Somebody has to Pay: Products Liability for Spyware

Jacob R Kreutzer
SOMEBODY HAS TO PAY: PRODUCTS LIABILITY FOR SPYWARE

Jacob Kreutzer*

It can be unsettling to discover that you have spyware (software that tracks user behavior and displays advertisements) installed on your computer. The signs of its presence can vary: a new toolbar may appear in your browser, or you may experience a proliferation of pop up advertisements. Whatever the symptoms, the cause is the same: a piece of software that was inconspicuous (or invisible) at the time of its installation is now dedicating itself to disrupting your use of your computer.

A persistent consumer will likely be able to identify spyware that is present on his computer, whether through technological sleuthing or by using one of the numerous spyware detectors available on the Internet. The first question on almost every consumer’s mind will be: “How do I get rid of this thing?” After they have gone through the detection and removal process several times, a different question may come to mind: “How can I make somebody pay for what they’ve done to my computer?”

A survey of the current literature, law, and pending legislation on this subject reveals that the answer to that question has consistently been “you can’t.” All of the currently contemplated attempts to impose liability focus on the act of installing spyware as the trigger for liability. As experience has shown, it is very difficult to separate good installations from bad, at least when we are only considering installations that are accompanied by user consent to licensing agreements. This article will show it is more productive to focus on the behavior of spyware after it is installed. By doing so, we can apply the basic principles of products liability to find that the answer to the question of “How can I make somebody pay?” may well be found in a suit based on product liability.

* Associate, Fried, Frank, Harris, Shriver & Jacobson, LLP. The author would like to thank Barry Adler, Jordan Barry, Sarah Blackman, Jeffrey Lipshaw, Sarah Trombley, and Katrina Wyman for their helpful guidance, discussion, and comments.
INTRODUCTION

I. BACKGROUND
   A. FACTUAL BACKGROUND
   B. OTHER LEGAL APPROACHES TO THE SPYWARE PROBLEM
      1. Contract Based Claims
      2. Trespass to Chattels
      3. Lawsuits by Advertisers
      4. Legislative Responses
      5. Federal Trade Commission Litigation

II. SPYWARE AS A DEFECTIVE PRODUCT
   A. PRODUCT
   B. DEFECT
   C. HARM
      1. Pure Economic Loss
      2. The Other Product Exception
      3. Applying the Theory
      4. Calculating Damages

III. SOFTWARE DEFECTIVENESS GENERALLY
   A. OBVIOUS DEFECTS
   B. CLASSIFYING DEFECTS
   C. IDENTIFYING DEFECTS
      1. Negligence Analysis Example
      2. Strict Liability Analysis Example
      3. Calibrating Liability

CONCLUSION
INTRODUCTION

It can be unsettling to discover that you have spyware installed on your computer.\(^1\) The signs of its presence can vary: a new toolbar may appear in your browser, your bookmarks may be changed to direct you to strange new web sites, or you may experience a proliferation of pop up advertisements. Whatever the symptoms, the cause is the same: a piece of software that was inconspicuous (or invisible) at the time of its installation is now dedicating itself to disrupting your use of your computer.

A persistent consumer will likely be able to identify spyware that is present on his computer, whether through technological sleuthing or by using one of the numerous spyware detectors available on the Internet. The first question on almost every consumer’s mind will be: “How do I get rid of this thing?” After they have gone through the detection and removal process several times, a different question may come to mind: “How can I make somebody pay for what they’ve done to my computer?” This article addresses one possible answer to this second question by analyzing whether a victim of spyware can seek relief under consumer product liability doctrine.

“Spyware” is a catch-all term for software that behaves in a manner hostile to the interests of the owner of the computer on which it is installed.\(^2\) The two most common forms of hostile behavior are to track a user’s behavior (and communicate it to a central location) and to create pop up advertisements.\(^3\) Spyware is commonly installed through the exploitation of security breaches and outright fraud.\(^4\) However, the method of choice for the largest players in the spyware industry is bundling,\(^5\) wherein a spyware

---

\(^1\) “Spyware” is the general term used to describe a broad variety of software that operates covertly on personal computers to track user behavior and display advertising. See infra notes 27-29 and accompanying text.


\(^2\) See Comment, supra note 1 at 882.


\(^4\) Comment, supra note 1 at 893-96.

\(^5\) Id. at 896.
program is incorporated into a package of useful software. Accordingly, this article will focus on spyware installed through bundling.

A survey of the current literature, law, and pending legislation on this subject reveals that the answer to the question of “How can I make somebody pay?” has consistently been “you can’t.” Courts have routinely held that consumers’ assent to End User License Agreements (or “EULAs”) suffice to show consent to the installation of spyware, foreclosing any action based on a contract theory of liability. The EULA disclaimers also prevent any action based on trespass to chattels. State legislative efforts to outlaw spyware are largely ineffective, unconstitutional, or both. Federal legislation has been “right around the corner” for three years now, and even if the currently contemplated legislation were to pass, it is unlikely to be effective. FTC enforcement efforts have had a mixed record of success, and in any event have focused on spyware installed through force or fraud.

All of these attempts to impose liability focus on the act of installing spyware as the trigger for liability. As experience has shown, it is very difficult to separate good installations from bad, at least when we are only considering installations that are accompanied by EULAs. This article will show that it is more productive to focus on the behavior of spyware after it is installed. By focusing on spyware as a product, we can apply the basic principles of products liability to find that the answer to the “How can I make somebody pay?” question may well be found in a suit based on product liability.

Manufacturers are generally liable under product liability doctrine when they produce a defective product that causes harm to the consumer. The three elements of product, defect, and harm all must be satisfied for the consumer to be able to recover. In most product liability cases

---

6 Id.
7 Winn, supra note 1, at 1355; Comment, supra note 1, at 914-21.
8 An End User License Agreement is a contract that accompanies software, and typically require a user to click “I Agree” before installation can proceed. See Davidson & Assoc. DBA Blizzard Entertainment Inc. v. Jung and Internet Gateway, 422 F.3d 630, 634-35 (8th Cir. 2005) (describing the End User License Agreement that accompanied Blizzard’s software).
9 See i.Lan Systems, Inc. v. Netscout Service Level Corp., 183 F. Supp. 2d 328, 337 (D. Mass. 2002) (“The only issue before the Court is whether clickwrap license agreements are an appropriate way to form contracts, and the Court holds they are.”); Comb v. PayPal, Inc., 218 F. Supp. 2d 1165, 1169, 1173 (N.D. Cal. 2002) (accepting that the clickwrap process created an agreement, but finding the agreement unconscionable as a contract of adhesion).
10 In a contractual relationship, trespass to chattels occurs only when the scope of the contract is exceeded. See City of Amsterdam v. Daniel Goldreyer, Ltd., 882 F.Supp. 1273, 1280-81 (E.D.N.Y. 1995).
11 Comment, supra note 1, at 901-09.
12 Id. at 910-11.
13 Id. at 912-13.
14 Id. at 915-16.
15 Restatement (Third) of Torts: Product Liability § 1.
16 Id.
(involving, e.g., rocking chairs, vaporizers, candy, or underwear), the result turns on the issue of defect—it is usually obvious that the consumer bought a product and that he or she was injured. Spyware cases invert these issues: establishing defectiveness in spyware is fairly straightforward, while showing that a compensable injury has occurred is more involved.

Once we focus on the behavior of spyware after it is installed, it quickly becomes clear that spyware is a defective product. Almost all spyware includes features designed solely to prevent detection and removal by the end user. Since removing these features would impose no burden on the producer or consumer of the software, they render spyware defective if it does any harm compensable under products liability. While the irritation of seeing pop up ads likely is not compensable in tort, avoiding removal and thus occupying computer resources against the will of the user likely does rise to the level of harm that triggers tort liability.

We still might reasonably conclude that it would not be worthwhile for any particular consumer to bring a products liability suit, given the relatively small nature of the likely damages award. However, product liability causes of action involving spyware lend themselves to class action suits, which would certainly be worth bringing, and which could have a real impact on manufacturer behavior.

Moving beyond the limited case of spyware, this article concludes with a discussion of how we should evaluate software in general under products liability law. More particularly, it addresses the question of how we should evaluate defectiveness when confronted with a piece of software that is erratic in its behavior rather than malicious in its design. I tentatively conclude that a strict liability standard should be applied, as it is necessary to give manufacturers the proper incentives to produce safe software. Concerns about overly extensive litigation are better addressed by a review of the difficulties involved in showing causation and harm by software, rather than by imposing an impossible test of defectiveness.

Part I of this article will provide background information, discussing what spyware is, how it behaves, and the legal approaches that have been adopted or suggested to curtail its excesses. Part II will analyze spyware as

---

17 See Matthews v. Lawnlite Co., 88 So.2d 299 (Fla. 1956).
20 See Nichols v. Union Underwear Co., 602 S.W.2d 429 (Ky. 1980).
22 See infra notes 115-120 and accompanying text.
23 As it is a purely economic loss, rather than damage to a protected interest. See infra notes 121-206 and accompanying text.
24 Being damage to an interest that is protected by tort law. See infra notes 181-83 and accompanying text.
25 Software is mass produced and mass distributed, and the damage done in all cases is nearly identical.
26 See infra notes 207-244 and accompanying text.
a product, discussing the rules of product liability as they are and as they should look when applied to digital products. Ultimately this part will conclude that most spyware programs are defective products, and that consumers should be able to recover against the producers and distributors of spyware under a products liability theory. Part III will then discuss the more general problem of determining when software is defective, particularly when the software is merely error-prone rather than borderline malicious.

