

REASONABLE FORESEEABILITY IN INFORMATION SECURITY LAW: A FORENSIC ANALYSIS

by
MEIRING DE VILLIERS*

I.	INTRODUCTION.....	102
II.	PRINCIPLES OF MALEVOLENT SOFTWARE.....	106
A.	COMPUTER VIRUSES	107
B.	COMPUTER WORMS	110
C.	GENERIC STRUCTURE OF A WORM.....	111
III.	VIRUSES AND WORMS AS THREATS TO INFORMATION SECURITY	115
IV.	COMPUTER SECURITY VULNERABILITIES.....	117
V.	DUTY TO SAFEGUARD CONFIDENTIAL INFORMATION	122
A.	STATUTORY DUTY OF CARE	122
B.	COMMON LAW TORT	124
C.	FORESEEABILITY.....	127
D.	LEGAL MEANING OF FORESEEABILITY	128
VI.	REASONABLE FORESEEABILITY IN INFORMATION SECURITY	130
A.	COMMON LAW BACKGROUND	130
B.	FORENSIC ANALYSIS OF SECURITY VULNERABILITIES.....	143
VIII.	A FORESEEABILITY METRIC	153
IX.	DISCUSSION AND CONCLUSION.....	158

* John Landerer Faculty Fellow, University of New South Wales School of Law. BSc Elec Eng (U. Pretoria), P.Eng. (Canada), J.D. (Stanford Law School), Ph.D. (Stanford Dept. Economics).

To Measure Is to Know
- James Clark Maxwell¹

I. 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.²

Malevolent code, such as computer viruses and worms, are powerful weapons in the hands of cyber rogues. Viruses and worms can be programmed to corrupt, delete, or steal sensitive information. Also, cyber terrorists can exploit malevolent code to disrupt elements of the national critical information infrastructure, such as banking, transportation, communications, and energy provision systems. A recent denial of service attack³ on the Port of Houston, for instance, made crucial navigating data on the port's Web service temporarily unavailable to shipping pilots and mooring companies, creating risks of substantial collision and other threats.⁴

1. Quoted in ANDREW JAQUITH, SECURITY METRICS REPLACING FEAR, UNCERTAINTY, AND DOUBT xv (2007).

2. See DANIEL J. SOLOVE, THE DIGITAL PERSON: TECHNOLOGY AND PRIVACY IN THE INFORMATION AGE 16-26 (2004) (describing history and use of databases in private sector); J. Howard Beales, III, Remarks, *Symposium on the Patriot Act, Consumer Privacy, and Cybercrime*, 5 N.C. J.L. & TECH. 1, 2 (2003) See also Danielle Keats Citron, *Reservoirs of Danger: The Evolution of Public and Private Law at the Dawn of the Information Age*, 80 S. CAL. L. REV. 241 (2007) ("A defining problem of the Information Age is securing computer databases of ultra sensitive personal information. These reservoirs of data fuel our Internet economy but endanger individuals when their information escapes into the hands of cyber criminals."); A Chronology of Data Breaches, www.privacyrights.org/ar/ChronDataBreaches.htm (last modified April 2, 2008).

3. A Distributed Denial of Service (DDoS) attack aims to deprive legitimate users of a resource or service provided by a system, by overloading the system with a flood of data packets, thus preventing it from processing legitimate requests. See, e.g., Meiring de Villiers, *Distributed Denial of Service: Law, Technology & Policy*, 39 WORLD JURIS LAW/TECH. J. 1 (2006) (an interdisciplinary analysis of allocation of liability among multiple tortfeasors for distributed denial of service attacks on vulnerable systems).

4. See STEVE GIBSON, THE STRANGE TALE OF THE DENIAL OF SERVICE: ATTACKS AGAINST GRC.COM, available at www.crime-research.org/library/grcdos.pdf.

Viruses and worms gain unauthorized access to a system, such as the Port of Houston's Web service, by exploiting security lapses, such as unpatched security vulnerabilities.⁵ A security vulnerability is an error in an information system that an intruder can exploit to violate the system's security policy.⁶ A security vulnerability in an automated bank teller program, for instance, may allow a rogue to capture sensitive information such as personal identity ("PIN") numbers of previous users.⁷

This article presents an analysis of civil liability for failure to safeguard confidential information. It focuses on situations where database owners fail to patch a computer security vulnerability, which facilitates compromise of sensitive information. Professor Robert Rabin has termed wrongdoing of this kind an "enabling tort."⁸ An enabling tort occurs when a negligent act by a primary tortfeasor sets the stage for an intervening tortfeasor to commit a tort or crime.⁹ Professor Mark Grady has developed a theory explaining that a primary tortfeasor's liability will be preserved for enabling conduct that foreseeably encouraged intervening tortfeasors who are so-called "free radicals."¹⁰ Free radicals are individuals who are not deterred by the threat of liability, and include persons such as mentally incompetent people, terrorists, and criminals.¹¹ Research has shown that cyber rogues generally fit the profile of free radicals.¹²

Foreseeability of an intervening tortfeasor's action is essential to imposing liability on a primary tortfeasor for enabling the intervenor's

5. A "patch" is a software update which is overlaid on an existing program in order to fix a vulnerability in the program. A patch is usually a temporary remedy, to be used until a more permanent remediation of the vulnerability becomes available. See ROBERT SLADE, *DICTIONARY OF INFORMATION SECURITY* 139 (2006).

6. See William L. Fithen et al., *Formal Modeling of Vulnerability*, BELL LABS TECH. J. Feb. 5, 2004, at 173, 173-74.

7. See CHARLES P. PFLEEGER & SHARI L. PFLEEGER, *SECURITY IN COMPUTING* 116 (4th ed., 2007).

8. See Robert L. Rabin, *Enabling Torts*, 49 DEPAUL L. REV. 435 (1999).

9. *Id.* at 437 ("Beyond the immediate perpetrator of harm, the victim perceives the individual, or more often, the enterprise, that set the stage for the suffering that unfolded. The Enabler."). See, e.g., *Kline v. 1500 Mass. Ave. Apartment. Corp.*, 439 F.2d 477 (D.C. Cir. 1970) (owner of apartment complex alleged to have enabled criminal activity by carelessly allowing security measures at the building to deteriorate).

10. See Mark F. Grady, *The Free Radicals of Tort*, 11 S. CT. ECON. REV. 189 (2004).

11. *Id.* at 191. See also Rabin, *supra* note 8, at 439 ("The key factor counseling liability . . . is that defendant paved the way for a truly reckless individual to be imposing serious risks of injury on the public at large.").

12. See generally Meiring de Villiers, *Free Radicals in Cyberspace: Complex Liability Issues in Information Warfare*, 4 N.W. TECH. & INTELL. PROP. L. J. 13 (2005).

behavior.¹³ In a civil suit against a database owner for failure to patch a vulnerability, foreseeability of exploitation of the vulnerability is a key element of the liability analysis, and the focus of this article. The article provides judicial decision-makers with a theoretical basis and practical methodology to make an informed and rational decision about reasonable foreseeability in the context of an information security breach.

The main contribution of the article to legal scholarship is its analysis of the law and technology of cyber attacks that exploit computer security vulnerabilities. The analysis identifies features that make exploitation of a computer security vulnerability reasonably foreseeable. The article studies traditional tort cases where a primary tortfeasor enabled a crime or tort perpetrated by an intervenor, and identifies factors that, in the view of the courts, made the intervenor's behavior foreseeable. The analysis develops cyber analogues of these features, and shows that vulnerabilities are likely to be exploited if they are easy to exploit; are technically closely aligned with the objectives of cyber attackers; provide unauthenticated access to a target system; provide remote access; provide anonymous access; and exhibit low access complexity.

*Richardson v. Ham*¹⁴ illustrates the meaning of these "foreseeability features" and describes their common law origins. In *Richardson*, the defendants left an unlocked bulldozer parked overnight in a public area. Three inebriated young men started one of the bulldozers, set it in motion, and then abandoned it, allowing the runaway bulldozer to plough through a residential area.¹⁵ Plaintiffs who suffered injuries and property damage filed suit against the owners of the bulldozer, alleging that they were negligent in leaving the machine unattended and unlocked.¹⁶ In imposing a duty of care on the defendants, the court emphasized factors that made the intermeddling foreseeable.

The bulldozer was easily accessible to intermeddlers, as it was left unlocked and unattended in a public area. A relatively unskilled person could start the machine and set the bulldozer in motion, even though he may not have the skill to stop it. Once intermeddlers gained basic access to the bulldozer, they faced no additional hurdles to starting the engine and setting it in motion. The bulldozer was fuelled up, and the engine could be started with the bulldozer in gear by simply pushing in a lever and stepping on the starter. If so started, the bulldozer would be set in motion

13. See Grady, *supra* note 10, at 214; Rabin, *supra* note 8, at 446 (calling foreseeability in workplace hazard enabling torts as "the battleground . . . of the third party intervenor story"). See also Rabin, *supra* note 8, at 447, 450.

14. *Richardson v. Ham*, 285 P.2d 269 (Cal. 1955).

15. *Id.* at 270.

16. *Id.*

immediately. In addition, the bulldozer was left parked overnight, which provided intermeddlers with a significant time window of opportunity. Bulldozers are relatively uncommon and present a greater attraction to intermeddlers than other vehicles. The public's fascination with bulldozers made unattended bulldozers a more foreseeable target of intermeddlers than, for instance, an ordinary automobile with its key left in the ignition.

The factors that made intermeddling foreseeable in *Richardson* have direct analogues in cyberspace. The intermeddlers had unauthenticated access to the bulldozer. "Unauthenticated access" generally refers to access without a valid key or identity validation.¹⁷ In information security, authentication refers to procedures by which a computer system verifies the identity of a party from whom it has received a communication, such as a login procedure. A system with a vulnerability that allows the authentication barrier to be bypassed is the cyber analogue of an unlocked vehicle in a public place, such as the bulldozer in *Richardson*.

A computer security vulnerability is considered easy to exploit if it requires minimal technical sophistication to leverage.¹⁸ A system with a vulnerability that is easy to exploit would be accessible to a relatively unskilled hacker, just as in *Richardson*, where a relatively unskilled person could start the bulldozer and set it in motion. Common sense suggests, and the *Richardson* decision confirms, that an easily exploitable vulnerability is relatively foreseeably exploited.

A computer security vulnerability has low access complexity if an attacker faces no additional barriers to exploiting the vulnerability after gaining basic access to a target system.¹⁹ Low access complexity is characterized by features such as a large window of opportunity to gain access and no complications beyond basic access.²⁰ The *Richardson* intermeddlers faced low access complexity: the bulldozer was conveniently fuelled up, and the bulldozer could be set in motion at the push of a lever and a step on the starter. Additionally, the bulldozer was left parked overnight, which provided intermeddlers with a significant window of opportunity. Low access complexity is clearly an attractive property to

17. See, e.g., THE OXFORD THESAURUS (American Edition, 1992) (Defining "authenticate" as "verify, validate, certify, substantiate, endorse, vouch for, corroborate.").

18. See JAMES C. FOSTER ET AL., BUFFER OVERFLOW ATTACKS 10 (2005); SYMANTEC INTERNET SECURITY THREAT REPORT TRENDS FOR JULY – DECEMBER 06 90 (2007), available at http://eval.symantec/mktginfo/enterprise/ent-whitepaper_internet_security_threat_report_xi_03_2007.en-us.pdf.

19. See, e.g., Peter Mell, Karen Scarfone, & Sasha Romanosky, *Common Vulnerability Scoring System*, IEEE SECURITY & PRIVACY, Nov./Dec. 2006, at 85, 86.

20. *Id.*

cyber attackers as well as their real-space siblings, such as the *Richardson* intermeddlers.

An opportunity is scarce or unique if it is aligned with the objectives of a criminal or tortfeasor and if there are few equivalent alternative opportunities available. The unusual fascination of the public with bulldozers in *Richardson* made it a scarce opportunity to intermeddlers. A computer security vulnerability may likewise constitute a scarce opportunity to an attacker who uses a specialist virus or worm that is programmed to exploit the particular vulnerability. The attractiveness of such a vulnerability over ordinary vulnerabilities without this feature is analogous to the public's heightened fascination with bulldozers, as compared to ordinary automobiles.

The article concludes by proposing a numerical metric of the degree to which a particular cyberspace vulnerability is foreseeably exploitable. The metric is a function of quantitative proxies of the "foreseeability features" identified by the analysis. The article includes a numerical example illustrating the application of the metric to vulnerabilities that have actually been exploited in cyber attacks.

Following this introduction, Section 2 provides a background on malevolent software and Section 3 discusses how viruses and worms are threats to the confidentiality, integrity, and availability of information online. Next, Section 4 discusses the role of computer security vulnerabilities in cyber attacks. Section 5 analyzes the legal duty of database owners to safeguard confidential information and the role of reasonable foreseeability in the liability analysis. Section 6 then provides a forensic analysis of foreseeability in the context of information security. Section 7 subsequently presents a numerical metric of the degree to which a particular vulnerability is foreseeably exploitable, as well as an example illustrating the application of the metric to "real world" vulnerabilities. Finally, Section 8 discusses and concludes.

II. Principles of Malevolent Software

Malevolent software is a term for computer code that is designed to disrupt the operation of a computer system.²¹ Computer viruses and its common variant, the worm, are the most common of these rogue programs. Other forms of malicious software include so-called logic bombs,²² Trojan horses,²³ and trapdoors.²⁴

21. See SLADE, *supra* note 5, at 118.

22. 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

A. Computer Viruses

The term “virus,” Latin for “poison,” was first formally defined by Dr. Fred Cohen in 1983.²⁵ The concept, however, originated in John von Neumann’s studies of self-replicating mathematical automata in the 1940s.²⁶ 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 clones 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.²⁷ As the definition suggests, a typical computer virus consists of three basic modules or mechanisms, namely an infection module, a payload trigger, and a payload.

1. Infection Module

An infection module enables a virus to reproduce and attach copies of itself onto target hosts, such as a computer or network.²⁸ This mechanism is the most salient technical property of a computer virus.²⁹ The first task of the infection mechanism is to locate a prospective host program. Once a suitable host is found, the virus may take precautions, such as checking

not viral, but a virus may contain a logic bomb as a payload.” PETER SZOR, *THE ART OF COMPUTER VIRUS RESEARCH AND DEFENSE* 30 (2005).

23. A Trojan horse is a program that appears to be beneficial, but contains a harmful payload. Slade, *supra* note 5, at 663.

24. A trapdoor, or backdoor, is a function built into a program or system to allow unauthorized access to the system. *Id.* at 643. See also DOROTHY E. DENNING & PETER J. DENNING, *INTERNET BESIEGED* 75-78 (1998).

25. FRED COHEN, *COMPUTER VIRUSES* (1984) (PhD dissertation, University of Southern California).

26. See Rick Lehtinen et al., *COMPUTER SECURITY BASICS* 83 (2006) (“[T]he roots of the modern computer virus go back to 1949. This was when computer pioneer John Von Neumann presented a paper on the ‘Theory and Organization of Complicated Automata,’ in which he postulated that a computer program could reproduce.”). See also DENNING & DENNING, *supra* note 24, at 74; Jeffrey O. Kephart et al., *Fighting Computer Viruses*, *SCIENTIFIC AMERICAN*, November 1997, at 56.

27. JOHN MACAFEE & COLIN HAYNES, *COMPUTER VIRUSES, WORMS, DATA DIDDERS, KILLER PROGRAMS, AND OTHER THREATS TO YOUR SYSTEM*, 26 (1989); FREDERICK B. COHEN, *A SHORT COURSE ON COMPUTER VIRUSES 1-2* (2d ed. 1994). In his PhD dissertation, Dr. Cohen defined a virus simply as any program capable of self-reproduction. This definition appears overly general. A literal interpretation of the definition would classify even programs such as compilers and editors as viral. DENNING & DENNING, *supra* note 24, at 75.

28. See ED SKOUDIS, *MALWARE: FIGHTING MALICIOUS CODE* 31-37 (2003).

29. DENNING & DENNING, *supra* note 24, at 73-75; DAVID HARLEY, ROBERT SLADE & URS E. GATTIKER, *VIRUSES REVEALED* 87 (2001) (“The infection mechanism is the code that allows the virus to reproduce and infect a target host, and *thus to be a virus.*”) (emphasis added); ROGUE PROGRAMS: VIRUSES, WORMS, TROJAN HORSES 247 (Lance J. Hoffman ed., Van Nostrand Reinhold, 1990) (“[t]he ability to propagate is essential to a virus program”).

whether the host has already been infected.³⁰ The virus then installs a copy of itself on the host.³¹ Once settled, the virus may take steps to protect itself from detection by changing its form.³² When the host program runs, control is 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 files on the system or network, and even across networks.³³

A virus may infect a computer or a network through several possible points of entry, including via an infected file downloaded from the Internet, web browsing, removable media such as writable compact disks and DVDs, infected files in shared directories, an infected email attachment, or even through infected commercial shrink-wrapped software.³⁴ Early viruses targeted the boot sectors of floppy disks, and this trend continued into the 1990s.³⁵ Floppy disks are no longer widely used to share files, and viruses are increasingly transmitted via email.³⁶ In a 1996 national survey, for instance, approximately 9 percent of respondents listed email as the medium of infection of their most recent virus incident, while 71 percent blamed infected diskettes. In 2004, the corresponding numbers were 92 percent for email, and zero percent for diskettes.³⁷

30. Viruses, known as sparse infectors, may try to slow down the rate of infection to avoid detection, while fast infectors, on the other hand, may attempt to infect as many hosts as possible in a short period of time. See HARLEY, SLADE & GATTIKER, *supra* note 29, at 87.

31. There are three mechanisms through which a virus can infect a host program. A virus may attach itself to its host as a shell, an add-on, or as intrusive code. A shell virus forms a layer ("shell") around the host code so that the latter effectively becomes an internal subroutine of the virus. The host program is then replaced by a functionally equivalent program that includes the virus. The virus executes first, and then allows the host code to execute. Boot program viruses are typically shell viruses. Most viruses are of the add-on variety. They become part of the host by appending, or prepending, their code to the host code, without altering the host code. The viral code may alter the order of execution, allowing itself to execute first and then the host code. Macro viruses are typically add-on viruses. Intrusive viruses, in contrast, overwrite some or all of the host code, replacing it with its own code. See DENNING & DENNING, *supra* note 24, at 81; PHILIP FRITES, PETER JOHNSTON & MARTIN KRATZ, *THE COMPUTER VIRUS CRISIS* 73-75 (2d ed. 1992).

32. A virus' capability to change its form is known as polymorphism. Detecting polymorphic viruses requires a more complex algorithm than simple pattern matching. See DENNING & DENNING, *supra* note 24, at 89. See also HARLEY, SLADE & GATTIKER, *supra* note 29, at 87-88.

33. See SKOUDIS, *supra* note 28, at 31-37.

34. See DENNING & DENNING, *supra* note 24, at 81; FRITES, JOHNSTON & KRATZ, *supra* note 31, at 73-75.

35. See LARRY BRIDWELL, ICSA LABS 10TH ANNUAL COMPUTER VIRUS PREVALENCE SURVEY 15, tbl. 5 (2004), available at <http://www.icsalabs.com/icsa/docs/html/library/whitepapers/VPS2004.pdf>.

