

University of New South Wales
University of New South Wales Faculty of Law Research Series
2009

Year 2009

Paper 34

Information Security Standards

Meiring de Villiers*

*University of New South Wales

This working paper is hosted by The Berkeley Electronic Press (bepress) and may not be commercially reproduced without the permission of the copyright holder.

<http://law.bepress.com/unswwps-flrps09/art34>

Copyright ©2009 by the author.

Information Security Standards

Meiring de Villiers

Abstract

Businesses, non-profit organizations and government agencies may be held liable for failure to safeguard sensitive information in their possession. The threat of liability creates incentives to improve security standards, but uncertainty about the required standard and its judicial application may result in under- or overcompliance. Perfect security is neither possible nor the goal of tort law, but where does the law draw the line? This article analyzes the legal standard of information security that must be achieved to avoid liability. A numerical example illustrates its implementation.

INFORMATION SECURITY STANDARDS

MEIRING de VILLIERS

**University of New South Wales, School of Law
Sydney, NSW 2052
AUSTRALIA**

mdv@unsw.edu.au

ABSTRACT

Businesses, non-profit organizations and government agencies may be held liable for failure to safeguard sensitive information in their possession. The threat of liability creates incentives to improve security standards, but uncertainty about the required standard and its judicial application may result in under- or overcompliance. Perfect security is neither possible nor the goal of tort law, but where does the law draw the line? This article analyzes the legal standard of information security that must be achieved to avoid liability. A numerical example illustrates its implementation.

1. INTRODUCTION

Information is the lifeblood of modern society. Businesses, non-profit organizations and government agencies regularly compile and maintain electronic databases of information about individuals who interact with these institutions. Computerized data include contact information, personal histories, financial records and official identifiers such as social security numbers. This wealth of information allows business and government to operate more efficiently, but also exposes the persons to whom the information relates to risks such as identity theft, monetary losses, loss of intellectual property, loss of privacy and reputation, stalking and blackmail.¹

Database owners may be held liable for failure to safeguard the confidentiality, integrity, and availability of sensitive information in their possession. The threat of liability creates incentives to improve security standards, but uncertainty about the required standard and its judicial application may result in under- or overcompliance.² Perfect security is neither possible³ nor the goal of tort law,⁴ but where does the law draw the line? This article analyzes the legal standard of care in information security.

A victim of an information security breach may pursue a civil action under a negligence theory, the most widely used theory of liability in the law of torts. Negligence is generally defined as the failure to take reasonable precautions against a foreseeable risk of harm to another. Negligence may consist of an act, such as careless disposal of confidential customer information in a public place, or an omission, such as failing to remedy a known security vulnerability in a computer system.

¹ See e.g. *Remarks of J. Howard Beales, III, Director, Bureau of Consumer Protection, Federal Trade Commission, Before the 2003 Symposium on the Patriot Act, Consumer Privacy, and Cybercrime*, 5 N.C. J.L. & TECH. 1, 2.

² See e.g. John E. Calfee & Richard Craswell, *Some Effects of Uncertainty on Compliance With Legal Standards*, 70 VA. L. REV. 965, 965.

³ See e.g. Nicholas Weaver et al., *A Taxonomy of Computer Worms*, 2003 ACM Workshop on Rapid Malcode, Wash. D.C., p. 16 ("It may be tempting to say that we could build secure systems which will not have exploitable vulnerabilities. However, even highly secure software systems with reputations for robustness and which have received considerable security scrutiny including multiple code reviews ... have contained major security holes.")

⁴ See e.g. Michael L. Rustad and Thomas H. Koenig, *The Tort of Negligent Enablement of Cybercrime*, 20 BERKELEY TECH. L. J. 1553, 1598 ("[T]he overarching goal of tort law is to control the costs of accidents rather than to eliminate them.")

The plaintiff defines the standard of care in negligence law, by identifying and pleading an untaken precaution. The plaintiff must then prove that the precaution is cost-effective, and that the defendant's failure to take the precaution was the actual and proximate cause of real harm to the plaintiff. The untaken precaution forms the basis of the plaintiff's case, and defines the standard of care that the defendant, the court hearing the case, and perhaps a subsequent appellate court will use.

The due care standard in information security is an issue of law, economics and technology. The untaken precaution must have been technically capable of preventing the plaintiff's harm, if taken. The plaintiff typically selects among precautions such as a better firewall, an intrusion detection system or a virus scanner. The cost-effectiveness of the selected precaution is determined by balancing the benefits it provides in risk reduction against its cost. The sociology of human-machine interaction also plays a role in the liability analysis. The article argues that a defendant may be held liable for negligently enabling a cyber crime if the criminal can be characterized as a free radical. Free radicals, in this context, are individuals who are not deterred by the threat of tort or criminal liability, because they are shielded from liability by factors such as anonymity, insufficient assets, lack of mental capacity or lack of good judgment. Research shows that cyber wrongdoers generally fit the profile of a free radical.

The article is organized as follows. Section 2 introduces the principles of computer viruses and worms, the weapons of choice of cyber wrongdoers. It also discusses the most common virus detection technologies as potential untaken precautions. Section 3 analyzes the standard of due care in a negligence cause of action, and shows how it is determined in an information security environment. A numerical example illustrates the cost-benefit calculus of an untaken precaution. A final section concludes.

2. PRINCIPLES OF MALEVOLENT SOFTWARE

Malevolent software is a term for computer code that is designed to disrupt the operation of a computer system. The most common of these rogue programs are the computer virus and its common variant, the worm. Other forms of malicious software include so-called logic bombs,⁵ Trojan horses,⁶ and trap doors.⁷ Viruses and worms can be programmed to

⁵ A logic bomb is "a section of code, preprogrammed into a larger program, that waits for a trigger event to perform a harmful function. Logic bombs do not reproduce and are therefore not viral, but a virus may contain a logic bomb as a payload." Peter Szor, *THE ART OF COMPUTER VIRUS RESEARCH AND DEFENSE* (2005), at 30.

corrupt, delete or steal sensitive information. Malevolent code can also be exploited by cyber terrorists to disrupt elements of the national critical information infrastructure, such as banking, transportation, communications and energy provision systems, by attacking the computer systems at the heart of these structures.