I. BACKGROUND

A. Factual Background

Spyware is a catch-all term for software that runs in the background on a user’s machine without the user’s knowledge, and that does things that are hostile to the user’s interests.27 Some of these programs gather information about the user’s computer usage, some pop up advertisements whenever the user is on the Internet, and many do both.28 The companies that make and distribute spyware profit by selling information about user behavior and by selling advertising space.29

Consumers generally do not seek to install spyware on their system. Spyware is most commonly installed through security breaches in a consumer’s system (so called “drive-by installations”),30 through fraud,31 or by bundling.32 A number of legal theories are available for victims of drive-by installations or fraud, as such behavior is patently indefensible. However, installation via bundling is a much more common practice and much bigger business,33 and thus is the focus of this article.

In a bundling installation, the spyware will be part of a package of software downloaded by the user.34 Typically the installation of the spyware will be disclosed in the license accompanying the software.35 While such disclosure is legally sufficient to constitute consent, it’s fair to assume that most users are not consciously aware of the installation of the spyware.36 However, this article is not aimed at challenging the general

27 See Stafford & Urbaczewski, supra note 3, at 292 (“Spyware is the name given to the class of software that is surreptitiously installed on a user’s computer and monitors a user’s activity and reports back to a third party on that behavior”).
28 See id. at 292-93.
29 See id.
30 See Comment, supra note 1 at 893.
31 Id. at 896.
32 Id. at 895.
33 Id. at 896.
34 Id.
35 Id. at 897.
36 See Stafford & Urbaczewski, supra note 3, at 292.
rule regarding disclosure, and will assume that users have knowingly consented to spyware installation. Even conceding that the spyware installation was consensual, its behavior after installation can create liability.

Unsurprisingly, most people do not like having spyware on their computer, and will do what they can to remove it once they discover its presence. To prevent this, most spyware programs take active steps to avoid removal. In response, a wide variety of programs have been developed that allow users to engage in technological self-help and remove spyware automatically. This has led to a sort of arms race, with spyware removed by anti-spyware programs, which are then stymied by more sophisticated spyware, leading to another generation of anti-spyware programs, and so forth. The resulting ecosystem bears some resemblance to the world of computer viruses and anti-virus programs. The primary difference is that in the case of spyware, the party creating the “virus” is a large corporation that derives significant revenues from the propagation of the spyware.

Some commentators have suggested that this ecosystem is a market solution to the spyware problem, and that this is an acceptable state of affairs. This view is flatly inconsistent with the standard view of product manufacture, which sees any product that forces consumers to invest in self-help for safety reasons as unacceptably flawed. Manufacturers should not be allowed to profit by imposing costs on consumers, no matter how many clever solutions the market subsequently comes up with to minimize those costs.

37 This fact is somewhat at odds with the assumption that the user knowingly consents to the installation of spyware. See New.net, Inc. v. Lavasoft, 356 F.Supp.2d 1071, 1078 (C.D. Cal. 2003)
38 See Stafford & Urbaczewski, supra note 3, at 297-98.
39 Id. at 299.
40 See Crawford, supra note 1, at 1468. See also Stefan Saroiu et al., Measurement and Analysis of Spyware in a University Environment, Usenix Association Proceedings of the First Symposium on Networked Systems Design and Implementation at (2004) (describing how “[m]any instances of spyware have the ability to self-update, or download new versions of themselves automatically… [self-updating] may be used to evade anti-spyware tools”). Some companies have even adopted the tactic of marketing their spyware programs as anti-spyware tools. See Timothy L. O’Brien & Saul Hansell, Barbarians at the Digital Gate, N.Y. TIMES, September 19, 2004 (“Spyware Labs Inc., a Hawaiian company, promotes itself as a vendor of anti-spyware tools but peddles a product called Virtual Bouncer that experts like Mr. Porter say functions as spyware and adware once it is installed on a computer”).
41 Some have used this comparison to argue against regulation of spyware. See Emily C. Kumler, What’s the Best Way to Stop Spyware?, PC World, Apr. 20, 2004, available at http://www.pcworld.com/news/article/0caid,115765,00.asp (quoting Andrew McLaughlin, senior policy council for Google as saying “We didn’t look to laws to stop viruses, we looked to technology like Norton Antivirus that stops viruses”).
42 See Timothy L. O’Brien & Saul Hansell, Barbarians at the Digital Gate, N.Y. TIMES, September 19, 2004 (noting that in 2003, Claria “earned about $35 million on $90.5 million in revenue - an enviable profit margin”).
43 This sort of contributory negligence argument (that the problems caused by the spyware are the fault of the consumer for his lack of vigilance) is routinely rejected. See, e.g., Bexiga v. Havir Manufacturing Corp., 290 A.2d 281 (1972) (“The asserted negligence of the plaintiff… was the very eventuality the safety devices were designed to guard against.”).
Three distinct parties are involved in the production and distribution of spyware: the manufacturer, the distributor, and the advertiser. Each of these parties has their own role to play in maintaining a successful spyware network.

The manufacturer designs and produces the spyware. He also operates the network of installed spyware, coordinates the display of advertisements on user’s computers, and uses the data collected by the installed base of spyware. The spyware manufacturer is directly analogous to the manufacturer in a traditional product liability case.

The distributor can be any web site owner or can even be an unaffiliated software designer. The manufacturer will pay the distributor based on the number of copies of its spyware that the distributor is able to get user’s to install. To do this the distributor can engage in the various tactics described above; as mentioned above, the focus of this article is on bundling. Again, the distributor is closely analogous to the retailer in a traditional products liability setting.

The advertiser is where the money comes from in this system. In exchange for his payment, the manufacturer will ensure that his ads will be displayed on the computers of users who have spyware installed. Interestingly, the advertiser does not have a close analogue in traditional products liability. It is not clear whether, should spyware be treated as a defective product, advertisers could be held liable for any damages caused by such spyware.

B. Other Legal Approaches to the Spyware Problem

1. Contract Based Claims

One might expect the purchaser of defective software to be able to recover against the software manufacturer on a warranty or other contract claim. After all, goods sold carry with them an implied warranty of merchantability and fitness for their intended purpose under the UCC. In addition, bringing a claim under contract law allows the plaintiff to recover for the sort of purely economic harms that are not compensable under tort-based theories of recovery.
However, under the UCC manufacturers are able to disclaim these implied warranties, and software manufacturers have done so en masse. While there are some doubts as to whether shrinkwrap licenses are always enforceable, courts have uniformly upheld click-through licenses. Not surprisingly, click-through licenses (in the form of End User License Agreements, or “EULAs”) are now almost universal.

These licenses have been upheld even when that magnitude of the alleged damages is quite high. In one case, a company that had lost almost two million dollars because of a software error was not able to recover because of a shrinkwrap license on the purchased software. When spyware is installed via bundling, the EULA accompanying the bundle will invariably disclose the proposed installation of the spyware as well as disclaiming any liability associated with the spyware. Accordingly, it seems unlikely that most consumers will ever be able to recover under contract law for the damage that software does to their computers.

2. Trespass to Chattels

Given the intrusive behavior of spyware programs, it seems natural to bring an action for trespass to chattels against spyware distributors. The tort of trespass to chattels has been applied to several forms of anti-social behavior in the digital world. In particular, sending large amounts of spam email, or using a large amount of automated “robots” to download information from a web site, have both been found to be trespass to chattels. In the case of spyware installed via security breaches or through fraud, it is reasonable to apply trespass to chattels, as the spyware distributor has tortiously intermeddled with the consumer’s chattel.

However, trespass to chattels is not a good way to attack spyware that has been installed through bundling. When applied between parties that have a contractual relationship, trespass to chattels exists only when the contractual relationship is exceeded. The only interaction between the

53 A shrinkwrap license is a license that is included in a package of software that the user does not see until he opens the box. Most such licenses state that the user accepts their terms by opening the package. The general trend is towards enforceability of such licenses. Bellia, supra note 1, at 2242-43.
54 See i.Lan Systems, Inc. v. Netscout Service Level Corp., 183 F. Supp. 2d 328, 337 (D. Mass. 2002) (“The only issue before the Court is whether clickwrap license agreements are an appropriate way to form contracts, and the Court holds they are.”); Comb v. PayPal, Inc., 218 F. Supp. 2d 1165, 1169, 1173 (N.D. Cal. 2002) (accepting that the clickwrap process created an agreement, but finding the agreement unconscionable as a contract of adhesion).
55 See Stafford & Urbaczewski, supra note 3, at 298.
57 Id.
61 One such case survived a motion to dismiss, although it ultimately settled before judgment. See Sotelo v. DirectRevenue, LLC, 384 F.Supp.2d 1219, 1229-33 (N.D. Ill. 2005).
consumer and the distributor is the installation of the spyware, to which the user consents, however tenuously. The actions after installation that so badly annoy the consumer are problems with the product as installed, not problems with the behavior of the manufacturer.

This line can be blurred somewhat with spyware that installs other spyware, or spyware that downloads updates to itself from a central server. In those cases the manufacturer is using the consensually installed software to make changes to the consumers’ computer which are not consensual (although such behavior may be disclosed by the original EULA, it’s not clear exactly how far such consent can be stretched). However, when spyware limits itself only to reporting on user behavior and popping up advertisements, a trespass to chattels action should not succeed, and is in fact unlikely to succeed.

3. Lawsuits by Advertisers

Some spyware operates not by creating its own pop up ads out of whole cloth, but by intercepting pop up ads sent out by web sites and replacing them with its own ads. For example, a web site might attempt to send an advertisement for Pepsi, which the spyware program intercepts and replaces with an advertisement for Coke. The consumer would be unaware of any irregularity, and the spyware distributor would receive a payment from the company whose ad it distributes. The advertiser, in turn, would be happy to pay a premium rate for the chance to lure consumers away from its competitor. However clever this trick might be, it understandably raised the ire of online advertisers, who brought suit against spyware distributors on a theory of misappropriation of resources.