36. *Id.*

37. *Id.* at 15, Table 5 and Fig. 10.

Email is currently the most widely used medium for exchanging files and sharing information, but it has also become a convenient and efficient vehicle for virus and worm propagation. Fast-spreading viruses, such as ExploreZip and Melissa, for instance, exploited automatic mailing programs to spread within and across networks.³⁸ Melissa typically arrived in the email inbox of its victim disguised as an email message with a Microsoft Word attachment. When the recipient opened the attachment, Melissa executed. First, it verified whether the recipient had the Microsoft Outlook email program on its computer. If Outlook were present, Melissa would mail a copy of itself to the first fifty names in the Outlook address book. This email would appear to the fifty new recipients as a personal email message sent by the user of the infected system. Melissa would then repeat the process with each of the fifty recipients of the infected email message (provided they had Outlook), by automatically transmitting clones of itself to fifty more people.³⁹ Melissa attacks frequently escalated and resulted in clogged email servers and system crashes.⁴⁰

2. Payload

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.⁴¹ A payload can perform a wide range of functions, depending on the aims and objectives of the virus author.⁴² A payload can be programmed to perform destructive operations such as corrupting, deleting, and stealing information. A payload may also create a backdoor⁴³ that allows unauthorized access to the infected machine.⁴⁴ Some payload effects are immediately obvious, such as a system crash, while others are subtle, such as transposition of numbers and

38. Andy Bisset and Geraldine Shipton, *Some Human Dimensions of Computer Virus Creation and Infection*, 52 INT'L J. HUM.-COMPUTER STUD. 899 (2000); Richard Ford, *No Surprises in Melissa Land*, 18 COMPUTERS AND SECURITY 300, 300-02 (1999).

39. See, e.g., Ford, *supra* note 38, at 302.

40. HARLEY, SLADE & GATTIKER, *supra* note 29, at 406-10.

41. JAN HRUSKA, *COMPUTER VIRUSES AND ANTI-VIRUS WARFARE*, 17-18 (Ellis Horwood Ltd., 1990) (in addition to self-replicating code, viruses often also contain a payload, which is capable of producing malicious side-effects). See also COHEN, *supra* note 27, at 8-15 (including examples of malignant viruses and their functions); MACAFEE & HAYNES, *supra* note 27, at 61.

42. See, e.g., Nicholas Weaver et al., *A Taxonomy of Computer Worms*, 2003 ACM Workshop on Rapid Malcode, Wash. D.C., <http://portal.acm.org/citation.cfm?id=948190> ("The payload is limited only by the imagination of the attacker.").

43. A backdoor is a method of gaining remote access to a computer without passing through normal security controls on a system. See SLADE, *supra* note 5, at 19.

44. SKOUDIS, *supra* note 28, at 27; HARLEY, SLADE & GATTIKER, *supra* note 29, at 88-89; Meiring de Villiers, *Computer Viruses and Civil Liability: A Conceptual Framework*, TORT TRIAL & INS. PRAC. L.J., 123, 172 (2004) (discussion of damage due to virus infection).

alteration of decimal places.⁴⁵ Subtle effects tend to be dangerous because their presence may not be detected until after substantial harm has been done. Payloads, however, are often relatively harmless and do no more than entertain the user with a humorous message, musical tune, or graphical display.⁴⁶

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.⁴⁷ More recently, the first CodeRed worm alternated between continuing its infection cycle, remaining dormant, and attacking the official White House Web page, depending on the day of the month.⁴⁸ In the simplest case, a payload executes whenever the virus executes, without a trigger event. Viruses do not always have a payload module, but even viruses without a payload may harm their environment by consuming valuable computing resources.⁴⁹

B. Computer Worms

Worms are similar to viruses, but differ from them in two important respects. Worms propagate autonomously across networks without human intervention, and they replicate and spread without infecting a host program.⁵⁰ For instance, the CodeRed worm propagated by injecting

45. MACAFEE & HAYNES, *supra* note 27, at 61. *See also* SZOR, *supra* note 22, at 302-03 (describing “data diddlers” as viruses that “do not destroy data all of a sudden in a very evident form . . . but [that] slowly manipulate the data, such as the content of the hard disk”).

46. 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 (2000) (describing the W95.LoveSong.998 virus, which was designed to trigger the host to play a love song on a particular date).

47. *See, e.g., id.* at 217, n. 176.

48. *See generally* de Villiers, *supra* note 12.

49. Viruses can cause economic losses by replicating and spreading, such as when they fill up available memory space, slow down the execution of important programs, and lock keyboards. The Melissa virus, for instance, mailed copies of itself to everyone in the victim’s email address book, resulting in clogged email servers and even system crashes. *See, e.g.,* FRITES, JOHNSTON & KRATZ, *supra* note 31, 23-24 (“The Christmas card (virus) stopped a major international mail system just by filling up all available storage capacity.”); HARLEY, SLADE & GATTIKER, *supra* note 29, at 88 (“A virus does not necessarily need to have either a trigger or payload. A virus with a trigger and payload but no replication mechanism, on the other hand, is not a virus, but may be described as a Trojan.”).

50. *See* United States v. Morris, 928 F.2d 504 (2d Cir. 1991) (Defining a worm as “a program that travels from one computer to another but does not attach itself to the operating system of the computer it infects. It differs from a virus, which is also a migrating program, but one that attaches itself to the operating system of any computer it enters and can infect any other computer that uses files from the infected computer.”); Weaver et al., *supra* note 42, at 11-18

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.⁵¹ 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.⁵²

The original worm was implemented by scientists at Xerox PARC in 1978.⁵³ The so-called Morris Worm, however, created by Cornell University graduate student, Robert T. Morris, was the first worm to become a household name.⁵⁴ 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 alerted the world to the dangers of computer security vulnerabilities, such as the buffer overflow flaw that enabled the Morris worm to paralyze the Internet.⁵⁵

C. Generic Structure of a Worm

A typical worm consists of the following basic components: (1) an activation mechanism, (2) a target selection algorithm and scanning engine, (3) a warhead, (4) a propagation engine, and (5) a payload.⁵⁶ The activation mechanism triggers execution of the worm on the target computer. The target selection algorithm identifies new potential targets, and the scanning engine narrows down the selection by identifying the vulnerable subset. The warhead penetrates the target, paving the way for the propagation engine to move the worm body to the target. The payload, if present, is programmed to cause harm, such as launching a denial of service attack.

(defining a worm as “a program that self-propagates across a network exploiting security or policy flaws in widely used services”).

51. Szor, *supra* note 22, at 398-401.

52. See generally John F. Schoch & Jon A. Hupp, *The “Worm” Programs - Early Experience with a Distributed Computation*, COMM. OF THE ACM, March 1982, at 172.

53. PARC Milestones, *Innovation Milestones*, <http://www.parc.xerox.com/about/history/default.html> (last visited Feb. 6, 2008).

54. See HARLEY, SLADE & GATTIKER, *supra* note 29, at 347-52.

55. Ari Takanen, et al., *Running Malicious Code By Buffer Overflows: A Survey of Publicly Available Exploits*, EICAR 2000 BEST PAPER PROCEEDINGS, 158, 162, (2000), available at <http://www.ee.oulu.fi/research/ouspg/protos/sota/EICAR2000-overflow-survey/paper.pdf> (“The day when the world finally acknowledged the risk entailed in overflow vulnerabilities and started coordinating a response to them was the day when the Internet Worm was introduced, spread and brought the Internet to its knees.”).

56. See SKOUDIS, *supra* note 28, at 79-80.

1. *Activation Mechanism*

A worm may be activated a number of different ways. The Melissa worm relied on a tantalizing message to persuade a user to open an e-mail attachment that launched the worm.⁵⁷ Worms such as Iloveyou⁵⁸ and Benjamin⁵⁹ employed similar tactics. The CodeRed worm self-activated by automatically searching for and exploiting network vulnerabilities to inject itself into the memory of a target server.⁶⁰

2. *Target Selection Algorithm and Scanning Engine*

A worm needs to locate new targets in order to continue spreading. Its target selection algorithm selects Internet Protocol (IP) addresses⁶¹ of potential targets. A scanning algorithm then determines whether the computer at the selected address contains a suitable vulnerability.⁶²

The most basic target selection algorithm chooses an IP address at random. The fast-spreading Slammer worm, for instance, generated random IP addresses and sent a packet to each target, without first verifying the validity of the IP address.⁶³ More sophisticated worms, such as CodeRed, are programmed to scan a network for vulnerable IP addresses and “fingerprint” a remote system to ascertain its vulnerability.⁶⁴ E-mail worms such as W97M/Melissa@mm read e-mail addresses on a system and mail copies of themselves to each address.⁶⁵ Worms may also harvest e-mail

57. The user was tempted with a subject line, such as “*Here is that document you asked for . . . don’t show anyone else ;-)*.” When the recipient opened the attachment, Melissa executed. See CERT Advisory CA-1999-04 Melissa Macro Virus, <http://www.cert.org/advisories/CA-1999-04.html> (last modified March 31, 1999). See also HARLEY, SLADE & GATTIKER, *supra* note 29, at 406-410. Other worm infection propagators include e-mail attachment inserters, instant messaging attacks, and SMTP attacks. See SZOR, *supra* note 22, at section 9.4, at 331-38.

58. CERT Advisory CA-2000-04 Love Letter Worm Virus, <http://www.cert.org/advisories/CA-2000-04.html> (last modified May 9, 2000).

59. *W32.Benjamin.Worm*, Symantec.com, <http://securityresponse.symantec.com/avcenter/venc/data/w32.benjamin.worm.html>. (last visited Mar. 26, 2008)

60. See SZOR, *supra* note 22, at 315.

61. Each computer on the Internet is uniquely identified by its IP address. FRED T. HOFSTETTER, INTERNET TECHNOLOGIES AT WORK 19 (2005) (“Every computer on the Internet has a unique Internet Protocol (IP) address. Each packet of information that gets transmitted over the Internet contains the IP address of the computer that sent it and the IP address of the computer to which it is being sent.”).

62. SKOUDIS, *supra* note 28, at 84-87.

63. David Moore et al., Inside the Slammer Worm, IEEE SECURITY & PRIVACY, July/August 2003, 33. Check source

64. SZOR, *supra* note 22, at 315.

65. The W97M/Melissa@mm worm propagated itself widely by exploiting the Microsoft Outlook e-mail program. SZOR, *supra* note 22, at 319, 334.

addresses from a mail server, a DNS server, or use search engines to harvest addresses on the Internet.⁶⁶

3. Warhead

The first step towards taking over a target computer is gaining access to the target machine. A worm accomplishes this through its warhead, typically by exploiting a vulnerability in the target system. Commonly employed penetration techniques include buffer overflow exploits, e-mail penetration, file sharing, and backdoor attacks.

a. Buffer Overflow

The buffer overflow is currently (and has been for over a decade) the most commonly exploited vulnerability to get unauthorized access to a system.⁶⁷ A buffer overflow vulnerability allows executable malevolent code to be copied into the memory of a target computer. A skillful attacker can then manipulate the vulnerability to remotely execute the malevolent code.⁶⁸

b. E-mail Penetration

E-mail is a popular penetration technique. E-mail worms transmit themselves to a target via an executable infected e-mail attachment. Sophisticated e-mail worms, such as W32/Nimda.A@mm are programmed to activate automatically when an infected e-mail message is read or merely previewed.⁶⁹

c. File Sharing Techniques

Some viruses and worms propagate through file-sharing mechanisms, such as the peer-to-peer (P2P) services, which include Gnutella and Kazaa.⁷⁰ Each member (“peer”) of a P2P network maintains a shared folder with files made available to other peers for downloading. Files that

66. See *id.* at 319-24.

67. See ERIC CHIEN AND PETER SZOR, BLENDED ATTACK EXPLOITS, VULNERABILITIES AND BUFFER-OVERFLOW TECHNIQUES IN COMPUTER VIRUSES (2002), available at <http://www.peterszor.com/blended.pdf>. See *infra* § IV(A) for a discussion of the buffer overflow.

68. See *infra* § IV(A) for a discussion of the buffer overflow.

69. See SZOR, *supra* note 22, at 414,-15.

70. See SKOUDIS, *supra* note 28, at 51; Kim Zetter, *Kazaa Delivers More Than Tunes*, WIRED, Jan. 1, 2004, <http://www.wired.com/techbiz/media/news/2004/01/61852> (reporting that 44 percent of executable files downloaded through a Kazaa client application is infected by malicious code).

are exchanged over a P2P network may be infected with malevolent code capable of infecting a user's computer when downloaded and opened.⁷¹

d. Propagation Through Backdoor Interfaces

Some worms use backdoor interfaces to propagate themselves. A backdoor is a software or hardware mechanism that can be used to gain remote access to a computer without passing through normal security controls.⁷² An attacker can exploit a backdoor to take control of a system and compromise sensitive information.⁷³ A backdoor may, for instance, consist of a password recognition routine installed in the computer as a modification of a legitimate program. The routine would enable a hacker who provided the correct password to gain access to confidential files and programs on the computer.⁷⁴ Worms that utilize backdoor interfaces include the Nimda worm, which took advantage of a backdoor opened by CodeRed. The W32/Borm worm used network scanning and fingerprinting techniques to locate backdoor-compromised systems.⁷⁵

4. Propagation Engine

The propagation engine moves the body of the worm to the target. The warhead may, for instance, execute a program such as the File Transfer Protocol,⁷⁶ in order to move the worm's code. The transported worm then installs itself on the machine, loads its code into the memory, and prepares to run on the system. An efficient worm carries its entire body of code within its warhead. In the case of e-mail worms, such as the SQL Slammer, the entire body is usually included in the e-mail attachment.⁷⁷

5. Payload

The payload is an optional, but common component of computer worms. It consists of special code designed to achieve a specific aim of the attacker. The payload may, for instance, display a simple one-time

71. SHIN ET AL., MALWARE PREVALENCE IN THE KAZAA FILE-SHARING NETWORK, 1 (2006), www.imconf.net/imc-2006/papers/p34-shin.pdf. Some authors prefer the term "virus" rather than "worm" for malevolent code that spreads via a P2P network, because of its reliance on human intervention. *Id.* at 3, n. 4.

72. See SLADE, *supra* note 5, at 19-20.

73. See SKOUDIS, *supra* note 28, at 190.

74. See SZOR, *supra* note 22, at 309-11. See also J. NAZARIO ET AL., THE FUTURE OF INTERNET WORMS, 6 (2001), <http://www.blackhat.com/presentations/bh-usa-01/JoseNazario/bh-usa-01-Joes-Nazario.pdf>.

75. See SZOR, *supra* note 22, at 331.

76. File Transfer Protocol, or FTP, is a popular file transfer program used to move files across networks. See SLADE, *supra* note 5, at 84-85.

77. See SKOUDIS, *supra* note 28, at 82-83.

message or graphic image. Some payloads perform more destructive acts, such as deleting, stealing, or corrupting information. Payloads may also install spyware, disable anti-virus software, and open up a backdoor to allow remote access to an attacker.⁷⁸

III. Viruses and Worms As Threats to Information Security

Modern information security has three basic components: (1) confidentiality, (2) integrity, and (3) availability.⁷⁹ Confidentiality refers to the prevention of unauthorized access to sensitive information.⁸⁰ Integrity refers to the protection of digital data from unauthorized change, such as corruption or deletion.⁸¹ Availability refers to procedures and safeguards ensuring that authorized users have access to information in a convenient format when it is needed.⁸² Computer viruses and worms threaten the components of information security through their capability to replicate, spread, and perhaps execute a payload.⁸³ The infection module also serves to export and multiply the effect of a payload, if the virus has a payload.

A. Malevolent Code Threatens the Confidentiality of Information

A virus or worm can be programmed to access and steal confidential information on a system.⁸⁴ The W32/Bugbear@mm family of viruses, for instance, was designed to exploit vulnerabilities in the Outlook e-mail program to gain access to machines, steal confidential information using a key-logging function, and interfere with antivirus software. It also created a backdoor for hackers to take over the machine and misappropriate passwords and confidential financial information. Some members of the Bugbear family specifically targeted financial institutions.⁸⁵

78. See HARLEY, SLADE, & GATTIKER, *supra* note 29, at 88-89.

79. See MATT BISHOP, INTRODUCTION TO COMPUTER SECURITY, 1-6 (2005); LEHTINEN ET AL., *supra* note 26, at 9.

80. See LEHTINEN ET AL., *supra* note 26, at 9 (“Data is confidential if it stays obscure to all but those authorized to use it.”).

81. See *id.* at 11.

82. See *id.* at 9; PFLEEGER & PFLEEGER, *supra* note 7, at 17-20.

83. See HARLEY, SLADE, & GATTIKER, *supra* note 29, at 97 (“Direct damage can be considered in terms of the classic tripartite model (namely) Availability, Integrity, Confidentiality. Viruses . . . have an impact across all three areas described by this model, as well as other areas, such as accountability.”).

84. See SKOUDIS, *supra* note 28, at 34 (viruses can “steal files from your machine, especially sensitive ones containing personal, financial, or other sensitive information”). Viruses also monitor user keystrokes and transmit information about the user’s computing habits, Web sites visited, and financial information to the attacker. *Id.* at 3.

85. *Virus Makes Unwelcome Return*, BBC NEWS BULLETIN, June 5, 2003, <http://news.bbc.co.uk/1/hi/technology/2965924.stm>. See also July, 2003 CCS News: Monthly Virus Update: Klez.H, Sobig.E and Bugbear.B Worms Dominate, www.fbi.gov/ITSD/CIS

Viruses and worms often use fake e-mail addresses and Web sites to deceive users into disclosing confidential information. This technique is called “phishing.” The W32/Mimail.I@mm worm, for instance, displayed dialogues, purportedly from PayPal, requesting financial information from unwitting users. The stolen information would then be encrypted and transmitted to the attacker.⁸⁶

B. Malevolent Code Threatens the Integrity of Information

Viral payloads can be programmed to delete, modify, or corrupt information on infected computers.⁸⁷ The infamous Michelangelo virus, for instance, was programmed to overwrite part of the hard disk of its host.⁸⁸ The Hungarian Filler virus exhibited a misplaced sense of humor. Its payload deleted data and replaced the missing sectors with 0/1 characters arranged in the form of large smiley faces.⁸⁹

In January, 2003, a young Welshman, Simon Vallor, was sentenced to two years imprisonment for releasing fast-spreading viruses via e-mail that were designed to corrupt data on the hard drives of infected computers.⁹⁰ Viruses often corrupt information by replicating and spreading alone, without the help of a payload. Leading anti-virus researcher, Peter Szor, writes, “[v]irus replication has many side-effects. This includes the possibility of accidental data loss when the machine crashes due to a bug in the virus code or accidental overwriting of a part of the disk with relevant data. Virus researchers call this kind of virus a *no payload* virus.”⁹¹ The so-called Stone virus was a “no payload virus” which destroyed data by causing machines to crash, merely by prolific replication and spreading.⁹²

/compnews/2003/July/08-MVU.html (describing the W32.Sobig and Klez worms, which have been programmed to steal confidential information on infected machines); Gregg Keizer, *Virus Posing as Microsoft e-Mail Spreads Fast*, TECH WEB NETWORK, Sept. 19, 2003, <http://www.techweb.com/wire/story/TWB20030919S0005> (describing a fast-spreading worm which attempts to steal confidential information from infected systems).

