes of its initial detection.[142]

(vi) A Large V Is Legally Mandated*—Sophisticated antivirus technology makes a large V* feasible.[143] V* is a legal concept, though, and encompasses more than technological feasibility. V* is, by definition, the set of virus

138. At the time of writing (2004), the latest version of Symantec's Norton AntiVirus was available for less than $200. *See also* Dunham, *supra* note 1, at 158–59.

139. See discussion of heuristic detection technologies in Section II.B.4, *supra.*

140. *See also* http://www.symantec.com/nav/nav_mac/; Dunham, *supra* note 1, at 158–59.

141. Gerald Tesauro et al., *Neural Networks for Computer Virus Recognition*, 11:4 IEEE Expert 5–6 (Aug. 1996). *See also* Press Release, IBM, IBM Awarded Patent for Neural Network Technology, *available at* http://www.av.ibm.com/BreakingNews/Newsroom/97-10-27/.

142. J.O. Kephart et al., *Computers and Epidemiology*, 30:5 IEEE Spectrum 20–173 (May 1993).

143. A scanner with a properly updated signature database can detect close to 100 percent of known virus strains. Heuristic scanners, such as Symantec's Bloodhound technology, can detect 70 to 80 percent of unknown viruses. IBM's neural network virus detection technology can capture 75 percent of new boot sector viruses. Innovative research promises that the trend toward "perfect" virus detection and elimination will continue and perhaps accelerate.

strains whose elimination is both technologically feasible as well as cost-effective. This subsection draws on the economics of virus precaution to show that a large V* is not only technologically feasible but also cost-effective, hence within the scope of due care.

The Learned Hand formula, $B \geq P \times L$, dictates that, to avoid liability, investment in antivirus precautions, B, should at least equal the expected harm avoided, $P \times L$. We have argued that the high danger level associated with virus attacks (L), as well as a significant and increasing probability of a virus attack (P), mandates a high investment in antivirus technology. We now explore estimates of the numerical value of $P \times L$ (and thus of B) and obtain a quantitative estimate of the proportion of all virus strains avoidable by the Learned Hand efficient level of precaution. This proportion is a direct estimate of the relative size of V*.

The ICSA survey reports that 92 of 300 respondents experienced at least one incidence of a virus disaster over the one-year survey period, with an average recovery cost of $100,000.[144] The survey states that the recovery cost figure likely underestimates the true cost by a factor of seven or eight, when considering direct as well as indirect costs.[145] An adjusted recovery cost figure per disaster, therefore, in reality, may be closer to $700,000 to $800,000. In addition to disasters, the survey data also show an average of 108 "ordinary" virus infections per month, per site.

If we take the recovery costs of a disaster to be $750,000 and 92/300 as the probability that a particular site will experience a disaster in a given year, then the ex ante expected annual monetary loss from a disaster is $230,000. This is a conservative estimate. It assumes, for instance, that each of the respondents who reported experiencing at least one disaster during the survey year did experience only one disaster. It also does not include the cost associated with ordinary infections (not disasters), which are much more numerous than disasters and also capable of significant damage.

A conservative estimate of the annual expected harm to an institution from virus attacks amounting to a disaster is $230,000. This corresponds to the term $P \times L$ in the Learned Hand formula and has to be balanced by the same amount of precaution, B, to avoid liability. How much protection does $230,000 buy? A recent competitive analysis of leading antivirus vendors shows that Symantec's premium antivirus product, the Symantec AntiVirus Enterprise edition, is available at a fee of approximately $700,000 for a four-year/5,000-seat license with premium support. A similar product, Sophos Corporate Connect Plus, is available for $156,250,

144. The survey defines a "virus disaster" as "25 or more PCs infected at the same time with the same virus, or a virus incident causing significant damage or monetary loss to an organization." ICSA Labs 9th Annual Computer Virus Prevalence Survey 2003, *supra* note 10, at 1.

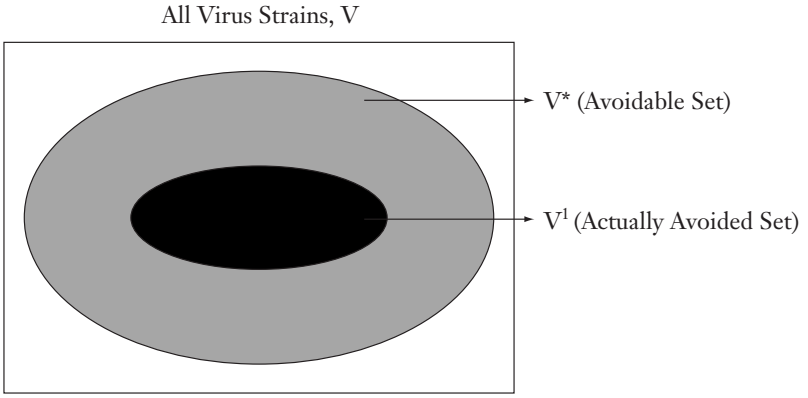
145. *Id.* at 2.

under similar terms.[146] Both Symantec and Sophos are recipients of Virus Bulletin's 100 Percent Award. Products receive this award if they successfully detect all the strains in a database compiled by Virus Bulletin, suggesting that they are capable of detecting virtually all known strains.[147] These products also contain heuristic algorithms that enable them to detect more than 80 percent of unknown virus strains.

Assuming, conservatively, that the Sophos and Symantec products are capable of preventing 80 percent of disasters,[148] then an investment of between $39,000 (Sophos) and $175,000 (Symantec) in antivirus precautions will prevent expected damage amounting to $0.8 \times \$230{,}000 = \$184{,}000$. Both antivirus products are cost-effective, and therefore within the scope of due care.