Ultimately these suits were settled out of court, and appear not to have had a major impact in how spyware companies conduct their business. In any event, such suits could only serve as a solution in the limited context of “hijacked” advertisements, and would have no effect on the larger area of consumers’ problems with spyware.

---

64 See Stafford & Urbaczewski, supra note 3, at 297 (describing how “[t]he Hertz rental car agency sued over pop-up ads promoting their competitors upon customer visits to Hertz”). See also Stefanie Olson, Web Sites Prey on Rivals’ Stores, CNET News.com, August 7, 2001.
66 Id.
4. Legislative Responses

The consumer who discovers spyware on his computer can not look to any legislative enactments to help him seek compensation (or punishment) for the harms inflicted on him by spyware.67 Most state laws on the subject do not apply to software installed pursuant to an EULA,68 while the one state law that did attempt to apply a more comprehensive regulatory regime to spyware was found to violate the dormant commerce clause of the constitution.69

Federal legislation on this subject has not been passed, and the legislation under consideration offers little help to the injured consumer. Both the SPY ACT70 and the I-SPY71 bill focus on regulating unauthorized installation and insufficient disclosure in EULAs. However, neither dictates a particular form, leaving open the possibility that new disclosures will be buried deep in complicated EULAs alongside already existing disclosures.72 Neither bill addresses the problem of spyware that evades or resists removal.73

5. Federal Trade Commission Litigation

The Federal Trade Commission (the “FTC”) has brought a number of lawsuits against spyware manufacturers.74 In general, these suits have focused on deceptive trade practices—that is, on spyware that was installed either through security breaches or through fraud.75 At best, these cases will have a deterrent effect on spyware installed through the same methods. They offer little hope to the consumer frustrated by spyware installed pursuant to a click through EULA.76

II. SPYWARE AS A DEFECTIVE PRODUCT

Product liability is an attractive path for recovery for an injured individual. Recovery based on product liability is not limited by the End

---

67 See Comment, supra note 1, at 921.
68 Id. at 902-04.
72 Comment, supra note 1, at 911-913.
73 Id. at 913-14.
74 Id. at 914.
75 Id. at 915-17.
76 Id.
User License Agreements accompanying software, and can be applied to any party in the chain of distribution of spyware. In addition, the same principle that allows consumers to recover from retailers may be extended to allow consumers to recover from advertisers who use spyware to deliver their messages. But before reaching this point, there remains the question of whether spyware is properly treated as a defective product.

Product liability generally applies when a defective product causes harm to the consumer. The three elements of product, defect, and harm all must be satisfied for the consumer to be able to recover. The archetypal product liability case involved an exploding soda bottle. In that case, as in most product liability cases (involving, e.g., rocking chairs, vaporizers, candy, or underwear), the result turns on the issue of defect—the consumer obviously bought a product and was obviously injured. Notably, product liability claims are not barred by disclaimers or license agreements accompanying consumer goods.

Showing that software is a defective product requires some effort be expended to show that software is a product that caused harm, in addition to the usual struggle of proving defectiveness. Focusing on spyware lets us set aside the issue of defect, thus reversing the usual set of contested issues—the program is clearly defective, but it is not obvious that it is a product or that it has caused compensable harm. A survey of the case law shows that applying products liability to spyware will require some extension of existing law, but no more than is usually required to translate legal concepts to the digital world.

Relatively few products liability cases have been filed alleging that harm has been done by defective software. This lack of cases does not necessarily show that society lacks for defective software. Most of the dearth of cases is likely explained by the fact that software failure generally leads to lost time and money rather than personal injury, leading the disappointed party to seek redress under the UCC or some other contract theory. Such an avenue of recovery will likely be open to large companies making heavy investments in software, as contracts negotiated between two entities with similar bargaining power usually have reasonably

77 See RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 18.
78 See RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 2.
79 See RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 1.
80 Id.
82 See Matthews v. Lawnlite Co., 88 So.2d 299 (Fla. 1956).
85 See Nichols v. Union Underwear Co., 602 S.W.2d 429 (Ky. 1980).
86 RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 18.
equitable remedies for failure to perform. This leaves little incentive for corporate plaintiffs to attempt to recover in tort. Individual consumers who are precluded from recovering in contract because of click-through or shrink-wrap licenses will rarely suffer sufficient harm to justify the expense of bringing a tort suit (particularly since the overwhelming majority of them suffer no personal injury as a result of software failure).88

The few defective software cases brought as tort claims have generally been dismissed as only involving economic damages. For example, a restaurant was not able to recover from a software developer when the restaurant management software malfunctioned, causing the loss of customers and profit.89 Similarly, an oil-and-gas drilling company was not able to recover the costs of drilling a dry well based on faulty information provided by specialized software.90 In one case involving allegedly defective office management software the plaintiffs were able to get a class certified based on a theory of “breach of contract, fraud, constructive fraud, fraud in the inducement, negligent misrepresentation and promissory estoppel”;91 however, the class was decertified by the Supreme Court of Texas.92

Most analysis of software as a potentially defective product has focused on the potential for physical injury caused by defective software (most likely caused by the hypothetical failure of software integrated into another product as part of a control system causing, for example, a plane crash).93 However, the analysis required to establish liability in such cases does not differ from the usual product liability analysis, as the defect causes physical harm. Establishing liability for spyware is more difficult because it can only cause harm to the computer that it occupies.

A. Product

Of course, products liability is not applied to things people buy which are not products (such things are usually referred to as “services”).94 The product/service distinction commonly arises in cases involving medical

89 Gus’ Catering Inc. v. Menusoft Systems, 762 A.2d 804, 807 (Vt. 2000) (holding that loss of business profits, customers, and time are economic losses, not physical harms, and therefore not recoverable under negligence theory).
92 Id. at 700-01 (finding failure to demonstrate “that the questions of law or fact common to the members of the class predominate over any questions affecting only individual members, and that a class action is superior to other available”).
94 See RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 19 Comment f.
procedures. For example, a patient may pay a doctor to install some sort of prosthesis. Should the prosthetic later harm the patient in some way, he will generally have to recover from the doctor for negligence in providing medical services, rather than on a product liability theory regarding the prosthesis. If software is more properly classified as a service than as a product, then customers can’t possibly recover for harms done by software on a product liability theory.

A spyware manufacturer could try to use this to escape liability. Claria, for example, claimed to provide consumers with the service of displaying context sensitive advertisements for products that Claria’s software determined the consumers were likely to want. On this theory, the spyware program itself is only an adjunct to the provision of the promotional service, and product liability is therefore inappropriate.

The hypothetical argument presented above is not frivolous, but is ultimately unpersuasive. Spyware is pretty clearly a product under either of the commonly used tests.

The general rule is that tangible goods are products, and that intangible goods are products when “the context of their distribution and use is sufficiently analogous to the distribution and use of tangible personal property that it is appropriate to apply the rules [of product liability].” As discussed below, software in the modern marketplace qualifies as a product under both methods of evaluation.

1. Tangibility

Under the first prong of the restatement test, tangible goods are recognized as products. Using this test, one might argue that software is intangible, since it isn’t something that you can see or touch—particularly so with spyware, which is usually delivered through the Internet rather than through some physical intermediary.

In most states, software is almost certainly sufficiently tangible to be considered a product rather than a service. Software that is purchased from a store comes in some tangible form, whether it be on compact discs, DVDs, diskettes, or even punch cards. Many courts and commentators have latched on to this fact as a hook for treating software as a product.
However, today software can be purchased and delivered entirely through the Internet, which means that it never takes a form more tangible than as magnetic impulses inside the consumer’s computer.\footnote{See generally Dirk Van den Poel & Joseph Leunis, Consumer Acceptance of the Internet as a Channel of Distribution, 45 JOURNAL OF BUSINESS RESEARCH 249 (1999).}

However, even the magnetic impulses transmitted over the Internet are usually sufficiently tangible to serve as a basis for liability. Most states treat electricity as a product.\footnote{See, e.g. Ransome v. Wisconsin Elec. Power Co., 275 N.W.2d 641 (Wis. 1979).} Anything that is as tangible as electricity is therefore also a product. Since an arrangement of magnetic impulses that reside on a computer is at least as tangible as an electrical current running through various appliances, we would expect states with this rule to show no hesitation in treating software as a product.

In the remaining states, the treatment of electricity as a service is generally seen as a comment (k) exception driven by public policy, rather than any reasoned determination that electricity is too intangible to be a product.\footnote{RESTATEMENT (SECOND) OF TORTS § 402A Comment k (recognizing an exception to strict liability for “unavoidably unsafe” products).} Unless there is some independent public policy reason to exempt software from being subject to products liability, these states should also recognize that software is sufficiently tangible to be a product.

2. Software Distribution and Use

Even if software is considered to be intangible, the circumstances of its distribution and use dictate that it be considered a product. Software is mass produced like a product, sold individually like a product, and is clearly a product from the consumer’s perspective.

a. Generally

Companies produce software by employing a team of developers to work on the project until the software is done. The software is developed before individual customers have been identified (although the company will probably do some market testing and identify its niche, it will not produce software to an individual buyer’s specifications). Once completed, copies of the software are distributed to those who are willing to pay for it.

Once a software project is complete, many copies of the software are produced and distributed, either to retail outlets or over the Internet.\footnote{See generally Dirk Van den Poel & Joseph Leunis, Consumer Acceptance of the Internet as a Channel of Distribution, 45 JOURNAL OF BUSINESS RESEARCH 249 (1999).} Every consumer purchases the same product as every other consumer, and the product very rarely includes any service beyond technical support.
The mass production and distribution of software is sufficiently analogous to that of more tangible products as to clearly qualify it for treatment as a product under the standard described by the Restatement (Third). It has none of the qualities of customization and personal attention that characterize services.

b. Licensing v. Selling

Another argument against treating software as a product is that software is licensed rather than sold, apparently on a theory that consumer products are not delivered through licensing, while services are. However, the software license looks, from a consumer’s perspective, more like a sale of goods than a negotiated service contract. The consumer is presented with the opportunity to buy (or download for free) a particular product on a particular set of terms. He does not enjoy any sort of personalized customization of the license or of the product provided.