86. SZOR, *supra* note 22, at 309.

87. See *Computer Virus*, THE COLUMBIA ENCYCLOPEDIA, (6th ed. 2007), <http://www.bartleby.com/65/co/computer-vir.html> (“Although some viruses are merely disruptive, others can destroy or corrupt data or cause an operating system or applications program to malfunction.”).

88. See SZOR, *supra* note 22, at 301.

89. *Id.* at 302.

90. *Computer Virus Author Jailed*, BBC NEWS BULLETIN, Jan. 21, 2003, http://news.bbc.co.uk/2/hi/uk_news/wales/2678773.stm.

91. SZOR, *supra* note 22, at 296, 297.

92. *Id.* at 297.

C. Malevolent Code Threatens the Availability of Information

Fast-spreading viruses make infected systems unavailable to legitimate users by monopolizing valuable computational resources.⁹³ A recent denial of service attack on the Port of Houston, for instance, made crucial navigating data on the port's Web service temporarily unavailable to shipping pilots and mooring companies, creating substantial risks of collisions and other issues.⁹⁴ The W32/Slammer worm overloaded Internet routers and slowed down networks worldwide, making it difficult to use e-mail. The paralyzing effect of Slammer on the Internet also caused ATM failures and interfered with elections.⁹⁵ The Sasser worm scanned so aggressively for new target computers that it caused networks to become congested and slow down. In Australia, Sasser disrupted Railcorp trains and brought down the computer system of a major Australian financial institution, Westpac Bank. In the UK, Sasser caused flight delays and brought down the computerized mapping systems of several coastguard stations.⁹⁶

IV. Computer Security Vulnerabilities

A security vulnerability is an error in an information system that an intruder can exploit to violate the system's security policy.⁹⁷ A system's security policy protects the confidentiality, integrity, and availability of information contained in the system by controlling access to the system.⁹⁸ The information system of a bank may, for instance, allow a customer to

93. See HARLEY, SLADE & GATTIKER, *supra* note 29, at 94 ("Network and mail viral programs carry, in a sense, their own payloads. The reproduction of the programs themselves use the resources of the hosts affected and, in the cases of both the Morris Internet and CHRISTMA worms, went so far as to deny service by using all available computing or communications resources."); GREG HOGLUND & GARY MCGRAW, EXPLOITING SOFTWARE: HOW TO BREAK CODE 20 (2004) ("Worms allow an attacker to carpet bomb a network in an unbridled exploration that attempts to exploit a given vulnerability as widely as possible. This amplifies the overall effect of an attack and achieves results that could never be obtained by manually hacking one machine at a time."). See also SZOR, *supra* note 22, at 306-307.

94. See generally GIBSON, *supra* note 4.

95. SZOR, *supra* note 22, at 306.

96. *Worm Brings Down Coastguard PCs*, BBC NEWS BULLETIN, May 4, 2004, <http://news.bbc.co.uk/2/hi/technology/3682803.stm>.

97. Fithen et al., *supra* note 6, at 174 (defining a vulnerability as "an unplanned system feature that an intruder may exploit, if he/she can establish certain preconditions, to achieve particular impacts on that system that violate its security policy"). See also SYMANTEC INTERNET SECURITY THREAT REPORT TRENDS FOR JULY 05 – DECEMBER 05 45 (2006), <http://www.symantec.com/business/theme.jsp?themeid=threatreport> (follow link for Vol. IX, March 2006) (defining vulnerabilities as "design or implementation errors in information systems that can result in a compromise of the confidentiality, integrity, and/or availability of information stored upon or transmitted over the affected system").

98. Fithen et al., *supra* note 6, at 174.

access her own account through login authentication, but restrict her from accessing any other information. A system vulnerability facilitates violations of such a security policy. A backdoor in an automated bank teller program, for instance, may allow a rogue who enters a special number on the keypad to capture sensitive information, such as the personal identity numbers (“PIN”) of previous users.⁹⁹

Fast-spreading worms often propagate via specific, known vulnerabilities. The W32/CodeRed worm, for instance, propagates on vulnerable servers running Microsoft’s Internet Information Server (IIS) software. CodeRed infects a server by exploiting a buffer overflow vulnerability¹⁰⁰ in Microsoft’s IIS to inject itself into the server’s memory. It then executes and propagates further by searching IP addresses for new vulnerable Web servers to infect.¹⁰¹ CodeRed depends on the presence of this specific vulnerability to propagate. More flexible worms, such as W32/Welchia, are capable of exploiting multiple vulnerabilities to invade any system on which at least one of the exploitable vulnerabilities is present.¹⁰²

Vendors usually promptly issue software patches to fix newly discovered vulnerabilities. Users tend to be slow in implementing vendor-provided patches, however, so that vulnerabilities often remain susceptible to exploitation.¹⁰³ Successive generations of CodeRed, for instance, continued to plague the Internet despite the fact that details of the attacks and the exploited vulnerability had been widely publicized, and that a security patch to fix the vulnerability had been made available even before the original CodeRed attack.¹⁰⁴

99. PFLEEGER & PFLEEGER, *supra* note 7, at 116.

100. See *infra* IV(A) for a discussion of the buffer overflow vulnerability.

101. PFLEEGER & PFLEEGER, *supra* note 7, at 137.

102. See SZOR, *supra* note 22, at 98.

103. See Stephen E. Henderson & Matthew E. Yarbrough, *Suing the Insecure?: A Duty of Care in Cyberspace*, 32 N.M. L. REV. 11, 16 (2002) (many computer systems are knowingly insecure, in part on account of failure to install readily available software patches). See also George V. Hulme, *One Step Ahead*, INFORMATIONWEEK, May 20, 2002 available at http://www.informationweek.com/news/software/development/showArticle.jhtml;jsessionid=TC5IJZGD1Z1EMQSNLPCKH0CJUNN2JVN?articleID=6502396&_requestid=693698 (citing estimates suggesting that even if 90 percent of the users of a particular technology with a newly discovered vulnerability could be trusted to implement the security patch issued by the vendor, the remaining unpatched systems could still allow enough hijackings to launch a denial of service attack on millions of other systems and networks). The CodeRed attacks occurred shortly after Microsoft had discovered the vulnerability and issued a patch to fix it.

Virus News, BOULDER.NET, www.cuisine.net/virus.htm (last visited Feb. 19, 2008).

104. See JOSE NAZARIO, DEFENSES AND STRATEGIES AGAINST INTERNET WORMS 98 (2004).

A. The Buffer Overflow

The so-called buffer overflow has been exploited by worms such as the Morris worm and CodeRed, amongst many others, and is (currently) the most commonly exploited security vulnerability.¹⁰⁵ A buffer overflow vulnerability allows executable malevolent code to be copied into the memory of a target computer. A skillful attacker can then manipulate the invaded computer to remotely execute the injected code.

1. What is a buffer?

Buffers are data storage areas in computer memory with limited capacity. Buffers often function as temporary storage for data to be transferred between two devices that are not operating at the same speed. The purpose of the temporary storage is to coordinate speed differentials between the adjacent devices. A printer, for instance, is not capable of printing data at the speed at which it receives the data from a computer. A buffer in the interface between the computer and printer resolves this bottleneck, by receiving the data from the computer and temporarily storing it. The buffer then relays the information to the printer, at the printer's speed, while the computer is freed up to carry on with other tasks.¹⁰⁶

A buffer overflow occurs when a program attempts to fill a buffer with more data than it was designed to hold. A buffer overflow is analogous to pouring ten ounces of water into a glass designed to hold eight ounces. The water must obviously overflow somewhere and create a mess. The glass represents a buffer and the water the application or user data.¹⁰⁷ The excess data typically overflows into adjacent memory locations where it can corrupt existing data, possibly changing the instructions, which results in unintended executions.

The "unintended executions" could be relatively harmless, but could also be malicious by design. In a relatively benign scenario, the buffer overflow will merely cause the corrupted program to abort, without much further harm.¹⁰⁸ In a darker scenario, a buffer overflow could allow a

105. See Benjamin A. Kuperman et al., *Detection and Prevention of Stack Buffer Overflow Attacks*, COMM. OF THE ACM, November 2005 at 50 (describing the buffer overflow vulnerability as "a security vulnerability that has been discussed for 40 years yet remains one of the most frequently reported types of remote attacks against computer systems"); SZOR, *supra* note 22, at 413. See also CHIEN & SZOR, *supra* note 67.

106. WILLIAM S. DAVIS & T.M. RAJKUMAR, *OPERATING SYSTEMS: A SYSTEMATIC VIEW*, 27, 28 (6th ed., 2004).

107. MARK E. DONALDSON, *INSIDE THE BUFFER OVERFLOW ATTACK: MECHANISM, METHOD, & PREVENTION* 3 (GSEC vers. 1.3, 2002), available at https://www2.sans.org/reading_room/whitepapers/securecode/386.php?portal=9d07851dff8251a34a4bcaacc23dcee3.

108. A buffer overflow may, for instance, abort the application program, resulting in a segmentation fault and core dump. See, e.g., RANDAL E. BRYANT ET AL., *COMPUTER SYSTEMS:*

hacker to remotely execute malicious code in a target computer. The next subsection describes how an attacker can exploit a buffer overflow vulnerability to achieve this objective.

2. *Exploitation of a Buffer Overflow*

A computer program consists of a set of instructions and a set of data on which the instructions operate. In the case of the common login procedure, for instance, the data consist of the user-provided identification name and password. The instructions of the login program parse the input data, and authenticate the stated identity of the user by validating the password, provided the password and identity match. When its identity is authenticated, the user is allowed to access the system.

A program occupies a memory buffer consisting of a text segment and a stack segment. The text segment contains the program instructions, and the stack contains data.¹⁰⁹ A final instruction in the buffer contains the return address, which specifies the instruction to be executed next.

A program such as the login procedure that accepts external input is a possible entry point for an attacker's malicious code.¹¹⁰ Instead of providing a valid user name, an attacker could enter characters representing malicious code. The computer would read the attacker's input into the stack segment in order to determine its validity. The attacker has at this point injected his malicious code into a memory buffer allocated to the currently active program, the login procedure. The attacker now needs to instruct the computer to execute the (malicious) contents of the buffer.

A carefully crafted attack strategy would not only fill the stack with malicious code, but also, overflow the stack and overwrite the adjacent return address. The return address contains an instruction pointer that

A PROGRAMMER'S PERSPECTIVE, 593 (2003). A core dump is the recorded state of the working memory of a computer program at a specific time, generally when the program has terminated abnormally (crashed). In practice, other key pieces of program state are usually dumped at the same time, including the processor registers, which may include the program counter and stack pointer, memory management information, and other processor and operating system flags and information. The name comes from the once-standard memory technology core memory. Core dumps are often used to diagnose or debug errors in computer programs. On many operating systems, a fatal error in a program automatically triggers a core dump, and by extension the phrase "to dump core" has come to mean, in many cases, any fatal error, regardless of whether a record of the program memory is created. The term is used in jargon to indicate any circumstance where large amounts of unedited data are deposited for further examination.

109. In the case of a login program, the text segment contains the instructions that examine a user's input to determine its validity. The stack segment stores the user input.

110. See SZOR, *supra* note 22, at 413 ("Computer worms typically attack service processes and daemon programs that are waiting to handle incoming requests by listening on various TCP/UDP ports. Any such communication service could potentially contain flaws, as did the fingerd (in the case of the Morris worm), the BIND (in the case of the ADM worm), and the Microsoft IIS (in the case of the CodeRed worm).").

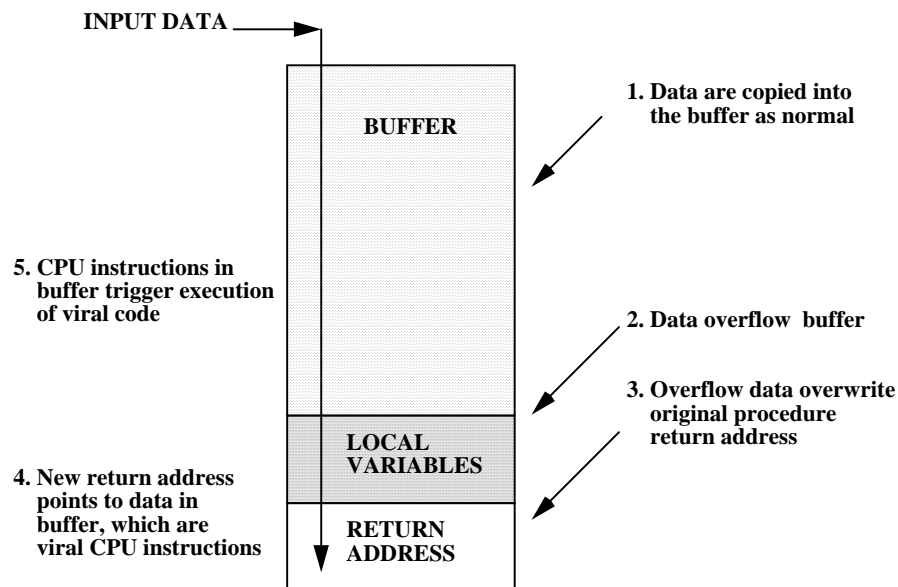
defines the next instruction to be executed. By overwriting this pointer with the address of the stack segment, the attacker can instruct the computer to execute the malicious content of the stack.

The attacker has now injected malicious code into memory and ensured that control will pass to the malicious code. Furthermore, the attacker has also ensured that the code will execute at the privilege level of the running program. If, for instance, the running program has control over confidential information, such as passwords and financial data, the malicious code will have the same privilege.

In summary, the most basic elements of a buffer overflow attack are as follows: (1) data are copied into the buffer, (2) the data overflows the buffer, (3) the overflow data overwrites the original procedure return address, (4) the new return address now points to the new data in the buffer, which may include malevolent instructions, and (5) these instructions trigger execution of the virus.¹¹¹

Schematically, this process works as illustrated below:¹¹²

BASIC BUFFER OVERFLOW MECHANISM



111. This section has described the classic buffer overflow exploit mechanism. Buffer overflow exploits come in many varieties. See SZOR, *supra* note 22, at Chapter 10, 365-421.

112. The diagram is adapted from ROB ENDERLE AND JASMINE NOEL, THE NEW APPROACH TO WINDOWS SECURITY 7 (2004), available at [http://globalwatchtech.com/resources/New_Approach_to_PC_SecurityFinal_\(2\).pdf](http://globalwatchtech.com/resources/New_Approach_to_PC_SecurityFinal_(2).pdf).

V. The Duty to Safeguard Confidential Information

Cyber intruders may be subject to criminal¹¹³ and civil liability.¹¹⁴ Victims may recover damages from attackers under common law tort,¹¹⁵ as well as under the civil liability provisions of the federal Computer Fraud and Abuse Act.¹¹⁶ Hackers, however, are often judgment-proof because it is difficult to identify them and to subject them to jurisdiction.¹¹⁷ For these reasons, database owners, such as deep-pocketed financial institutions or medical service providers who negligently fail to prevent an attack, are usually the preferred targets for civil lawsuits.¹¹⁸ Whether a database owner is liable under tort depends on whether he owes a legal duty of care to safeguard the data contained in the database. Such a duty may be imposed by statute or by common law tort principles.¹¹⁹

A. Statutory Duty of Care

A statute may impose a duty to exercise due care on a database owner to protect data from intruders, either expressly,¹²⁰ or implicitly through

113. See CRITICAL INFORMATION INFRASTRUCTURE PROTECTION AND THE LAW: AN OVERVIEW OF KEY ISSUES 37-39 (Stewart D. Personick & Cynthia A. Patterson eds., 2003), available at <http://www.nap.edu/openbook.php?isbn=030908878X>. See generally 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).

114. See Robin A. Brooks, *Detering the Spread of Viruses Online: Can Tort Law Tighten the Net?*, 17 REV. OF LITIG. 343 (1998); Michael L. Rustad, *Private Enforcement of Cybercrime on the Electronic Frontier*, 11 S. CAL. INDERDISC. L.J. 63, 66 (2001) (stating that tort remedies “will play an increasingly important role in punishing and deterring fraud, hacking, and other wrongdoing on the Internet”).

115. See generally Brooks, *supra* note 114; de Villiers, *supra* note 33, at 123 (an analysis of common law tort principles, including damages, in the information security context); Meiring de Villiers, *Virus ex Machina Res Ipsa Loquitur*, 1 STAN. TECH. L. REV. 1 (2003) (quantitative analysis of circumstantial evidence of negligent virus transmission).

116. 18 U.S.C. § 1030(g) (2000) (allowing civil action against violator “to obtain compensatory damages and injunctive relief or other equitable relief”).

117. See de Villiers, *supra* note 12, at § 4(A)

118. See *id.* (an analysis of civil liability for enablement of cyber crime and tort); W. REID WITTLIFF, *COMPUTER HACKING AND LIABILITY ISSUES: WHEN DOES LIABILITY ATTACH?* 11, available at http://www.gdhn.com/pdf/wrw-hack_article.pdf (Discussing tort theories that may be utilized to recover damages from defendants whose negligence enabled a cyber attack.).

119. See, e.g., Citron, *supra* note 2, at 261-68; de Villiers, *supra* note 33, at 123; Henderson & Yarbrough, *supra* note 103, at 11; Sarah Faulkner, Comment, *Invasion of the Information Snatchers: Creating Liability for Corporations with Vulnerable Computer Networks*, 18 J. MARSHALL J. COMPUTER & INFO. LAW 1019 (2000); W. Reid Wittliff, *supra* note 118.

120. See RESTATEMENT (THIRD) OF TORTS § 14 cmt. b (Proposed Final Draft No. 1, 2005) (discussing express and implied statutory causes of action).

legal precedent.¹²¹ California's Security Breach Information Act (SBIA),¹²² for instance, expressly creates a civil cause of action against a person or entity who fails to protect customers' personal information.¹²³ It also provides that aggrieved customers may recover damages for the defendant's breach of that duty.¹²⁴ 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."¹²⁵ The legislation further provides, "[a]ny customer injured by a violation of this title may institute a civil action to recover damages."¹²⁶

The civil cause of action created by the SBIA is rooted in negligence principles, because the duty it imposes is based on reasonableness. The statute requires, for instance, implementation of "reasonable security procedures and practices"¹²⁷ that are "appropriate to the nature of the information."¹²⁸ The essence of the standard of care imposed by negligence law is reasonableness and the duty to act as a reasonable person would under the circumstances.¹²⁹ Negligence is generally described as conduct "which falls below the standard established by law for the protection of others against unreasonable risk of harm."¹³⁰ Professors Michael L. Rustad and Thomas H. Koenig comment that, "[a]lthough courts vary in what impact a statutory violation has on the adjudication of negligence, they may employ civil statutes to set standards in negligent

121. See VINCENT R. JOHNSON & ALAN GUNN, *STUDIES IN AMERICAN TORT LAW* 305-06 (3d ed., 2005) ("In the one case, the court is saying that the legislation sets the standard because the legislature implicitly intended it to do so, and in the other case, the court acknowledges that the statute sets the standard because the court thinks that it is a good idea. Either way, if the statute does not expressly create a cause of action, the essential inquiry is the same: was the law intended to protect this class of persons from this type of harm.").

