Fordham Intellectual Property, Media and Entertainment Law Journal

Volume 18 *Volume XVIII*
Number 3 *Volume XVIII Book 3*

Article 8

2008

Are the Current Computer Crime Laws Sufficient or Should the Writing of Virus Code Be Prohibited?

Robert J. Kroczynski
Fordham University School of Law

Are the Current Computer Crime Laws Sufficient or Should the Writing of Virus Code Be Prohibited?

Cover Page Footnote

Alexander Southwell, Shari Sckolnick

Are the Current Computer Crime Laws Sufficient or Should the Writing of Virus Code Be Prohibited?

Robert J. Kroczynski*

A PDF version of this article is available online at http://law.fordham.edu/publications/article.ihtml?pubID=200&id=2738. Visit http://www.iplj.net for access to the complete Journal archive.

* J.D. candidate, Fordham University School of Law, 2008; B.S., Chemistry and Physics, Montclair State University, 1991; M.S., Chemistry, University of Stony Brook, 1994; M.Eng., Chemical Engineering, Stevens Institute of Technology, 2004. The author wishes to thank Professor Alexander Southwell for reviewing the original draft and making helpful suggestions as well as Shari Sckolnick and her team for their editorial contributions.

INTRODUCTION

Cybercrime is a problem that has developed with the increased use of computers and the Internet. At first, viruses plagued only the few mainframe computers, but this annoyance[1] expanded as personal computers became more readily available throughout the 1980s and 1990s.[2] The proliferation of viruses continued as stand-

[1] Most viruses were considered an "annoyance" when personal computers were rather rare, owned only by those with a true interest in their operation and usefulness, and when even professionally written software and operating systems contained many bugs, which hampered the reliable use of such systems. *See generally* SNORRE FAGERLAND ET AL., THE NORMAN BOOK ON COMPUTER VIRUSES 35 (2003), *available at* http://www.lan-aces.com/Norman_Book.pdf.

[2] Personal computers first became available with the Altair 8800, released in 1975, and the Apple 1 developed by Steve Wozniak and Steve Jobs in 1976. *See 1973 AD to 1981 AD The First Personal Computers*, (abstracted from CLIVE MAXFIELD & ALVIN BROWN, BEBOP BYTES BACK: AN UNCONVENTIONAL GUIDE TO COMPUTERS (1998)), *available at* http://www.maxmon.com/1973ad.htm (providing a basic time line of early personal computer development) (last visited Nov. 2, 2007).

alone systems became inter-connected, through bulletin board systems, and eventually via the Internet, thus increasing the potential for damage to computer systems.

To combat the harm caused by these small yet malicious computer programs, state and federal governments attempted to prosecute the people causing this damage. At first, prosecutors relied upon the statutes used in prosecuting standard real world crimes, but these laws were ineffective because they were not written to address the unique aspects of computer crimes.[3] New statutes focusing strictly on computer crimes were therefore passed with language directed at the particular activities involved with developing computer technology.[4] However, even these new statutes have been unable to eliminate the damage caused by malicious programs.[5]

This Note examines why current computer crime laws are ineffective, and will continue to be ineffective, in preventing the damage caused by virus and worm computer programs unless significant changes are made. This Note then presents an alternative approach to fighting cybercrime that would prohibit the writing of virus and worm programs.[6] Part I outlines the issues involving computer systems, the Internet and malicious software and introduces the concept of cybercrime. Part I.A describes

---

[3] *See* Orin S. Kerr, *Cybercrime's Scope: Interpreting "Access" and "Authorization" in Computer Misuse Statutes*, 78 N.Y.U. L. REV. 1596, 1605–07 (2003) [hereinafter *Cybercrime's Scope*]; *see also* Aaron Busstein, *A Survey of Cybercrime in the United States*, 18 BERKELEY TECH. L.J. 313, 315 (2003) (describing the general approach law enforcement took when first confronted with cybercrimes).

[4] *See* 18 U.S.C. § 1030 (1984); *see generally* ORIN S. KERR, COMPUTER CRIME LAW: CASES AND MATERIALS (West Publishers 2006) [hereinafter COMPUTER CRIME LAW].

[5] Many believe the amount of damage caused by computer viruses is greatly inflated by those reporting it. However, it is also believed that the number of systems affected is greatly underreported to avoid embarrassment and loss of client or consumer confidence. *See* Marc D. Goodman & Susan W. Brenner, *The Emerging Consensus on Criminal Conduct in Cyberspace*, 10 INT'L J.L. & INFO. TECH. 139, 155–57 (2002) (describing some of the difficulties in quantifying the number of cybercrimes committed and the amount of damage sustained).

[6] The approach is not new; it was previously suggested by some computer professionals, but was not seen at the time as a viable or legal alternative to preventing the damage caused by malware. *See* Kim Zetter, *Freeze! Drop That Download! The Words Are the Bomb*, PCWORLD, Nov. 16, 2000, *available at* http://www.pcworld.com/news/article/0,aid,34406,pg,2,00.asp.

malware and explains the technical details of how viruses and worms work. Part I.B explains how viruses and worms are released to infect other systems. Part II examines the threat posed by viruses and worms to computer users and society. Part III presents how cybercrime laws currently seek to curb the proliferation of virus code and protect the businesses and individuals potentially harmed by virus outbreaks. Part III.A outlines the general approach taken to combat cybercrime. Part III.B presents the current approaches taken by the federal and state cybercrime laws including the Federal Computer Frauds and Abuse Act of 2002. Part IV examines the possible results of prohibiting the writing of virus and worm programs. Part IV.A considers the problems and shortcomings of the current laws. Part IV.B discusses how a new law could address the problems and shortcomings of the current laws. Parts IV.C and D considers the issues that outlawing the actual writing of computer virus code might raise with the computer-using community, and whether the losses are balanced by the gains. This Note concludes by arguing that virus writing itself can and should be made illegal.