While it is true that the consumer who purchases software does not get the right to do whatever he wants with it (most notably, most software is licensed with a prohibition on copying and modification), this limitation on the consumer is not enough by itself to exempt the manufacturer from liability for a defective product. There is no general requirement that the consumer have a particular set of rights regarding the product that injures him in order to recover. A consumer who purchases an item that is protected by a patent is forbidden from copying the product, but can still recover from the manufacturer if it injures him. An individual can be injured by a product that does not belong to him, or that is being used by another. In either case, the injured person can recover from the manufacturer despite lacking ownership or possession of the product at the time of the injury. The fact that a software consumer has only purchased a limited array of rights from the manufacturer does not have any bearing on whether the manufacturer should be held liable for injuries caused by the product, and is only tangentially related to the question of whether software is a product.

106 See RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 19 ("[intangible] items… are products when the context of their distribution and use is sufficiently analogous to the distribution and use of tangible personal property that it is appropriate to apply the rules stated in this Restatement").
107 This argument is discussed in Wolpert, supra note __, at 520.
109 Matthews v. Lawnlite Co., 88 So.2d 299, 300-01 (Fla. 1956) (allowing recovery for a store patron whose finger was snipped off by an in-store display model of a rocking chair).
c. Software that is Given Away

Products that are distributed non-commercially are not subject to products liability. Accordingly, injuries caused by most software that is given away for free should not be subject to recovery under a products liability theory. However, there are many methods of software distribution that are commercial in nature, though the software is given away to the end user. Spyware distribution is an example of this—while the software is given away for free, the manufacturer makes money by charging for ad space or by selling the user’s private information. The commercial purpose of the transaction is enough to trigger products liability, in the same way that a product manufacturer is liable if it distributes free samples that turn out to be defective.

B. Defect

Establishing defect for software in general is a complicated issue that will be discussed further in Part III of this article. For spyware in particular, the question of defect is relatively straightforward, since there is no cost to remediing the most common defect found in spyware.

The most stringent test of defect commonly in use is the risk-utility test. The risk utility test weighs the risk created by a particular design feature against the utility of having the feature (for example, if a car’s airbag were alleged to be defectively designed, we would weigh the cost of modifying the airbag against the risk created by its current design). The test is equivalent to the test for negligence first laid out by Learned Hand, which asks if the burden that must be incurred to avoid a risk (B) is less than the danger posed by the risk (the probability P of the harm coming to pass, multiplied by the loss L if the harm occurs).

Spyware programs as a whole may provide some utility to the consumer. However, the risk-utility analysis focuses not on the product as a whole, but on the particular feature alleged to be defective. For spyware,
the most obvious design defect is that the program actively avoids removal. The utility of this feature to the end user is zero, since it only exists to thwart the user’s desires. The risk the consumer faces is the inability to configure and use his computer as he sees fit. It may be that the potential harm caused by spyware is not compensable by the tort system, so that the risk the consumer faces also counts as zero, and the product is not defective. However, if there is any recognizable harm at all, the spyware must be considered defective. Accordingly, for spyware the outcome of the defect analysis is controlled by the analysis of the harm caused by the program, and does not require further independent consideration.

C. Harm

Generally, consumers can not recover for harms caused by defective products when the harms are purely economic in nature. For example, if an assembly line breaks down and forces a factory to shut down, the factory owner can not recover his lost profits from the assembly line manufacturer. Nor can consumers recover for product failure that only results in damage to the product itself. These sorts of losses are a result of disappointed consumer expectations, the core concern of contract law, and the restrictions on recovery prevent contracts from “drown[ing] in a sea of tort.” Since spyware does not cause physical injury to the consumer, this rule presents a serious obstacle to recovery.

However, while plaintiffs can not recover for loss that is purely economic in nature, they can recover for losses that don’t involve personal injuries if the defective product has damaged some other product owned by the plaintiff. The damage must be physical in nature: in the example above, the defective assembly line might “damage” the factory by reducing its overall value, but such economic harm does not suffice for the other product exception. However, if some defect in the assembly line caused it to catch fire and the fire did some damage to the factory, such damage would fall under the other product exception.

Since spyware doesn’t cause physical injury to the consumer, the only way to apply products liability to spyware is if it does damage to an “other product” owned by the consumer. This section will show that, if we apply alternative design would, at reasonable cost, have reduced the foreseeable risks of harm posed by the product…”}

---

120 See America Online, Inc. v. St. Paul Mercury Insurance Co. 347 F.3d 89, 96 (4th Cir. 2003) (finding that damage to software is not physical damage within the meaning of the term as used in an insurance contract).
121 RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 21.
122 This rule was first laid out in the admiralty case East River S.S. Corp. v. Transamerica Delaval Inc., 476 U.S. 858 (1986), and has been widely followed by state courts.
123 RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 21 cmt. d.
124 Id. § 21 cmt. e.
traditional tort principles to determine what counts as harm to another product, the damage that spyware inflicts on personal computers should be treated as such harm.

1. Pure Economic Loss

Products liability is generally concerned with protecting consumers. Protecting them from personal injury is covered by allowing recovery for personal injuries caused by defective products. Note that this is true whether we regard consumer protection as ultimately serving the role of compensation or of deterrence—changing our view of the goal of products liability might change the test we apply to determine when a product is defective, but only changes in our view of what injuries the consumer should be protected against change the types of harm that trigger recovery. Allowing recovery for anything besides personal injury must be motivated by a desire to protect consumers from that sort of injury, and can not be justified by referring back to the general goal of protecting consumers from personal harm.

The pure economic loss rule inevitably follows from the conclusion that consumers should not be protected in tort from the economic harm of owning a product that fails to perform as expected. This lack of protection is consistent with most torts law—consumer disappointment with a failure of a product to work is essentially disappointment that the consumer is not getting the benefit of his bargain. Such a concern is appropriately addressed in a contract law setting.

The general rule is that if a consumer is unhappy because he was harmed by a product, he can recover in tort. If he is unhappy because the product does not work, he can only recover in contract. Sitting in between products that harm the consumer and products that only harm themselves are products that harm other property owned by the consumer. As it turns out, the consumer can recover for the harms caused by these “in between” products, as well. The next section will explore the source and extent of this protection, which are vital for determining whether spyware causes damage cognizable under product liability.

---

127 Id.
128 East River, 476 U.S. at 868.
129 Id. at 871.
130 Id.
131 Id.
2. The Other Product Exception

Determining the scope of the other product exception is essential to determining whether spyware gives rise to products liability. Spyware resides on a computer and can not physically injure a consumer. It can only cause tortious harm if the effect it has on personal computers constitutes damage to an “other product” under the other product exception.

As discussed above, even when the consumer is not harmed, he may still recover in products liability if another product he owns is harmed by a defective product. There are several possible explanations for this exception, each with important consequences for potential product liability for spyware. First, it is possible that the other product exception simply serves as a proxy for physical risk. As will be seen below, this does not appear likely; the exception is more plausibly explained as providing protection for consumers’ interests in the security of their property.

There are two classes of property interests that are generally protected in tort law: one is those that are so identified with the consumer himself that attacking them is analogous to battery, while the other is the general interest in the security of property that is protected by trespass to chattels and negligence liability. Of these justifications, the general protection of property interests best fits existing doctrine.

Other possible rationales for the exception are the magnitude of losses caused by damage to other products, or an informational gap between consumers and manufacturers that requires that consumers be protected. Neither holds up to scrutiny.

The other product exception can not be justified by the magnitude of the losses in question. Purely economic harms can be enormous (an assembly line break down can shut down an entire factory, even though the only product physically damaged is the assembly line itself), while damages that easily qualify as harms to other products (and thus are compensable) can be quite trivial (e.g., a defective lawnmower blade that breaks the living room window).

In addition, there is no clear information gap here—while the manufacturer might know more about the dangers attending the use of its product, the consumer knows better than the manufacturer whether the

---

133 Id.
134 A similar line of argument is that the proper role of torts law is to regulate calamitous events, which led to the “sudden accident” exception to the economic loss rule. See, e.g. Moorman Mfg. Co. v. National Tank Co., 91 Ill.2d 69, 86 (1982) (“Tort theory is appropriately suited for personal injury or property damage resulting from a sudden or dangerous occurrence…”). Courts have since largely moved away from examining of the manner in which the injury occurred to examining what has been damaged. See, e.g. Trans States Airlines v. Pratt & Whitney Canada, Inc. 177 Ill.2d 21, 42 (1997) (“[W]e answer the inquiry as to whether there may be tort recovery for damage to a single product resulting from a sudden and calamitous event in the negative.”).
135 See GEISTFELD, supra note 126 at 202.
product will be used around other products that could be damaged by some malfunction. As there is no clear information gap, there can be no justification for a tort duty predicated on such a gap.

The remainder of this section will discuss the three plausible lines of reasoning that could justify the other product exception to the pure economic loss rule.

i. Physical Damage as Proxy for Personal Risk

One way of looking at damages to other products is as a proxy for physically endangered consumers. To take a hypothetical example, a defective lawnmower blade that snaps off of a lawnmower and smashes through a living room window creates liability for damage to an other product. Now, the broken window might not be a big deal, but the nature of the damage does indicate that a piece of metal was flying through the air that could have injured the consumer. It probably is true that product malfunctions that result in physical damage to another product are on average more dangerous than product malfunctions that do not. However, the argument that plaintiffs can recover for damages to other property because damage to other property serves as a proxy for physical risk to the consumer suffers from two infirmities: first, damage to other products is not a reliable proxy for physical risk (being both over and under inclusive), and second, consumers generally can not recover absent damage to themselves or other property even when they prove that they have been exposed to physical risk.