122. Cal. Civ. Code §1798.81 *et. seq.* (West Supp. 2005).

123. *Id.*

124. *Id.*

125. *Id.* at §1798.81.5.

126. *Id.* at §1798.84(b).

127. *Id.*

128. *Id.* at §1798.81.5(c).

129. RESTATEMENT (SECOND) OF TORTS § 283 (1965).

130. See Richard W. Wright, *The Standards of Care in Negligence Law*, in *PHILOSOPHICAL FOUNDATIONS OF TORT LAW* 249 (David G. Owen ed., 1995) (stating that negligence "is generally described as behavior that creates unreasonable foreseeable risks of injury"). See also RESTATEMENT (SECOND) OF TORTS, § 282 (1965); PROSSER & KEETON, *PROSSER AND KEETON ON THE LAW OF TORTS* § 31 (5th ed., 1984).

enablement lawsuits. An unexcused violation of a statute requiring reasonable security is itself negligence, that is, negligence per se.”¹³¹

B. Common Law Tort

Common law tort principles may also impose on a database owner a duty to safeguard information from intruders.¹³² A person whose private information has been compromised may pursue a civil action under a theory of negligence for this harm, which is the most widely used theory of liability in the law of torts.¹³³

Negligence is generally defined as a breach of the duty not to impose an unreasonable risk on society.¹³⁴ It applies to any risk that can be characterized as unreasonable, including risks associated with information security breaches. 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.¹³⁵

The plaintiff in a negligence action has to prove the following elements to establish her claim: (1) the existence of a legal duty on the part of the defendant not to expose the plaintiff to unreasonable risks, (2) a breach of the duty, namely a failure on the part of the defendant to conform to the norm of reasonableness, (3) a causal connection between defendant’s conduct and plaintiff’s harm,¹³⁶ and (4) actual damage to the plaintiff resulting from the defendant’s negligence.

131. Michael L. Rustad and Thomas H. Koenig, *The Tort of Negligent Enablement of Cybercrime*, 20 BERKELEY TECH. L.J. 1553, 1592. *See also* RESTATEMENT (SECOND) OF TORTS § 285, cmt. c (“Even where a legislative enactment contains no express provision that its violation shall result in tort liability, and no implication to that effect, the court may, and in certain types of classes customarily will, adopt the requirements of the enactment as the standard of conduct necessary to avoid liability for negligence. The same is true of municipal ordinances and administrative regulations.”). *See also* 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.”).

132. Mark J. Maier, *Backdoor Liability from Internet Telecommuters*, 6 COMP L. REV. & TECH. J. 27, 55 (2001).

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

134. PROSSER & KEETON, *supra* note 130, at § 31. RESTATEMENT (SECOND) OF TORTS, § 282 (describing negligence as conduct “which falls below the standard established by law for the protection of others against unreasonable risk of harm”).

135. DAN B. DOBBS, *THE LAW OF TORTS*, 258 (2005) (The plaintiff can assert that *any* conduct counts as negligence.).

136. This element includes actual as well as proximate cause. Defendant’s negligence is the actual cause of the plaintiff’s harm if, but for the negligence, the harm would not have occurred.

Multiple lawsuits, including class actions, have been filed against database possessors for enabling cyber intrusions that resulted in compromise of confidential information.¹³⁷ In June 2005, CardSystems, Inc., a credit card processor, publicly acknowledged that hackers had illegally accessed its credit card database.¹³⁸ A class of 40 million plaintiffs promptly filed a class action suit¹³⁹ alleging that CardSystems had failed to adequately protect sensitive information from unauthorized access.¹⁴⁰

The Federal Trade Commission (FTC) regularly brings lawsuits against companies for enabling theft of confidential customer information by failing to patch foreseeably exploitable vulnerabilities.¹⁴¹ The FTC's authority to enforce information security practices derives from Section 5 of the FTC Act,¹⁴² which prohibits unfair or deceptive acts or practices, and the FTC's Gramm-Leach-Bliley Safeguards Rule.¹⁴³

In its lawsuit against the clothing manufacturer Guess?, the FTC alleged that the company had breached its promise to safeguard the confidentiality of customer information collected through its Web site.¹⁴⁴ According to allegations in the FTC complaint, the Guess? Web site had been vulnerable to reasonably foreseeable attacks from intruders seeking access to confidential customer information.¹⁴⁵ In particular, Guess? had allegedly failed to patch a security vulnerability that made the site susceptible to SQL injection attacks by the SQL Slammer worm.¹⁴⁶ According to the FTC complaint, the fast-spreading Slammer foreseeably exploited a buffer overflow vulnerability in several Microsoft products,

The proximate causation element requires the defendant's conduct to be reasonably related to the plaintiff's harm.

137. See, e.g., *Harrington v. Choicepoint, Inc.*, No. CV05 1294 SJO (C.D. Cal. filed Feb. 22, 2005); *Parke v. Cardsystems Solutions, Inc.*, No. CGC-05-442624 (Cal. Super. Ct. filed June 27, 2005), available at <http://www.techfirm.com/cardsystems.pdf>.

138. See Jonathan Krim and Michael Barbaro, *40 Million Credit Card Numbers Hacked: Data Breached at Processing Center*, WASH. POST, June 18, 2005, at A01.

139. *Parke*, No. CGC-05-442624.

140. *Id.* at 2, 5, 12, 19-20, 23, and 25.

141. See Beales, *supra* note 2, at 20 (when companies promise security, "they have a legal obligation to take reasonable and appropriate steps to guard against reasonably anticipated vulnerabilities").

142. 15 U.S.C. § 45 (2006)

143. 16 C.F.R. § 314.1-314.5 (2008).

144. In the Matter of GUESS?, Inc. and GUESS.com, Inc. Docket No. C-4091 (C.D. Cal. 2003), available at <http://www.ftc.gov/os/2003/06/guesscmp.htm> [hereinafter *Guess? Complaint*].

145. *Id.*

146. *Id.* For details on the Slammer worm, see David Moore et al., *Inside the Slammer Worm*, IEEE SECURITY & PRIVACY, July/August 2003, at 33.

including Microsoft's SQL server,¹⁴⁷ resulting in access to a Guess? customer database containing 191,000 credit card numbers.¹⁴⁸ The FTC complaint further alleged that Guess? had been aware of the vulnerabilities since October 2000, that the stolen information was sensitive, and that the fix was relatively inexpensive.¹⁴⁹ The case settled in June 2003.¹⁵⁰

In an administrative proceeding decided by the Maine Public Utilities Commission (PUC), the telecommunications company Verizon applied for a waiver of wholesale performance metrics.¹⁵¹ Verizon pleaded that it had failed to meet its performance standards because the SQL Slammer worm had attacked its servers.¹⁵² The PUC rejected Verizon's arguments, concluding that the telecommunications company had acted unreasonably by failing to apply a security patch issued six months earlier by Microsoft.¹⁵³ The PUC reasoned that cyber attacks by viruses and worms, such as those experienced by Verizon, were foreseeable, and that Microsoft had made patches available for vulnerabilities in their products, including the vulnerability specifically exploited by Slammer.¹⁵⁴ The PUC noted that companies such as AT&T and WorldCom had installed the Microsoft patches, and consequently escaped the Slammer attacks.¹⁵⁵

147. See Vulnerability Note, US-CERT, Microsoft SQL Server 2000 Contains Heap Buffer Overflow in SQL Server Resolution Service, VU#399260, <http://www.kb.cert.org/vuls/id/399260>.

148. Guess? Complaint, at 3-4.

149. *Id.* at 3.

150. Press Release, FTC, Guess Settles FTC Security Charges: Third FTC Case Targets Claims about Information Security, (June 18, 2003), available at <http://www.ftc.gov/opa/2003/06/guess.htm> (last modified June 25, 2007). See generally STEWART BAKER AND MAURY SHENK, A PATCH IN TIME SAVES NINE: LIABILITY RISKS FOR UNPATCHED SOFTWARE, (2003), available at www.stepto.com/assets/attachments/631.pdf.

151. *In re Verizon Related Reduction Claim*, Me Pub. Util. Comm'n, Docket No. 2000-849 (April 30, 2003).

152. Inquiry Regarding the Entry of Verizon-Maine Into the InterLATA Telephone Market Pursuant To Section 271 of Telecommunications Act of 1996, Maine Pub. Util., Dkt. No. 2000-849 (Apr. 30, 2003) (Order).

153. *Id.*

154. *Id.* See generally, Welcome to CERT, <http://www.cert.org> (last visited Feb. 19, 2008); SANS Institute – SANS Top-20 Security Risks 92007 Annual Update, <http://www.sans.org/top20.htm> (last visited Feb. 19, 2008).

155. *In re Verizon Related Reduction Claim*, Docket No. 2000-849. See also Citron, *supra* note 2, at 256, n.82 (discussing other information security breach cases pursued by the FTC during 2005-06).

C. Foreseeability

Foreseeability is a crucial concept in negligence law.¹⁵⁶ Foreseeability defines whether the defendant owed a duty to the plaintiff,¹⁵⁷ whether the defendant breached a duty,¹⁵⁸ and whether the defendant's breach proximately caused the plaintiff's injury.¹⁵⁹ A person who commits an affirmative act that creates a risk of harm, for instance, owes a duty of care to everyone foreseeably within the scope of the risk.¹⁶⁰ Courts have denied a duty based on absence of foreseeability, even where the defendant's conduct created a risk of harm.¹⁶¹

Courts require a plaintiff to prove breach of duty in a negligence action by identifying an untaken precaution, and showing that the precaution would have yielded greater benefits in risk reduction than its cost.¹⁶² The role of the untaken precaution in negligence law is well

156. See generally PROSSER & KEETON, *supra* note 130, at § 43 (discussing foreseeability in torts). See also W. Jonathan Cardi, *Reconstructing Foreseeability*, 46 B.C. L. REV. 921, 921-922 (2005) ("The concept of foreseeability is fast devouring the negligence cause of action. Foreseeability of a risk of injury has for centuries rested at the heart of court determinations of whether a defendant breached its duty of care. More recently, foreseeability of a particular plaintiff's injury has become central to the element of proximate cause. Foreseeability's most aggressive advance of late, however, has been into the realm of duty.").

157. See 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, 755 (2005) (Section 2. "Duty and Foreseeability"); Cardi, *supra* note 156, at 921, 923 ("Foreseeability remains a pervasive consideration in many courts' duty analyses.").

158. See Cardi, *supra* note 156, at 921 ("Foreseeability of a risk has for centuries rested at the heart of court determinations of whether a defendant breached its duty of care."). See also RESTATEMENT (THIRD) OF TORTS § 4 (Discussion Draft 1999) ("Primary factors to consider in ascertaining whether conduct lacks reasonable care are the foreseeable likelihood that it will result in harm, the foreseeable severity of the harm that may ensue, and the burden that would be borne by the actor and others if the actor takes precautions that eliminate or reduce the possibility of harm.").

159. See PROSSER & KEETON, *supra* note 130, at § 43. See also Mark F. Grady, *Proximate Cause Decoded*, 50 UCLA L. REV. 293, 323 (2002).

160. See, e.g., *Brennen v. Docusearch, Inc.*, 591 P.2d 719, 723 (Or. 1979) (stating that a duty is created where the defendant "created a foreseeable risk of harm to others"); *Greater Houston Transp. Co. v. Phillips*, 801 S.W.2d 523, 525 (Tex. 1990) ("In determining whether the defendant was under a duty, the court will consider several interrelated factors, including the risk, foreseeability, and likelihood of injury weighed against the social utility of the actor's conduct, the magnitude of the burden of guarding against the injury, and the consequences of placing the burden on the defendant. Of all these factors, foreseeability of the risk is the foremost and dominant consideration.").

161. See, e.g., *Herrera v. Quality Pontiac*, 73 P.3d 181, 187 (N.M. 2003). See also Cardi, *supra* note 156, at 930 ("Foreseeability has become so central a concept in many courts' duty analyses that a ruling on foreseeability is outcome-determinative.").

162. Mark F. Grady, *Untaken Precautions*, 18 J. LEGAL STUD. 139, 146 (1989). See also Cardi, *supra* note 156, at 925 (the breach calculus turns on (1) "the foreseeable likelihood that the defendant's actions might result in injury", (2) the severity the range of foreseeable injuries, and (3) the cost of available precautions that would have prevented the injury).

illustrated in *Cooley v. Public Service Co.*¹⁶³ In *Cooley*, the plaintiff suffered harm from a loud noise over a telephone wire. She suggested that the defendant could have taken two untaken precautions to prevent the harm: (1) strategically positioned a wire mesh basket on the wires and (2) insulated the wires.¹⁶⁴ The court ruled that neither untaken precaution constituted a breach of duty.¹⁶⁵ Both precautions would have increased the risk of electrocution to passersby sufficiently to outweigh the benefits in harm reduction. Any foreseeable risk can be included in the cost-benefit calculation, as long as it would have been reduced by the untaken precaution.¹⁶⁶

The proximate cause doctrine limits the plaintiff's recovery to harm reasonably related to the defendant's wrongdoing.¹⁶⁷ Foreseeability plays a central role in proximate cause.¹⁶⁸ Professor Jonathan Cardi has commented on the role of foreseeability, stating "[A] plaintiff may fail to survive the 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."¹⁶⁹

D. Legal Meaning of Foreseeability

An event is the foreseeable result of an action if the action ex ante systematically increased the likelihood of the event. This would be the case if the action either created the risk of the event or increased the likelihood of its materialization.¹⁷⁰ In *Bunting v. Hogsett*,¹⁷¹ for instance, the driver of a dinky train negligently failed to pay proper attention upon approaching a railroad intersection and collided with a passenger train at the intersection. Injured passengers on the passenger train filed suit against the owner of the dinky line. The court found the driver of the dinky train was negligent in

163. 10 A.2d 673 (N.H. 1940).

164. *Id.* at 675.

165. *Id.* at 676.

166. Grady, *supra* note 162, at 146.

167. See DOBBS, *supra* note 135, at 444.

168. See, e.g., MARC A. FRANKLIN AND ROBERT L. RABIN, TORT LAW AND ALTERNATIVES 399 (7th ed. 2001). See generally Grady, *supra* note 162.

169. Cardi, *supra* note 157, at 749.

170. See Grady, *supra* note 162, at 323 (stating plaintiff must show that the untaken precaution would have reduced the risk of the accident at issue. If not, the plaintiff fails on proximate cause grounds). See also Stephen R. Perry, *Responsibility for Outcomes, Risk, and the Law of Torts*, in PHILOSOPHY AND THE LAW OF TORTS 72, 97 (Gerald Postema, ed., 2001) (reasonable foreseeability is "a function of two separate effects: (1) the objective probability of an event occurring, and (2) a reasonable person's knowledge and beliefs about that probability."); *Id.* at 98 (explaining that proximate causation views the plaintiff's harm "from the standpoint of an appropriately general description of the risk created by the defendant.").

171. 21 A. 31 (Pa. 1891).

failing to keep a proper lookout. The dinky engineer's failure to look out for the passenger train created the exact risk that caused the plaintiff's injuries, namely the risk of a collision.¹⁷²

The basic test of foreseeability in negligence law can also be described as "whether one can see a systematic relationship between the type of accident that the plaintiff suffered" and the defendant's action.¹⁷³ An illustration of foreseeability as a systematic causal relation is found in *O'Malley v. Laurel Line Bus Co.*¹⁷⁴ In *O'Malley*, the defendant's bus driver let a passenger off in the middle of a street instead of at the regular bus stop. It was a dark and stormy night so that the passenger did not realize where he was being let off. The court held the defendant liable for injuries sustained when a car struck the passenger. There is a systematic causal relationship between letting people off in the middle of a street under such conditions that they cannot ascertain the risks of dangerous traffic, and their being struck by a passing car.

Coincidental harm is not foreseeable. Suppose, for instance, a defendant negligently exceeds the speed limit and arrives at a spot just in time to be struck by a falling tree. Although a plaintiff such as an injured passenger may argue credibly that falling trees are foreseeable, the accident is likely outside the scope of the risk created by the defendant's speeding. The defendant's speeding created risks of traffic accidents, but it neither created the risk that materialized nor made it more likely. The accident was therefore not within the scope of foreseeable risk created by the defendant's conduct. The accident was coincidental and not systematically related to the defendant's negligence, and was hence, unforeseeable. The outcome would likely have been different if, instead, a tree had fallen in front of the speeding driver, and the car crashed into it. If it can be shown that the accident could have been avoided had the driver traveled at a reasonable speed, then the speeding driver's negligence may have been a proximate cause of the accident. Failure to stop within a short time window is a foreseeable risk of speeding.¹⁷⁵

Foreseeability is not necessarily a reflection of the objective probability of an event, but rather, is a reflection of what a reasonable person would foresee under the circumstances.¹⁷⁶ This degree of foresight may be equal to the objective probability of the event, or it may be a

172. *Id.* at 31. The case is much more interesting than this brief description reveals. See Grady, *supra* note 159, at 304-05 (discussing case in more detail).

173. See Grady, *supra* note 159, at 323.

174. 166 A. 868 (Pa. 1933).

175. See *Berry v. Borough of Sugar Notch*, 191 Pa. 345 (1899). See also Grady, *supra* note 159, at 324.

176. See ARTHUR RIPSTEIN, *EQUALITY, RESPONSIBILITY, AND THE LAW* 94 (1999).

fraction thereof.¹⁷⁷ This fraction may be zero if the defendant is reasonably ignorant of the systematic relationship between her act and a plaintiff's injury. The "Reasonable Ignorance of the Relationship" doctrine, proposed by Professor Mark Grady, formalizes this scenario.¹⁷⁸ Under this doctrine, the defendant's liability is cut off when, even though in hindsight there is clearly a systematic relationship between the defendant's untaken precaution and the plaintiff's harm, scientists would not have predicted the relationship. The following case illustrates the doctrine.

In *Doughty v. Turner Manufacturing Co.*,¹⁷⁹ foreseeability turned on scientific state of the art. A worker negligently knocked the cover of a vat containing molten sodium cyanide into the molten liquid in the vat. The plaintiffs were injured when a chemical reaction between the molten sodium cyanide and the cover, which was made of a combination of asbestos and cement, known as sindayo, caused an eruption. The risk that the cover might splash the molten liquid onto someone was known and foreseeable, but the chemical reaction that actually caused the harm was unknown and unpredictable at the time of the accident. Scientists later demonstrated that at sufficiently high temperatures the sindayo compound would undergo a chemical change which creates steam. It was steam created in this manner that caused the eruption that injured the plaintiff in *Doughty*. None of this was known at the time of the accident. The court therefore held for the defendant, stating that the defendant was reasonably ignorant of the chemical reaction that caused the injuries.¹⁸⁰ The defendant thus escaped liability under the Reasonable Ignorance doctrine.