The detection of most viruses is not only technologically feasible but also cost-effective. Most virus strains belong to V*. In fact, at least 80 percent, perhaps in excess of 90 percent, of all strains, known as well as unknown, belong to V*. Having established that V* is large, we now argue that V^1 is small.

(vii) V^1 Is Small—The diagram, below, represents the avoidable and unavoidable virus strains associated with a typical computing environment. V* represents the avoidable set, as previously defined, and V^1 represents the set of viruses that will actually be prevented.



V^1 is smaller than V*, because a rational, profit-maximizing defendant, such as a software provider, has an economic incentive to fall short of the

146. *Total Cost of Ownership: A Comparison of Anti-Virus Software*, SOPHOS White Paper, *available at* http://www.sophos.com/link/reportcio.

147. Dunham, *supra* note 1, at 150–51 (Table 6.3).

148. The 80 percent figure is a conservative estimate. The technology we discuss is capable of detecting and eliminating *at least 80 percent* of *unknown* viruses and virtually 100 percent of known ones.

legal standard of due care, resulting in the transmission of some virus strains in V*. The grey area, between V* and V^1, represents the viruses that should be prevented, because they belong to V*, but will not, because of the precautionary lapse. The precautionary lapse is likely due to an inadvertent compliance error.

c. Compliance Error

In order to understand the nature and origin of a compliance error, we distinguish between durable and nondurable precautions against harm. A durable precaution typically has a long service life, once it is installed. Use of a durable precaution must usually be complemented by shorter-lived, nondurable precautions, which have to be repeated more frequently than durable precautions. A medical example illustrates the distinction between durable and nondurable precautions. A kidney dialysis machine is a typical durable precaution. A dialysis machine has a long service life once it is installed, but it cannot function properly without complementary nondurable precautions, such as regular monitoring of the hemodialytic solution.[149]

Antivirus precautions consist of a durable as well as nondurable component. Durable precautions, such as a virus scanner and signature database, must be complemented by nondurable precautions, such as regularly updating and maintaining the signature database and monitoring the output of the scanner.[150] A "compliance error" is defined as a deviation from perfect compliance with the (Learned Hand) nondurable precaution rate.[151]

A compliance error is efficient, even though the courts equate it to negligence. A rational, profit-maximizing entity such as a commercial software provider will systematically fail to comply with the legally required nondurable antivirus precaution rate.

(i) Compliance Error Is Rational—Results in the law and economics literature predict that there will be no negligent behavior under a negligence rule of liability, in the absence of errors about legal standards, when precaution is not random and when private parties have identical precaution costs.[152] It seems, therefore, that the frequent occurrence of negligence in society must be explained in terms of nonuniform precaution costs, or errors by courts and private parties about the relevant legal standards, or that precaution has a random or stochastic component.

149. Mark F. Grady, *Why Are People Negligent? Technology, Nondurable Precautions, and the Medical Malpractice Explosion*, 82 Nw. U. L. Rev. 293, 299 (1988).

150. A scanner reads software code and searches for known virus patterns that match any of the viral patterns in its database. *See* Section II.B, *supra*, for a review of virus detection technologies.

151. Mark F. Grady, *Res Ipsa Loquitur and Compliance Error*, 142 U. Penn. L. Rev. 887.

152. *Id.* at 889–91.

Dean Mark Grady has argued that none of these theories explains the prevalence of negligence entirely satisfactorily. Grady has proposed a theory according to which there is a pocket of strict liability within the negligence rule. According to the theory, a rational injurer may find an occasional precautionary lapse economically efficient and thus preferable to perfectly consistent compliance with the legal standard of due care. The frequency of such lapses will increase as the due care standard becomes more burdensome. The occasional lapse is rational and profit maximizing, as we argue below, but will nevertheless be classified as negligence by the courts, because of the courts' inability to distinguish between efficient and inefficient lapses.

The level of investment in durable and nondurable antivirus precautions required by negligence law is determined according to the Learned Hand formula.[153] Scanners, for instance, come in a variety of degrees of sophistication (and cost), ranging from basic systems that detect only known strains, to heuristic artificial intelligence–based systems capable of detecting polymorphic viruses and even unknown strains. The optimal Learned Hand level of investment in scanning technology would be determined by balancing the cost of acquiring and operating the technology against the expected harm avoided. The optimal nondurable precaution level, such as frequency of viral database updating, is determined similarly.

The courts require perfectly consistent compliance with the Learned Hand precautions to avoid a finding of negligence. If, for instance, the courts require a viral signature database to be updated twice daily, then even one deviation, such as one skipped update over, say, a two-year period, would be considered negligent.[154] When the courts apply the Learned Hand formula to determine an efficient precaution level and rate, the calculation weighs the costs and benefits of the precaution *each time* it is performed but ignores the cost of consistently performing it *over time*. Consider a numerical example. Suppose the cost of a daily update is $10, and the marginal benefit of the update is $11. Failure to perform even one such update would be viewed as negligence by the courts. Over, say, 300 days,

153. *See* Section II.B, *supra*, on breach of duty.