I. BACKGROUND OF CYBERCRIME AND VIRUSES

Cybercrime encompasses all criminal acts that use a computer.[7] This category of offenses include both acts where the computer is a key element of the offense,[8] and where the computer helps facilitate a crime that would be more difficult or impossible without it.[9] Cybercrime does not include ordinary crimes that use a computer to record or otherwise do something that could be accomplished by ordinary means, such as an accountant's journal

---

7 *See generally* COMPUTER CRIME LAW, *supra* note 4, at v–vi.

8 *Id.* at 1 (presenting the division between computer misuse crimes and traditional crimes committed using computers). The dissemination of a computer virus or computer hacking is a computer misuse crime because a computer system is a necessity to effectuate the criminal act. This differs from the dissemination of child pornography or fraud, neither of which require a computer but instead utilize them to facilitate the execution of the crime.

9 *Id.* Both of these activities would fall under the heading of substantive computer crime law because the methods of perpetrating the crime involve computer technologies, which must be addressed in a statute.

to record illegal profits, pencil and paper to draw a diagram for a robbery, or snail mail[10] for communication between accomplices.

The dissemination of viruses and worms is a computer misuse crime, because it could not exist without computers.[11] This crime involves creating and executing computer code that can transfer copies of this computer code to other users' computer systems.[12] This unwanted transfer of computer code typically results in some form of harm to the recipient's computer system.[13] The unwanted transfer of code is only one facet of computer crimes, which federal and state laws attempt to deal with.[14]

Even with state and federal computer crime laws in place,[15] there are very few prosecutions for the damage done by viruses and worms released into the wild.[16] This is because it is difficult to

---

[10] "Snail mail" is defined as physical letters delivered by the U.S. Post Office, or some other delivery system, as opposed to some form of electronic mail. *See* Snail Mail, THE AMERICAN HERITAGE DICTIONARY OF THE ENGLISH LANGUAGE (4th ed. 2000), *available at* http://dictionary.reference.com/browse/snail%20mail (last visited Nov. 14, 2007).

[11] Currently, the closest physical world analogy to a computer virus is a robot programmed to produce copies of itself which then move to new locations and replicate only to have the replicates repeat the process. *See* PETER SZOR, THE ART OF COMPUTER VIRUS RESEARCH AND DEFENSE 5–7 (Addison-Wesley 2005) (describing John Von Neumann's theory of self-reproducing automata, the 'Universal Machine,' and self-replicating machines including nano-bots).

[12] *See infra* Part I.A.

[13] This harm could be the loss of application programs or data, as well as the loss of confidence in the safety and security of the computer system.

[14] Computer crimes span the range of online stalking and extortion to online fraud schemes, accessing child pornography, and "hacking" into other users' computer systems for fun and profit. *See* Goodman & Brenner, *supra* note 5, at 144–49.

[15] The federal statute that most computer crimes are prosecuted under is the Computer Fraud and Abuse Act. The first version of this statute was passed in 1984. 18 U.S.C. § 1030.

[16] *See* Ronald B. Standler, *Examples of Malicious Computer Programs* (2002), *available at* http://www.rbs2.com/cviru s.htm (identifying five prosecutions and convictions made against virus writers). Of the few perpetrators who have been caught, most have pleaded guilty to the charges. This resulted in very few trial and appellate opinions clarifying the state and federal cybercrime laws. Various experts believe these prosecutions were only possible because the perpetrators made the mistake of remaining in jurisdictions where they could be apprehended. *See also* Kelly Cesare, *Prosecuting Computer Virus Authors: The Need for an Adequate and Immediate International Solution*, 14 TRANSNAT'L LAW. 135, 152–53 (2001) (discussing how David Lee Smith was only successfully apprehended for the release of the 'Melissa' virus in 1999 because he wrote the virus in the United States and remained in the country after its release).

identify and track down perpetrators.[17] The anonymity of cyberspace allows a perpetrator to conceal his identity, and cover his electronic tracks in ways that make it much more difficult for law enforcement to uncover information as compared to real space crimes. Additionally, it is difficult to apply laws to prosecute cybersuspects without a proper understanding and recognition of what has actually resulted from the suspect's acts.[18] The enforcement officer must recognize that a theft can occur without the original article missing, a trespass can occur without the person being on the same premises as the computer system, and a computer or its data can be rendered inoperable without being physically vandalized.[19]

A. Definition of Viruses and Technical Descriptions

The following section will provide a detailed description of viruses and worms to help in understanding their nature and identifying them in the digital world. An understanding of the technical aspects of a virus code is important so that one may determine what type of programming should be outlawed. It is also important to create awareness that some forms of programming and dissemination should not be completely protected speech.[20]

---

[17] *See* Susan Brenner, *Toward a Criminal Law for Cyberspace: A New Model of Law Enforcement*, 30 RUTGERS COMPUTER & TECH. L.J. 1, 25–32 (2004) (identifying the different characteristics of cybercrime which make enforcement much more difficult than "real space" crimes). These differences include lack of any proximity to the location of the computer crime, the scale of the crime committed by a single individual, the speed at which the crime can be carried out, and the lack of physical constraints to limit the crime. *See* Goodman & Brenner, *supra* note 5, at 142 (describing some of the difficulties in fighting cybercrime). *See also* Cesare, *supra* note 16, at 151–53 (discussing the problems of enforcing cybercrime laws).