VI. Reasonable Foreseeability in Information Security

A. Common Law Background

This article focuses on situations where a database owner fails to patch a security vulnerability, thereby paving the way for a cyber attacker to obtain unauthorized access to confidential information. Professor Robert Rabin has termed wrongdoing of this kind an "enabling tort."¹⁸¹ An enabling tort is a negligent act by a primary tortfeasor that sets the stage for

177. See Stephen R. Perry, *Responsibility for Outcomes, Risk, and the Law of Torts*, in *PHILOSOPHY AND THE LAW OF TORTS* 72, 322 (Gerald J. Postema ed., 2001) ("foreseeability is often referred to as epistemic (or knowable) probability.").

178. See Grady, *supra* note 159, at 328.

179. [1964] 1 Q.B. 518.

180. *Id.*, at 520, 525.

181. See generally Rabin, *supra* note 8.

an intervening tortfeasor to commit a tort or crime.¹⁸² A construction worker may, for instance, negligently leave an unattended scaffold beside an open window, which enables a burglar to obtain access to valuables inside the building.¹⁸³

Professor Mark Grady has developed a theory explaining that a primary tortfeasor's liability will be preserved for conduct that foreseeably encouraged intervening tortfeasors who are so-called "free radicals."¹⁸⁴ Free radicals are individuals who are not deterred by the threat of liability because they are judgment-proof, anonymous, immature, or strongly motivated by ideological considerations. Free radicals include persons such as mentally incompetent people, terrorists, and criminals.¹⁸⁵ The free radical doctrine recognizes that the prospect of negligence liability is ineffective against defendants who are shielded from, or otherwise undeterred by, the prospect of liability. The deterrence rationale of negligence law would be defeated if responsible people who encourage free radicals were allowed to escape judgment by shifting liability to undeterrable individuals. Common law negligence rules therefore impose liability on a primary tortfeasor, even when intentional or criminal behavior by a free radical intervenes.¹⁸⁶ Research has shown that cyber rogues generally fit the profile of free radicals.¹⁸⁷

"Reasonable foreseeability" is an essential element of enabling torts. An intervening tortfeasor's behavior must have been foreseeable before the primary tortfeasor can be held liable for enabling the intervening tort or crime.¹⁸⁸ This article now turns to an analysis of factors that have convinced courts of the existence of the requisite foreseeability in enabling tort cases.

182. See Rabin, *supra* note 8, at 437 ("Beyond the immediate perpetrator of harm, the victim perceives the individual, or more often, the enterprise, that set the stage for the suffering that unfolded. The Enabler.").

183. See *Stansbie v. Troman*, [1948] 2 K.B. 48.

184. See Grady, *supra* note 10, at 189. See also Rabin, *supra* note 8, at 439 ("The key factor counseling liability . . . is that defendant paved the way for a truly reckless individual to be imposing serious risks of injury on the public at large.").

185. See Grady, *supra* note 10, at 191.

186. See Grady, *supra* note 10, at 196 ("When the encouraged people predictably lack exposure to tort law deterrence, the courts have concluded that more responsible people should be deterred from encouraging them."). See also Rabin, *supra* note 8, at 444. ("The main deterrence gap is the inability to effectively reach the putative wrongdoer himself, either through criminal or tort sanctions. This is the crux of the matter and the link to creating responsibility for enabling behavior.").

187. See generally de Villiers, *supra* note 12.

188. See Grady, *supra* note 10, at 447. See also RESTATEMENT (THIRD) OF TORTS § 17 (Discussion Draft 1999) ("The conduct of a defendant can lack reasonable care insofar as it can foreseeably combine with or bring about the improper conduct of the plaintiff or a third party.").

1. *Scarce Opportunity*

Courts are more likely to impose liability when the defendant has created a tempting or “scarce” opportunity that does not exist for the intervening tortfeasor in the “normal background of incitements and opportunities.”¹⁸⁹ A scarce opportunity is one that meets a wrongdoer’s objectives when he has few, if any equivalent, alternative opportunities conveniently available to him.

In *Sun Trust Banks, Inc. v. Killebrew*,¹⁹⁰ appellee Stephen Killebrew was shot by a robber on Sun Trust Bank’s premises after using an automated teller machine at night. Killebrew sued Sun Trust for failing to exercise due care to keep its premises safe in light of a prior similar criminal incident. The prior incident was reported to the police but not to the bank. The trial court granted summary judgment in favor of Sun Trust, but was reversed on appeal. Sun Trust appealed to the Supreme Court of Georgia.¹⁹¹

In his concurring opinion, Justice Sears concluded that, even absent evidence of prior criminal activity, Sun Trust should have foreseen criminal activity at its ATMs because of the unique opportunity for criminal activity they present. Justice Sears referred to studies by the banking industry suggesting that ATMs are reliable sources of funds not only for bank

189. See *Isaacs v. Huntington Mem'l Hosp.*, 38 Cal. 3d 112, 130 (1985) (imposing duty to protect against criminal acts, stating “defendants may be said to have created ‘an especial temptation and opportunity for criminal misconduct.’” (quoting *Gomez v. Ticor*, 145 Cal. App. 3d 622, 628 (1983))); *Sharon P. v. Arman, Ltd.*, 65 Cal. Rptr. 2d 640 (1997) (emphasizing that the unique nature of the parking complex in which the plaintiff was assaulted invited such acts); *Spitzak v. Hylands, Ltd.*, 500 N.W.2d 154, 156-57 (Minn. App. 1993) (declining to impose liability on landlord where building neither exposed tenants to unusual risks, nor presented a unique opportunity for criminal activity); *Grady*, *supra* note 159, at 310. See also Michael J. Yelnosky, Comment, *Business Inviter’s Duty to Protect Invitees From Criminal Acts*, 134 U. PA. L. REV. 883, 891 (1986) (noting that “courts frequently state that a duty to protect arises if a business provides a ‘unique climate for crime.’”).

190. 464 S.E.2d 207 (Ga. 1995).

191. The Court was presented with the issue of “whether a prior crime that was unreported and unknown to a property owner, but was reported to the police, was sufficient to give the property owner knowledge of a risk of criminal activity on its property so as to require it to take reasonable precautions to protect customers from similar risks.” *Id.* at 207. The Court held that the criminal incident did not create actual or constructive knowledge of the risk of criminal activity on the part of the defendant. It held further, that no evidence was presented that established the existence of a duty on the property owner to “search police records for reports of criminal activity on its premises.” In his concurring opinion, Justice Sears concluded that a jury question nevertheless remained regarding whether Sun Trust reasonably could have foreseen the risk of a robbery on its premises, on grounds other than the prior incident. Justice Sears concluded that, even absent evidence of prior criminal activity, Sun Trust should have foreseen criminal activity at its ATMs because of the unique opportunity for criminal activity presented by automated teller machines.

customers, but also, for criminals.¹⁹² A criminal can generally not be certain whether a person or premises he is attempting to rob is carrying any money, but a customer who has withdrawn money from an ATM is a guaranteed source of cash.¹⁹³ Automated teller machines, therefore, present a scarce opportunity to robbers, because ATMs are weakly secured and their patrons guaranteed to have money.¹⁹⁴

In *Home Office v. Dorset Yacht Co.*,¹⁹⁵ inmates who had been sentenced to working in a boot camp for juvenile offenders were working under supervision of three Home Office guards. One evening, in breach of their instructions to watch the delinquents, the guards simply went to bed, leaving the inmates unsupervised. The inmates swam out to an unattended yacht moored nearby and managed to set it in motion. They collided with another yacht owned by the plaintiffs, who sued the Home Office for the resulting damage. The trial court ruled in favor of the plaintiff, and the Court of Appeal affirmed.¹⁹⁶

The defendants' liability turned on the foreseeability of the delinquents' behavior.¹⁹⁷ Several factors made exploitation of the escape opportunity foreseeable. The inmates were juvenile offenders with records including convictions for breaking and entering, larceny, and grand theft auto. Five of the seven had a record of previous escapes from boot camp, and they likely were biding their time for another chance to break out. The escape opportunity was easy to exploit due to the total absence of supervision. It was also a scarce opportunity. Under normal circumstances, the only realistic way out for the inmates would be for them to physically incapacitate the guards or break out surreptitiously, both of which are more complicated operations.

If, unlike the situation in *Dorset Yacht*, a wrongdoer already has several equally attractive opportunities available for harmful behavior, the defendant's encouragement likely does not amount to a scarce opportunity.

192. See generally Richard E. Vogel, Note, *Institutional Liability for Attacks on ATM Patrons*, 1994 U. ILL. L. REV. 1009 (1994). See also Yelnosky, *supra* note 189, at 886 ("Commercial crime is more prevalent than household and personal victimization, and crime rate for retail stores and proprietary structures is higher than for other businesses. The parking lot seems to present unique opportunities for crime. Customers are typically in possession of money and recently purchased items. In this respect, the would-be assailant in search of valuable(s) need not take a chance on the unknown assets of some passerby.").

193. See Daniel J. Smith, *Liability of Bank For Criminal Attack at ATM or Night Depository*, in AM. JUR. 3d *Proof of Facts* § 5, at 514 (1989).

194. See Vogel, *supra* note 192, at 1010 (in light of the fear of ATM crime "and the danger presented by the combination of money, nighttime, and lax security, banks need to consider means of protecting the users of their money machines.").

195. [1970] 2 A.C. 1004 (appeal taken from Eng.)

196. *Id.*, at 1005.

197. *Id.*, at 1008-1010.

In *Gonzalez v. Derrington*¹⁹⁸ for instance, the defendant sold five gallons of gasoline in an open pail to a customer in violation of a municipal ordinance that prohibited sales of gasoline in open containers and in excess of two gallons. The purchaser subsequently used the gasoline to commit arson. The court declined to impose liability on the defendant for enabling the crime. Providing the gasoline to the arsonist did not constitute a rare opportunity, as he could have siphoned the gasoline he needed from a car, or bought it piecemeal.¹⁹⁹

In *Segerman v. Jones*,²⁰⁰ a school teacher leaving normal school children unsupervised for few minutes was considered insufficient encouragement hold her liable for mischief that occurred during her absence. Similar cases have denied liability for leaving a stake at a construction site,²⁰¹ for leaving a screwdriver out in a yard,²⁰² and for leaving a load of dirt clods out in a backyard.²⁰³ The courts apparently did not consider these opportunities to be particularly scarce, and hence not foreseeably exploitable. A rogue intent on harming another does not have to wait for a pile of dirt clods or a screwdriver to become available.

2. Unauthenticated Access

Valuable property is usually secured by measures that require authentication, such as through a valid key, an access card, or proof of identity. Lawsuits are frequently brought against primary tortfeasors who provide criminals with unauthenticated access to a secured location. In *Stansbie v. Troman*²⁰⁴ for instance, the defendant, an interior decorator, neglected to lock the door of the house of a client. A burglar entered through the open door and stole the plaintiff's jewelry. The court held the defendant liable for the loss. The defendant had provided the burglar with unauthenticated access—valuables are normally kept under lock and key.²⁰⁵

198. 363 P.2d 1 (Cal. 1961).

199. See Grady, *supra* note 10, at 211-13 (“Often the best way to see whether a defendant has encouraged free radicals is look at the world from their perspective. Has the defendant created some tempting opportunity that does not normally exist for them?”).

200. 259 A.2d 794 (Md. 1970).

201. *Cole v. Hous. Auth.*, 385 N.E.2d 382 (Ill. 1979).

202. *Dennis ex rel. Evans v Timmons*, 437 S.E.2d 138 (S.C. 1993).

203. *Donehue v. Duvall*, 243 N.E.2d 222 (Ill. 1969). The defendants had hauled loads of dirt into their backyard. Children from the neighborhood frequented the pile and threw clods of dirt at each other, and the defendants knew about this. One of the children threw a dirt clod at the five-year old defendant, and injured his eye. The trial court dismissed the complaint, and the Illinois Supreme Court affirmed.

204. [1948] 2 K.B. 48.

205. See also PROSSER & KEETON, *supra* note 130, at 203 (stating that it is foreseeable that valuable property will be stolen if left unguarded and in public view) (incl cases cite at note 2).

In a roughly similar case,²⁰⁶ the defendant put a scaffold in place next to the plaintiff's apartment building. Armed robbers used the scaffold to gain entry to the plaintiff's apartment and stole his goods. The New York Supreme Court denied the defendant's petition for summary judgment.²⁰⁷ The defendant encouraged wrongdoers by providing a tempting, and therefore, foreseeably exploitable opportunity. In an analogous case involving information security, the bookseller Barnes and Noble allegedly permitted cyber rogues to gain unauthorized access to confidential client information through security vulnerabilities in its website. Barnes and Noble entered into a settlement with the New York Attorney General in April 2004.²⁰⁸

Situations such as an unlocked door, strategically positioned scaffold, or electronic access, are valuable opportunities to a criminal, because they enable a level of access that normally requires authentication. The opportunity also lowers a wrongdoer's transaction cost. A thief can use brute force to break into a house, but exploiting an unlocked door or conveniently placed scaffold requires less physical exertion, produces faster results, and is less likely to attract attention than a more forceful entry.

3. Access Complexity

Opportunities that provide wrongdoers with basic access to valuables, but that leave significant remaining barriers, are less attractive, and thus less foreseeably exploitable than opportunities without such barriers. Courts and scholars classify opportunities of the former kind as ones having significant "access complexity".²⁰⁹ An enabling opportunity exhibits high access complexity when it is exploitable only during a narrow time frame, when its successful exploitation requires specific fortuitous circumstances, or when it requires "cooperation" from the victim.²¹⁰ Consider, for example, a conveniently positioned scaffold by an open window which provides access to a room containing valuables locked in a vault. The scaffold provides a burglar with basic access to the room, but the locked vault is a significant remaining barrier to the burglar's target, the

206. *Russo v. Grace Institute*, 546 N.Y.S.2d 509 (Sup. Ct. 1989).

207. *Id.* at 510.

208. See Press Release, Office of the New York State Attorney General, Attorney General reaches Agreement with Barnes and Noble on Privacy and Security Standard, April 29, 2004, available at http://www.oag.state.ny.us/press/2004/apr/apr29a_04.html.

209. The term is borrowed from computer science literature. See, e.g., Mell, Scarfone, & Romanosky, *supra* note 19, at 86.

210. See *id.*

valuables. The scaffold is therefore an opportunity with high access complexity.

Access complexity plays a role in so-called “key in ignition cases.” In these cases, a defendant typically leaves the key in the ignition of an unlocked car. The car is stolen, the thief’s negligent or reckless driving injures the plaintiff, and the plaintiff sues the defendant for negligently enabling the crime that injured her. Courts have declined to impose liability in the absence of special circumstances that made intermeddling by a thief foreseeable.²¹¹ Commenting on the evolved rule governing the key-in-car issue in California, Professor Mark Grady states, “[l]eaving the keys in unusually dangerous or difficult-to-manage vehicles will yield liability if they are parked under circumstances that make theft or meddling probable.”²¹²

In *Palma v. U.S. Industrial Fasteners*,²¹³ a case where a truck had been left parked overnight in a dangerous neighborhood with the key in the ignition, the court found special circumstances that made intermeddling foreseeable. The court imposed a duty on the defendant based on several factors, including the fact that free radical alcoholics and derelicts frequented the neighborhood, that the truck had been parked in the location for a relatively long period of time, that the truck was large, and that it was difficult to control.²¹⁴ The fact that the vehicle was left unlocked provided unauthenticated access to the vehicle, and the significant time window of opportunity provided low access complexity. If, in contrast, the key had been left unattended only momentarily while the owner ran a quick errand, the case may have been decided in favor of the defendant.

In *Richardson v. Ham*,²¹⁵ the defendants were engaged in construction work in San Diego County. In their earth moving operations they used two 26-ton Allis-Chalmers bulldozers. The bulldozers could be started if a person pushed a compression lever in and stepped on the starter. The machines could be started in gear, in which case they would commence to move as soon as the engine started.²¹⁶

211. See, e.g., *Richards v. Stanley*, 271 P.2d 23 (Cal. 1954) (finding for the defendant, holding that absent special circumstances, the owner of an automobile does not owe a duty to remove the ignition key to protect other motorists from the negligent driving of a thief).

212. See Grady, *supra* note 10, at 199; Rabin, *supra* note 8, at 440 (“Some states that allow recovery recognize a duty only under special circumstances.”).

213. 681 P.2d 893 (Cal. 1984).

214. *Id.* at 902. See also *Hergenrether v. East*, 393 P.2d 164 (Cal. 1964) (reinstating verdict for plaintiff where defendant left key in ignition of truck for extended period of time in dangerous neighborhood).

215. 285 P.2d 269 (Cal. 1955).

216. *Id.* at 270.

At the end of a working day, workers parked the bulldozers on top of a mesa. One of the machines was locked with a lock provided by the dealer, while the other was left practically unlocked.²¹⁷ The following evening three inebriated young men decided to go to the mesa for the purpose of racing the bulldozers. They were unable to start the locked bulldozer but succeeded in starting and setting the other one in motion. They drove it around the mesa, and being unable to stop it, headed it toward a canyon, and abandoned it. The bulldozer went off the edge of the mesa and caused considerable damage, until it was finally stopped by a retaining wall.²¹⁸

The plaintiffs, who suffered personal injuries and property damage, brought action for damages against the defendants. They alleged that defendants were negligent in leaving the bulldozer unattended and unlocked.²¹⁹ The trial jury found for the defendants, but the court granted the plaintiffs' motions for a new trial on grounds of insufficiency of evidence and jury misconduct. The defendants appealed.²²⁰

The defendants relied on a precedent of the Supreme Court of California in *Richards v. Stanley*,²²¹ a key-in-car case in which the court held that absent special circumstances, the owner of an automobile does not owe a duty to remove the ignition key to protect other motorists from the negligent driving of a thief.²²² The Court distinguished *Richards* from *Richardson*, and held that the dangers associated with a bulldozer in uncontrolled motion and the foreseeability of intermeddling justify imposing a duty on the owner to take reasonable care to prevent such intermeddling.²²³ The court held that these circumstances are generally not present in automobile cases, including *Richards*.²²⁴ Several factors in *Richardson* made the intermeddling foreseeable: the bulldozer presented a scarce opportunity to intermeddlers; it could be accessed without authentication; it was easy to exploit; and it offered low access complexity.

Compared to automobiles and other "ordinary" vehicles, a bulldozer is a scarce opportunity. Bulldozers are relatively uncommon and present a

217. *Id.*

218. *Id.*

219. *Id.*

220. *Id.*

221. 271 P.2d 23 (Cal.1954).

222. *Id.* at 27

223. 285 P2d, at 271 ("The extreme danger created by a bulldozer in uncontrolled motion and the foreseeable risk of intermeddling fully justify imposing a duty on the owner to exercise reasonable care to protect third parties from injuries arising from its operation by intermeddlers.").