154. In *Kehoe v. Central Park Amusement Co.*, 52 F.2d 916 (3d Cir. 1931), an amusement park employee had to apply a brake to control the speed of the car each time the rollercoaster came around. When he failed to do so once, the car left the track. The court held that the compliance error by itself constituted negligence, i.e., the court required perfect compliance and considered anything less as negligence. *Id.* at 917 ("If the brake was not applied to check the speed as the car approached . . . it was clear negligence itself."). For other cases, see *Grady*, *supra* note 151, at 901. In *Mackey v. Allen*, 396 S.W.2d 55 (Ky. 1965), plaintiff opened a "wrong" exterior door of a building and fell into a dark storage basement. The court held the owner of the building liable for failing to lock the door. *But see* Myers v. Beem, 712 P.2d 1092 (Colo. Ct. App. 1985) (an action brought against an attorney for legal malpractice, holding that lawyers are not required to be infallible).

the courts expect 300 updates, because each of those updates, *by itself*, is Learned Hand efficient. However, the courts do not consider the cost of consistency, i.e., of *never* forgetting or lapsing inadvertently. Human nature is such that over a 300-day period, the person in charge of updating will occasionally inadvertently fail to implement an update.

Human nature, being what it is, dictates that perfection is (perhaps infinitely) expensive.[155] Perfect consistency, i.e., ensuring that 300 updates will actually be achieved over 300 days, would require additional measures, such as installing a monitoring device alerting the operator to a lapse, or perhaps additional human supervision, all of which are costly. Even assuming (heroically) that such measures would assure consistency, their cost may nevertheless be prohibitive to a rational software provider. Suppose, for instance, that such a measure would add an additional $2 to the cost of an update. The marginal cost of an update ($12) is now more than the marginal benefit ($11). Hence, perfect consistency is not in society's interest.

An occasional lapse is also reasonable from the viewpoint of the software provider: The marginal cost of perfect consistency is greater than the marginal increase in liability exposure due to efficient negligence. The courts nonetheless would consider such an efficient lapse to be negligence. Courts act as if they ignore the additional cost of $2 to achieve perfect consistency. Efficient lapses can be expected to become more likely and more frequent, the more demanding and difficult the Learned Hand nondurable precaution rate, i.e., the more expensive perfect consistency becomes.

A major reason for the courts' insistence on perfect compliance, in spite of the inefficiency of such perfection, is that it is impossible or expensive to determine whether any given deviation from perfect compliance is efficient. Who can judge, for instance, whether a software provider or website operator's mistake or momentary inattentiveness was an economic or uneconomic lapse? Courts, therefore, do not acknowledge efficient noncompliance where it is difficult to distinguish between efficient and inefficient noncompliance.[156]

155. *See, e.g.*, Peterson, *supra* note 40, at 194 ("Even under the best of circumstances, our brains don't function perfectly. We do forget. We can be fooled. We make mistakes. Although complete failures rarely occur, neural systems often suffer local faults.").

156. The policy rationale behind the courts' insistence on perfect compliance was expressed by Lord Denning in Froom v. Butcher, 3 All E.R. 520, 527 (C.A. 1975) ("The case for wearing seat belts is so strong that I do not think the law can admit forgetfulness as an excuse. If it were, everyone would say: 'Oh, I forgot.'"). Instead of incurring the considerable measurement cost to distinguish between efficient and inefficient failures to comply, courts simply equate any and all noncompliance to negligence. *See also* Grady, *supra* note 151, at 906; W. Landes & R. Posner, The Economic Structure of Tort Law 73 (1987). Courts tend to be forgiving, however, where the cost of ascertaining the efficiency of noncompliance is low or zero. In cases where the deviation is demonstrably efficient or unavoidable, such as an accident resulting from a defendant's (provable) temporary physical incapacitation, courts have not imposed liability. *See, e.g.*, cases cited in Grady, *supra* note 151, at 887 n.26. *See also*

We argue that an efficient lapse, a compliance error, in antivirus precautions is particularly likely, due to the nature of the technology and economics of viruses and virus detection.

(ii) Virus Transmission Likely Involves Compliance Error—Negligence in antivirus precautions can occur in two ways, namely durable precautions below the Learned Hand level and compliance errors.

A formal economic analysis of compliance error in the context of virus prevention has shown that a rational software provider will invest in durable antivirus precautions at the due care level required by negligence law. However, the provider will invest in nondurable precautions at a level below the due care level. It is cheaper to the provider to spend less on nondurable precautions and risk liability exposure, rather than incurring the even higher cost of achieving perfectly consistent compliance with the legally imposed due care standard.[157]

Rational agents therefore will not fail in durable precautions but will likely commit compliance errors. Investing in durable precautions up to the efficient Learned Hand level is profit-maximizing because such investment reduces the provider's liability exposure by more than it costs. A compliance error is efficient due to the high cost of perfect consistency, hence, likewise profit-maximizing. Most negligent behavior on the part of rational, profit-maximizing software and service providers, therefore, will be the result of compliance errors.

We now argue that virus prevention technology is particularly susceptible to compliance error. Compliance error has a high likelihood where precautions are characterized by a high durable level, complemented by high levels and intense rates of nondurable precautions. These conditions make it harder to achieve perfectly consistent compliance with the due care standard and characterize virus prevention technology.

(iii) Antivirus Precautions Consist of Durable Precautions Complemented by a Significant Nondurable Component—Technical defenses against computer viruses consist of a durable precaution, complemented by essential nondurable precautions.[158] Durable antivirus precautions come in four main categories, namely pattern scanners, activity monitors, integrity monitors,

Ballew v. Aiello, 422 S.W.2d 396 (Mo. Ct. App. 1967) (finding defendant not liable for negligence because he was half asleep at the time he was allegedly negligent); Grady, *supra* note 151, at 887 n.59 ("For faints and other slips, it is possible for courts to judge whether they should have been avoided. Indeed, courts' measurement of unusual slips reintroduces the negligence component back into the negligence rule.").

157. *See* de Villiers, *supra* note 110 (mathematical analysis of compliance error in virus context). *See generally* Grady, *supra* note 151 (seminal article on compliance error).