COMPUTER VIRUSES

The term "virus," Latin for "poison," was first formally defined by Dr. Fred Cohen in 1983,⁸ even though the concept originated in John von Neumann's studies of self-replicating mathematical automata in the 1940s.⁹ A computer virus is a series of instructions (a program) that (i) infects a host program by attaching itself to the host, (ii) executes when the host is executed, and (iii) spreads by cloning itself, or part of itself, and attaching the new versions to other host programs. In addition, many viruses have a so-called payload capable of harmful side-effects, such as deleting, stealing or modifying digital information.¹⁰ As the definition suggests, a typical computer virus consists of three basic modules or mechanisms, namely an infection module, payload trigger, and payload.

Infection module

The infection mechanism is the most salient technical property of a computer virus.¹¹ It enables a virus to reproduce and spread, by locating a prospective host program, and installing a copy of the virus onto it.¹² When the host program runs, control is eventually

⁶ A Trojan horse is a program that appears to be beneficial, but contains a harmful payload. Peter Szor, *THE ART OF COMPUTER VIRUS RESEARCH AND DEFENSE* (2005), at 663.

⁷ A trapdoor, or backdoor, is a function built into a program or system to allow unauthorized access to the system. Peter Szor, *THE ART OF COMPUTER VIRUS RESEARCH AND DEFENSE* (2005), at 643.

⁸ FRED COHEN, *COMPUTER VIRUSES*. PhD dissertation, University of Southern California (1985).

⁹ See e.g. R. Lehtinen et al., *COMPUTER SECURITY BASICS* (O'Reilly, 2006), at 83.

¹⁰ See e.g. FREDERICK B. COHEN, *A SHORT COURSE ON COMPUTER VIRUSES* (Wiley, 1994, 2d ed.), at 1-2.

¹¹ LANCE J. HOFFMAN (ed.), *ROGUE PROGRAMS: VIRUSES, WORMS, TROJAN HORSES* (Van Nostrand Reinhold, 1990), at 247.

¹² See e.g. DOROTHY E. DENNING and PETER J. DENNING, *INTERNET BESIEGED* (ACM Press, New York, 1998), at 81.

passed to the resident virus code, allowing it to execute. The executing virus repeats the infection cycle by automatically replicating itself and copying the newly-created clones to other executable host files on the system or network, and even across networks.¹³

A virus may infect a computer or a network through several possible points of entry, including via a downloaded infected file, through web browsing, through removable media such as writable compact disks and DVDs, via infected files in shared directories, via an infected e-mail attachment, and even hidden in infected commercial shrinkwrapped software.¹⁴

E-mail is the most widely used medium of exchanging files and sharing information, but it has also become a convenient and efficient vehicle for virus and worm propagation.¹⁵ Fast-spreading viruses such as ExploreZip and Melissa, exploit automatic mailing programs to spread within and across networks.¹⁶ Melissa typically arrived in the e-mail Inbox of its victim, disguised as an e-mail message with a Microsoft Word attachment. When the recipient opened the attachment, Melissa executed. First, it verified whether the recipient had the Microsoft Outlook e-mail program on its computer. If Outlook were present, Melissa would mail a copy of itself to the first fifty names in Outlook's address book, creating the appearance to the fifty new recipients that the user of the infected system had sent them a personal e-mail message. Melissa would then repeat the process with each of the fifty recipients of the infected e-mail message (provided they had Outlook), by automatically transmitting clones of itself to fifty more people. Melissa attacks frequently escalated and resulted in clogged e-mail servers and system crashes.¹⁷

Payload

¹³ See Ed Skoudis, *MALWARE FIGHTING MALICIOUS CODE* (Prentice Hall, 2004), at 31-37.

¹⁴ See e.g. DOROTHY E. DENNING and PETER J. DENNING, *INTERNET BESIEGED* (ACM Press, New York, 1998), at 81.

¹⁵ ICSA Labs 10th Annual Computer Virus Prevalence Survey 2004, Table 5 and Fig. 10, p. 15 (Survey demonstrating popularity of e-mail as medium of virus attack.)

¹⁶ A. Bisset and G. Shipton, *Some Human Dimensions of Computer Virus Creation and Infection*, 52 *INTERNATIONAL J. HUM. COMPUTER STUD.* (2000), 899; R. Ford, *No Surprises in Melissa Land*, 18 *COMPUTERS AND SECURITY*, 300-302.

¹⁷ David Harley et al., *VIRUSES REVEALED UNDERSTAND AND COUNTER MALICIOUS SOFTWARE* (Osborne/McGraw-Hill, 2001), 406-410.

In addition to replicating and spreading, viruses may be programmed to perform specific harmful actions. The module that implements this functionality is known as the payload.¹⁸ A payload can perform a wide range of functions, depending on the aims and objectives of the virus author.¹⁹ A payload can be programmed to perform destructive operations such as corrupting, deleting and stealing information.²⁰ A payload may also create intrusion-enabling devices, such as a backdoor²¹ that allows unauthorized access to the infected machine.²² Some payload effects are immediately obvious, such as a system crash, while others are subtle, such as transposition of numbers and alteration of decimal places.²³ Subtle effects tend to be dangerous because their presence may not be detected until substantial harm has been done. Payloads are often relatively harmless and do no more than entertain the user with a humorous message, musical tune, or graphical display.²⁴

The payload is triggered when a specific condition is satisfied. Triggering conditions come in a variety of forms, such as a specified number of infections, a certain date, or specific time. The Friday-the-13th virus, for instance, only activated its payload on dates with the cursed designation.²⁵ More recently, the first CodeRed worm alternated

¹⁸ JAN HRUSKA, *COMPUTER VIRUSES AND ANTI-VIRUS WARFARE*, (Ellis Horwood Ltd., 1990), at 17, 18.

¹⁹ See e.g. Nicholas Weaver et al., *A Taxonomy of Computer Worms*, 2003 ACM Workshop on Rapid Malcode, Wash. D.C., p. 4 ("The payload is limited only by the imagination of the attacker.")

²⁰ See e.g. Meiring de Villiers, *Computer Viruses and Civil Liability: A Conceptual Framework*, TORT TRIAL AND INSURANCE PRACTICE LAW JOURNAL, Fall 2004 (40:1), 123, 172 (Discussion of damage due to virus infection.)

²¹ A backdoor is a method of gaining remote access to a computer without passing through normal security controls on a system. See Robert Slade, *DICTIONARY OF INFORMATION SECURITY* (2006), at 19.

²² Ed Skoudis, *MALWARE FIGHTING MALICIOUS CODE* (Prentice Hall, 2004), at 27.