224. *Id.* at 271. ("Since, however, the kinds of foreseeable intervening conduct by third parties as well as the risks created by such conduct in this case are materially different from those considered in the Richards case, that case is not controlling here.").

greater fascination to onlookers.²²⁵ Evidence showed that the defendant's bulldozers attracted curious spectators, both during daytime operations, as well as when they were parked overnight. Bulldozers also present a different kind and level of risk from being set in motion when left unattended.²²⁶ The court concluded that the public's fascination made it reasonably foreseeable that the defendants' bulldozers might be tampered with when left unattended.²²⁷

The bulldozer could be accessed without authentication because it was left unattended in a public area and lacked an effective lock. The Court observed that the risk of intermeddling with the unlocked bulldozer and consequent harm could have been avoided by the use of a simple but effective lock.²²⁸ The intermeddlers attempted to tamper with the other bulldozer, which was locked, but failed to set it in motion.

The bulldozer was also easy to exploit and offered low access complexity. Once the intermeddlers gained basic access to the bulldozer, they faced few significant hurdles to starting the engine and setting it in motion. A relatively unskilled person could start the machine and set the bulldozer in motion, even though he may not have the skill to stop it.²²⁹ The bulldozer was fuelled up. A person could start the engine when the bulldozer was in gear simply by pushing in a lever and stepping on the starter. If so started, the bulldozer would be set in motion immediately. The bulldozer was left parked overnight, which provided intermeddlers with a significant time window of opportunity.

In summary, intermeddling with the unlocked and unattended bulldozer was foreseeable, because the machine was easy to exploit, presented a scarce opportunity, was accessible without authentication, and was exploitable with low access complexity. The Court granted plaintiffs' motion for a new trial,²³⁰ concluding that the foreseeable risk of intermeddling fully justified imposing a duty on the owners.²³¹

4. Ease of Exploitation

An opportunity that can be exploited conveniently and without significant effort and skill on the part of a criminal or tortfeasor is

225. *Id.* at 274 (bulldozers have "a fascination which an ordinary automobile would not have for the average person").

226. *Id.*

227. *Id.* at 271.

228. *Id.*

229. *Id.* at 274 ("It could be inferred from this that it would be a simple matter for a curious person to start the machine without knowing how to stop it.").

230. *Id.* at 272.

231. *Id.* at 269.

characterized by courts and scholars as “easy to exploit.” The unattended bulldozer in *Richards v. Ham*²³² presented an easily exploitable opportunity because a relatively unskilled person could start the machine and set the bulldozer in motion. Courts are more likely to impose liability on an enabler who provided a relatively easily exploitable opportunity to an intervening tortfeasor. The common law pattern of cases involving workplace hazards illustrates this tendency.

Industrial machinery is typically manufactured with safety guards to protect operators from injury. Factory owners are sometimes tempted to remove a guard to expedite production and increase profitability, but do so at the expense of greater risk of injury to workers. Foreseeability—“the manufacturer’s reasonable anticipation that the product will be altered by removal of the safety guard in the quest for greater profitability”—is the crucial determinant of a manufacturer’s liability in these cases.²³³ Thus, in civil actions, manufacturers of machinery with easily removable guards face a greater likelihood of liability than manufacturers of machinery with a guard that requires significant effort and skill to alter.²³⁴

In *Lopez v. Precision Papers, Inc.*,²³⁵ the plaintiff’s employer had removed the overhead guard on a forklift to improve its maneuverability. An unprotected employee was struck by a falling object and rendered paraplegic.²³⁶ In a subsequent lawsuit against the manufacturer, the appellate court confirmed denial of defendant’s motion for summary judgment.²³⁷ The appellate court considered evidence of the ease with which the safety guard could be removed, and concluded that “the forklift was purposefully manufactured to permit its use without the safety guard.”²³⁸

In *Robinson v. Reed-Prentice Division*,²³⁹ a manufacturer designed machinery with a complicated interlock system that made removal of a safety shield difficult.²⁴⁰ The employer of the plaintiff circumvented this

232. *Id.*

233. *See* Rabin, *supra* note 8, at 447.

234. In lawsuits by injured workers, the manufacturer of the machine is usually a more convenient target than the employer. *See id.* at 447 (“[T]he employer is shielded from tort responsibility, and arguably, as a consequence, meaningful safety incentives, by the workers’ compensation laws. For this reason, the pragmatic attraction of enablers’ liability - the ability to target a realistic candidate for deterrence pressure, rather than the more egregious but tort-proof employer - is a salient feature of the workplace scenario . . .”).

235. 492 N.E.2d 1214 (N.Y. 1986).

236. *Id.* at 1215.

237. *Id.* at 1214..

238. *Id.* at 1215.

239. 403 N.E.2d 440 (N.Y. 1980).

240. *Id.* at 441.

difficulty by cutting holes in the safety shield so as to render the shield ineffective. The plaintiff suffered injuries, and sued the manufacturer of the machinery. The court found for the defendant, holding that the employer's substantial modifications to the product shielded the manufacturer from liability for defective design.²⁴¹ Although it is foreseeable that employers will attempt to remove shields to increase profitability, the defendant in this case had reduced the likelihood of tampering by making the shield difficult to remove.²⁴² The enabling opportunity provided by the defendant was not easily exploitable, as the intervenor had to make "substantial modifications" to achieve its goal.²⁴³

5. Anonymity

Bricks-and-mortar criminals²⁴⁴ and cyber attackers²⁴⁵ alike are emboldened by anonymity. Even "ordinary" law-abiding people tend to exhibit out-of-character behavior in an anonymous crowd. In a classic nineteenth century study of mass behavior, sociologist Gustave Le Bon concluded that crowd behavior is impulsive and uncritical, and that people act very differently in crowds than they do as individuals.²⁴⁶ Le Bon's insights about the behavior of individuals in crowds have been confirmed by more recent empirical studies by behavioral economists and social psychologists.²⁴⁷

Courts have recognized this phenomenon and the effect of anonymity on anti-social behavior. In *Guille v. Swan*,²⁴⁸ the defendant descended in a

241. *Id.* at 444.

242. *Id.* ("Material alterations at the hands of a third party which work a substantial change in the condition in which the product was sold by destroying the functional utility of a key safety feature, however foreseeable that modification may have been, are not within the ambit of the manufacturer's responsibility.")

243. *Id.*

244. *See* Grady, *supra* note 10, at 198.

245. ALLEN HOUSEHOLDER ET AL., MANAGING THE THREAT OF DENIAL-OF-SERVICE ATTACKS 23 (10th ed., 2001), available at http://www.cert.org/archive/pdf/Managing_DoS.pdf ("It is easy for attackers to avoid getting caught by hiding their identity. They command their attack network from stolen dial-up accounts and other compromised systems, and they use spoofed source addresses for attack traffic. Victim sites and law enforcement face a daunting and frequently unfeasible task to identify and prosecute attackers. Suffering few consequences—if any—for their actions, attackers continue their work. The combination of all these factors provide a fertile environment for DoS agents."); JELENA MIRKOVIC ET AL., INTERNET DENIAL OF SERVICE: ATTACK AND DEFENSE MECHANISMS 30 (2005); de Villiers, *supra* note 3, at 9-16 (noting that cyber criminals are emboldened by anonymity-preserving technologies of the Internet).

246. GUSTAVE LE BON, THE CROWD: A STUDY OF THE POPULAR MIND 31 (2002).

247. *See* RICHARD A. THALER, QUASI RATIONAL ECONOMICS (1993); Herbert A. Simon, *Theories of Bounded Rationality*, in DECISION AND ORGANIZATION: A VOLUME IN HONOR OF JACOB MARSCHAK (C.B. McGuire and Roy Radner eds., 1972).

248. 19 Johns. 381 (N.Y. 1822).

balloon over New York City into plaintiff's garden in a manner that attracted a crowd. The balloon dragged over the plaintiff's garden, but the crowd did most of the damage to the garden. The defendant argued that he should be responsible only for his share of the damages, and not for that caused by the crowd, but the court held him responsible for all the damages.²⁴⁹ The court held that the behavior of the crowd was foreseeable in that particular situation.²⁵⁰ The defendant's mode of arrival foreseeably attracted the crowd, and the relative anonymity and diminished accountability inspired the crowd's irresponsible behavior.²⁵¹ Commenting on *Guille*, Professor Grady explains, "courts are especially sensitive to the fact that people behave differently in crowds. One reason must be that being part of a crowd creates anonymity and makes it difficult for an injured plaintiff to assign fault. Thus, a responsible person becomes a free radical by joining an unruly crowd."²⁵²

An individual's relative anonymity on the Internet appears to have a similar behavioral effect.²⁵³ Otherwise upstanding citizens in the physical world behave in antisocial and even criminal ways in the relatively anonymous world of the Internet.²⁵⁴ Professor Jelena Mirkovic comments, "[t]his disassociation and lack of physical proximity encourages people to participate in illegal activities in the Internet, such as hacking, denial of service, or collecting copyrighted material. They do not feel that in reality they are doing any serious harm."²⁵⁵ The anonymity of the Internet further emboldens cyber rogues by complicating the task of detecting computer crimes and tracking down offenders, which makes it hard for authorities to obtain evidence against a wrongdoer.²⁵⁶ The Internet provides the

249. *Id.* at 381-83.

250. *Id.* at 383 ("Now, if his descent, under such circumstances, would, ordinarily and naturally, draw a crowd of people about him, either from curiosity, or for the purpose of rescuing him from a perilous situation; all this he ought to have foreseen, and must be responsible for.")

251. 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. *Id.* See also Grady, *supra* note 10, at 201-02.

252. See Grady, *supra* note 10, at 214.

253. See HOUSEHOLDER ET AL., *supra* note 245, at 23; de Villiers, *supra* note 3, at 9-16 (noting that cyber criminals emboldened by anonymity-preserving technologies of the Internet).

254. John R. Suler and W. Phillips, *The Bad Boys of Cyberspace: Deviant Behavior in Online Multimedia Communities and Strategies for Managing It*, 1 *CyberPsychology and Behavior*, 275 (1998), available at <http://www.rider.edu/~suler/psycyber/badboys.html>. See also JOHN .P. DAVIS, THE EXPERIENCE OF 'BAD' BEHAVIOR IN ONLINE SOCIAL SPACES: A SURVEY OF ONLINE USERS, available at <http://research.microsoft.com/scg/papers/Bad%20Behavior%20Survey.pdf>.

255. MIRKOVIC et al, *supra* note 245, at 30.

256. HOWARD F. LIPSON, TRACKING AND TRACING CYBER ATTACKS: TECHNICAL CHALLENGES AND GLOBAL POLICY ISSUES 49 (2002), available at <http://www.sei.cmu.edu/publications/documents/02.reports/02sr009.html> ("Although promising, research on tracking and

technological platform and opportunity to a skilled operator to assume different identities, erase his digital footprints, and transfer incriminating evidence electronically to innocent computers, often without leaving a trace.²⁵⁷

Courts have recognized the perverse incentives and law enforcement problems created by anonymous defendants in cyberspace. In *Religious Technology Center v Netcom On-Line Communication Services, Inc.*,²⁵⁸ which dealt with the misappropriation of trade secrets, the court cautioned that an anonymous or judgment-proof defendant can do significant harm and leave the plaintiff without recourse.²⁵⁹ In enabling cases where anonymity makes the intervening tortfeasor hard to reach, courts that impose liability on the primary tortfeasor are often concerned with deterrence. Professor Rabin comments, for instance, that “the main deterrence gap is the inability to effectively reach the putative wrongdoer

tracing cyber-attacks is in a nascent state. The lack of proven techniques for effectively and consistently tracking sophisticated cyber-attacks to their source (and rarely to the individuals or entities responsible) severely diminishes any deterrent effect. Perpetrators feel free to act with nearly total anonymity.”); MIRKOVIC ET AL, *supra* note 245, at 14 (“Very few attackers have been caught and prosecuted [One] factor is the ease of performing a DoS attack without leaving many traces for investigators to follow Another type of DoS perpetrator is a sophisticated hacker who uses several means to obscure her identity and create subtle variations in traffic patterns to bypass defenses.”); Ian C. Ballon, *Alternative Corporate Responses to Internet Data Theft*, 471 PLI/Pat. 737, 739 (1997); Mary Calkins, *They Shoot Trojan Horses, Don’t They? An Economic Analysis of Anti-Hacking Regulatory Models*, 89 GEO. L.J. 171 (2000); Douglas. Lichtman and Eric Posner, *Holding Internet Service Providers Accountable* (John M. Olin Law & Economics Working Paper No. 217, 2004), available at <http://www.cato.org/pubs/regulation/regv27n4/v27n4-7.pdf> (“Sophisticated saboteurs use the Internet’s topology to conceal their tracks by routing messages and information through a convoluted path that is difficult for authorities to uncover.”).

257. *Spammers and Viruses Unite*, BBC NEWS, April 30, 2003, available at <http://news.bbc.co.uk/1/hi/technology/2988209.stm> (last visited Feb., 8, 2008) (describing an anonymity-preserving computer hijacking program named Proxy-Guzu). See also Noah Levine, *Note: Establishing Legal Accountability for Anonymous Communication in Cyberspace*, 96 COLUM. L. REV. 1526, 1530-33; Jay Lyman, *Authorities Investigate Romanian Virus Writer*, LINUXINSIDER.COM, Sep. 4, 2003, available at <http://www.linuxinsider.com/perl/story/31500.html> (referring to “the difficulty of tracking down virus writers, particularly when they are skilled enough to cover their digital tracks, [so that] few offenders are ever caught”).

258. 923 F. Supp. 1231 (N.D. Cal. 1995).

259. *Id.* at 1255-57. See also Christopher Butler, *Plotting the Return of an Ancient Tort to Cyberspace: Toward a New Federal Standard of Responsibility for Defamation for Internet Service Providers*, 6 MICH. TELECOMM. & TECH. L. REV. 247, 260 (2000) (discussing *Zeran v America Online, Inc.*, 129 F.3d 327 (4th Cir. 1997), and commenting that a plaintiff injured by anonymous speech of an ISP subscriber “was left without recourse once the court held AOL to be immune from liability as a distributor of third party information content because the messages had been posted by an anonymous person whose identity was never able to be traced”).

himself, either through criminal or tort sanctions. This is the crux of the matter and the link to creating responsibility for enabling behavior.”²⁶⁰

Summary

The common law pattern shows that enablers foreseeably encourage tortious and criminal behavior when they provide opportunities that are scarce, easily exploitable, aligned with the objectives of the criminal, and that provide the criminal with anonymity, unauthenticated access, and access with low complexity. The next subsection develops and analyzes analogous features of computer security vulnerabilities that make the vulnerabilities foreseeably exploitable.

B. Forensic Analysis of Security Vulnerabilities

1. Ease of Exploitation

Easily exploitable security vulnerabilities are more likely to be targeted by attackers.²⁶¹ A target’s ease of exploitation depends on the availability and quality of exploit code to leverage the vulnerability.²⁶² This feature varies substantially across different vulnerabilities.²⁶³ A vulnerability that allows a hacker to access a university’s confidential student records by simply clicking on a hyperlink posted on the Internet is easy to exploit. In contrast, a vulnerability that can only be exploited by a highly skilled and technically sophisticated attacker is difficult to exploit.

260. Rabin, *supra* note 8, at 444 (referencing *Kline v. 1500 Mass. Ave. Apartment Corp.*, 439 F.2d 477 (D.C. Cir. 1970)).

261. *See, e.g.*, SYMANTEC SECURITY UPDATE – JUNE 2005 4 (2005), http://www.symantec.com/avcenter/reference/SSU_AMS_06_2005.pdf (“The threat of severe vulnerabilities is increased if and when an associated exploit is released publicly or if the vulnerability can be exploited trivially.”). *See also* Scott Carpenter, *Vulnerability Management Solutions*, ITDEFENSE MAGAZINE, April 2006, at 2, *available at* http://www.itdefensemag.com/8_06/issue_cover.php (“If a proof of concept exploit is made available the same day the vulnerability is published (i.e. zero day exploit), and a patch is not yet available, a virus or worm likely will be created from this vulnerability.”). *See also* SYMANTEC, COMPREHENSIVE THREAT MANAGEMENT: A SYMANTEC SOLUTION FOR MODERN-DAY ATTACK PROTECTION 4-5, *available at* https://www4.symantec.com/Vrt/offer?a_id=20197 (explaining that the combination of financial incentives and easy access to malevolent code has increased the number of computer security breaches).

262. *See* SYMANTEC INTERNET SECURITY THREAT REPORT TRENDS FOR JULY – DECEMBER 06, *supra* note 18, at 90; FOSTER ET AL., *supra* note 18, at 10; PETER MELL, KAREN SCARFONE, & SASHA ROMANOSKY, A COMPLETE GUIDE TO COMMON VULNERABILITY SCORING SYSTEM VERSION 2.0 10 (2007), *available at* <http://www.first.org/cvss/cvss-guide.pdf> (“Public availability of easy-to-exploit code increases the number of potential attackers by including those who are unskilled . . .”).

263. *See also* SZOR, *supra* note 22, at 547 (“Some vulnerabilities are easily exploited by the attackers, while others take months to develop.”).

The following classification defines ease of exploitation of a vulnerability in relation to its exploit code.²⁶⁴

1. Unproven: No exploit code is available.

2. Proof of concept: The only exploit code available for vulnerabilities in this category is “proof-of-concept.” Proof-of-concept code is a program with the sole purpose of verifying the existence of a vulnerability. It is not sufficiently functional to exploit the vulnerability.

3. Functional: Functional exploit code is publicly available for vulnerabilities in this category. Attackers can exploit some incidences of the vulnerability without having to use sophisticated programming techniques or other technical skills of their own.

4. High: Functional exploit code is available for every exploitable incidence of the vulnerability. It is reliably exploitable, and there have been instances of actual exploitation by self-propagating malevolent code.

Vulnerabilities in the latter two categories are easy to exploit, because leveraging those vulnerabilities requires minimal technical sophistication. Vulnerabilities in the first two categories are difficult to exploit, because attackers must create exploit code or develop existing proof-of-concept code to leverage the vulnerability.²⁶⁵

Many incidences of the buffer overflow vulnerability belong to the “easy to exploit” category. Writing an original effective exploit for a vulnerability such as the buffer overflow generally requires considerable programming skill, but exploit code is often publicly available and accessible to individuals without technical sophistication. Using publicly available exploit code often requires little more than basic programming skills and standard software tools. As new buffer overflow vulnerabilities are discovered, exploits are habitually published shortly after the discovery.²⁶⁶

A vulnerability in the Solaris KCMS Library Service System for instance, belongs to the “easy to exploit” category. It can be exploited by drawing on a standard and widely available software tool and basic

264. See Presentation, Ivan Arce, On the Quality of Exploit Code (June 2004), available at <http://www.coresecurity.com/files/attachments/CSI-NetSec2004.ppt>; SYMANTEC INTERNET SECURITY THREAT REPORT TRENDS FOR JULY – DECEMBER 06, *supra* note 18, at 90.