158. Cohen emphasizes the importance of nondurable precautions in an antiviral strategy: "Suppose we want to protect our house from water damage. It doesn't matter how good a roof we buy . . . We have to maintain the roof to keep the water out. It's the same with protecting information systems." Cohen, *supra* note 8, at 148.

and heuristic scanners.[159] The durable precautions are complemented by nondurable precautions. An activity monitor, for instance, halts execution or issues a warning when it senses viruslike behavior. This requires nondurable precautions in the form of human intervention, consisting of observation and interpretation of monitor alerts and an appropriate response.

Virus scanners operate by searching for virus patterns in executable code and alerting the user when an observed pattern matches a virus signature stored in a signature database. Nondurable precautions complementary to a scanner include regular maintenance and updating of the virus signature databases, monitoring scanner output, and responding to a pattern match. An inadequately maintained signature database would reduce the effectiveness of a scanner, and virus alarms are worthless if ignored.

Several factors make compliance burdensome. Integrity checkers and heuristic scanners produce fewer false negatives but far more false positives than regular scanners. A large number of false positives make compliance more burdensome and efficient lapses more likely. False positives tend to diminish the effectiveness of the antivirus strategy, perhaps to the point of undermining confidence in the precaution. If the probability of a false alarm were high enough, it may be rational and efficient for a human operator to ignore some alarms. An ignored alarm may turn out to be real and result in the transmission of a virus. If the Learned Hand precautionary level required attention to all alerts, the courts would view such a lapse as negligence, even if the compliance error were efficient from the viewpoint of the human operator.

Scanners require a frequently updated viral pattern database, as new viruses are discovered at a high rate.[160] By the Learned Hand formula, the high danger rate associated with viral infection imposes a demanding nondurable precaution rate, such as a high database updating frequency and diligent monitoring of and responding to all alarms, regardless of the frequency of prior false alarms. Some critical applications may require virtually continuous updating, incorporating new virus strains in real time, as they are discovered.

159. *See* Section II.B, "Technical Antivirus Defenses," *supra*.

160. IBM's High Integrity Computing Laboratory reported, for instance, that by June 1991, new signatures were added to their collection at the rate of 0.6 per day. By June 1994, this rate had quadrupled to 2.4 per day and has since quadrupled yet again to more than 10 a day. Kephart et al., *supra* note 21, at 179–94. *See also* Steve R. White et al., *Anatomy of a Commercial-Grade Immune System*, IBM Thomas J. Watson Research Center research paper, *available at* http://www.av.ibm.com/ScientificPapers/White/Anatomy/anatomy.html (in the late 1990s, new viruses were discovered at the rate of eight to ten per day); Dunham, *supra* note 1, at xix ("[A]n estimated 5 to 10 new viruses are discovered daily, and this number is increasing over time."); Jennifer Sullivan, *IBM Takes Macro Viruses to the Cleaners*, Wired News (Dec. 4, 1997) ("It is estimated that 10 to 15 new Word macro viruses . . . are discovered each day.").

This discussion of antivirus precautions suggests that they consist of a high durable component, complemented by high rates and intense levels of nondurable precautions. The result is a high likelihood of a compliance error. The higher and more intense the rate of precaution, the more burdensome, hence more costly the cost of perfect compliance and the greater the likelihood of a compliance error.[161]

d. Conclusion

Most virus strains are avoidable, which implies that most cases of virus infection involve negligence. Furthermore, most cases of virus infection governed by the negligence rule involve a compliance error. When a virus penetrates a network and causes harm, failure to detect it in time is therefore likely due to a compliance error. Liability of the individual who exposed network users to the compliance error will likely be preserved under the dependent compliance error paradigm.

This conclusion remains valid, by a preponderance of the evidence, even in cases where the culprit virus cannot be reliably identified as avoidable or unavoidable. Even when the virus is not identifiable,[162] it is likely avoidable and likely involves a compliance error.

2. Paradigms in Reasonable Foresight Doctrine

The reasonable foresight doctrine governs multiple risks cases. The doctrine includes five mutually exclusive paradigms, namely (i) minimal systematic relationship, (ii) reasonably foreseeable harm, (iii) reasonable ignorance of the relationship, (iv) correlated losses, and (v) adverse selection.[163]

Under the minimal systematic relationship paradigm, an inadvertently negligent tortfeasor would not be held liable for coincidental harm that results from his or her negligence. To illustrate this paradigm, suppose a hypothetical defendant negligently exceeds the speed limit and arrives at a spot just in time to be struck by a falling tree. Although an injured passenger plaintiff may argue credibly that falling trees are foreseeable, the (coincidental) accident is likely outside the scope of risk created by the defendant's speeding. The defendant's speeding created risks of traffic accidents, but it neither created the risk of the falling tree nor increased the probability of its occurrence. The accident was therefore not within the scope of the risk created by the defendant's conduct, and liability fails on proximate cause grounds. It is coincidental and not systematically related to the defendant's negligence.

161. *See* de Villiers, *supra* note 110, ¶¶ 8–14 (describing possible complications in identifying the exact virus strain responsible for certain harm).