Using damage to other products as a proxy for an endangered consumer is over-inclusive because there are many ways in which a product can be defective and damage another product without endangering the consumer. Many products can malfunction while the consumer is not even present. Consider a defective coffee maker that malfunctions while the consumer is in another room, allowing boiling hot water to escape onto the kitchen counter and destroy all manner of small appliances. The consumer was in no physical danger, but will be able to recover damages to compensate him for his damaged possessions.

Damage to other products is an under-inclusive proxy for physical risk

---

136 This appears to be the theory underlining the “calamitous event” line of cases. See, e.g. Moorman Mfg. Co. v. National Tank Co., 91 Ill.2d 69, 86 (1982) (“Tort theory is appropriately suited for personal injury or property damage resulting from a sudden or dangerous occurrence...”).

137 See, e.g. Pfizer, Inc. v. Farsian, 682 So.2d 405 (Ala.1996) (holding that plaintiff could not recover despite receiving a defective heart valve because the plaintiff’s heart valve had not actually failed). The exception is in the minority of states that allow recovery for “calamitous failure”; however, few such states remain. See RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 21 cmt. d.

138 RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 21.
because it is quite common for a consumer to be physically endangered by a defective product without suffering damage to himself or to another product. Consider Pfizer, Inc. v. Farsian. Mr. Farsian underwent surgery that involved having a heart valve installed in order to regulate the flow of blood through his heart. The manufacturer discovered that the valve was defectively designed, and there was a chance that it would fail catastrophically and kill or injure Mr. Farsian. He was physically endangered by the product, but because he had suffered no physical harm (to himself or to anything else that he owned), he was unable to recover.

The Farsian case illustrates the second problem with the proxy argument, which is that plaintiffs who are merely endangered by defective products are not able to recover damages. While a plausible argument could be made that they should be able to recover for such damages, the fact that they can not—and that they can recover for damages to other products—suggests that some other rationale must underlie the other product exception to the economic loss rule.

ii. Security Interest Analogous to Battery

Some products are so closely tied up with a person’s identity that striking the products is equivalent to striking the person. For example, knocking a cane out of a person’s hand is offensive battery, even if the defendant does not strike the person himself. One possible rationale for the other products rule is that it simply extends this protection into the realm of products liability.

However, if this were the rationale then we would expect the other product exception to be drawn much more narrowly than it actually is. To take one example, when a boat caught fire and sank, destroying fishing nets that the owner had attached to the boat, the owner was able to recover for the value of the lost nets. Fishing nets are not an item so bound up in the owner’s personality that an assault on them would result in an offensive battery. Similarly, a consumer was able to recover when their coin collection was stolen out of a defective safe; while a coin collection may

---

139 682 So.2d 405.
140 In that there was an unacceptably high risk of a strut fracture occurring. Id. at 406.
141 Two thirds of strut fracture incidents lead to death. Id. at 406.
142 Id. at 407.
143 Id.
144 See GEISTFELD, supra note 126, at 201.
145 See RESTATEMENT (SECOND) OF TORTS § 18 (c)
146 Respublica v. De Longchamps, 1 U.S. 111 (Pa. 1784). Similar behavior was sanctioned in Clark v. Downing, 55 Vt. 259 (1882) (striking horse that plaintiff was riding).
be valuable, it is not particularly bound up in its owner’s personality.

The relatively broad applicability of the other products rule suggests that some other rationale must underlie the rule; while the analogy to battery supports some of the instances of the rule, it can’t possibly support all (or even most) of the applications of the rule.

### iii. Security Interest in Property

Personal interests in private property have long been protected by the actions of trespass to chattels and conversion. Such diverse actions as altering silverware and interfering with a mooring buoy have given rise to actions for trespass to chattels. Even where the parties have a contractual relationship, exceeding the scope of consent to physically modify a chattel owned by can give rise to a tort claim of trespass to chattels or conversion, not merely to a breach of contract. Plaintiffs in trespass to chattels and conversion cases can recover for all damages flowing from the intentional tort, and are not limited to the value of the chattel that was interfered with.

Negligent damage to personal property has also long been compensable through the torts system. Plaintiffs can not only recover for repair costs, but can also recover for damages flowing from the loss of the property, such as the costs of obtaining a temporary replacement. Once they have established the physical damage to their property, they can even recover for lost business that occurred as a consequence of the damage.

This historic concern with protecting a security interest in chattels justifies the other product exception to the economic loss rule in products liability. While plaintiffs have rarely been able to recover in tort for disappointed expectations or non-physical harms, they have always been able to recover for damaged property. When a consumer’s property is damaged because of a manufacturer’s negligence, it doesn’t matter whether the negligence damaged the property directly, or whether the negligence

---

1. 1174 (4th Cir. 1997).
2. 149 Both actions allow for recovery for damages flowing from an intentional interference with a property right. The distinction between the two is mainly a matter of degree- trespass to chattels becomes conversion when the interference is so great as to require the defendant to pay for the full value of the property interfered with. See Restatement (Second) of Torts §216-17 (describing trespass to chattels), Restatement (Second) of Torts §222A (defining conversion).
8. 155 Schuester, 209 S.W. at 542, Reynolds, 345 P.2d at 928.
resulted in a defective product that then damaged the consumer’s property. In either case, tort law protects the plaintiff’s security interest in his possessions.

Another way of thinking about the doctrine is that the protected interests match up with the tort protection that would exist if the product were a person hired to do a job. That is, actions that if done by a hired worker would lead to contract liability (such as failure to perform) do not create damages for the purposes of product liability, while actions that would lead to tort liability if done by a hired worker (such as physically damaging some other product) do create damages for the purposes of product liability. It makes sense that the doctrine falls out this way, because the product is in a sense simply an instrument through which the manufacturer may commit a tort against the consumer.

Accordingly, determining the extent of the protection provided by product liability requires that we explore the protection generally provided by tort law to digital products.

3. Applying the Theory

Because software defects generally do not cause personal injuries, they often have been held to cause harm that falls in the category of pure economic loss. However, even without causing physical injuries, the damage caused by spyware could be compensable as long as it fits into the “other property” exception to the economic loss rule. The important question in deciding a software products liability claim is whether the damage caused by the software can be accurately characterized as damage to other products.

The foregoing analysis leaves us with some difficulty in applying the rule of damage to “other products” to software. For example, at first glance, a software program that wipes out a user’s hard drive has clearly damaged another product. The program that deleted all of the other programs was one product, while the programs that were deleted were another product. It looks like a straightforward application of the doctrine is all that is needed to establish liability. However, recall the requirement that a product not only harm another product, but that it physically harm the other product. We are left with the following question: when program A deletes program B, is the damage done more like the assembly line that stops working (reducing the value of other products around it), or is it more

---

157 See, e.g., Gus’ Catering Inc. v. Menusoft Systems, 762 A.2d 804, 807 (Vt. 2000) (holding that loss of business profits, customers, and time are economic losses, not physical harms, and therefore not recoverable under negligence theory), Hou-Tex, Inc. v. Landmark Graphics, 26 S.W.3d 103 (Tex. 2000), America Online, Inc. v. St. Paul Mercury Insurance Co. 347 F.3d 89, 96 (4th Cir. 2003) (finding that damage to software is not physical damage within the meaning of the term as used in an insurance contract).
like the assembly line that catches fire and burns products placed nearby? This sort of question can lead to a “battle of analogies” that is not terribly helpful in arriving at a satisfactory answer. However, in applying product liability to digital products, we may be able to find help in other cases applying tort liability to digital products.

Products liability generally uses the same definition for damages as does tort law generally—as discussed above, the “other products” exception largely tracks the interests that are protected by the tort of trespass to chattels (and protected from negligent behavior). While it is difficult to find cases involving electronic damage caused by negligence, courts have recently extended trespass to chattels to the electronic world. We can use the shape of that doctrine as a description of the shape of the interests protected by product liability involving software.

a. Trespass to Chattels in the Digital Context

Trespass to chattels has been applied to cover telephone networks and, more recently, computers that are used as Internet servers. The common thread through all of these cases is that actions that interfere with the owner’s ability to use his physical property create liability, even if the defendant only directly affects software.

The first major case involving an electronic trespass to chattels was Thrifty-Tel v. Bezenek. In that case, two children obtained an access code which, when used with a separate authorization code, would allow them to place free long-distance calls over Thrifty-Tel’s network. They then employed various methods in an attempt to discover an authorization code, culminating in the use of a computer and automated software to dial over one thousand three hundred potential codes in a short span of time. This last attempt overburdened Thrifty-Tel’s system, denying some subscribers access to phone lines.