²³ See e.g. Peter Szor, *THE ART OF COMPUTER VIRUS RESEARCH AND DEFENSE* (2005), at 302-303 (Describing "data diddlers" as viruses that "do not destroy data all of a sudden in a very evident form, ... but slowly manipulate the data, such as the content of the hard disk.")

²⁴ E.J. Sinrod and W.P. Reilly, *Cyber-Crimes: A Practical Approach to the Application of Federal Computer Crimes Laws*, 16 SANTA CLARA COMPUTER & HIGH TECH. L.J. 117, 218 (describing the W95.LoveSong.998 virus, designed to trigger a love song on a particular date.)

²⁵ See, e.g., Eric J. Sinrod and William P. Reilly, *Cyber Crimes A Practical Approach to the Application of Federal Computer Crime Laws*, 16 SANTA CLARA COMPUTER & HIGH TECH. L. J. 177 (2000), at 217, n. 176.

between continuing its infection cycle, remaining dormant, and attacking the official White House Web page, depending on the day of the month.²⁶ In the simplest case, a payload executes whenever the virus executes, without a triggering event. Viruses do not always have a payload module, but even viruses without a payload may harm their environment by consuming valuable computing resources, by for instance, filling up available memory space and slowing the execution of important programs.

Computer worms

Worms are similar to viruses, but differ in two important respects. Worms propagate autonomously across networks without human intervention, and they replicate and spread without infecting a host program.²⁷ The CodeRed worm, for instance, propagated by injecting copies of itself into the memory of a remote system by exploiting a security vulnerability in the target system. It located potential targets by scanning the Internet for vulnerable systems, to which it propagated automatically.²⁸ The typical virus in contrast, needs to attach itself to an executable file, and then relies on human interaction to propagate across networks. Like viruses, worms may carry destructive payloads, but even without a destructive payload a fast-spreading worm can do significant harm by slowing down a system through the prolific network traffic it generates.²⁹

The original worm was implemented by scientists at Xerox PARC in 1978,³⁰ but the so-called Morris Worm, created by Cornell University graduate student, Robert T. Morris, was the first to become a household name.³¹ The 1989 Morris worm used a security flaw in a UNIX program to invade and shut down much of the Internet. By some accounts, this event first woke the world up to the dangers of computer security

²⁶ See Meiring de Villiers, *Free Radicals in Cyberspace: Complex Liability Issues in Information Warfare*, 4 NORTHWESTERN TECH & IP L. J. 13, 13-14 (Fall, 2005).

²⁷ See *United States v. Robert Tappan Morris*, 928 F.2d 504 (1991); Nicholas Weaver et al., *A Taxonomy of Computer Worms*, 2003 ACM Workshop on Rapid Malcode, Wash. D.C., pp. 11-18. <http://doi.acm.org/10.1145/948187.948190>.

²⁸ See Peter Szor, *THE ART OF COMPUTER VIRUS RESEARCH AND DEFENSE* (2005), at 398-401.

²⁹ See, generally, John F. Schoch and Jon A. Hupp, *The "Worm" Programs - Early Experience with a Distributed Computation*, COMM. ACM, Vol 25, No 3, March 1982, 172.

³⁰ <http://www.parc.xerox.com/about/history/default.html>.

³¹ See David Harley, Robert Slade, and Urs E. Gattiker, *VIRUSES REVEALED* (2001), at 347-52.

vulnerabilities, such as the buffer overflow flaw that enabled the Morris worm to paralyze the Internet.³²

VIRUS DETECTION

Technical anti-virus defenses come in four varieties, namely signature scanners, activity monitors, integrity checkers, and heuristic techniques.³³ Scanners detect specific, known viruses by indentifying patterns that are unique to a particular virus strain. Activity monitors look out for virus-like activity in a computer. Integrity checkers sound an alarm when detecting suspicious modifications to computer files. Heuristic techniques combine virus-specific scanning with generic techniques, to provide a significantly broadened range of detection.

Scanners

Scanners are the most widely used anti-virus defense. A scanner reads executable programs and searches for the presence of virus patterns, known as "signatures." A virus signature consists of patterns of hexadecimal digits embedded in viral code that are unique to a particular virus strain.³⁴ These signatures are created by human experts at institutions such as IBM's High Integrity Computing Laboratory, who scrutinize viral code and extract sections of code with unusual patterns.³⁵ The selected byte patterns are collected in a signature database and used in anti-virus scanners. A scanner detects a virus in a program by comparing the program to its database of signatures, and announcing a match as a possible virus.³⁶

³² Takanen et al., *Running Malicious Code By Buffer Overflows: A Survey of Publicly Available Exploits*, 162. EICAR 2000 Best Paper Proceedings. Available at <http://www.papers.weburb.dk>.

³³ See Peter Szor, *THE ART OF COMPUTER VIRUS RESEARCH AND DEFENSE* (2005), p. Ch. 11.

³⁴ JAN HRUSKA, *COMPUTER VIRUSES AND ANTI-VIRUS WARFARE* (Ellis Horwood, Ltd., 1990), at 42.

³⁵ See Ed Skoudis, *MALWARE FIGHTING MALICIOUS CODE* (Prentice Hall, 2004), 53, 54.

³⁶ JEFFREY O. KEPHART ET AL., *Automatic Extraction of Computer Virus Signatures*, Proceedings of the 4th Virus Bulletin International Conference, R. Ford, ed., Virus Bulletin Ltd., Abingdon, England, 1994, pp. 179-194, at 2.

An ideal virus signature would issue neither false negatives nor false positives.³⁷ Although this ideal is unachievable in practice, anti-virus researchers pursue optimal solutions within practical constraints. The IBM High Integrity Computing Laboratory, for instance, has developed an optimal statistical signature extraction technique that examines all sections of code in a virus, and selects the byte strings that optimize the tradeoff between false positives and negatives.³⁸

Scanners are easy to use, but they are limited to detecting known signatures. Furthermore, a scanner's signature database has to be continually updated as new viruses are discovered and their signatures catalogued, a burdensome requirement in an environment where new viruses appear daily. Modern antivirus vendors have attempted to lighten the burden on users by distributing signature updates directly to their customers via the Internet.³⁹ As the number of known viruses grows, the scanning process will inevitably slow down as an ever-increasing set of possibilities has to be evaluated.

Activity monitors

Activity monitors are resident programs that monitor activities in a computer for behavior commonly associated with viruses. Suspicious activities include operations such as attempts by a program to delete information and mass mail copies of itself. When suspicious activity is detected, the monitor may simply halt execution and alert the user, or take definite action to neutralize the activity.⁴⁰ Activity monitors, unlike scanners, do not need to know the signature of a virus to detect it. Its function is to recognize generic suspicious behavior, not the precise identity of the culprit.