[18] *See* Marc D. Goodman, *Why the Police Don't Care about Computer Crime*, 10 HARV. J.L. & TECH 465, 486 (1997). A person cannot be charged with damaging a computer if the malware did not cause recognizable damage. Nor can someone be charged with theft if there was nothing in the code to facilitate the taking of information or data from an infected system.

[19] *Id.* at 482.

[20] *See generally* Eugene Volokh, *Crime-Facilitating Speech*, 57 STAN. L. REV. 1095, 1098–103 (2005) (discussing aspects of free speech protection that allow the furtherance of crimes and how different types of crime are interconnected under a rubric of free speech).

The term malware is short for malicious software.[21] It encompasses a wide range of program types including viruses, worms, logic bombs, Trojan horses, keyloggers, zombie programs, and backdoors.[22] Each of these programs has a different structure and overall purpose, but there can be overlap.[23] The term malware is now also used in reference to cookies and other forms of spyware when it operates without the user's knowledge or against his wishes.[24]

Viruses and worms damage or destroy programs and data files located on infected computers. The use of keyloggers[25] allow the misappropriation of secret information to be used for financial or other gain later by the miscreant. Other types of malicious software such as backdoors, [26] Trojan horses,[27] and zombie programs are capable of allowing access into a computer system and its sensitive and confidential information. This type of software provides an opportunity to damage or hijack the machine while being able to eliminate any evidence of the crime. Even though there are other categories of malware that can cause damage to computer systems, they do not have the same potential to cause the widespread damage that viruses or worms do. Thus, only viruses and worms are directly addressed in the remainder of this Note.[28]

---

[21] SZOR, *supra* note 11, at 28.

[22] *See id.* at 28–36 (defining each of the different forms of malicious software).

[23] *See id.* (introducing the terminology used in describing the various computer viruses and worms).

[24] *See* Definitions of Malware on the Web, http://www.google.com/search?q=define:malware (providing numerous web definitions) (last visited Nov. 5, 2007).

[25] SZOR, *supra* note 11, at 36. A keylogger is a program that records each key as the computer user types and then relays the information to the perpetrator. *Id.* The criminal's hope is that the keylogger is able to obtain information such as a credit card or bank account numbers that he can then exploit later.

[26] *Id.* at 331 (stating back doors listen for a connection from the attacker and then allow access to the system). Back Orifice was the most familiar form of this type of malware. *Id.*

[27] *Id.* at 31–32. Trojan Horse programs masquerade as legitimate versions of commercial software, but they contain secret code allowing a cyber-criminal access to the computer system through a back door.

[28] This does not imply that these programs do not pose a serious risk to computer use, but only that they do not have the characteristics pertinent to this discussion. *See* Yury Mashevsky, *Malware Evolution 2005*, Feb. 8, 2006, *available at* http://www.viruslist.com/en/analysis?pubid=178949694 (showing that Trojans now make up the largest portion of malicious software being encountered).

1. Viruses

Similar to viruses that may infect a living organism, computer viruses can self-replicate.[29] A virus makes copies of itself in order to spread to new systems against the user's wishes and without his knowledge.[30] The virus program accomplishes this by writing a set of machine instructions,[31] which are attached to another executable file[32] in some manner when the program in which it is embedded is executed by the computer's central processing unit (CPU).[33] Viruses must be a part of a program, which the computer identifies as a set of instructions to be executed.[34] When the newly infected program is run, the process repeats itself.[35]

Virus code can be added to an existing executable file in a variety of ways.[36] A plain text file or image file does not contain any executable instructions.[37] Since the CPU does not expect any

[29] *See* SZOR, *supra* note 11, at 18. Dr. Frederick Cohen, who first coined the term "virus," defined it as "a program that is able to infect other programs by modifying them to include a possibly evolved copy of itself." *Id.*

[30] *Id.* at 20 ("Computer viruses are self-automated programs that, against the user's wishes, make copies of themselves to spread themselves to new targets.").

[31] Machine instructions are a set of binary digits of a predetermined length which the computer recognizes as a particular operation to be preformed followed by the address or value to be operated on. This is referred to as the opcode. *See* ANDREW S. TANENBAUM, STRUCTURED COMPUTER ORGANIZATION 251–54 (4th ed.1999); *see also* SZOR, *supra* note 11, at 53–54 (explaining the dependency of virus code on the particular Central Processing Unit and its opcodes).

[32] An executable file is one that the computer interprets as instructions to perform specific operations as defined within the machine's hardware. *See* M. MORRIS MANO, COMPUTER SYSTEM ARCHITECTURE 251–254 (2d ed. 1982).

[33] Technical details about the design and operation of a computer's central processing unit (CPU) should be looked up in textbooks on computer architecture and assembly language. *See generally id.*; DAVID A. PATTERSON AND JOHN L. HENNESSEY, COMPUTER ORGANIZATION AND DESIGN: THE HARDWARE/SOFTWARE INTERFACE (3d ed. 2004); TANENBAUM, *supra* note 31, at 39–56; RICHARD C. DETMER, INTRODUCTION TO 80X86 ASSEMBLY LANGUAGE AND COMPUTER ARCHITECTURE (Jones & Bartlett 2001).