The court in Bezenek determined that trespass to chattels was the appropriate cause of action, as the Bezenek’s had interfered with Thrifty-Tel’s possession of its physical property. The court required, however,
that Thrifty-Tel prove the actual damages caused to it by the trespass.167

The next major case involving electronic trespass to chattels was \textit{CompuServe Inc. v. Cyber Promotions, Inc.}168 This case was brought by CompuServe in an attempt to enjoin a company from continuing to send spam email to CompuServe customers.169 The court held that the defendant’s behavior was trespass to chattels, as it interfered with CompuServe’s ability to use the computers to serve its subscriber base.170

The scope of trespass to chattels was broadened further in \textit{eBay, Inc. v. Bidder’s Edge, Inc.}171 Here, eBay sued to enjoin an auction aggregator from using an automated web crawler to crawl eBay’s site.172 Although the automated crawler conducted many more searches per minute than could a human user, it still never amounted to more than two percent of the requests received by eBay during the time in question.173 While the court acknowledged that Bidder’s Edge’s activity did not amount to serious harm to eBay, it granted relief on the theory that failing to do so would expose eBay to harm caused by other auction aggregators who would be encouraged to engage in similar behavior.174

There followed two cases that restricted the scope of trespass to chattels. First, \textit{Ticketmaster Corp. v. Tickets.com, Inc.} rejected the issuance of an injunction, finding that there was no reason to believe that there was a danger of dozens of parasites joining the fray.175 Then, in \textit{Intel Corp. v. Hamidi},176 Intel sought to enjoin a former employee from sending email critical of the company to current employees through the Intel email system.177 The California Supreme Court refused to extend trespass to chattels to encompass “an electronic communication that neither damages the recipient computer system nor impairs its functioning.”178

These cases establish that in order to recover for an electronic trespass to chattels, the plaintiff must show that there is something that he wanted to do with his chattel that he was prevented from doing by the defendant’s actions, or that he could potentially be prevented from doing if actions like those done by the defendant are permitted.179 To be consistent with this standard in the context of a products liability suit involving a personal

\begin{itemize}
  \item \textit{Id.} at 474-75.
  \item \textit{Id.} at 1015 (S.D.Ohio,1997).
  \item \textit{Id.} at 1021.
  \item \textit{Id.} at 1022 (“any value CompuServe realizes from its computer equipment is wholly derived from the extent to which that equipment can serve its subscriber base”).
  \item 100 F.Supp.2d 1058 (N.D.Cal. 2000). 
  \item \textit{Id.} at 1062-63.
  \item \textit{Id.} at 1063.
  \item \textit{Id.} at 1066.
  \item 30 Cal.4th 1342 (2003).
  \item \textit{Id.} at 1346-47.
  \item \textit{Id.} at 1347.
\end{itemize}
computer, the plaintiff should have to show that the program substantially impaired his ability to use his computer (or some other software program) as he saw fit, causing him actual damage.180

The following sections will evaluate several different types of damage that computers might suffer using the test derived from the trespass to chattels cases. It will start by examining the damage done by spyware of resisting removal from a computer. They will then turn to file erasure, crashing of the entire computer, and crashing of the program itself.

b. Damage: Resisting Removal

As discussed above, most spyware resists removal by the end user.181 It accomplishes this through various technical means, none of which serve any purpose other than to thwart the will of the user.182

Such behavior is closely analogous to the trespass to chattels cases where parties engage in a consensual transaction, but one party exceeds the boundaries set by the property owner. While spyware arrives on the computer through some form of consumer consent (whether actual consent, or constructive consent based on the content of the click through license), remaining on the computer after the user tries to remove it clearly exceeds the boundaries of any such consent.

The most closely analogous cases involving digital trespass are the cases involving email.183 In these cases, companies wished to stop individuals from sending email to people using the companies’ email servers. While the companies generally held out their email servers for public use, they later withdrew permission to use the servers from the defendants. The courts split in their resolution of the cases: one held that any use of the plaintiff’s email servers without permission was trespass to chattels, while the other held that the imposition was so slight as to not rise to the level of trespass to chattels.

The email cases show that electronically exceeding the boundaries set by a property owner gives rise to liability. Where the cases diverged was in their evaluation of the seriousness of the harm. One court held that the plaintiff was prevented (or potentially prevented) from doing something he wanted to do by the defendant’s actions, while the other court held that he

---

180 Some support for this can be seen in Sotelo v. DirectRevenue, LLC, 384 F. Supp. 2d 1219, 1229–33 (N.D. Ill. 2005). There, plaintiff alleged that the installation of spyware was itself trespass to chattels. Though the case ultimately settled, the court did refuse to dismiss the case for failure to state a claim.

181 See Sudhindra Shakla & Fiona Fui-Hoon Nah, Web Browsing and Spyware Intrusion, 48 COMMUNICATIONS OF THE ACM 85, 87 (2005) (“Removal violation occurs when there is either no uninstaller provided for the software (spyware) or the uninstaller is nonfunctional. The uninstaller may be poorly coded or could be a buggy uninstaller”)

182 See id.

was not. Here, the spyware is clearly preventing the user from doing something that he wants to do. It is doing something (whether displaying advertising or tracking the user’s behavior) that is sufficiently troubling to the consumer to cause him to attempt to remove it specifically from his computer. The spyware then prevents him from accomplishing the desired removal. While the mere use of some processing resources may not rise to the level of interference with the user’s enjoyment of his property, active interference with the use of a personal computer certainly should do so.

c. Damage: Erased Files

The most clear cut form of damage that software can do to a computer is to erase files stored on the user’s hard drive. This sort of dramatic damage gave rise to the highest profile case attempting to categorize damage caused by software: *America Online, Inc. v. St. Paul Mercury Insurance Co.*

The court in that case ultimately held that file erasure is not tangible damage; an analysis of the court’s reasoning will show where it diverged from standard tort principles and why file erasure should be considered damage to an other product giving rise to product liability.

The case arose after the distribution of AOL’s Version 5.0 software. Many of AOL’s customers filed suit, alleging in short “that Version 5.0 altered the plaintiffs’ existing software, disrupted their network connections, caused the loss of stored data, and caused their operating systems to crash.” Any single one of these disruptions likely qualifies as an interference with the plaintiffs’ use of their property under the electronic trespass to chattel cases, but I focus here (as did the court) on the loss of the stored data and the damage done to other stored data.

AOL tendered the defense of the class action suits to its insurance company, St. Paul and Underwriters and Lloyd’s of London. St Paul denied coverage, stating that the claims “do not allege damage to ‘tangible’ property and are not property damage as defined by the St. Paul [commercial general liability] policy.” AOL promptly sued St. Paul, seeking a declaratory judgment that St. Paul was obligated to defend and indemnify AOL.

The court ruled in favor of St. Paul, holding that the underlying plaintiffs did not allege any physical damage to tangible property. While the court described the physical magnetic material on a computer’s hard drive...
drive that is used to store software as being tangible, it stated that the particular arrangement of information on the hard drive is not.\textsuperscript{190} It reasoned that:

By analogy, when the combination to a combination lock is forgotten or changed, the lock becomes useless, but the lock is not physically damaged. With the retrieval or resetting of the combination—the idea—the lock can be used again. This loss or alteration of the combination may be a useful metaphor for damage to software and data in a computer. With damage to software, whether it be by reconfiguration or loss of instructions, the computer may become inoperable. But the hardware is not damaged. The switches continue to function to receive instructions and the data and information developed on the computer can still be preserved on the hard drive. While the loss of the idea represented by the configuration of the computer switches or the combination for the lock might amount to damage, such damage is damage to intangible property. It is not damage to the physical components of the computer or the lock, i.e., to those components that have “physical substance apparent to the senses.”\textsuperscript{191}

Alteration of a combination lock is indeed an apt analogy to modifications in software. However, what the court fails to note in its opinion is that tampering with a combination lock that belongs to another person is a tort. Physically altering the property of another in a way that makes it less valuable is the very definition of trespass to chattels.\textsuperscript{192} As discussed above, damage that would be trespass to chattels if done directly generally trigger products liability when done indirectly.\textsuperscript{193} A hypothetical product that as an avoidable side effect of its operation altered the combination of nearby combination locks would be defective, and the damage done by that product could be recovered through product liability.

The court may have been constrained to rule as it did because it was interpreting the word “tangible” as used in a particular contract, and not attempting to define the appropriate scope of products liability doctrine.\textsuperscript{194} While the court’s holding is against treating damage to software as tangible injury, its reasoning provides a powerful argument for holding software manufacturers liable for damage that their software does to other software.

d. Damage: Crashing Computer

Some defective programs can cause personal computers to crash. That is, they will cause the computer to shut down immediately, but do not inflict any lasting harm on the computer once it has been restarted.

\textsuperscript{190} Id. at 95.
\textsuperscript{191} Id. at 96.
\textsuperscript{192} See RESTATEMENT (SECOND) OF TORTS § 217 (b).
\textsuperscript{193} See supra notes 133-56 and accompanying text.
\textsuperscript{194} See id. at 92-93.
Causing a computer to crash will interrupt the use of any other programs on the computer. It interferes with the user’s ability to use the computer, and clearly rises to the level of trespass to chattels (just as it would be trespass to chattels for some other person to unplug the computer without the user’s permission).

Recognizing this sort of harm as being compensable under product liability may raise concerns about overly expansive tort liability. However, several mitigating factors should prevent liability for causing crashes from getting out of hand. First of all, the damaged user must prove defect. Since few (if any) manufacturers will deliberately design their programs to crash, the user will have to establish that the manufacturer should have done something to prevent the crash that he didn’t do. This represents a substantial technical challenge and, as discussed in Section III, could present a difficult legal challenge as well. In addition, even if the user proves the software is defective (or a strict liability standard is applied that excuses him from presenting such proof), he still has to prove causation. It will not be easy to prove exactly which piece of software caused a crash, and the task will be made more difficult by the fact that it is hard to even present evidence regarding computer crashes, as they tend to be somewhat random in nature. Especially if the crash results from some combination of unstable software, it will be challenging to isolate which product is defective and the “true” cause of the crash. The unpredictable nature of such crashes could also pose substantial obstacles to any attempt to bring a class action suit over crashing software because of the procedural requirements regarding the similarities that must exist between class members.

e. Damage: Crashing Program

Another common result of poorly written software is that it crashes itself only. That is, the program itself ceases functioning immediately, but it does not affect any of the other programs running at the same time, and it does not have any lingering effect on other programs or on the computer itself.

The lack of effect on other programs prevents this sort of harm from fitting under the other products exception. It is an example of a product simply failing to work or, at worst, inflicting physical harm on itself only. It can not support a product liability suit.