The greatest strength of activity monitors is their ability to detect unknown virus strains, but they also have significant weaknesses. They can only detect viruses that are actually executing, possibly after substantial harm has been done. A virus may, furthermore, execute before the monitor code does, and do harm before it is detected. A

³⁷ See e.g. Jeffrey O. Kephart et al., *Blueprint for a Computer Immune System*, IBM Thomas J. Watson Research Center Report, at 11.

³⁸ See e.g. Jeffrey O. Kephart et al., *Automatic Extraction of Computer Virus Signatures*, Proceedings of the 4th Virus Bulletin International Conference, R. Ford, ed., Virus Bulletin Ltd., Abingdon, England, 1994, pp. 179-194.

³⁹ See e.g. Ed Skoudis, *MALWARE FIGHTING MALICIOUS CODE* (Prentice Hall, 2004), 54.

⁴⁰ Sandeep Kumar and Eugene H. Spafford, *A Generic Virus Scanner in C++*, Technical report CSD-TR-92-062, Dept. of Computer Science, Indiana University, at 3-4.

virus may also be programmed to alter monitor code on machines that do not have protection against such modification.

The effectiveness of activity monitors is limited by the lack of unambiguous rules defining "suspicious" activity. This ambiguity may result in false alarms when an activity monitor picks up legitimate activities which resemble virus-like behavior.⁴¹ False negatives may result when an activity monitor fails to recognize viral activity which does not fit the monitor's programmed definitions.⁴² Recurrent false alarms may ultimately lead users to ignore warnings from the monitor.

Integrity verification

An integrity verifier applies the electronic equivalent of a tamper-proof seal to protected programs, and issues an alert when the seal has been broken, presumably by a virus intrusion. An integrity verification program generates a code, known as a "checksum," for protected files. A checksum may be an arithmetic calculation based on variables such as the total number of bytes in a file, the numerical value of the file size and its creation date. A checksum is periodically recomputed and compared to the original. When a virus infects a file, it usually modifies the contents, resulting in a change in the checksum. When the recomputed value does not match the original, the file is presumed to have been modified since the previous inspection, and an alert is issued.⁴³

The advantage of integrity checking is that it detects most instances of viral infection, as infection usually alters the target file. Its main drawback is its tendency to generate false alarms, as a file can change for "legitimate" reasons unrelated to virus infection.⁴⁴ Integrity checking software therefore presents a high likelihood of false positives, because of the general difficulty of determining whether a program

⁴¹ See e.g. ROBERT SLADE, ROBERT SLADE'S GUIDE TO COMPUTER VIRUSES (Springer, 2d ed., 1996), at 40-41.

⁴² JAN HRUSKA, COMPUTER VIRUSES AND ANTI-VIRUS WARFARE, (Ellis Horwood Ltd., 1990), at 75.

⁴³ See Ed Skoudis, MALWARE FIGHTING MALICIOUS CODE (Prentice Hall, 2004), 58.

⁴⁴ PHILIP FRITES, PETER JOHNSTON AND MARTIN KRATZ, THE COMPUTER VIRUS CRISIS (Van Nostrand Reinhold, New York, 2d ed., 1992), at 125; Ed Skoudis, MALWARE FIGHTING MALICIOUS CODE (Prentice Hall, 2004), 58.

modification is legitimate or due to a virus.⁴⁵ Integrity checking works best on static files, such as system utilities, but it is, of course, an inappropriate technique for files that naturally change frequently, such as Word documents.

Heuristic detection

Modern virus detectors increasingly use heuristic methods. Heuristic rules solve complex problems "fairly well" and "fairly quickly," but less than perfectly. Virus detection is an example of a complex problem that is amenable to heuristic solution. It has been proven that it is mathematically impossible to write a virus detection program that is capable of consistent perfect detection.⁴⁶ Heuristic virus detection methods accept such limitations and attempt to achieve a heuristic solution, namely a detection rate that is below the (unachievable) perfect rate, but representing an optimal tradeoff between detection accuracy, speed and computational expense.⁴⁷

Heuristics detect novel viruses by examining the structure and logic of executable code for evidence of virus-like behavior. The heuristic program then assesses the likelihood that a scrutinized program constitutes a virus by calculating a score based on the number and type of virus-like characteristics detected. If the score exceeds a certain threshold, the scanner classifies the program as malevolent code, and notifies the user. Instructions to send an e-mail message with an attachment to every listing in an address book, for instance, would add significantly to the score. Other high-scoring routines include capabilities to replicate, hide from detection, and execute some kind of payload.⁴⁸

A heuristic scanner typically operates in two phases. The scanning algorithm first narrows the search by identifying the location most likely to contain a virus. It then analyzes code from that location to determine its likely behavior upon execution. A static heuristic scanner compares the code from the "most likely" location to a database of byte

⁴⁵ ROBERT SLADE, ROBERT SLADE'S GUIDE TO COMPUTER VIRUSES (Springer, 2d ed., 1996) 157.

⁴⁶ Diomidis Spinellis, *Reliable Identification of Bounded-Length Viruses is NP-Complete*, IEEE TRANSACTIONS ON INFORMATION THEORY, 49(1), 280, 282 (January 2003) (Stating that theoretically perfect detection is in the general case undecidable, and for known viruses, NP-complete.); Chess & White, *Undetectable Computer Virus*, IBM Research Paper, available at <http://www.research.ibm.com/antivirus/SciPapers/VB2000DC.htm>.

⁴⁷ See e.g. Carey Nachenberg, *Future Imperfect*, VIRUS BULLETIN, August 1997, 6.

⁴⁸ See e.g. Peter Szor, THE ART OF COMPUTER VIRUS RESEARCH AND DEFENSE (2005), at 472-74.

sequences commonly associated with virus-like behavior, and decides whether to classify the code as viral.⁴⁹ A dynamic heuristic scanner uses CPU⁵⁰ emulation.⁵¹ It loads suspect code into a virtual computer, emulates its execution and monitors its behavior. Because it is only a virtual computer, virus-like behavior can be safely observed in what is essentially a laboratory setting, with no need to be concerned about real damage.⁵² Although dynamic heuristics can be time-consuming due to the computationally intensive CPU emulation process, they are sometimes superior to static heuristics. This will be the case when the suspect code is obscure and not easily recognizable as viral in its static state, but clearly reveals its viral nature in a dynamic state.