[34] *See* Carolyn P. Meinel, *Introduction to Computer Viruses Part I*, GUIDE TO (MOSTLY) HARMLESS HACKING, July 19, 1998, *available at* http://www.happyhacker.org/gtmhh/vol3no71.shtml.

[35] *See* Viruslist.com, http://www.viruslist.com/en/virusesdescribed?chapter=152540474 (last visited Nov. 12, 2007).

[36] *See* SZOR, *supra* note 11, at 129–57 (describing the ways virus code can be introduced to a program or system).

[37] This does not include "macros" which were added to word processing programs and other application programs to automate certain tasks. Virus writers found this

executable code in such a file, it does not look for opcodes when opening one.[38] The application program used to open the file interprets any formatting, instructions, or macros. Implementation of macros, however, is another method of embedding virus code for execution by the application software.[39]

The most blatant way to infect the program is to erase the entire executable program and insert the virus code in its place.[40] When this is done, the original file can no longer perform its original function. In fact, running the program will only retrigger the virus code.[41] This effect may be considered a compromise of the system's integrity.[42] This type of virus infection, however, is fairly easy to detect. The program file changes in size from what it was originally, and the program does not produce any of the expected results when attempts are made to run it.[43] These aspects of the virus infection make it rather easy to detect and quarantine the malicious code, thus preventing its spread or propagation.[44] This keeps down the overall amount of damage caused by this virus type.

One method to avoid the shortcomings of a virus code that overwrites its target program is to attach the virus code to the beginning or end of the program's code.[45] This method still has the problem of noticeably changing the file's size, however, the program's original code continues to function, thus masking the fact it has been infected by the virus.[46] This allows the attached virus code to be executed many more times because there is no

functionality useful for writing malicious code that could operate on many computers and damage documents and other files when opened. *See id.* at 66–69 (explaining how virus code is dependent on the programming environment).

38 *See supra* note 31 (introducing the concept of opcodes).

39 *See* SZOR, *supra* note 11, at 66–69 (describing how macro viruses are created and spread).

40 *Id.* at 130–31 (describing overwriting viruses).

41 *Id.* at 130.

42 See *infra* Part III.B.4.

43 *See* SZOR, *supra* note 11, at 130–131 (commenting on the shortcomings of this type of virus).

44 *Id.*

45 *Id.* at 132–35 (describing appending and prepending viruses).

46 *Id.*

immediate evidence of a problem.[47] The original program continues to function as the user desires, although both the program and system's integrity could again be considered compromised.[48] When a user realizes that his system is infected, he will attempt to remove the virus code from the system. This effort results in costs that could be considered consequential damages.[49]

Another means of infecting a program without altering its size is by placing the virus code within one or more cavities[50] within the host.[51] This avoids simple detection methods revealing the presence of the virus code.[52] It also allows the original program to continue functioning.

Virus code can also be placed in a disk's boot sector.[53] When the virus code is located in a boot sector, it takes direct control of the system away from the owner or user and tricks the CPU into loading the virus writer's code on start-up.[54] Inserting such an instruction may compromise the integrity of the computer system, because it directly alters the way the system functions.[55] This is subtly different from an executable virus because of the level at

---

[47] This considers only the execution of the virus code itself, and not any payload, which may cause noticeable damage outside the infected program. *See* Meinel, *supra* note 34.

[48] *See* SZOR, *supra* note 11, at 66–69 (describing how macro viruses are created and spread).

[49] Contract law defines consequential damages as those foreseeable to the parties at the time the contract was formed. *See* JOHN D. CALAMARI & JOSEPH M. PERILLO, THE LAW OF CONTRACTS 547–48 (4th ed. 1998). Losses that do not "flow directly and immediately from an injurious act, but that result indirectly from the act" BLACK'S LAW DICTIONARY 416 (8th ed. 2004).

[50] Cavities consist of sections of code containing zeroes, spaces, holes, or other null values. *See* SZOR, *supra* note 11, at 136–39.

[51] *See id*. (describing cavity and fractionated cavity viruses).

[52] More advanced anti-virus software can detect these viruses through a checksum analysis. *See* Vasselin Bontchev, *Possible Virus Attacks Against Integrity Programs and How to Prevent Them*, PROC. 2ND INT'L VIRUS BULL. CONF. (1992), *available at* http://www.people.frisk-software.com/~bontchev/papers/attacks.html.

[53] The boot sector is the location on a hard drive or floppy disk where the computer looks for instructions on loading the operating system or other files located on the disk. By placing the correct type of instruction in a boot sector, the computer can be instructed to load a virus into memory before an operating system or anti-virus program is loaded. *See* SZOR, *supra* note 11, at 122–29.

[54] *See id*. at 125 (describing how a Master Boot Record can become infected).

[55] *See infra* note 111 and accompanying text.

which the boot sector virus works.[56] It supersedes all other software priorities by taking control of the computer system before any other software is loaded. An executable virus operates on top of the operating system and any other memory resident programs.[57]

Each of these computer virus infections needs a method of spreading to additional systems just as a real microbe needs a vector to spread to new hosts.[58] Viruses, unlike worms, do not self-propagate. In order to spread, a human agent must distribute the virus to new systems.[59] A virus typically spreads when an infected program is shared with others. Initially, this was accomplished by physically passing along a program on a portable media,[60] which had a boot sector virus embedded in it, or an infected file saved on it. With the development of bulletin board systems accessed through modem and telephone lines, this physical transfer was no longer the only means of transferring files. Software could be directly uploaded and downloaded between individual computers electronically. The Internet further increased the speed and volume of these electronic transfers using e-mail, which can send a file to multiple recipients almost instantaneously.[61]

---

[56] *See* SZOR, *supra* note 11, at 122–29 (describing boot viruses generally).