162. *Id.*

163. Mark F. Grady, *Proximate Cause Decoded*, 50 UCLA L. Rev. 293, 322 (2002).

Suppose, on the other hand, that the tree had fallen in front of the speeding driver and the car crashed into it. If it can be shown that the impact could have been avoided had the driver traveled at a reasonable speed, then the speeding driver's negligence may have been a proximate cause of the accident. Failure to stop with a short reaction time is a foreseeable risk of, and systematically related to, speeding.[164]

The reasonably foreseeable harm paradigm, described as the default paradigm under the reasonable foresight doctrine, imposes liability where an ex ante known systematic relationship exists between the defendant's negligence and the plaintiff's harm.[165] In *O'Malley v. Laurel Line Bus Co.*,[166] for instance, the defendant's bus driver let a passenger off in the middle of a street, instead of at the regular bus stop. It was a dark and stormy night so that the passenger did not realize where he was being let off. The court held the defendant liable for injuries sustained when the passenger was struck by a car. Letting people off in the middle of a street under such conditions that they cannot ascertain the risks of dangerous traffic does have a foreseeable systematic relationship to their being struck by a car.

Under the reasonable ignorance of the relationship paradigm, proximate causality is broken when, even though ex post there is clearly a systematic relationship between the defendant's untaken precaution and the harm, scientists would not have predicted the relationship ex ante. This paradigm is particularly relevant in a virus context, where scientific and technological state of the art evolves rapidly and often unpredictably.[167]

The issue of ex ante scientific knowledge is illustrated in the following classic case, known as the "Wagon Mound."[168] A ship was anchored in Alaska's Anchorage harbor. It negligently discharged oil into the water, but there was no apparent fire hazard, because the oil was of a type that required extremely high heat to ignite. Some debris, with a piece of cotton attached to it, floated on the water under the oil layer. The debris was covered by the oil and invisible to any observer. A welder's torch set off sparks that struck the cotton. The cotton smoldered for a while and eventually acquired sufficient heat to ignite the oil, causing a fire that burned down the dock. The dock owner sued the owner of the ship for damages under a negligence theory.

The oil spill created several risks, including hazards associated with water pollution and fire. The fire hazard was unforeseeable, because of the nature

164. Berry v. Borough of Sugar Notch, 191 Pa. 345 (1899); *see also* Grady, *supra* note 163, at 324.

165. Grady, *supra* note 163, at 326.

166. 166 A. 868 (Pa. 1933).

167. See Section IV.B, "Breach and Actual Cause Satisfied, but Proximate Cause Failed," *infra*, for a discussion and example of the role of reasonable ignorance of the relationship in a virus context.

168. Overseas Tankship (U.K.), Limited v. Morts Dock & Eng'g Co., Ltd. (The Wagon Mound), [1961] A.C. 388 (Privy Council 1961).

of the oil and the fact that the debris and cotton were out of sight. The risk of pollution was foreseeable but did not cause the harm.

The court accepted the testimony of a distinguished scientist who testified that the defendants could not reasonably have foreseen that the particular kind of oil would be flammable when spread on water.[169] The Privy Council therefore properly denied liability, and the suit failed on proximate cause grounds, namely reasonable ex ante ignorance of the relationship between defendant's untaken precaution and the harm.[170]

The correlated losses/moral hazard and adverse selection paradigms are mainly of historical interest, although they are based on sound public policy arguments that may be applicable in negligence cases.[171] The New York fire rule, which only permits recovery by the owner of the first property to which a fire spread, is a classic example of denial of liability under the correlated losses paradigm.[172] The adverse selection paradigm denies liability where, due to a heterogeneity of risks, the plaintiff would have received a better insurance bargain than others.[173]

The final element of a negligence cause of action is actual damages, to which we now turn.

169. *Id.* at 413 ("The raison d'etre of furnace oil is, of course, that it shall burn, but I find the [appellants] did not know and could not reasonably be expected to have known that it was capable of being set afire when spread on water.").

170. *See also* Doughty v. Turner Mfg. Co., [1964] 1 Q.B. 518 (C.A.), a case where proximate causality also turned on scientific state of the art. In *Doughty*, a worker negligently knocked the cover of a vat containing molten sodium cyanide into the molten liquid in the vat. The plaintiffs were injured when a chemical reaction between the molten sodium cyanide and the cover, which was made of a combination of asbestos and cement known as sindayo, caused an eruption that resulted in injuries to the plaintiffs. The risk that the cover might splash the molten liquid onto someone was known and foreseeable, but the chemical reaction that actually caused the harm was unknown and unpredictable at the time of the accident. Scientists later demonstrated that at sufficiently high temperatures the sindayo compound underwent a chemical change that creates steam, which in turn caused the eruption that injured the plaintiff. None of this was known at the time of the accident. The court therefore held for the plaintiff, stating that the defendant was reasonably ignorant of the chemical reaction that caused the injuries. *Id.* at 520, 525. The defendant escaped liability under the reasonable ignorance paradigm.

171. Grady, *supra* note 163, at 330–31.

172. *See*, *e.g.*, Homac Corp. v. Sun Oil Co., 180 N.E. 172 (N.Y. 1932); Ryan v. N.Y. Cent. R.R., 35 N.Y. 209 (1866) (Defendant negligently ignited its own woodshed, from which the fire spread to the plaintiff's house. The court denied liability, reasoning that first-party insurance by homeowners would be more efficient than imposing unlimited liability on a defendant for mass fires caused by its own inadvertent negligence. Such liability would constitute a "punishment quite beyond the offence committed." *Id.* at 216–17). The fire rule seems to have been limited to New York. Other courts have allowed recovery even when fire spread over great distances and over obstacles. *See*, *e.g.*, Cox v. Pa. R.R., 71 A. 250 (N.J. 1908) (recovery allowed for damage from fire that had spread beyond several buildings from its origin before destroying the plaintiff's building). Even in New York, the doctrine was not always followed. *See*, *e.g.*, Webb v. Rome, Watertown & Ogdensburgh R.R. Co., 49 N.Y. 420 (1872). Consistent with the "extent of harm" rule, it may apply to secondary victims of virus infection. *See also* Prosser & Keeton on the Law of Torts, *supra* note 3, at 282–83 (Time & Space).