A heuristic assessment is less than perfect by design, and will inevitably provide false positives and negatives. A low scanner threshold will result in false positives. A scanner with a threshold that is set too high, on the other hand, will fail to detect viruses that are malicious but that do not exactly match the unrealistically tight specifications, resulting in false negatives.⁵³ As in the case of activity monitors, the term "suspicious" is ambiguous. Many legitimate programs, including even some anti-virus programs, perform operations that resemble virus-like behavior.⁵⁴ Nevertheless, state-of-the-art heuristic scanners achieve a 70-80 percent success rate at detecting unknown viruses.⁵⁵

A major advantage of heuristic scanning over generic anti-virus technologies such as behavior monitoring and integrity checking, is its ability to detect viruses before they execute and cause harm. Its advantage over conventional signature scanners lies in its capability to detect novel virus strains whose signatures have not yet been catalogued.

⁴⁹ Sandeep Kumar and Eugene H. Spafford, *A Generic Virus Scanner in C++*, Technical report CSD-TR-92-062, Dept. of Computer Science, Indiana University, at 4-5.

⁵⁰ The CPU, or central processing unit, of a computer is responsible for data processing and computation.

⁵¹ See e.g. JAN HRUSKA, *COMPUTER VIRUSES AND ANTI-VIRUS WARFARE*, (Ellis Horwood Ltd., 1990), at 115; D. BENDER, *COMPUTER LAW: EVIDENCE AND PROCEDURE* (1982), §2.02, at 2-7, -9.

⁵² Sandeep Kumar and Eugene H. Spafford, *A Generic Virus Scanner in C++*, Technical report CSD-TR-92-062, Dept. of Computer Science, Indiana University, at 4.

⁵³ Ed Skoudis, *MALWARE FIGHTING MALICIOUS CODE* (Prentice Hall, 2004), 56.

⁵⁴ Francisco Fernandez, *Heuristic Engines*, Proc. 11th Intl. Virus Bulletin Conference, September 2001, Virus Bulletin Ltd., Abingdon, England, 1994, at 409.

⁵⁵ Carey Nachenberg, *Future Imperfect*, *VIRUS BULLETIN*, August 1997, at 7; *Understanding Heuristics: Symantec's Bloodhound Technology*, Symantec White Paper Series, v. XXXIV, at 9.

Heuristic scanners are also capable of detecting complex virus families, such as polymorphic viruses which complicate detection by changing their signatures from infection to infection.⁵⁶

The explosive growth in new virus strains has made reliable detection and identification of individual strains very difficult and costly, making heuristics more important and increasingly prevalent.⁵⁷ Commercial heuristic scanners include IBM's AntiVirus boot scanner and Symantec's Bloodhound technology.

3. LIABILITY ANALYSIS

Cyber intruders may be subject to criminal,⁵⁸ as well as civil liability.⁵⁹ However, cyber rogues are often judgment-proof and may be difficult to identify or subject to jurisdiction.⁶⁰ A deep-pocketed defendant, such as an organization whose lax security negligently enabled a cyber crime, is therefore usually the preferred target for a civil lawsuit.⁶¹ This section discusses liability issues related to information security failures. It focuses on the paradigmatic case where a defendant negligently enabled a crime by a third party against the plaintiff.

⁵⁶ Polymorphic viruses have the ability to "mutate" by varying the code sequences written to target files. To detect such viruses requires a more complex algorithm than simple pattern matching. See, e.g., DOROTHY E. DENNING and PETER J. DENNING, *INTERNET BESIEGED* (ACM Press, New York, 1998), at 89.

⁵⁷ Carey Nachenberg, *Future Imperfect*, VIRUS BULLETIN, August 1997, at 9.

⁵⁸ See S.D. Personick, ed., *Critical Information Infrastructure Protection and the Law: An Overview of Key Issues*, 37-39; Brent Wible, *A Site Where Hackers are Welcome: Using Hack-In Contests to Shape Preferences and Deter Computer Crime*, 112 YALE L. J. 1577, 1581-85 (2003).

⁵⁹ See Michael L. Rustad, *Private Enforcement of Cybercrime on the Electronic Frontier*, 11 S. CAL. INTERDISC. L.J. 63, 66; Robin A. Brooks, *Deterring the Spread of Viruses Online: Can Tort Law Tighten the Net?*, 17 REV. LIT. 343 (1998); 18 U.S.C. 1030(g) (2000) (Provision in Computer Fraud and Abuse Act allowing civil action against wrongdoer "to obtain compensatory damages and injunctive relief or other equitable relief.")

⁶⁰ See Meiring de Villiers, *Free Radicals in Cyberspace: Complex Liability Issues in Information Warfare*, 4 NORTHWESTERN TECH & IP L. J. 13 (Fall, 2005), § 4.

⁶¹ See Meiring de Villiers, *Free Radicals in Cyberspace: Complex Liability Issues in Information Warfare*, 4 NORTHWESTERN TECH & IP L. J. 13 (Fall, 2005), § 3.3 (Analysis of civil liability for enablement of cyber crime and tort.)

A plaintiff may pursue a civil action under a negligence theory, the most widely used theory of liability in the law of torts.⁶² Negligence is generally defined as a breach of the duty not to impose an unreasonable risk on society.⁶³ It applies to any risk that can be characterized as unreasonable, including risks associated with information security failures. A victim of an information security breach may therefore bring legal action under a negligence theory against anyone who contributed to the risks associated with the breach, including those who failed in their duty to reduce or eliminate the risk.⁶⁴

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

1. A legal duty on the part of the defendant not to expose the plaintiff to unreasonable risks.
2. A breach of the duty, namely a failure on the part of the defendant to conform to the norm of reasonableness.
3. A causal connection between the defendant's conduct and the plaintiff's harm. This element includes actual as well as proximate cause. A defendant's negligence is the actual cause of the plaintiff's harm if, but for the negligence, the harm would not have occurred.⁶⁵ The proximate causation element requires the defendant's conduct to be reasonably related to the plaintiff's harm.⁶⁶
4. Actual damage resulting from the defendant's negligence.⁶⁷

Duty

⁶² See James A. Henderson, *Why Negligence Law Dominates Tort*, 50 UCLA L. REV. 377 (2003).

⁶³ PROSSER AND KEETON ON THE LAW OF TORTS (5th ed., West Publ. Co., 1984), § 31. Second Restatement of Torts, § 282.