[57] In fact, executable virus code relies on an operating system being loaded in order to function as designed, and is typically operating system specific. *See id.* at 55 (explaining operating system dependency of virus programs).

[58] *See* Vector (biological), Wikipedia, http://en.wikipedia.org/wiki/Vector_%28biology%29 (last visited Dec. 19, 2007); *see also* Virus, Wikipedia, http://en.wikipedia.org/wiki/Virus (last visited Dec. 19, 2007).

[59] One example is the sneaker-net, referring to the physical walking of an infected disk over to another person. *See* Sarah Gordon, *Technologically Enabled Crime: Shifting Paradigms for the Year 2000*, 14 COMPUTERS & SECURITY 5, 393 (1995) *available at* http://vx.netlux.org/lib/pdf/Technologically%20Enabled%20Crime%3A%20Shifting%20Paradigms%20for%20the%20Year%202000.pdf.

[60] Portable media includes floppy disks, compact discs (CDs), ZIP disks, flash cards, or any other magnetic or optical storage device.

[61] E-mail attachments do not have a boot sector, so this vector cannot transmit boot sector viruses.

2. Worms

Worms differ from viruses in two fundamental ways.[62] First, they do not need to infect other programs in order to spread.[63] Second, they can propagate without a human agent.[64] They travel between computer systems across network connections by exploiting holes or flaws in the programming.[65] Like viruses, however, they may create modified copies of themselves. Worms are considered a subclass of virus.[66]

In order to function, a worm must have two essential parts: a target locator component and an infection component.[67] The target locator examines files on an infected system in order to find available systems to which it could send itself.[68] The basic means of accomplishing this is by locating the e-mail address book on the infected system.[69] Alternatively, a worm program can be designed to search for e-mail addresses on network servers[70] or by using Internet search engines.[71] This method is similar to the one used by spammers.[72]

Once a worm locates these target e-mail addresses, it must exploit some weakness or bug in the programs that support the

---

[62] Eugene H. Spafford, *Computer Viruses as Artificial Life*, ARTIFICIAL LIFE, VOL. 1, NUM. 3, § 2.1 (1994), *available at* http://www.scs.carleton.ca/~soma/biosec/readings/spafford-viruses.pdf.

[63] *See* Fred Cohen, *Computer Viruses - Theory and Experiments*, COMPUTERS & SECURITY, VOL. 6, § 2 (1984), *available at* http://vx.netlux.org/lib/pdf/Computer%20Viruses%20-%20Theory%20and%20Experiments.pdf.

[64] *See* Eugene H. Spafford, *The Internet Worm Incident*, TECH. REPORT CSD-TR-933, Department of Computer Science, Perdue University (1988), *available at* http://homes.cerias.purdue.edu/~spaf/tech-reps/933.pdf (discussing the technical details of the worm released by Robert T. Morris in 1988).

[65] *Id.*

[66] *See* SZOR, *supra* note 11, at 314–15 (describing the structure of computer worms compared to viruses).

[67] *See id.* at 315–16 (describing the components of a worm program).

[68] *See id.* at 319 (describing harvesting of e-mail information from address books).

[69] *Id.*

[70] *See id.* at 320–21 (describing ways to obtain e-mail addresses from network servers).

[71] *See id.* at 321–22 (describing ways to obtain e-mail addresses with Internet search engines).

[72] *See id.* at 323–24 (describing how the methods of obtaining e-mail addresses can be combined in a worm).

computer network.[73] Worms use exploits[74] to transfer its code directly over the network, thereby avoiding the need to infect some carrier program.[75] The simplest form of weakness used by a worm to infect a system is social engineering using an enticing e-mail header or file name to trick a receiving party into opening the letter or attachment.[76] Upon opening the attachment, the worm program is executed on that computer.[77] This is also one of the hardest exploits to counter, because it involves protecting the system user from himself.[78] No software package can prevent a user from purposely granting access to malicious code.

In each of these instances, the issue of damage caused by a worm is questionable. Without executing some form of malicious code, the worm simply takes up residency on the system, and in some cases this is only temporary.[79] However, there is no question that a worm compromises a computer system's integrity. The worm code immediately causes the computer to behave in a manner that is against the owner's wishes and without his

---

[73] Robert Morris's worm program capitalized on two weaknesses and one bug in the programs used to allow the network to function. The bug was located in the *fingerd* program used to gain information on network users. The program code allowed buffer overruns from overly long input strings. The first weakness was a *debugger* function available in the *sendmail* program, which was typically left accessible by network administrators as a matter of convenience. The second weakness involved *trusted hosts*. This feature allowed someone on a system marked as trusted to access other systems without use of a password. The third method of gaining access to systems involved a brute force method of guessing passwords on secured systems. *See* Eugene H. Spafford, *The Internet Worm Program: An Analysis*, TECH. REPORT CSD-TR-823 § 3, Department of Computer Sciences, Purdue University (1988) [hereinafter *The Internet Worm Program*] (describing in computer science terms the technical details of each of the flaws exploited by the worm).

[74] An exploit is a flaw in the system programming or configuration that allows the worm code to access another computer, which its user would otherwise consider safe and secure. *See* FFIEC Information Technology Examination Handbook Glossary, http://www.ffiec.gov/ffiecinfobase/html_pages/gl_01a.html (last visited Dec. 20, 2007).