265. SYMANTEC INTERNET SECURITY THREAT REPORT TRENDS FOR JULY – DECEMBER 06, *supra* note 18, at 90.

266. For a review of publicly available exploits, see Takanen et al., *supra* note 55. See, e.g., NATHAN P. SMITH, STACK SMASHING VULNERABILITIES IN THE UNIX OPERATING SYSTEM, (1997), available at <http://destroy.net/machines/security/nate-buffer.pdf> (last visited Feb. 8, 2008) (thorough academic survey, covering history and terminology, various vulnerabilities and related technologies, as well as solutions). See also DAVID LITCHFIELD, EXPLOITING WINDOWS NT 4 BUFFER OVERRUNS, available at <http://www.ngssoftware.com/papers/ntbufferoverflow.html> (last visited Feb. 19, 2008).

computer literacy.²⁶⁷ Not all vulnerabilities are easy to exploit, though, and the time and skill required to create an exploit vary among vulnerabilities.²⁶⁸ A buffer overflow vulnerability in NOD32 antivirus software, for instance, is relatively difficult to exploit, because only proof-of-concept exploit code has been released. The code requires substantial modification by a skilled attacker in order to exploit the vulnerability.²⁶⁹

Computer security vulnerabilities that are easy to exploit are analogous to the escape opportunity given to the inmates in *Home Office v Dorset Yacht Co.*,²⁷⁰ or the circumstances in *Richardson v. Ham*,²⁷¹ which made intermeddling with a bulldozer foreseeable.

2. Scarcity of Opportunity

An opportunity is “scarce” if it meets the objectives of a wrongdoer and no reasonably equivalent opportunity is readily available. The unlocked door in *Stansbie*²⁷² is an example of a scarce opportunity. Valuables are usually locked up, and the only alternative means of access available to the thief was entry in a way more likely to attract attention.

A computer security vulnerability is a scarce opportunity to an attacker who is using a virus or worm which is programmed to exploit the specific vulnerability.²⁷³ Malevolent code is often designed to target a specific system and vulnerability type. The Morris worm,²⁷⁴ for instance, had a three-pronged attack vector. It first invaded user accounts on the target machine by “guessing” passwords, encrypting them, and comparing

267. See AusCERT Alert, Vulnerability in Solaris 2.5 KCMS, May 1, 1997, <http://www.auscert.org.au/render.html?it=69>. For further examples of easily exploitable vulnerabilities, see Bill Brenner, *The Exploits of August*, SECURITY NEWS, Aug. 12, 2005, http://searchsecurity.techtarget.com/news/article/0,289142,sid14_gci1115378,00.html# (“MS05-039 is a remote RPC vulnerability that . . . is a very easy to exploit vulnerability.”) and Ryan Naraine, *Exploit Allows Windows XP Piracy*, EWEEK.COM, May 23, 2005, <http://www.eweek.com/c/a/Windows/Exploit-Allows-Windows-XP-Piracy/>, (Describing a vulnerability in Microsoft’s Windows Genuine Advantage as uncomplicated and easy to exploit.”).

268. SMITH, *supra* note 266, at 2; SZOR, *supra* note 22, at 547 (“Some vulnerabilities are easily exploited by the attackers, while others take months to develop.”).

269. See National Vulnerability Database (CVE-2003-0062), <http://nvd.nist.gov/nvd.cfm?cvename=CVE-2003-0062> (last modified Dec. 21, 2006). See also SZOR, *supra* note 22, at 402 (describing the OpenSSL buffer overflow vulnerability as challenging to exploit).

270. [1970] 2 A.C. 1004 (appeal taken from Eng.).

271. 285 P.2d 269 (Cal 1955).

272. [1948] 2 K.B. 48.

273. See Weaver et al., *supra* note 42, at 2 (explaining that certain worms are designed to exploit specific vulnerabilities, and are programmed to search and locate hosts that contain these vulnerabilities).

274. See, e.g., NAZARIO, *supra* note 104, at 39-41 (history of the Morris worm).

the encrypted words to the system's encrypted password file.²⁷⁵ The worm simultaneously exploited a buffer overflow vulnerability in the *Fingerd* program,²⁷⁶ and also attempted to exploit a trapdoor in the *sendmail* mail handler.²⁷⁷

A scarce opportunity is also characterized by a paucity of equally attractive alternative opportunities. If most instances of a targeted vulnerability have been remediated, the few remaining unremediated ones will be scarce opportunities to scavenging worms looking for them. A vendor typically remediates a vulnerability by providing a patch to fix the vulnerability and issuing security alerts to potentially affected users. Such notification heightens awareness of the vulnerability among legitimate users and rogues alike, further increasing the likelihood of exploitation of unpatched versions of the vulnerability. The scarcity of the opportunity presented by a vulnerability exploited in a cyber attack, therefore, depends on whether the virus used in the attack specialized in the specific vulnerability, and the prevalence of the vulnerability on the Internet.

This article adopts the following standard classification of the remediation level of a vulnerability.²⁷⁸

1. Official fix: A final official patch is available from the vendor that eliminates the vulnerability, or an upgrade is available that does not contain the vulnerability.

2. Temporary fix: The vendor has released a temporary patch for a vulnerability in this category.

3. Workaround: A patch is available, but it is temporary and not officially issued by the vendor. A user or group of users may, for instance, have created and distributed a fix.

4. Unavailable: There is no practicable solution available.

Category (1) represents the highest remediation level and Category (4) represents the lowest. A remediation score assigns a numerical value to each remediation category. The "scarcity score" for a vulnerability is then obtained by multiplying the vulnerability's remediation score by a scaling

275. SZOR, *supra* note 22, at 395-97.

276. *Fingerd* is a continuously running program that responds to requests for information about system users. See PFLEEGER & PFLEEGER, *supra* note 7, at 136.

277. The *sendmail* program normally runs in the background, awaiting a command to transmit mail. It typically receives a command and the message, as well as a destination address. The Morris worm, however, caused the *sendmail* program to receive and execute malevolent code instead of a destination address. See generally BRYAN COSTALES AND ERIC ALLMAN, *SENDMAIL* (3d ed., 2002).

278. See Victor-Valeriu Patriciu, Justin Priescu, & Sebastian Nicolaescu, *Security Metrics for Enterprise Information Systems*, 1 JOURNAL OF APPLIED QUANTITATIVE METHODS 154 (2006).

factor. The scaling factor is a binary value which measures whether the vulnerability was targeted or not.

3. *Unauthenticated Access*

“Authentication” refers to procedures by which a computer system verifies the identity of a party from whom it has received a communication.²⁷⁹ The login procedure is a well-known authentication procedure. It authenticates the identity of a user whose password and stated identity match. If they do not match, the user is restricted from accessing the system. Other examples of authentication include the requirement of confirmation e-mail to activate an on-line account, ATM access procedures, cryptographic authentication of a digitally signed contract, and biometric identification in applications such as Internet banking.²⁸⁰

Authentication provides a line of defense against unauthorized access to sensitive information. A vulnerability that allows a user to bypass this line of defense is said to allow unauthenticated access. Network vulnerabilities, including buffer overflows, often allow unauthenticated remote access to unauthorized individuals.²⁸¹ A vulnerability in Fusion News, for instance, a news management program for web servers, allows remote unauthenticated attackers to create arbitrary user accounts on the Fusion News server by sending a specially crafted request to the server. If properly structured, the request could also be used to gain administrative access. Exploitation of the vulnerability is trivial, as a ready-to-use sample server request is available on the Internet.²⁸² This vulnerability contains several critical elements favorable to a cyber attacker: It is easily exploitable, and allows unauthenticated access, root access, and remote access.

279. See PFLIEGER & PFLIEGER, *supra* note 7, at § 4.5.

280. *Id.* at 219-220; Citron, *supra* note 2, at 249-50.

281. See, e.g., Advisory, Next Generation Security Software, Microsoft NetDDE Service Unauthenticated Remote Buffer Overflow (Jan. 21 2005), <http://www.ngssoftware.com/advisories/netddefull.txt> (Reporting a vulnerability in the Microsoft DDE service that allows a remote attacker to execute arbitrary code on a system without authentication.). Microsoft has released an update addressing the vulnerability. Security Bulletin, Microsoft TechNet, Vulnerability in NetDDE Could Allow Remote Code Execution (841533), <http://www.microsoft.com/technet/security/bulletin/MS04-031.msp>. See also DOROTHY E. DENNING, INFORMATION WARFARE AND SECURITY 214 (1999) (explaining that hackers can execute malicious code remotely, without logging in and providing a valid password, by exploiting buffer overflow vulnerabilities).

282. DarkKnight, *Fusen News 3.3. Account Add Vulnerability*, NEOHAPSIS, Aug. 15, 2003, <http://archives.neohapsis.com/archives/bugtraq/2003-08/0201.html>. See also SZOR, *supra* note 22, at 410.

A computer security vulnerability that grants unauthenticated access is the cyber analogue of an unlocked door or conveniently placed scaffold, both of which provide access to an unauthorized entrant.

4. Access Complexity

“Access complexity” measures the complexity of attack required to exploit a vulnerability beyond basic access. A vulnerability has high access complexity if an attacker faces additional barriers to exploiting the vulnerability once the attacker has gained basic access to a target system.²⁸³ In many cases, once a system has been penetrated, exploitation of the vulnerability is straightforward. A simple buffer overflow vulnerability in a server program that runs continuously, for instance, has low access complexity. Once an intruder has leveraged the vulnerability to inject malevolent code, there are no additional barriers to executing the code. The attacker can hijack the server program at will and run the attack code, often at the same level of privilege as the server program.²⁸⁴

A vulnerability with high access complexity is characterized by specialized access conditions. It may, for instance, be exploitable only during a narrow time frame, or its successful exploitation may require non-default technical conditions or special “cooperation” from the victim.²⁸⁵ For example, a vulnerability has been reported in Microsoft Windows which was not remotely exploitable by default, but could be exploited via a Web page “if the user has installed an application with custom URL handlers and then uninstalled that application, and the uninstall failed to correctly remove the application completely.”²⁸⁶ In addition to this fortuitous sequence of events, the attacker would have to create an HTML Web page according to narrow technical specifications in order to exploit the vulnerability.²⁸⁷ This vulnerability is therefore exploitable only under specific non-default conditions, a characteristic of high access complexity.

283. See MIKE SCHIFFMAN, A COMPLETE GUIDE TO THE COMMON VULNERABILITY SCORING SYSTEM (CVSS) 3 (2005), available at <http://www.packetfactory.net/papers/CVSS/guide/index.html> (last modified June 7, 2005).

284. *Id.* at 4.

285. See generally Mell, Scarfone, & Romanosky, *supra* note 19; Presentation, Mike Schiffman, The Common Vulnerability Scoring System (Feb. 2005), available at <http://www.packetfactory.net/papers/CVSS/cvss-ppt.pdf>.

286. See Security Bulletin, AUSCERT, *Unchecked Buffer in Windows Shell Could Lead to Code Execution*, (Mar. 11, 2002), available at <http://www.auscert.org.au/render.html?it=1779>. See also Security Advisory: Microsoft Security Bulletin MS02-014 – securites vulnerabilities database, <http://securityvulns.com/docs2607.html> (last modified Oct. 3, 2002).

287. Security Bulletin, AUSCERT, *Unchecked Buffer in Windows Shell Could Lead to Code Execution*, (Mar. 11, 2002), available at <http://www.auscert.org.au/render.html?it=1779>.

It is therefore less foreseeably exploitable than an equivalent vulnerability without these complicating factors.²⁸⁸

A simple buffer overflow vulnerability with low access complexity is analogous to the situation in *Richardson v. Ham*,²⁸⁹ the bulldozer case. In *Richardson*, the intermeddlers had unauthenticated access to a bulldozer, because the machine was left unlocked and parked in a public place. The intermeddlers also enjoyed low access complexity, because there were no significant additional barriers to exploiting the bulldozer. The bulldozer was fueled up, was easy to start, and it could be started in gear, which automatically set it in motion. It was also left parked overnight, so that the intermeddlers had a significant time window of opportunity.

5. *Remote Access*

A vulnerability is remotely exploitable when it enables a user to access and execute commands on a target system across a network. A vulnerability that grants only local access requires physical access or a local account on the target system. An attacker with remote access may effectively take over a remote console and enter keystrokes and commands as if the attacker had local access. Some versions of the *rlogin* program for instance, contain a vulnerability that allows attackers to launch a remote attack on any vulnerable system connected to the Internet.²⁹⁰

Remote access offers obvious advantages to an attacker, such as convenience, fewer physical risks than on-site entry, and anonymity.²⁹¹ The anonymity and geographic distance also help the attacker by

288. See also SANS CRITICAL VULNERABILITY ANALYSIS, Feb. 3, 2004, http://www.sans.org/newsletters/cva/vol2_4.php. The advisory describes the SpamAssassin buffer overflow as “challenging to exploit.” To succeed, a would-be attacker would have to identify and target victims who are using a vulnerable spam filter.

289. 285 P.2d 269 (Cal 1955).

290. The *rlogin* program, which is available on most UNIX systems, allows a legitimate user to access the UNIX system from a remote terminal. Many versions of the *rlogin* program, however, contain a vulnerability that allows attackers to inject and remotely execute arbitrary code on the vulnerable machine, and run the attack code with root privileges. Such an attack can be launched remotely from any location with an Internet connection to a targeted institution’s Web site. See LAWRENCE R. ROGERS, RLOGIN(1): THE UNTOLD STORY 3, 16, 23 (1998), available at <http://www.sei.cmu.edu/publications/documents/98.reports/98tr017/98tr017abstract.html>; SZOR, *supra* note 22, at 341.

291. PFLEEGER & PFLEEGER, *supra* note 7, at 397 (“An attacker can mount an attack from thousands of miles away and never come into direct contact with the system, its administrators, or users. The potential attacker is thus safe behind an electronic shield.”); SYMANTEC SECURITY UPDATE - JUNE 2005, *supra* note 261, at 4 (“Remotely exploitable buffer overflow vulnerabilities are particularly dangerous, as skilled attackers can carry out exploitation without alerting a target user to the attack.”).

complicating law enforcement efforts to establish jurisdiction, obtain extradition, and procure evidence.²⁹²

Vulnerabilities, such as the buffer overflow, are useful springboards to gain remote access to a target system because they allow a remote attacker to inject malevolent code directly into the execution path of a remote system.²⁹³ The newly injected viral code can then create further opportunities for other remote attackers.²⁹⁴ Remotely exploitable buffer overflow vulnerabilities have recently been reported in well-known products, such as Sendmail, various Microsoft products, and, ironically, PGP.²⁹⁵ A vulnerability in Microsoft's Internet Explorer browser, for instance, allowed a properly formatted HTML document to cause a buffer overflow. This flaw could be remotely exploited to execute arbitrary code on the affected system with the privileges of the user running Internet

292. See, e.g., Henderson & Yarbrough, *supra* note 103, at 11 (“[T]oday’s Internet can be crippled by distributed denial-of-service attacks launched by relatively unsophisticated and judgment-proof parties.”); *Id.* at 16 (“Similarly, in the case of a DDoS attack, the person who uses the weapon . . . is generally clearly liable, but that person is often either impossible to locate, is judgment-proof, or both.”); HOUSEHOLDER ET AL., *supra* note 245, at 23 (“It is easy for attackers to avoid getting caught by hiding their identity. They command their attack network from stolen dial-up accounts and other compromised systems, and they use spoofed source addresses for attack traffic. Victim sites and law enforcement face a daunting and frequently unfeasible task to identify and prosecute attackers. Suffering few consequences—if any—for their actions, attackers continue their work.”).

293. SZOR, *supra* note 22, at 68. The so-called backdoor is a widely used device to gain unauthorized remote access to a system. A back door system such as the infamous Back Orifice, allows an attacker to obtain information about the system on which it is installed, including information on currently running programs and the contents and nature of files and directories on the system. It also allows the remote intruder to download files from the system and submit commands to it. Cyber attackers frequently employ viruses and worms to leverage backdoors to penetrate and compromise target systems. Malevolent code that utilizes backdoor interfaces include worms such as Nimda, which exploited a backdoor opened by CodeRed. The W32/Borm worm used network scanning and fingerprinting techniques to locate exploitable backdoor-compromised systems. *Id.* at 309-311, 331; HARLEY, SLADE & GATTIKER, *supra* note 29, at 74; Thomas Chen, *Trends in Viruses and Worms*, INTERNET PROTOCOL J., available at http://enr.smu.edu/~tchen/papers/Cisco%20IPJ_sep2003.pdf.

294. The Nimda worm, for instance, attacked via backdoors left by worms such as CodeRed. See SZOR, *supra* note 22, at 309-311, 331.

295. See, e.g., Security Response, Symantec, Sendmail Header Processing Buffer Overflow Vulnerability, (Mar. 3, 2003), available at <http://securityresponse.symantec.com/avcenter/security/Content/3.3.2003.html> (“A remotely exploitable vulnerability has been discovered in Sendmail. This vulnerability is due to a buffer overflow condition in the SMTP header-parsing component. Remote attackers may exploit this vulnerability by connecting to target SMTP servers and transmitting them malformed data.”). Pretty Good Privacy, popularly known as PGP, is a computer program that provides cryptographic privacy and authentication. PGP is often used for signing, encrypting and decrypting e-mails to increase reliability for e-mail communications. It was originally created by Philip Zimmermann in 1991. See, e.g., BRUCE SCHNEIER, APPLIED CRYPTOGRAPHY 587 (1995).

Explorer.²⁹⁶ Some vulnerabilities are not remotely exploitable in a default installation, but can be remotely exploited under unusual conditions and with application of significant technical expertise by an aspiring attacker.²⁹⁷

A vulnerability without remote access offers limited scope for a virus or worm attack, and is therefore less likely to be exploited.²⁹⁸ A recent Microsoft Security Bulletin advises of the existence of an image parsing vulnerability in Microsoft Office that allowed an attacker to install programs, create new accounts with full user rights, and change or delete data on a client workstation. This was a local vulnerability, and not remotely exploitable. The Security Bulletin concludes, “we do not expect to see widespread exploitation of these vulnerabilities in current operating system versions.”²⁹⁹

Courts have recognized that anonymity encourages wrongdoing.³⁰⁰ The common law position is supported by empirical research which suggests that cyber criminals are encouraged by anonymity-preserving technologies, such as the Internet.³⁰¹ A vulnerability that enables a remote attack behind the Internet’s cloak of anonymity is therefore foreseeably exploitable, as a matter of common law precedent as well as technology-created economic incentives.

6. Root Access

Cyber rogues favor vulnerabilities that closely meet their objectives. An intruder intent on stealing sensitive information from a financial

296. Advisory, CERT, Buffer Overflow in Microsoft Internet Explorer (Feb. 25, 2002), available at <http://www.cert.org/advisories/CA-2002-04.html> (last modified Apr. 2, 2002). This document also advises on other remotely exploitable vulnerabilities in Microsoft Internet Explorer.

297. *See id.*

298. *See* Scott Carpenter, *Vulnerability Management Solutions*, ITDEFENSE MAGAZINE, Apr. 2006, at 2, available at http://www.itdefensemag.com/8_06/issue_cover.php (“If the vulnerability is not remotely exploitable, the likelihood of a virus based on the vulnerability alone is minimized.”).