⁶⁴ Dan B. Dobbs, *The Law of Torts*, at 258 (The plaintiff can assert that *any* conduct counts as negligence.)

⁶⁵ See e.g. Meiring de Villiers, *Computer Viruses and Civil Liability: A Conceptual Framework*, TORT TRIAL AND INSURANCE PRACTICE LAW JOURNAL, Fall 2004 (40:1) 123, 141-42 (Analysis of causality in context of virus attack.)

⁶⁶ See e.g. David G. Owen, *Idea: The Five Elements of Negligence*, 35 HOFSTRA L. REV. 1671, 1681 (Defining proximate cause as "a reasonably close connection between a defendant's wrong and the plaintiff's injury, a connection that is not remote.")

⁶⁷ See e.g. Meiring de Villiers, *Computer Viruses and Civil Liability: A Conceptual Framework*, TORT TRIAL AND INSURANCE PRACTICE LAW JOURNAL, Fall 2004 (40:1) 123, 172-74 (Discussion of damage due to virus infection.)

Negligence liability of a defendant depends first and foremost on the existence of a legal duty of care to the plaintiff. A duty of care may be imposed by common law tort principles.⁶⁸ A duty may also be imposed by statute, either expressly,⁶⁹ or by legal precedent, if the statute does not expressly provide for civil liability.⁷⁰ California's Security Breach Information Act (SBIA),⁷¹ for instance, expressly creates a civil cause of action for a business' failure to protect customers' personal information, and provides that aggrieved customers may recover damages for breach of that duty. The relevant provision states: "A business that owns or licenses personal information about a California resident shall implement and maintain reasonable security procedures and practices appropriate to the nature of the information, to protect the personal information from unauthorized access, destruction, use, modification, or disclosure."⁷² The legislation further provides, "Any customer injured by a violation of this title may institute a civil action to recover damages."⁷³

The civil cause of action created by the SBIA is rooted in negligence, because the duty it imposes is based on reasonableness, the essence of the standard of care imposed by negligence law.⁷⁴ The statute requires, for instance, implementation of "reasonable

⁶⁸ See e.g. Vincent R. Johnson, *Cybersecurity, Identity theft, and the Limits of Tort Liability*, 57 S.C. L. REV. 255, 272-82; Michael J. Rustad & Thomas H. Koenig, *Extending Learned Hand's Negligence Formula to Information Security Breaches*, 3 ISJLP 237, 239-40 (Arguing that "companies have a duty to provide reasonable information security practices under the common law of torts.")

⁶⁹ See Restatement (Third) of Torts: Liability for Physical Harm 14 cmt. b (Proposed Final Draft No. 1, 2005) (Discussing express and implied statutory causes of action.)

⁷⁰ See e.g. RESTATEMENT (THIRD) OF TORTS § 12 (Discussion Draft 1999) ("An actor is negligent if, without excuse, the actor violates a statute that is designed to protect against the type of accident the actor's conduct causes, and if the accident victim is within the class of persons the statute is designed to protect."); Vincent R. Johnson and Alan Gunn, *STUDIES IN AMERICAN TORT LAW*, 305-06 (3d ed., 2005).

⁷¹ Act effective July 1, 2003, ch. 915, 2002 Cal. Legis. Serv. (West), available at CA LEGIS 915 (2002) (Westlaw).

⁷² Cal. Civ. Code 1798.81.5 (West Supp. 2005).

⁷³ Cal. Civ. Code 1798.81(b) (West Supp. 2005).

⁷⁴ RESTATEMENT (SECOND) OF TORTS § 283 (1965.); R.W. Wright, *The Standards of Care in Negligence Law*, in D.G. Owen (ed.), *PHILOSOPHICAL FOUNDATIONS OF TORT LAW* (Clarendon Press, 1995), at 249; Second Restatement of Torts, § 282.

security procedures and practices"⁷⁵ that are "appropriate to the nature of the information."⁷⁶

Breach of duty

"Breach of duty" refers to a violation of the duty to avoid unreasonable risks of harm to others. This element defines the standard of due care to which a defendant is held, and determines whether the standard has been violated. The traditional law and economics approach to negligence analysis defines due care as a level of precaution that minimizes a global cost function.⁷⁷ In practice, however, courts do not follow this approach. Instead, the common law requires the plaintiff to identify and plead an untaken precaution that would have prevented the accident, if taken. The defendant will then be considered to have breached her duty if the benefits of risk reduction provided by the pleaded precaution exceeded its cost.⁷⁸

The breach calculus weighs the cost of the untaken precaution against the value of the reduction in *all* foreseeable risks that the precaution would have provided, not just the risk that actually materialized.⁷⁹ For instance, the common computer security vulnerability known as a buffer overflow,⁸⁰ may enable a variety of cyber crimes, including identity theft, denial of service, and data corruption. A victim of identity theft enabled by a buffer overflow may argue that the defendant should have patched the

⁷⁵ Cal. Civ. Code 1798.81.5 (b) (West Supp. 2005).

⁷⁶ Cal. Civ. Code 1798.81.5(c) (West Supp. 2005).

⁷⁷ See e.g. John Prather Brown, *Toward an Economic Theory of Liability*, 2 J. LEGAL STUD. 323 (1973); William M. Landes & Richard A. Posner, *THE ECONOMIC STRUCTURE OF TORT LAW* 63 (1987); Oliver Wendell Holmes, Jr., *THE COMMON LAW* 111 (1881).

⁷⁸ See Mark F. Grady, *UNTAKEN PRECAUTIONS*, 18 J. LEGAL STUD. 139, 143 (1989); *Delisi v. St. Luke's Episcopal-Presbyterian Hosp., Inc.*, 701 S.W.2d 170 (Mo. App. 1985) (Plaintiff had to prove physician's breach of duty by specifying the treatment that should have been given, but was not.)

⁷⁹ See e.g. Restatement (Second) of Torts § 281(b), comment e (1965) ("Conduct is negligent because it tends to subject the interests of another to an unreasonable risk of harm. Such a risk may be made up of a number of different hazards, which frequently are of a more or less definite character. The actor's negligence lies in subjecting the other to the aggregate of such hazards."); Mark F. Grady, *UNTAKEN PRECAUTIONS*, 18 J. LEGAL STUD. 139, 146 (1989).

⁸⁰ See e.g. Meiring de Villiers, *Reasonable Foreseeability in Information Security Law: A Forensic Analysis*, 30 HASTINGS COMM. ENT. L.J. 419, 437-39 (Description of the buffer overflow vulnerability.)

vulnerability in a timely fashion, and plead this as an untaken precaution. To show breach, the plaintiff must balance the cost of eliminating the buffer overflow against the aggregate benefit of all risks so reduced, including risks related to denial of service and data corruption, not just the risk of identity theft.