[75] *See* SZOR, *supra* note 11, at 341–44 (discussing three modes of attack on targeted systems).

[76] *See id.* at 333–34 (discussing some tricks used by worms to get executed).

[77] Some might argue that this violates one of the definitions of a worm, because it requires human intervention in order to propagate similar to a standard virus.

[78] *See* SZOR, *supra* note 11, at 333–34 (discussing some tricks used by worms to get executed).

[79] *See id.* at 29–30 (defining rabbits as a worm variant which terminates its code on one system after infecting another).

knowledge. This is true even when the user is the one to activate the worm by opening an attachment because the result is unexpected and unwanted.

3. Payloads

A virus or worm may or may not have a payload.[80] The payload is additional code beyond what is needed for the virus to function. If there is a payload, it can be nondestructive,[81] somewhat destructive,[82] or highly destructive.[83] A nondestructive payload is typically some form of amusement including graphics or music.[84] Somewhat destructive payloads may alter files or affect system performance, but don't have any serious lasting effect.[85] The most serious payloads are highly destructive, and are of the most concern. These viruses may overwrite files or erase them from the disk altogether.[86] The most malicious payload does not do readily recognizable damage, but instead makes small modifications continuously over time until all the files are corrupted in some manner.[87] This kind of code causes more damage because of the subtle way it causes damage. It is difficult to detect early on, and when it is finally noticed, it has already permeated the entire system. The final form of damage is the attack of hardware.[88] The code actually alters programmable chips on hardware devices or containing the BIOS preventing the actual hardware or computer from functioning. All of the highly

[80] *See id.* at 296 (stating that the majority of viruses do not carry any form of code beyond that required to replicate, or at most a name or message to be found by anti-virus researchers).

[81] *See id.* at 297 (describing non-destructive virus and worm payloads).

[82] *See id.* at 300 (describing somewhat destructive virus and worm payloads).

[83] *See id.* at 301–06 (describing highly destructive virus and worm payloads).

[84] *See id.* at 298 (mentioning W95/Marburg as a virus of this type. Marburg randomly placed 256 icons on the desktop).

[85] *See id.* at 301 (mentioning the WM/Wazzu.A virus as a somewhat destructive virus. Wazzu randomly scrambled three words and placed "wazzu" into documents.).

[86] *See id.* (mentioning the Michelangelo virus as one of the well known viruses in this category).

[87] *See id.* at 302–03 (discussing data diddlers as a particularly malicious form of data corruption).

[88] *See id.* at 305–06 (discussing how viruses could alter a machine's Flash BIOS thereby preventing boot up).

destructive payload attacks cause damage to the files on a user's system. These are exactly the results the criminal statutes attempt to address.[89]

B. How Malware is Released

In order to perpetrate a crime through a virus or worm, a person must first create a malicious program. When a programmer attempts to create a computer virus or worm,[90] he has a very definite purpose in mind. A virus is unlike any other computer code. It is specifically designed to replicate itself onto uninfected machines.[91] The more complex the computer programming used to accomplish this by making the virus undetectable and resistant to treatment by anti-virus software,[92] the more obvious the programmer's intention to create a malicious form of program.[93] A worm differs in the manner in which it spreads, but there must be a similar intent to produce code with the sole purpose of obtaining unauthorized access to a computer system and then replicating and propagating itself.

---

[89] 18 U.S.C. § 1030(a)(5)(A)(i) (2002) ("intentionally causes damage . . ."); § 1030(a)(5)(A)(ii) ("recklessly causes damage . . ."); § 1030(a)(5)(A)(iii) ("causes damage . . .").

[90] Throughout this Note, the term virus will generally include worms. *See supra* note 66 and accompanying text.

[91] *See supra* Part I.A.

[92] *See* SZOR, *supra* note 11, at 220–47 (discussing methods of protecting virus code from anti-virus software by "armoring" them).

[93] It has been debated whether a virus is inherently malicious, and most researchers believe the malicious aspects of viruses are accidental rather than purposeful. *See, e.g.*, Meinel, *supra* note 34. A virus is defined by its ability to self-replicate and not by the damage it might do. Some computer scientists feel that mass media corrupted the definition of viruses to include a malicious nature. Some researchers have stated that very few viruses contain any malicious code. The resulting damage is usually caused by programming flaws, by the overtaxing of computer resources during propagation, and by the lack of control over the program's behavior once it has been released into the wild. *See, e.g.*, Vesselin Bontchev, *Are "Good" Computer Viruses Still a Bad Idea?*, PROC. EICAR'94 CONF., 25–47, *available at* http://www.people.frisk-software.com/~bontchev/papers/goodvir.html [hereinafter *Are "Good" Computer Viruses Still a Bad Idea*].

The second step is the virus or worm's release "into the wild,"[94] in order to cause harm. The release of a worm or virus can be accomplished in a number of different ways. The most obvious way to purposely cause a virus outbreak or computer infection is to activate the code on a system connected to the public through the Internet or used for downloading files. It could also be disseminated on a form of portable media to unsuspecting users. In both cases the person initiating the virus outbreak is doing so purposefully and with the hope and expectation that the virus will spread. A less direct method of causing an outbreak involves the virus writer uploading his executable code to a website or bulletin board.[95] This places the functioning virus program in the hands of some third party who may then initiate the outbreak by the same means available to the writer himself. The virus writer in this scenario does not know if the virus will be released by the other person, but expects that at least one person who accesses the program will in fact execute it. A third even more attenuated method of disseminating virus code involves providing the public, through a website or a bulletin board, with just the uncompiled source code as a text file.[96] This is similar to the previous scenario, but requires the person who acquires the code to go through an extra step of compiling the program into an executable file before releasing it. This method counts on the person having

[94] IBM researcher Dave Chess coined the phrase "into the wild." It covers virus code, which can function on commercial systems in general use by the public. *See* SZOR, *supra* note 11, at 26.