173. Grady, *supra* note 163, at 331.

E. *Damages*

Damage resulting from virus infection can be classified into two broad categories: pre-infection and post-infection damages.[174] Pre-infection damages include the cost of detecting, tracing, identifying, and removing a virus before it enters the system or network. Typical expenses include personnel and managerial expenditures associated with the implementation and maintenance of software designed to detect a virus automatically at the point of entry as well as expenses for tracing the source of the virus, advising the source, logging the incident, and communicating with the owner of the system on which the incident occurred.

Post-infection damages can be classified into two main categories: (i) impact of the presence of a virus on the computing environment, before execution of the payload, and (ii) damage caused by execution of the payload.

Viruses modify the computing environment when they install their code on a host program and overwrite or displace legitimate code. Partly overwritten systems programs may become dysfunctional. Corrupted boot sector code, for instance, may prevent an infected computer system from booting and garbled spreadsheet formulas may make the program virtually unusable. Theft of resources, such as clock cycles, may slow down processes and, in the case of time-critical processes, cause them to behave unpredictably. Macro viruses, for instance, often disable menu options of Microsoft Word. Viral invasion of space in main memory and on the hard disk may result in impaired performance and disablement of some programs, including time-critical processes and resource-intensive software. In the absence of virus detection software, these modifications are often unobservable until execution of the payload.[175] These viral actions nevertheless cause actual damage, by dissipating valuable computing resources and disabling or disrupting commercially valuable computer functions.

Virus attacks have effects beyond the money and other resources required to recover from the attacks. In a survey of organizational effects of virus encounters, participants were asked about the organizational effects of virus incidents on their company or working group. The following table is a partial list of their greatest concerns, with the percentage of respondents reporting each effect.[176]

174. David Harley, *Nine Tenths of the Iceberg*, Virus Bull. 12 (Oct. 1999).

175. *Id.* at 13 ("General incompatibility/de-stabilization issues can manifest themselves in several ways. System software/applications/utilities display unpredictable behavior due to conflicts with unauthorized memory-resident software. Symptoms include protection errors, parity errors, performance degradation, loss of access to volumes normally mounted and unavailability of data or applications.").

176. ICSA Labs 9th Annual Computer Virus Prevalence Survey 2003, *supra* note 10, at 13 (Table 9).

Response	*Percentage*
Loss of productivity	76%
Unavailability of PC	67%
Corrupted files	58%
Loss of access to data	50%
Loss of data	47%

Damage from execution of the virus payload comes in three categories: loss of availability, integrity, and confidentiality of electronic information.[177] Attacks on availability include renaming, deletion, and encryption of files. Attacks on integrity include modification and corruption of data and files, including garbling of spreadsheet formulas and destruction of irreplaceable information. Attacks on confidentiality include security compromises, such as capturing and forwarding of passwords, e-mail addresses, and other confidential files and information.

The ICSA 2003 survey on computer virus prevalence provides numerical estimates of the effects of virus attacks. The survey defines a "virus disaster" as "25 or more PCs infected at the same time with the same virus, or a virus incident causing significant damage or monetary loss to an organization."[178] Ninety-two participants in the survey reported disasters with average server downtime of seventeen hours.[179] Respondents also were asked how many person-days were lost during the virus disaster that struck their company. The median time for full recovery was eleven person-days, and the average was twenty-four person-days. The average dollar cost per disaster, including employee downtime, overtime to recover, data and information loss, lost opportunities, etc., was in excess of $99,000.[180]

Consequential, or secondary, damage is defined as (i) damage (both pre- and post-infection) due to secondary infection, namely damage to other computer systems to which the virus spreads; (ii) damage due to an inappropriate response, such as unnecessarily destroying infected files that could be cheaply disinfected and restored; (iii) psychological damage, such as loss of employee morale and opportunities lost due to a sense of insecurity, bad publicity, and loss of reputation and credibility; (iv) the cost of cleanup and disinfection, the cost of restoration of the computer system and impaired data, and expenses related to upgrading computer security; (v) legal risks, such as exposure to civil and criminal liability; and (vi) punitive

177. Harley, *supra* note 174, at 13.
178. ICSA Labs 9th Annual Computer Virus Prevalence Survey 2003, *supra* note 10, at 1.
179. *Id.* at 10.
180. *Id.* at 13.

action from parties with whom the victim had breached a contractual agreement.[181]

Certain viruses attempt to conceal their presence on the computer system. Such concealment action may itself cause damage to the computing environment, independently of any harmful effect from execution of a payload. A virus may, for instance, attempt to thwart attempts to track it down by looking out for attempts to read the areas it occupies in memory and crashing the system in order to shake its "pursuer."

No viruses have been known to cause direct damage to hardware (at least at the time of writing), and losses are usually limited to destruction of data and related direct and indirect costs. A virus may cause indirect physical harm to hardware. Certain viruses are, for instance, capable of impairing the operation of a computer by writing garbage to a computer chip. It is often cheaper to repair the damage by discarding the entire motherboard than to replace a soldered chip.[182]

A negligence theory of liability would be irrelevant if no damages were recoverable. A doctrine in tort law, the so-called economic loss rule, appears to significantly limit recovery for damages caused by virus infection. The doctrine denies a defendant's liability for pure economic loss, namely loss not based on physical harm to person or property. In a related article, we argue that damages related to viral infection, including pure economic losses such as data corruption, are likely to be recoverable, the economic loss rule notwithstanding, because (i) a virus may cause physical harm due to the malfunction of a computer system, in applications such as medical systems and aviation; (ii) a minority of jurisdictions have relaxed the rule against recovery for pure economic loss; and (iii) an increasing number, perhaps a majority, of jurisdictions recognize electronic information as legally protected property.[183]