The role of the untaken precaution in negligence law is illustrated in *Cooley v. Public Service Co.*⁸¹ In *Cooley*, the plaintiff sustained injuries from a loud noise over a telephone wire. She suggested two untaken precautions that would have prevented the harm, namely (i) a strategically positioned wire mesh basket, and (ii) insulating the wires. The court ruled that neither untaken precaution constituted a breach of duty. Both precautions would have increased the risk of electrocution to passersby sufficiently to outweigh the benefits in harm reduction.

The *Cooley* court noted that, although the suggested precautions did not succeed, there may exist a cost-effective precaution other than the ones pleaded, that would have satisfied the breach requirement. It is however, the plaintiff's burden to identify and plead such a precaution, if indeed it exists. In a negligence case, more than one untaken precaution may have greater benefits than costs, and the plaintiff may allege several precautions in the alternative.⁸² A court may base a finding of negligence on one or more of the pleaded precautions.⁸³

The following stylized numerical example illustrates the cost-benefit calculus needed to prove a breach of duty that enabled a cyber attack.

Numerical example

In this hypothetical, we assume that an organization uses a signature scanner to detect viruses in its computer network. An attacker launches a virus which evades detection and compromises the confidential information of a customer. The culprit virus is a novel strain that has been documented recently for the first time. It was not detected because its signature had not been included in the database of the organization's scanner at the time of the attack.

The victimized customer contemplates a negligence lawsuit against the organization. The plaintiff must prove the defendant's breach of duty by identifying an

⁸¹ 90 N.H. 460, 10 A.2d 673 (1940).

⁸² MARK F. GRADY, *Untaken Precautions*, 18 J. LEGAL STUD. 139, 144 (1989).

⁸³ See MARK F. GRADY, *Untaken Precautions*, 18 J. LEGAL STUD. 139, 144-45 (1989).

alternative cost-effective precaution that would have avoided the virus. The plaintiff has several pleading options. Potential untaken precautions include more frequent updating of the signature database, or perhaps use of a generic scanner that does not depend on an updated database. Each option has its own set of costs and benefits which must be balanced to determine its cost-effectiveness.⁸⁴

Suppose the plaintiff selects a policy of updating the signature database more frequently. The numbers in Table 1 suggest that this precaution is efficient.⁸⁵ The precaution would add 3 cents to the defendant's average cost of production, but would reduce its expected information security-related loss by 8 cents. The first column lists the alternative precautions at issue, namely continuing scanning at the current rate and scanning at the proposed increased rate, respectively. The second column lists the defendant's average production cost for each precaution. The third column lists the probabilities of virus transmission corresponding to the respective precautions, the fifth the expected losses from a virus attack, and the final column lists the full cost per unit of production, namely production cost plus expected losses due to a virus attack. We assume that a virus attack will result in expected damages of \$10,000.

TABLE 1

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

With the defendant's precaution at its current level, the production cost per unit is 40 cents, the probability of a successful virus infection is 1/100,000, and the loss if an infection occurs is \$10,000. The expected loss per unit due to a virus attack therefore equals 10 cents (1/100,000 X \$10,000), and the total cost per unit is 50 cents.

⁸⁴ See e.g. Fred Cohen, *A Cost Analysis of Typical Computer Viruses and Defenses*, COMPUTERS & SECURITY, 10 (1991).

⁸⁵ Based on an example in A.M. Polinsky, INTRODUCTION TO LAW AND ECONOMICS 98 (table 11) (1983).

If the defendant implemented the proposed precaution pleaded by the plaintiff, its production cost would increase to 43 cents, the probability of virus infection would decline to 1/500,000, and the expected loss due to a virus attack would be 2 cents, giving a total cost per unit of 45 cents. The plaintiff should therefore prevail on the issue of breach. Although the pleaded precaution would increase the defendant's cost by 3 cents per unit, it would lower expected information security-related losses by 8 cents.

The cost-benefit analysis draws on the law, economics and technology of information security, and it requires data on the types, nature and frequencies of cyber attacks. Such an analysis is feasible in principle, and it is aided by the appearance of cost-benefit models of viruses and anti-virus defenses in the computer security literature,⁸⁶ and the increasing availability of empirical data on cyber attacks and their economic impact.⁸⁷

Proximate cause

The plaintiff must prove that the defendant's failure to take the untaken precaution was the actual and proximate cause of real harm to the plaintiff. There are two doctrines of proximate cause in the common law, namely the direct consequences doctrine and the reasonable foresight doctrine.⁸⁸ The reasonable foresight doctrine considers whether the type of harm sustained by the plaintiff was a reasonably foreseeable consequence of the defendant's wrongdoing.⁸⁹ The direct consequences doctrine applies where the wrongful

⁸⁶ See, e.g., Fred Cohen, *A Cost Analysis of Typical Computer Viruses and Defenses*, COMPUTERS & SECURITY, 10 (1991); Robert W. Hahn & Anne Layne-Farra, *The Law and Economics of Software Security*, Working Paper 06-08, AEI Brookings Joint Center for Regulatory Studies; Huaqiang Wei et al., *Cost-Benefit Analysis for Network Intrusion Detection Systems*, CSI 28th Annual Computer Security Conference, October 2001; Meiring de Villiers, *Computer Viruses and Civil Liability: A Conceptual Framework*, TORT TRIAL AND INSURANCE PRACTICE LAW JOURNAL, Fall 2004 (40:1), 123, 161-63; TECH404, <http://www.tech-404.com/calculator.html> (Online cost calculator enabling estimation of financial impact of information security breaches.)

⁸⁷ See e.g. Robert Richardson, 2008 CSI COMPUTER CRIME & SECURITY SURVEY; ICSA Labs 10th Annual Computer Virus Prevalence Survey 2004; Anat Hovav & John D'Arcy, *Capital Market Reaction to Defective IT Products: The Case of Computer Viruses*, 24 COMPUTERS & SECURITY 409 (2005); Brian Cashell et al., *The Economic Impact of Cyber Attacks*, CRS Report for Congress, April 1, 2004; Shane Coursen, *The Financial Impact of Viruses*, INFORMATION SYSTEMS SECURITY 64 (Spring 1997). See however, Michael L. Rustad and Thomas H. Koenig, *The Tort of Negligent Enablement of Cybercrime*, 20 BERKELEY TECH. L. J. 1553, 1599 ("[N]o reliable data exists on the probability of cybercrime, its severity, or the costs of minimizing intrusions, so it is difficult to determine the most efficient level of precaution.")