[95] *See* Sarah Gordon, *Technologically Enabled Crime: Shifting Paradigms for the Year 2000*, COMPUTERS AND SECURITY § 3.1 (1995), *available at* http://www.research.ibm.com/antivirus/SciPapers/ Gordon/Crime.html (describing the use of bulletin boards, newsgroups, and websites for the dissemination of virus code).

[96] Text versions of virus source code are available on the same sites providing the executable code as well as in books on the topic. The available material can be either technical computer science treaties on how to write the components necessary for a virus, or underground publications containing the code for various existing viruses. *See* FREDRICK B. COHEN, A SHORT COURSE ON COMPUTER VIRUSES (ACS Publ'ns 1990); FREDRICK B. COHEN, IT'S ALIVE! (John Wiley & Sons 1994); JOHN R. KOZA, GENETIC PROGRAMMING: ON THE PROGRAMMING OF COMPUTERS BY MEANS OF NATURAL SELECTION, (MIT Press 1992); MARK LUDWIG, THE LITTLE BLACK BOOK OF COMPUTER VIRUSES (Am. Eagle Publ'ns 1991); MARK LUDWIG, THE GIANT BLACK BOOK OF COMPUTER VIRUSES (Am. Eagle Publ'ns 1995); MARK LUDWIG, THE GIANT BLACK BOOK OF COMPUTER VIRUSES (Am. Eagle Publ'ns 2d ed.1998); MARK LUDWIG, THE LITTLE BLACK BOOK OF E-MAIL VIRUSES: A TECHNICAL GUIDE (Am. Eagle Publ'ns 2002).

the software necessary to compile the source code into executable code. Finally, the virus might also escape accidentally from a writer's system if he does not keep it isolated from networks or carrier programs.[97]

In considering the intent and culpability of the virus writer, the first and last scenarios are cases where the virus has been released into the wild, but only in the first case could it be done purposefully. In the second and third scenarios, the virus could be considered purposefully distributed by the writer, but in neither case has the writer released it. The second case involves a functional form of the virus code which could be released without any further effort or expertise required by a third party. The third case involves a minimum level of effort by any third party that acquires the source code to put it into a functional form by compiling it.[98] There is a question of responsibility if a third party causes damage through the release of the virus code, particularly if the code is already in a functioning form.[99] The editing and compiling of source code requires an intervening human actor to put the code into a form, which is capable of causing damage.[100] Additionally, the writer may not know for certain whether the program will actually function the way it was meant to once it is installed on a system for which it was not specifically written.[101] However, the question of whether the program will work as envisioned by its creator is separate from his intentions in writing and releasing the code.[102]

---

[97] *See* SZOR, *supra* note 11, at 612 (discussing the importance of not introducing viruses to non-isolated systems).

[98] Some authors and scholars mistakenly believe that the computer program text or "source code" can directly infect another system by self-executing or through an interpreter program. This is not possible. Only executable code can be automatically loaded into a computer's random access memory and interpreted as instructions by the central processing unit.

[99] *See generally* WAYNE R. LAFAVE, CRIMINAL LAW §§ 13.1–2 (4th ed. 2003) (discussing the requirements for accessories and accomplices of a crime).

[100] *See* SANFORD H. KADISH & STEPHEN J. SCHULHOFER, CRIMINAL LAW AND ITS PROCESSES: CASES AND MATERIALS 536–37 (7th ed. 2001) (discussing causation and intervening human actions).

[101] Virus code which functions within a particular system environment, but not out on commercial systems, is termed a "zoo" virus. *See* SZOR, *supra* note 11, at 26.

[102] Many virus authors claim they did not know that the virus or worm program would behave the way it did, but this does not change their intent. *See* Standler, *supra* note 16

II. THE THREAT POSED BY VIRUSES AND WORMS

Society has identified malicious software including viruses and worms as one of the threats to computer systems. The outbreak and infection of computer systems by viruses and worms causes hundreds of millions if not billions of dollars in damage for each major occurrence.[103] It also has a social cost that is not easily measured—the computer and Internet-using public's lost faith in the safety and security of the online world. This fear and aversion is a psychological cost, which reduces the use of the Internet for its beneficial and commercial purposes.

III. CURRENT LEGAL EFFORTS TO FIGHT CYBERCRIME

A. Background of the Federal and State Cybercrime Statutes

The federal and state governments determined malicious software should be dealt with through criminal statutes. The statutes first appearing in the early 1980's approached the threats posed by malicious software and the behaviors of the persons responsible for these threats in a specific way.[104] The federal statute and most state statutes focused on the act of accessing a computer without authorization and thereby either causing damage or obtaining some form of protected information. This is because the earliest laws focused on the efforts of hackers to gain access to important governmental or private computer systems.[105] These initial statutes were modified over time to address the proliferation of viruses and worms, but the focus remained on the malicious program gaining unauthorized access to the computer system. While gaining access is the direct and specific act that can be

(explaining how the comments in the original source code of the Morris Worm indicated the author's actual intent despite his claims to the contrary).