IV. LITIGATION COMPLICATIONS

The unique and dynamic nature of virus technology may complicate a plaintiff's litigation strategy. To succeed in a negligence action, the plaintiff has to plead an untaken precaution that simultaneously satisfies the re-

181. HARLEY ET AL., *supra* note 18, at 97–100; DUNHAM, *supra* note 1, at 7 (a user who receives a virus warning "may shut off the computer incorrectly, potentially damaging files, the operating system, or even hardware components like the hard drive"). *See also* ICSA LABS 6TH ANNUAL COMPUTER VIRUS PREVALENCE SURVEY 2000, *supra* note 110, at 31 (Table 16) (22 percent of respondents named loss of user confidence as a significant effect of a virus encounter).

182. HARLEY ET AL., *supra* note 18, at 100. *See also* Bissett & Shipton, *supra* note 113, at 899, 903 (describing the CIH virus, which overwrites memory, necessitating replacement of the memory chip).

183. *See* de Villiers, *supra* note 110, § VI.B (economic loss rule).

quirements of breach of duty as well as actual and proximate cause. In other words, the untaken precaution must be cost-effective and capable of preventing the harm if taken, and failure to take it must be reasonably related to actual harm.

In a given case there may exist precautions that clearly satisfy at least one, perhaps several, of the elements but no precaution that simultaneously satisfies all the elements of a negligence action. Modifying the pleading strategy by selecting an alternative precaution may fill the gap but leave yet a different subset of elements unsatisfied.

Antivirus technology is varied and sophisticated, reflecting antivirus researchers' response to the equally volatile and sophisticated nature of the virus threat, and a plaintiff usually has a rich array of untaken precautions to choose from. There may nevertheless, in many cases, exist no choice that simultaneously satisfies all the elements necessary to build a negligence case. Such a Catch-22 dilemma can, of course, arise in any negligence case, but it is especially likely in virus cases, as we show in this section.[184]

A. *Breach Satisfied but Actual Cause Failed*

A plaintiff will prevail on the issue of breach if her pleaded untaken precaution is cost-effective. Breach can often be proved quite easily in a virus context, by pleading a trivial precautionary lapse with negligible marginal benefit, yet even smaller cost, hence efficient. Suppose a software provider who signs up for fifty-two signature database updates per year is offered four free updates. The software provider opts not to use some or all of the free updates. The marginal cost, therefore, of increasing the updating frequency from fifty-two to, say, fifty-three times per year is approximately zero so that the fifty-third update is almost certainly efficient. However, the more trivial the lapse, the harder it is, generally, to establish actual and proximate causality. The fifty-third update, although efficient, is unlikely to make a significant practical difference in computer security. Failure to implement the fifty-third update will likely fail the but-for test of actual causality of a virus attack.

Although the fifty-third update will likely fail the but-for test, there is ample scope for the plaintiff to rethink her pleading choice. The rich array of available antivirus precautions virtually ensures the existence of an alternative precaution that would have prevented the virus, and therefore satisfies actual causality. A generic technology, such as an activity monitor, for instance, does not need an updated signature database to detect a novel virus strain. The virus that the fifty-third update failed to detect would therefore likely have been snared by an activity monitor. Failure to use an

184. Grady, *supra* note 61, at 139.

activity monitor will be an actual cause of the virus infection. It may, however, not be cost-effective, hence, fail the breach requirement.

Generic virus detectors, such as activity monitors, are very efficient in certain computing environments and quite inefficient and resource-consuming in others. The particular environment in which the virus caused the harm may be of the latter kind. The costs of the activity monitor may outweigh its benefits, so that failure to use it does not constitute a breach of duty, even though such failure is the actual cause of the virus harm.

Several factors may diminish the cost-effectiveness of an activity monitor in a particular computing environment. Activity monitors do not perform well with viruses that become activated before the monitor code and escape detection until after they have executed and done their harm. Activity monitors are also ineffective against viruses that are programmed to interfere with the operation of activity monitors. Certain virus strains, for instance, are programmed to sabotage the operation of activity monitors by altering or corrupting monitor code. Some, but not all, machines and networks have protection against such modification. A further drawback of activity monitors is that they can only detect viruses that are actually being executed, which may be a significant detriment in sensitive applications where a virus can wreak havoc before being caught by an activity monitor.

A further disadvantage of activity monitors is the lack of unambiguous and foolproof rules governing what constitutes "suspicious" activity. This may result in false positive alarms when legitimate activities resemble viruslike behavior and false negative alarms when illegitimate activity is not recognized as such. The vulnerability of activity monitors to false alarms makes them relatively costly.[185] A high cost of dealing with false negatives and positives may outweigh the benefit provided by activity monitors in a particular environment. An activity monitor may therefore not be cost-effective because of any or all of these factors, even though it may have been technically capable of detecting the culprit virus.

B. *Breach and Actual Cause Satisfied, but Proximate Cause Failed*

The rapid and often unpredictable development of virus technology introduces an element of unforeseeability into the behavior of viruses. New virus creations often have the explicit goal of making detection harder and more expensive.[186] Innovations, undoubtedly designed with this goal in mind,

185. The technology is programmed to make a judgment call as to what constitutes "suspicious behavior." There are, however, no clear and foolproof rules governing what constitutes suspicious activity. False alarms may consequently occur when legitimate activities resemble viruslike behavior. Recurrent false alarms may ultimately lead users to ignore warnings from the monitor. Conversely, not all "illegitimate" activity may be recognized as such, leading to false negatives.