299. Security Bulletin, MicrosoftTechNet, Vulnerabilities in Microsoft Office Filters Could Allow Remote Code Execution (915384) (July 11, 2006), <https://www.microsoft.com/technet/security/bulletin/ms06-039.msp> (last modified Nov. 29, 2006). *See also* Security Bulletin, Adobe Coldfusion Sandbox Security Vulnerability (Sept. 12, 2006), available at <http://www.adobe.com/support/security/bulletins/apsb06-13.html> (reporting a vulnerability in Coldfusion software versions MX 7 and MX 7.01, which is not remotely exploitable).

300. *See, e.g.*, *Guille v. Swan*, 19 Johns. 381 (N.Y. 1822); *Religious Tech. Ctr. v Netcom On-Line Comm’n. Servs., Inc.*, 923 F.Supp. 1231, 1255-57 (9th Cir. 1995) (when referring to misappropriation of trade secrets, the court cautioned that an anonymous or judgment-proof defendant can do significant harm and leave the plaintiff without recourse).

301. *See, e.g.*, *MIRKOVIC ET AL*, *supra* note 245, at 30 (“This disassociation and lack of physical proximity encourages people to participate in illegal activities in the Internet, such as hacking, denial of service, or collecting copyrighted material. They do not feel that in reality they are doing any serious harm.”). *See also* *Suler & Philips*, *supra* note 254; *DAVIS*, *supra* note 254.

institution for instance needs access to the institution's information system at the privilege level of an administrator who has authority to access and transmit such information. This is usually the most privileged level of access, namely full root-level access.

"Root" is the conventional name of the super-user who has all rights in all modes on a computer system. This is usually the system administrator's account. The super-user has privileges that an ordinary user does not have, such as authority to change the ownership of files; install and run programs; change Web Server databases; add, change, or delete system files or data; and change or replace web pages.³⁰² An attacker who gains root access inherits these privileges. If a program is already running with root privileges, an appropriately configured vulnerability may be exploited to hijack the program and transfer root control to the attacker.³⁰³ The attacker can then effectively become the administrator of the network or system. Root access is the cyber analogue to real-space insider access to a company's information resources, usually obtained through bribery or blackmail of company insiders by foreign governments, competitors and organized crime.³⁰⁴

Many known vulnerabilities yield root access to an attacker. The Linux application DosEMU, for instance, contained a vulnerability that assisted an attacker in gaining root access.³⁰⁵ The Apple Mac OS Apple Filing Server Remote Buffer Overflow Vulnerability³⁰⁶ likewise allowed attackers to remotely compromise a vulnerable computer to gain administrator-level access. Attackers typically exploited this vulnerability to inject malevolent code into the vulnerable computer over a network or the Internet, and run it with root privileges.³⁰⁷ Not all vulnerabilities allow root access. A buffer overflow vulnerability reported in the IMAP Server, for instance, allowed remote exploitation, but not root access.³⁰⁸

302. HOGLUND & MCGRAW, *supra* note 93, at 151-53.

303. DENNING, *supra* note 259, at 214 (describing malicious code executed via a buffer overflow as executing "with the privileges of the program it exploits, which is often root").

304. *Id.* at 131.

305. See, e.g., SecuriTeam – DosEMU Buffer Overflow Assists in Gaining Root, <http://www.securiteam.com/exploits/2GUPVSAQ00.html> (last modified July 9, 1999).

306. See Apple Mac OS X Security Update 2005-006 Multiple Vulnerabilities, <http://www.securityfocus.com/bid/13899> (last visited Feb. 8, 2008).

307. See SYMANTEC SECURITY UPDATE – JUNE 2005, *supra* note 261, at 5.

308. See Security Advisory, Security Reason, UW-IMAP Netmailbox Name Parsing Buffer Overflow Vulnerability (May 10, 2005), <http://securityreason.com/securityalert/47>. See also SANS CRITICAL VULNERABILITY ANALYSIS, Dec. 8, 2002, http://www.sans.org/newsletters/cva/cva1_20.php ("The Cyrus IMAP server for UNIX contains a remotely exploitable buffer overflow that allows non-authenticated attackers to execute arbitrary code with the privileges of the server process (*typically not root*).") (Emphasis added).

Root level enablement is aligned with the objectives of cyber attackers who need high-level access to a company's most sensitive information. A feature in a vulnerability that allows root access to a system therefore contributes to the foreseeability of exploitation of the vulnerability.

Summary

The analysis presented in this section has identified technical features that make a computer security vulnerability foreseeably exploitable. The features, which are cyber analogues of traditional common law counterparts, are as follows: (1) the availability in the public domain of user-friendly tools to leverage the vulnerability to gain basic access to a targeted system, (2) the facilitation of unauthenticated access to the targeted system, (3) the facilitation of remote access, (4) the facilitation of anonymous access, (5) the insignificance of the complexity of the attack once basic access has been obtained, and (6) the vulnerability uniquely meets the objectives of the exploiter.

VIII. A Foreseeability Metric

This section proposes a numerical metric that combines values for each of the attributes described in the previous section into a composite score that represents the foreseeability of exploitation of a vulnerability. The attributes are listed below, together with proposed scores for each category within an attribute. The chosen numerical values are not arbitrary, but are consistent with industry standards, such as the Common Vulnerability Scoring System (CVSS), developed by the U.S. National Infrastructure Assurance Council.³⁰⁹ The values are also consistent with the objectives of the metric, in the sense that the numerical values associated with individual attributes increase with that attribute's contribution to foreseeability. A vulnerability for which there is no exploit code available, for instance, is assigned a relatively low value for "ease of exploitation," namely 0.85. A vulnerability for which functional exploit code is available for every exploitable incidence of the vulnerability, is assigned a maximum value of 1.0.

The vulnerability attributes on which the foreseeability metric is based, and the numerical indices associated with each category of the attributes, are listed below.

309. See, e.g., Mell, Scarfone, & Romanosky, *supra* note 19, at 85. CVSS was developed by the U.S. National Infrastructure Assurance Council, a group of industry leaders who advise the US Department of Homeland Security on critical information infrastructure security. *Id.*

A. Ease of Exploitation

The following classification defines ease of exploitation of a vulnerability in relation to its exploit code. It lists the various categories to which exploit code for a vulnerability may belong.³¹⁰ The numerical index in brackets represents the degree to which the category contributes to foreseeability of exploitation. The highest index (“1”), for instance, is assigned to a vulnerability for which functional exploit code is available for every exploitable incidence of the vulnerability, (the “High” category). The ease of exploitation scores are:

1. Unproven [0.85].
2. Proof of concept [0.9].
3. Functional [0.95].
4. High [1.00].

B. Scarcity

This metric depends on the level of remediation of the vulnerability and whether a worm that exploited it specifically targeted this vulnerability.³¹¹

1. Unavailable [0.87].
2. Workaround [0.9].
3. Temporary fix [0.95].
4. Official fix [1.00].

The composite scarcity score is obtained by multiplying the remediation level score by 1.0 if the vulnerability was targeted, and by 0.5 if it was not.

C. Access Complexity

1. High: Additional barriers to exploitation exist [0.8].
2. Low: The target is exploitable under general conditions [1.0].

D. Unauthenticated Access

1. The attacker does not need authentication to access and exploit the vulnerability [1.0].
2. The attacker needs authentication to access and exploit the vulnerability [0.6].

E. Remote Access

1. Vulnerability allows only local access [0.7].

310. The categories are defined and described, *supra* §VIII(A).

311. The remediation categories are defined and described, *supra* § VIII(B).

2. Vulnerability allows remote access [1.0].

F. Root Access

1. Vulnerability allows root access [1.0].
2. Vulnerability does not allow root access [0.6].

G. Information Security Impact

This metric measures the impact on confidentiality, integrity, and availability of information of a successful exploit of the vulnerability.

- A. Confidentiality impact: None [0]; Partial [0.7]; Complete [1.0].
- B. Integrity impact: None [0]; Partial [0.7]; Complete [1.0].
- C. Availability impact: None [0]; Partial [0.7]; Complete [1.0].

A, B, and C are equally weighted to calculate the information security impact metric.

In an environment where confidentiality is emphasized, the weights assigned to A, B, and C respectively, are [0.5; 0.25; 0.25]. If integrity is emphasized, the weights are [0.25; 0.5; 0.25], and when availability is emphasized, the weights are [0.25; 0.25; 0.5].

Composite Foreseeability Score

The composite foreseeability score for a vulnerability is calculated by applying the following steps: (1) determine the appropriate category and score for each of attributes I through VII for the vulnerability, (2) calculate the product of the scores, (3) multiply the score by 10, and (4) round the result to the nearest integer. The resultant score is a numerical index, on a scale from 1 to 10, of the foreseeability of exploitation of the vulnerability.

This article will now discuss illustrative examples of the application of the metric to real-world vulnerabilities.

- a. CVE:-2003-0818: Microsoft Windows ASN.1 Library Integer Handling Vulnerability.³¹²

This is a buffer overflow vulnerability in Microsoft Windows which allows an attacker to execute arbitrary code with root privileges. It allows unauthenticated access and is remotely exploitable. The vulnerability offers the additional advantage of low access complexity, because no additional effort or special circumstances are necessary for a successful exploit. Functional exploit code is publicly available. Microsoft released a patch (MS04-007) to remediate the vulnerability, so that the remediation

312. See National Vulnerability Database (CVE-2003-0818), <http://nvd.nist.gov/nvd.cfm?cvename=CVE-2003-0818> (last modified Mach 28, 2006).

level is classified as “Official Fix.”³¹³ The vulnerability’s multiple attractive properties (from an attacker’s viewpoint) suggest that exploitation of this vulnerability is highly foreseeable.

The foreseeability composite score is calculated as follows:

METRIC	EVALUATION	SCORE
Ease of exploitation	Functional	0.95
Scarcity	Official-Fix, Targeted	1.0
Access Complexity	Low	1.0
Unauthenticated Access	Yes	1.0
Remote Access	Yes	1.0
Root Access	Yes	1.0
Info Sec Impact	C/I/A, Equally Weighted	1.0
<p>The foreseeability score for this vulnerability is calculated as: $\text{ROUND}[10 \times 0.95 \times 1.0 \times 1.0 \times 1.0 \times 1.0 \times 1.0 \times 1.0] = 10.0.$</p>		

The high value for the foreseeability metric (10/10) is consistent with the prima facie impression that the vulnerability is a tempting target and likely to be exploited.

b. CVE-2003-0062: Buffer Overflow in NOD32 Antivirus.³¹⁴

This buffer overflow vulnerability was discovered in February 2003 in Linux and UNIX versions of NOD32, an antivirus application. The vulnerability allowed attackers to execute arbitrary code with the privileges of the user running NOD32. The vulnerability was not remotely exploitable and had significant access complexity. To execute malevolent code via the buffer overflow, an attacker had to wait for another user to scan a directory path of excessive length. If a user executed the scan, full compromise of the confidentiality, integrity, and availability of information on the target system was possible. Proof-of-concept level exploit code is available for the metric. The developer of NOD32 (Eset) has released

313. *Id.*

314. See National Vulnerability Database (CVE-2003-0062), <http://nvd.nist.gov/nvd.cfm?cvename=CVE-2003-0062> (last modified Dec. 21, 2006).

updated software, giving the vulnerability the highest remediation level of “Official Fix.”³¹⁵

The foreseeability score for this vulnerability is calculated as follows:

METRIC	EVALUATION	SCORE
Ease of exploitation	Proof-of-Concept	0.9
Scarcity	Official-Fix, Not Targeted	0.5
Access Complexity	High	0.8
Unauthenticated Access	Yes	1.0
Remote Access	Local	0.7
Root Access	Yes	1.0
Info Sec Impact	C/I/A, Complete	1.0
<p>The foreseeability score for this vulnerability is calculated as: $\text{ROUND } [10 \times 0.9 \times 0.5 \times 0.8 \times 1.0 \times 0.7 \times 1.0 \times 1.0] = 3.0$</p>		

This vulnerability contains several features that make it unattractive to an attacker, and is therefore not highly foreseeably exploitable. In order to successfully exploit the vulnerability, a prospective attacker must develop exploit code beyond the available proof-of-concept. This requires a level of skill and effort that would decrease the number of prospective exploiters. An attacker would have to contend with access complexity, such as having to wait for non-default conditions before a successful attack can be launched. An attacker would also have to work within the limitations of local access. The vulnerability has a high remediation level, but the fact that it is not specifically targeted makes it less likely to be exploited.

A vulnerability that contains each attribute at its highest level, in contrast, would have the highest possible foreseeability score, namely a “perfect 10.” A cyber rogue who contemplates designing a worm to exploit a specific vulnerability would likely prefer to target a “10” vulnerability, rather than, for instance, a “3.”

315. *Id.*

IX. Discussion and Conclusion

This article presents an analysis of civil liability for database owners' failure to safeguard confidential information. It focuses on situations where database owners fail to patch a computer security vulnerability, which facilitates compromise of sensitive information. In a civil action against a database owner, a key element of the liability analysis is the foreseeability of exploitation of the vulnerability at issue. This is the focus of the article. An analysis of the law and technology of a cyber attack identifies features that make a vulnerability foreseeably exploitable. The article further presents a numerical metric of the foreseeability of exploitation of a vulnerability. The metric combines numerical values for proxies of the "foreseeability features" into a composite score that represents the foreseeability of exploitation of the vulnerability.

The proposed metric is not intended to displace expert discretion and human judgment, but rather, to complement it.³¹⁶ The metric is designed to be flexible and allows expert users to adapt it to particular situations. Consider, for instance, the case of two identical security vulnerabilities, one in a computer system with defensive strategies such as firewalls and intrusion detection, and the other in an insecure system. The superior security has reduced the likelihood that a vulnerability in the secure system will be exploited. It may be more difficult to exploit in the secure system, because an attacker has to use substantial technical expertise to circumvent the firewall. The security configuration may also allow exploitation of the vulnerability only during an extremely limited time window, a feature known as "high access complexity." Even though the vulnerabilities are identical, the vulnerability in the secure system should be given lower foreseeability scores for "ease of exploitation" and "access complexity", compared to the vulnerability in the insecure system.

Other variables could plausibly have been included in the metric, but the chosen set reflects a compromise between completeness, parsimony, and accuracy. Some of the variables in the model can be refined. The authentication metric in the baseline model, for instance, does not differentiate between one or multiple authentication steps. It simply asks, in binary fashion, whether exploitation of a vulnerability does, or does not, require authentication. As such, it does not distinguish between a

316. See IAN AYRES, SUPER CRUNCHERS HOW ANYTHING CAN BE PREDICTED 116-124 (2007), (exploring the co-existence of expert discretion and statistical models and discussing a Virginia statute which allows the Virginia Department of Corrections to subject an offender who has served his full sentence to civil commitment in a state mental hospital. The statute specifies that the commitment process should be set in motion if a statistical algorithm predicts that the offender has a high likelihood of recidivism. The committee reviewing the case has the discretion to release the offender notwithstanding the result of the algorithm).

vulnerability that requires multiple authentications, such as authentication to an operating system as well as the application running on it, and a vulnerability that requires authentication only to the operating system. A more sophisticated authentication score might use three values that account for whether exploitation requires no authentication, a single authentication, or multiple authentications.³¹⁷

The foreseeability metric is not intended to provide a conclusive resolution of the issue of reasonable foreseeability, but it brings a measure of objectivity to an issue that is often clouded by distortions, such as hindsight bias. Research in behavioral psychology suggests, for instance, that people tend to overstate the predictability of past events,³¹⁸ and that after-the-fact decisions by judges and juries about what an individual knew or should have known may be influenced by knowledge of what actually occurred.³¹⁹ Social science researchers report that judges are as susceptible to hindsight bias as the general public.³²⁰

Empirical research suggests that hindsight bias extends to judgments of reasonable foreseeability.³²¹ Professors Susan LaBine and Gary LaBine report results of a field experiment in which a sample of community residents were asked to read different clinical scenarios involving treatment of potentially dangerous patients.³²² The scenarios presented different outcomes: (1) the patient became violent, (2) the patient did not become violent, or (3) no outcome was specified. The respondents were given identical scenarios except for the outcome, and were asked to rate the foreseeability of violence. The authors report that respondents who were told that the patient did in fact become violent, rated violence as more

317. See Mell, Scarfone, & Romanosky, *supra* note 19, at 87.

318. See, e.g., Jeffrey J. Rachlinski, *A Positive Psychological Theory of Judging in Hindsight*, 65 U. CHI. L. REV. 571, 571 (1998) (stating that “psychologists have demonstrated repeatedly that people overstate the predictability of past events - a phenomenon that psychologists have termed the ‘hindsight bias.’”).

319. See generally David A. Oliver, *Toxic Torts: Risk, Foreseeability and Causation*, TX. LAWYER, Oct. 13, 2003, available at <http://www.porterhedges.com/Toxic-Torts-Risk-Foreseeability-And-Causation.aspx>.

320. See, e.g., Chris Guthrie et al., *Inside the Judicial Mind*, 86 CORNELL L. REV. 777, 816 (2001).

321. Oliver, *supra* note 319, at 2 (“Judgments about foreseeability suffer similarly. By what is called hindsight bias, decisions about what was known or knowable are repeatedly distorted in favor of ‘they knew it all along.’ Hindsight bias occurs when knowledge of what actually occurred causes overestimation of how predictable the outcome was The more awful the outcome, the more judges and jurors will think it was foreseeable - regardless of how improbable it might have appeared at the time.”).

322. Susan J. LaBine & Gary LaBine, *Determinations of Negligence and the Hindsight Bias*, 20 LAW & HUM. BEHAV. 501 (1996).

foreseeable. Respondents who were told that the patient did not become violent reported that the absence of violence was more foreseeable.³²³

Cognitive biases in judicial decisions will likely be exacerbated where issues involve complex novel technologies unfamiliar to decision-makers,³²⁴ and also in cases involving a high profile event that was widely reported and that involved significant harm.³²⁵ These factors are often present in information security breaches, such as virus and worm attacks, identity theft, and denial of service attacks on high profile companies, government agencies, and the national critical information infrastructure. The analysis and results presented in this article attempt to contribute a rational framework and methodology to assist judicial decisionmaking on the issue of reasonable foreseeability in complex cases.

323. *Id.* at 511. Jeffrey Rachlinski cautions that, “(s)tudies on the hindsight bias [and its effect on foreseeability assessment] have documented its influence on probability estimates, not on *what* could have been envisioned.” Rachlinski, *supra* note 318, at 593.

324. See STEPHEN BREYER, ECONOMIC REASONING AND JUDICIAL REVIEW 12 (2003), available at <http://www.aei-brookings.org/publications/abstract.php?pid=672> (noting the difficulty courts often have in assessing reasonableness in cases involving complex technologies). See also Citron, *supra* note 2, at 265, n. 128 (“While lay juries ordinarily have difficulty assessing negligence in complicated technical cases, juries may have an especially challenging time assessing a database operator’s care over its security system given the rapid changes in technologies and new risks . . .”).

325. See, e.g., Oliver, *supra* note 319 (reporting that “the more awful the outcome, the more judges and jurors will think it was foreseeable - regardless of how improbable it might have appeared at the time.”).