---

195 See infra notes 208-44 and accompanying text.
196 The erratic behavior of crashing software would particularly cause difficulties with the requirement that the questions of law and fact be “common to the class”. FRCP 21 (a).
f. Effect of Relatively Easy Repairs

Damage to software is different from most physical damage in that it is relatively easy to repair, and that it is possible to repair completely. This ease of repair creates the temptation to argue against liability for things like an erased hard drive because the plaintiff can simply reinstall the software that was deleted. On this theory, the only real damage to the plaintiff is the time that he has to spend reinstalling the software, a classic example of pure economic loss. Ultimately this theory does not hold up, as the fact that a consumer is able to repair the harm caused by the defective product doesn’t change the fact that the manufacturer has committed a tortious act and should have to compensate the consumer.

The argument appears most plausible in the case where no unique files are stored on a computer, and the user has kept backup copies of his software and software serial numbers. At this point, whatever damage defective software has done to the computer can be fixed by reformatting and reinstalling. Reinstalling the erased files will be relatively straightforward and not consume too much of the user’s time. However, accepting that the availability of some sort of time for repair tradeoff prevents something from being considered compensable damage will lead to clearly unacceptable results.

Consider a computer where the only unique file stored on the computer was a spreadsheet created by the consumer. The deleted file was unique, and can not be replaced through a simple reinstallation procedure. However, the file can be replaced if the creator simply goes through the same steps that went in to creating it in the first place. Doing this is a drain on his time, but once he puts enough time in he will be in exactly the same situation as he was before the harm occurred.

Take the argument a step further. Suppose that D hits P’s car repeatedly with a baseball bat. While P is not capable of removing the dents himself, his mechanic can restore the car to its original state. Paying the mechanic requires P to work, costing him time, but ultimately restoring his product to its undamaged state.

As I believe this makes clear, even if P is able to repair the damage caused by D, D still caused the damage in the first place, and should have to pay for its repair. The security interest in most property is ultimately economic, as the property can simply be replaced for a price. However, we do not limit tort protection to irreplaceable items.

The point of protecting the security interest in property is to establish that the damaged party should not have to bear the costs of repairing damage negligently or intentionally caused by another. Ease of repair may reduce the compensation required after damage has occurred, but it does not
change the fact that the damage was inflicted tortiously. Whether the cost of repair involves paying somebody else to do it or taking the time to do it himself, the plaintiff should be compensated for having had to repair the product.

4. Calculating Damages

Once we conclude that spyware has caused a compensable harm to the consumer, there remains the problem of calculating the amount of damages required to be paid to make the plaintiff whole.\[197\]

Spyware invades the consumer’s privacy and impede the user’s ability to use his computer as he sees fit.\[198\] Also, programs that pop up advertisements impede the consumer’s ability to use the Internet, a major source of value for most computer users (particularly those that have spyware—a product only distributed via the Internet—installed).\[199\] The aggressive measures that most spyware programs take to resist removal prevent the user from configuring his computer as he wants.\[200\]

It may be difficult to establish the loss of time and loss of privacy as harms to the consumer. The spyware manufacturer would characterize these losses as the price paid by the consumer in exchange for the benefit that the spyware program provided.\[201\] While this argument may seem implausible for the many users that are unaware of the bargain they are entering into when they approve a clickwrap agreement, it is hard to argue that this sort of time for money tradeoff is one that should be foreclosed by the tort system.\[202\]

The resistance to removal is a different story. There is no plausible reason for a consumer to agree to install a program that makes itself almost impossible to remove. Particularly when the program has the potential to become highly aggravating and intrusive, the consumer should retain the basic ability to configure his computer as he sees fit. Quantifying the damage done to the end user can be done by researching the rates charged by computer professionals to decontaminate spyware infested computers.\[203\]

---

197 See Restatement (Second) of Torts § 901 cmt. a (“[T]he law of torts attempts primarily to put an injured person in a position as nearly as possible equivalent to his position prior to the tort.”).


200 See Shukla & Nah, supra note 181, at 87, Qing Hu & Tamara Dinev, Is Spyware an Internet Nuisance or Public Menace?, 48 COMMUNICATIONS OF THE ACM 61, 62 (2005) (“Moreover, cleaning spyware from infected systems is not any easier than ridding computers of viruses... many spyware programs have a feature that will automatically reinstall if removed. It takes a special breed of anti-spyware software to do the job properly.”).

201 As the functioning of the turbines in East River was simply part of the sale contract. East River S.S. Corp. v. Transamerica Delaval Inc., 476 U.S. 858, 866 (1986)

202 To do otherwise risks drowning contract in a “sea of tort.” Id.

This measure of damages would likely result in a relatively small award.\footnote{See Mark B. Schmidt & Kirk P. Arnett, Spyware: A Little Knowledge is a Wonderful Thing, 48 Communications of the ACM 67, 68 (2005) (noting that software vendors have made a great start at providing “low-cost and uncomplicated tools for spyware detection and removal”).} However, the fact that each version of the program would have exactly the same defect and cause very similar damages make this case very suitable for a class action. A class action suit against a spyware manufacturer could result in substantial damages, considering the number of copies of spyware that are out there. The prospect of such a large damage award would probably discourage future companies from making software with the same defective feature.\footnote{Such deterrence would be consistent with the overall purpose of products liability. See GEISTFELD, supra note 126, at 201.}

Additionally, there is the possibility that consumers could recover from advertisers as well as from manufacturers and distributors. The advertiser’s role in spyware does not have a close analogue in traditional products liability cases. However, there is no denying the fact that advertisers have a major impact in how spyware is produced and distributed, and could exercise influence in how spyware is designed. These are the same basic policy concerns that underlie strict liability for non-manufacturing distributors.\footnote{See, e.g., Zamora v. Mobil Corp., 704 P.2d 584 (Wash. 1985).} If anything, these concerns are stronger in the spyware example than in the case of a retailer who, for example, sells a defective soda bottle. Accordingly, it is quite possible that if a products liability action can be brought over spyware, it can be brought against advertisers who utilize spyware to spread their message.

As this section has established, applying the standard tools of product liability analysis to software offers the benefit of changing manufacturer behavior without the need for Congressional intervention. It also offers a graduated series of incentives where manufacturers will be most severely punished for creating particularly malicious software, without the need for any individual or group to create an a priori ranking of maliciousness. Treating spyware as a defective product makes sense and would ease the lives of computer users everywhere.

III. SOFTWARE DEFECTIVENESS GENERALLY

The problem of proving software defect in cases that do not involve consciously malicious design choices raises some interesting questions. When a software manufacturer like AOL produces a new version of their software, what sort of steps do they have to take before they release it into the world? How much testing is enough? How many crashes does the reasonable consumer expect the software to cause? This section will
address these questions.

A. Obvious Defects

Consumers do not always have to identify a specific defect, or even the type of defect (manufacturing or design) that they are alleging. This is because in a large class of cases plaintiffs can use the malfunction doctrine to establish defect.\(^{207}\) The malfunction doctrine essentially applies *res ipsa loquitur* to products liability—if the consumer can show that the type of accident that occurred happens more often than not because of a defective product, they establish a presumption of defect.\(^{208}\)

Applying the malfunction doctrine to software will identify a clear set of defective products—products that have obvious, self-defeating malfunctions. For example, a computer game program that routinely crashes and erases a user’s entire hard drive under normal usage is clearly defective. However, other products will not be as clearly defective (e.g. an email reader that makes it easy for viruses to infect the user’s computer), and will require a more involved analysis of defect.

B. Classifying Defects

The first step in most analyses of defects is to determine their proper classification. Under the Restatement (Third) of Torts, defects can be classified as manufacturing,\(^{209}\) design,\(^{210}\) or warning defects.\(^{211}\) Given the ubiquity of exhaustive click through licenses,\(^{212}\) warning defects probably don’t apply with most software, which means that when something goes wrong the question is whether the alleged defect is one of design or manufacture. Under the Restatement (Third), manufacturing defects give rise to strict liability,\(^{213}\) while design defects are subject to a risk-utility test that is essentially a negligence standard.\(^{214}\) Therefore, deciding to treat a particular alleged defect as one of design or manufacture can have a major impact on the outcome of a case. Making such a determination requires some examination of how software is produced.

The software creation process can be broadly broken down into four steps: specification, programming, testing, and distribution.\(^{215}\) The

\(^{207}\) Restatement (Third) of Torts: Products Liability § 3.

\(^{208}\) Id.

\(^{209}\) Restatement (Third) of Torts: Products Liability § 2 (a).

\(^{210}\) Id. § 2 (b).

\(^{211}\) Id. § 2 (c).

\(^{212}\) See Stafford & Urbaczewski, *supra* note 3, at 298.

\(^{213}\) Id. § 2 (a).

\(^{214}\) Id. § 2 (b).

The specification step involves deciding what the program is going to do.\textsuperscript{216} The programming step involves the design and creation of the program code that will accomplish the tasks laid out in the specification.\textsuperscript{217} Testing involves going over the completed program to try to find problems.\textsuperscript{218} Distribution is where the completed program is copied and given to the end user (whether in physical form or over the internet).\textsuperscript{219}

Some defects are introduced at the specification stage. For example, many spyware programs are designed with the goal of resisting removal.\textsuperscript{220} They will secure copies of themselves throughout the system and replace themselves when the user tries to delete them.\textsuperscript{221} This is clearly a design decision that should be evaluated under the risk-utility test.

On the other hand, many problems are inadvertently introduced in the programming stage, and then not caught during the testing stage.\textsuperscript{222} These problems could plausibly be characterized as occurring in either the ‘manufacture’ or ‘design’ phase.