⁸⁸ MARK F. GRADY, *Untaken Precautions*, 18 J. LEGAL STUD. 139, 151 (1989).

⁸⁹ See e.g. W. Jonathan Cardi, *Purging Foreseeability: The New Vision of Duty and Judicial Power in the Proposed Restatement (Third) of Torts*, 58 VAND. L. REV. 739, 749 ("[A] plaintiff may fail to survive the

acts of multiple tortfeasors were all necessary causes of the plaintiff's injury. The doctrine examines whether the subsequent tortfeasors have cut off the liability of the original tortfeasor.⁹⁰

The Encourage Free Radicals (EFR) doctrine,⁹¹ a paradigm within the direct consequences doctrine, preserves the liability of an original tortfeasor who has encouraged the opportunistic behavior of so-called "free radicals." Free radicals are individuals who are not deterred by the threat of tort or criminal liability, because they lack mental capacity and good judgment, and are shielded from liability by anonymity and insufficient assets. Such trouble-prone individuals are termed "free radicals" because of their tendency to bond with trouble. Examples of free radicals include children, anonymous crowds, criminals, mentally incompetent individuals, terrorists blinded by ideological or religious motivations, and cyber wrongdoers.⁹² The EFR doctrine recognizes that the deterrence rationale of negligence law would be defeated if responsible people who foreseeably encouraged free radicals to be negligent were allowed to escape judgment by shifting liability to the latter. Common law negligence rules therefore preserve the liability of the responsible individuals. The EFR doctrine imposes liability on the encourager, even when intentional or criminal behavior by a free radical intervened.⁹³

*Guille v Swan*⁹⁴ was possibly the original EFR case in the United States.⁹⁵ In *Guille*, the defendant descended in a balloon over New York City into the plaintiff's garden in a manner that attracted a crowd. The defendant's balloon dragged over the plaintiff's garden, but the crowd did much more damage to the garden. The defendant argued that he should be responsible only for his share of the damages, and not for that

proximate cause inquiry where the defendant's actions resulted in (1) an unforeseeable type of injury, (2) an injury occurring in an unforeseeable manner, or (3) injury to an unforeseeable plaintiff.")

⁹⁰ Mark F. Grady, *Proximate Cause Decoded*, 50 UCLA L. REV. 299 (2002).

⁹¹ See Mark F. Grady, *The Free Radicals of Tort*, SUPREME COURT ECONOMIC REVIEW 189 (2004).

⁹² Mark F. Grady, *Proximate Cause Decoded*, 50 UCLA L. REV. 293, 306-312 (2002).

⁹³ Mark F. Grady, *Proximate Cause Decoded*, 50 UCLA L. REV. 293, 308 (2002).

⁹⁴ 19 Johns. 381 (N.Y. 1822).

⁹⁵ See Mark F. Grady, *The Free Radicals of Tort*, SUPREME COURT ECONOMIC REVIEW, 2004, 189, 201, 201.

caused by the crowd, but the court nevertheless held him responsible for all the damages. The crowd were free radicals in that particular situation, and the defendant foreseeably encouraged their behavior. People who are otherwise perfectly rational often behave differently when they are shielded by the anonymity and diminished accountability of a crowd. Chief Justice Spencer stated that the defendant's manner of descent would foreseeably draw a crowd, with predictable consequences for which he should be held responsible, a classic description of the EFR doctrine.⁹⁶

Under the EFR doctrine, the liability of a defendant who enabled a cyber crime by, for instance, failing to correct a security vulnerability, may be preserved if it can be ascertained that the security vulnerability had foreseeably provided encouragement to free radicals. This will often be the case, as research shows that cyber wrongdoers have properties commonly associated with free radicals. They are often judgment-proof and shielded by the anonymity of the Internet architecture.⁹⁷ They appear undeterred by the threat of liability,⁹⁸ and unconcerned about the problems caused by their actions.⁹⁹ Furthermore, security breaches are under-reported, under-prosecuted and the probability of apprehending a cyber attacker is comparatively low.¹⁰⁰ Studies show that most virus authors would either be unaffected or, perversely, actually encouraged by stricter legislation against computer fraud and abuse.¹⁰¹ These factors are consistent with a free radical profile.

A defendant, such as an organization whose negligent security policies enabled a cyber crime, may therefore be held liable to victims of the crime, even though the criminal act of the attacker intervened between the defendant's wrongdoing and the plaintiff's harm. Such a result would be helpful to a plaintiff in cases where the primary

⁹⁶ Mark F. Grady, *The Free Radicals of Tort*, at 113.

⁹⁷ See Meiring de Villiers, *Free Radicals in Cyberspace: Complex Liability Issues in Information Warfare*, 4 NORTHWESTERN TECH & IP L. J. 13, 39-42 (Fall, 2005).

⁹⁸ See e.g. Sarah Gordon, *Virus Writers: The End of Innocence.*; R. Lemos (1999), *'Tis the Season for Computer Viruses*. www.zdnet.co.uk/news/1999/49/ns-12098.html.

⁹⁹ See Meiring de Villiers, *Free Radicals in Cyberspace: Complex Liability Issues in Information Warfare*, 4 NORTHWESTERN TECH & IP L. J. 13, 42-44. (Fall, 2005).

¹⁰⁰ See e.g. Jelena Mirkovic et al., INTERNET DENIAL-OF-SERVICE: ATTACK AND DEFENSE MECHANISMS 14 (2005).

¹⁰¹ See e.g. Sarah Gordon, *Virus Writers: The End of Innocence*. IBM White Paper, <http://www.research.ibm.com/antivirus/SciPapers/VB2000SG.htm>.

tortfeasor has sufficient assets to pay a judgment, but the actual perpetrator is judgment-proof or otherwise immune to liability.

4. CONCLUSION

This article analyzes the legal standard of due care in information security, as an issue of law, economics and technology. The court deciding the case does not determine the standard of care. The plaintiff in a negligence action defines the standard of care by identifying and pleading an untaken precaution, and proving that the precaution is cost-effective. The plaintiff must further prove that failure to take the precaution was the actual and proximate cause of the plaintiff's harm. The plaintiff's selection of untaken precaution therefore defines every aspect of the negligence analysis that will ultimately determine the defendant's liability.