186. *See, e.g.*, Spinellis, *supra* note 31, at 280 ("Even early academic examples of viral code were cleverly engineered to hinder the detection of the virus."). *See also* Ken L. Thompson, *Reflections on Trusting Trust*, 27:8 Comms. ACM 761–63 (Aug. 1984).

include stealth viruses,[187] polymorphic viruses, and metamorphic viruses.[188] As a consequence, some virus strains are capable of transforming into a shape and causing a type of harm very different from what was ex ante foreseeable.

These and other unpredictable aspects of viruses may cause a negligence action to fail on proximate cause grounds, where foreseeability is an issue. In a particular virus incident, an ex post obvious systematic relationship may exist between the evolved virus and the harm it has caused. If, however, computer scientists could not ex ante foresee or predict this dynamic relationship, proximate cause may be broken and defendant's liability cut off.

The following example illustrates this complication. Viruses can be roughly divided into two groups: those with a destructive payload and those without a payload, or with a relatively harmless payload, such as display of a humorous message. For the purposes of this example, we refer to the two types as "harmful" and "harmless" viruses, respectively.[189]

Suppose a hypothetical software provider decides not to scan for "harmless" viruses, perhaps to increase scanning speed and reduce costs, or because of a perceived low likelihood of exposure to liability and damages. The provider purchases only signatures of new viruses that are known to be harmful, at the time, for inclusion in his scanner database. The software provider then sells a software product containing a harmless virus strain that, by design, was not detected. This virus infects the computer network of the purchaser of the infected program.

The virus happens to be a metamorphic virus,[190] a type of virus capable of mutating into a totally different virus species. In fact, it mutates into a strain with a malicious payload capable of destroying data. The mutated strain, now transformed into a harmful virus, erases the hard disk of its host computer. The purchaser of the infected software contemplates a lawsuit against the vendor on a negligence theory.

187. Stealth virus strains are designed to evade detection by assuming the appearance of legitimate code when a scanner approaches. *See, e.g.*, Kumar & Spafford, *supra* note 25; *see also* David Ferbrache, A Pathology of Computer Viruses (1992), for a description of stealth viruses.

188. Polymorphic viruses change their signature from infection to infection, making them harder to detect. Metamorphic viruses are capable of changing not only their identity but also their entire nature and function. *See, e.g.*, Carey Nachenberg, *Understanding and Managing Polymorphic Viruses*, The Symantec Enterprise Papers, Volume 30. *See also* Spinellis, *supra* note 31, at 280 ("Viruses that employ these techniques, such as W32/Simile[,] can be very difficult to identify.").

189. Bissett & Shipton, *supra* note 113, at 899, 903 ("Viruses may be classified as destructive or nondestructive in their primary effect. The least destructive . . . simply print a . . . message and then erase themselves. . . . Destructive effects include halting a legitimate program. More destructive viruses erase or corrupt data or programs belonging to legitimate users of the computer.").

190. Metamorphic viruses are capable of changing not only their identity but their very nature. *See, e.g.*, Nachenberg, *supra* note 188.

The plaintiff could easily prove breach of duty by arguing that the trivial marginal cost to the software provider of scanning for "harmless" viruses is outweighed by the foreseeable harm from such viruses in the form of consumption of computing and personnel resources. Defendant, on the other hand, could credibly argue that proximate causality should be broken under the reasonable ignorance of the relationship paradigm. Although it is clear after the incident that a systematic relationship existed between the harm and the defendant's untaken precaution (failure to scan for harmless viruses), computer scientists were nevertheless unaware of this systematic relationship ex ante. This systematic relationship originates from the ability of harmless viruses to transform into harmful ones, which depends on the existence and feasibility of metamorphic virus technology. This technology was unknown ex ante, even to scientists.

C. *Attempt to Fix Proximate Causality Fails Breach Test*

The plaintiff in the foregoing metamorphic virus example may attempt to fix the proximate causality problem by rethinking his pleaded untaken precaution. Once the first harmless virus has morphed into a destructive one, the provider of the infected software can prevent further carnage by recalling all his previously sold software products and rescanning them for all viruses, harmful as well as harmless. A plaintiff therefore may plead that the defendant, once the infection came to his or her attention, could have taken this action. Failure to recall will be the proximate cause of any further (now foreseeable) harm from this type of virus, under the no intervening tort paradigm, or perhaps the reasonably foreseeable harm paradigm. Failure to recall is, of course, also an actual cause of all further harm caused by the virus.

The plaintiff nevertheless may still find him- or herself stuck in a legal Catch-22. Empirical studies on the economic impact of product recall strongly suggest that product recalls are very costly.[191] In cases where human lives are not at stake, as is usually the case with ordinary commercial software, product recall may very likely not be cost-effective and failing to undertake it would not be a breach of duty. The plaintiff who pleads product recall as an untaken precaution will likely be able to prove actual and proximate causality but, this time, fail on breach.

V. CONCLUSION

This article analyzes the elements of a negligence cause of action for inadvertent transmission of a computer virus. The analysis emphasizes the

191. *See, e.g.*, Paul H. Rubin et al., *Risky Products, Risky Stocks*, 12 REGULATION 1, which provides empirical evidence of the costs associated with a product recall and states that "[o]n the basis of this research, we conclude that product recalls are very costly, resulting in large drops in the stock prices of affected firms. . . [T]he health and safety benefits to consumers may not be worth the cost."

importance of an understanding of virus and virus detection technology, as well as the economics of virus prevention, in negligence analysis. The classic principles of negligence apply to a virus case, but a plaintiff's case may be significantly complicated by the unique and dynamic nature of the technologies involved.