The distinction between design and manufacture are not as clear cut in software development as it is with most products. The work of a programmer, in particular, straddles the line between design and manufacture. While the programmer is constrained in his behavior by an overarching design, he has considerable freedom in deciding how to implement that design.

Some have made the argument that a programmer implementing a set of specifications is simply manufacturing a product, in the same way that a line worker putting together an automobile according to directions is manufacturing a car.\textsuperscript{223} Under this view, any defects that are introduced after the product specifications are complete are manufacturing defects, and thus subject to strict liability.\textsuperscript{224}

I believe a more appropriate analogy would be to limit the stage at which “manufacturing defects” may be introduced to those times when a new instance of the product is created. Such a definition seems to more closely track the traditional view of a manufacturing defect manifesting as a product that differs from the other products produced on the same assembly

\textsuperscript{216} See id. at 331.
\textsuperscript{217} See id. at 328-329.
\textsuperscript{218} See id. at 335.
\textsuperscript{219} See id. at 335.
\textsuperscript{220} See id. at 335.
\textsuperscript{221} See Shukla & Nah, supra note 181, at 87.
\textsuperscript{222} See, e.g., America Online, Inc. v. St. Paul Mercury Insurance Co., 347 F.3d 89 (4th Cir. 2003) (detailing how the software “altered the plaintiffs’ existing software, disrupted their network connections, caused the loss of stored data, and caused their operating systems to crash”; presumably such behavior was unintentional).
\textsuperscript{224} Under the standard set in RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 2.
line.  

Even if one believes that programmer errors are best classed as manufacturing defects, the risk-utility test still ultimately turns out to be the appropriate measure of liability. This section will explore why this is so.

The foregoing section assumed that we could easily identify errors, leaving only the question of whether the problems rise to the level of defect. With that assumption, a classification of programmer errors as manufacturing defects means that software producers are liable every time a programmer makes a mistake. However, it is not always obvious that this is the case. In fact, in order to argue that a programmer has made a mistake (and thus introduced a defect), one must go through a sort of risk-utility analysis. That is, we will often have to do a negligence analysis before we can apply strict liability.

For example, consider a hypothetical email reading program. The specification says that the program should be “secure.” However, the actual program as installed by the end user allows an email-delivered virus to destroy many important files on the user’s hard drive. What should the user have to prove in order to recover damages from the manufacturer?

1. Negligence Analysis Example

Under a negligence analysis, the user would have to show that the email program did not provide a reasonable level of safety. To do this, he would need to show that some expenditure by the company could have prevented the problem, and that the cost of this expenditure would be less than the total safety benefits that it would create.

This would require the user to show how the virus was able to harm his computer through the email program, and to demonstrate some modification to the email program that would have prevented the harm from occurring. He would then have to show that the manufacturer could have made this

\[\text{Id. \S 2 cmt. c ("a manufacturing defect is a departure from a product unit’s design specifications").} \]

\[\text{Id. \S 2 cmt. d ("the test is whether a reasonable alternative design would, at reasonable cost, have reduced the foreseeable risks of harm posed by the product").} \]

\[\text{Id.} \]
modification at a low enough cost that it would have been cost-benefit justified to have done so.\textsuperscript{229} This would require extensive expert witness testimony, as well as a lot of forensic programming analysis.

The user may well attempt to avoid this daunting task by arguing that strict liability should be applied.

2. Strict Liability Analysis Example

Under a strict liability theory, the user still has to show that the program was defective.\textsuperscript{230} The mere fact that the program has allowed him to come to harm does not establish defect—people are harmed by non-defective products all the time.\textsuperscript{231} In order to show that the program is defective, he has to show that it allowed him to come to harm that it should have prevented. That is, he will need to show that a reasonably secure program would have prevented his injury. The only logical way to establish this is to show some reasonable step that could have been taken to prevent the harm from occurring. Imposing such a requirement brings us back into the world of risk-utility analysis, requiring the same sort of proof discussed in the previous section.\textsuperscript{232}

3. Calibrating Liability

a. Potential for Too Much Liability

Even though a negligence style test (and the difficulties that it places in front of any prospective plaintiff) will likely be required to prove defect, application of the test could devolve into nearly automatic liability for defendants. The creation of a software program bears some resemblance to the decision on what warnings to include, and thus software products liability could give rise to problems similar to those created by warning defects.\textsuperscript{233} In particular, the alleged defect will generally involve changing some line or lines of code—meaning that the only expense involved in fixing the defect would be the cost of having the programmer make some changes in his text-editing tool of choice.

The cost of changing a few lines of code is almost nothing—even less than the “ink and paper” costs of including a warning with a product, because there are no reproduction costs involved. The only cost is a few

\textsuperscript{229} Id.
\textsuperscript{230} See GEISTFELD, supra note 126, at 128 (describing how a strict liability standard can evolve into a negligence test).
\textsuperscript{231} Id.
\textsuperscript{232} Id.
\textsuperscript{233} Id. at 156 (discussing warning defects).
minutes of some employee’s time. In addition, that negligible amount of
time is amortized over all the copies of the software that have been sold,
further reducing the magnitude of the risk required to lead to a conclusion
of defect. Thus, if the change would lead to any safety benefits at all, juries
would be likely to find for the plaintiff.234

Consider again the example of the email program that allows a virus to
infiltrate the user’s computer. Any particular virus will likely exploit a
security hole that is created by relatively few lines of code, and that can be
fixed using even fewer. Essentially the plaintiff will be able to ask the
defendant “Why didn’t you make this simple fix, which would have taken
thirty minutes to type in and saved countless users from having to deal with
this computer virus?”235 The defendant is in the same situation as the
defendant confronted with the question of “Why didn’t you warn me about
this problem, it would have cost pennies to add the warning and prevented
this horrific injury?”236 In either case, there are few compelling responses
available.

b. Potential for Too Little Liability

On the other hand, applying the negligence rationale to software
products could result in a situation where plaintiffs are almost never able to
recover.237 This is because any defect that plaintiffs identify could easily be
characterized as a failure to find a flaw in the product before shipping it to
the consumer. If this characterization is accepted, the negligence inquiry
turns on the question of whether the company devoted a reasonable level of
resources to catching the flaw in its product before shipment.

Returning to the purportedly insecure email program example, the
defendant might point out that the particular virus that caused harm to the
plaintiff is only one of many possible viruses, and that the line of code that
the plaintiff says should be changed is but one of thousands of lines of code
that could possibly contain security holes. Exhaustively checking each line
of code would be prohibitively costly in both time and money. The
defendant would argue that he took reasonable steps to secure the program,
but that even a reasonable security process won’t catch everything.

It is very difficult for the plaintiff to rebut this sort of testimony. He
would have to prove not only that the defendant should be investing more
time and effort in its quality control program, but also that doing so would

234 Id.
235 See, e.g., Thomas M. Chen, Trends in Viruses and Worms, 6 The Internet Protocol Journal, Number 3
detailing under the “Mass E-Mailers” heading how viruses were able to spread rapidly due to the faulty design of
Microsoft Outlook).
236 See GEISTFELD, supra note 126, at 156 (discussing the problems with warning defects doctrine).
237 Id. at 30 (discussing the rationale for Escola).
more likely than not have resulted in them catching the flaw that caused the plaintiff’s loss. This is a task that is very similar to what the plaintiff in an exploding Coke bottle case would have faced under a strict application of res ipsa loquitur, and one that most plaintiffs would find impossible to satisfy.

c. Reevaluating How Much is Too Much

If it were impossible as a practical matter for plaintiffs to win under a negligence standard, then software manufacturers would be effectively immunized from products liability for any defects introduced by programmers that do not satisfy the malfunction doctrine. This immunization would leave no legal incentive for manufacturers to observe a reasonable standard of care, even though they are ostensibly subject to a burden of reasonable care.

The way to provide them with an incentive to conform to a burden of reasonable care is to forbid arguments based on the “failure to catch defects” theory, even though it may lead to what is in effect strict liability for programmer errors. Such a rule forces a manufacturer to internalize all of the costs of the damage his software causes. This gives manufacturers an incentive to make expenditures such that the expenditure is less than the total damage that it averts. That is, their behavior would correspond exactly to the negligence standard that would be enforced if it were practical for plaintiffs to meet such a standard.

We might find such liability excessive or ineffective compared to an ideal negligence regime. However, it may well prove impossible to provide enough information to ever support recovery under a negligence standard. Given the choice between strict liability and no liability whatsoever, users are better off under a strict liability regime.

CONCLUSION

Applying products liability analysis to spyware reveals that it is clearly a defective product that should create liability to its manufacturer. Imposing such liability will discourage manufacturers from creating such

\[\text{footnotesystem}]

\footnotetext{238}{Id.}
\footnotetext{239}{See Escola v. Coca Cola Bottling Co. of Fresno, 150 P.2d 436, 441 (Cal. 1944) (Traynor, J. concurring) ("An injured person, however, is not ordinarily in a position to refute such evidence or identify the cause of the defect, for he can hardly be familiar with the manufacturing process as the manufacturer himself is.").}
\footnotetext{240}{See Geistfeld, supra note 126, at 30.}
\footnotetext{241}{Id.}
\footnotetext{242}{Id.}
\footnotetext{243}{Id.}
\footnotetext{244}{Id.}
products, and the potential for big damage awards in class action suits will ensure that this liability will be sought once it is generally realized to be available. Imposing liability in this way is much more flexible than trying to legislate the problem away, and rectifies the unfairness inherent in simply waiting for the market to solve the problem.

Having established that spyware is a defective product, we can turn to the software world more generally. We know from our analysis of spyware that software can be a product and can cause tortious harm—however, it is difficult to tell what how to go about the determination of defect. This article attempted to describe the principles that should inform any such determination. Given the plethora of buggy software in the world, such analysis should prove useful in the future.

***