[103] *See* Standler, *supra* note 16 (listing the recent virus outbreaks and the estimated economic harm caused by each outbreak).

[104] The Computer Fraud and Abuse Act focuses on the unauthorized access of computer systems and the damage resulting from such access. *See* 18 U.S.C. § 1030(a) (2002) (specifying unauthorized access of a computer system).

[105] *See* Eric J. Sinrod & William P. Reilly, *Cyber-Crimes: A Practical Approach to the Application of Federal Computer Crime Laws*, 16 SANTA CLARA COMPUTER & HIGH TECH. L.J. 177, 199–201 (2000).

criminalized, there are other key actions which must first be taken by a person attempting to perpetrate a computer crime through the use of a virus or worm. These actions include creating the computer code and releasing it. Associated with these acts are certain mental states or mens rea, which is addressed later in this Note.

B. The Current Laws Directed at Cybercrime

1. Federal Computer Fraud and Abuse Act.

The federal government first enacted the Computer Fraud and Abuse Act ("CFAA") in 1984.[106] The CFAA has been modified many times since it was first enacted in order to address developing issues in cybercrimes.[107] The most recent embodiment of the Act has broadened its applicability to offer protection to the vast majority of computer users.[108] It also addresses the infection of these protected computers by viruses and worms. The federal statute 18 U.S.C. § 1030(a)(5)(A)(i) requires the person to

> knowingly cause the transmission of a program . . . and as a result of such conduct, intentionally cause damage without authorization to a protected computer, and . . . (B) by conduct described in clause (i) . . . of subparagraph (A) cause . . . (i) loss to 1 or more persons during any one-year period aggregating at least $5,000 in value.[109]

The CFAA attempts to cover a very broad range of activities, but focuses mostly on the issue of unauthorized access.[110] Section

---

[106] 18 U.S.C. § 1030 (2002).

[107] The CFAA was amended in 1986, 1988, 1989, 1990, 1994, 1996, 2001, and 2002. *See id.* (outlining the legislative history of the CFAA).

[108] The CFAA defines "protected computer" as "a computer . . . which is used in interstate or foreign commerce or communication, ... " 18 U.S.C. § 1030(e)(2)(B). This definition effectively covers any computer connected to the Internet or used for business.

[109] 18 U.S.C. § 1030 (2002).

[110] *See* 18 U.S.C. § 1030(a)(1) (including "[w]hoever—having knowingly accessed a computer without authorization or exceeding authorized access . . . ."); 18 U.S.C. § 1030(a)(2) (covering "[w]hoever—intentionally accesses a computer without authorization or exceeds authorized access . . . ."); 18 U.S.C. § 1030(a)(3) (covering "[w]hoever—intentionally, without authorization to access any nonpublic

(5)(A) of the criminal statute encompasses the purposeful or knowing release of a computer virus, but not the reckless or negligent release of such a program.[111] The statute does not address the writing of the virus program, but only its knowing release and the damage intentionally caused by it. This particular section of the statute allows a virus writer to create virus code on his system and risk its release through negligence.[112] In addition, by requiring damage to be caused intentionally or knowingly, this statute requires the virus program to either be designed with a malicious nature recognizable in its code or to be released with the intent of causing harm.

Much less difficult to perceive than a person's intent is the actual unauthorized access of a computer system or network, and the compromise of its integrity.[113] Both access and a compromise of integrity can occur without any damage having been caused to the computer system or its files. Unauthorized access is easy to recognize because the evidence of the infection and the loss of system integrity is the presence of the virus on the victimized system and is not in the details of the virus's code or in understanding the writer's mental state at the time of its release. The virus infection is an objective element of the crime rather than a subjective one. The unauthorized access can be shown by the presence of any malicious code on the user's system. Even if it was never activated due to programming bugs or incompatibility with the host system, it is still evidence of someone other than the owner affecting changes to the computer. This unwanted and unknown change to the system is exactly what is encompassed by the term compromise of integrity.

computer . . . ."); 18 U.S.C. § 1030(a)(4) (covering "[w]hoever—knowingly and with an intent to defraud, accesses a protected computer . . . .").

[111] The possible means of disseminating a computer virus was discussed and differentiated in Part I.B, *supra*. A virus may be released purposely by its creator, or negligently through accidentally activating the code on a computer system connected to the Internet.

[112] The level of culpability required in these sections of the statute must be more than negligence to constitute a crime. *See* KADISH & SCHULHOFER, *supra* note 100, at 210 (stating that negligence "is distinguished from purposeful, knowing, or reckless action in that it does not involve a state of awareness").

[113] "Integrity" is defined as "soundness." THE OXFORD DICTIONARY OF CURRENT ENGLISH (2nd ed. 1996).

2. An Example of the Application of the Computer Fraud and Abuse Act

Early virus releases have been dealt with in different ways. *United States v. Morris*[114] approached the infection of computers through the issue of unauthorized access and damages. In *Morris*, defendant Robert Morris supposedly intended the program to operate only as a flag indicating vulnerable machines on the network. When the project went awry, he was prosecuted for violating the Computer Fraud and Abuse Act.[115] The malware that Morris released was designed with certain "protections" in place to prevent multiple infections of the same system.[116] Morris made some initial calculations regarding the program's propagation through the network.[117] The worm contained no payload, so there was no obvious intent to cause damage revealed by the code itself.[118] All of these behaviors indicate a lack of culpable mens rea regarding the damages element required by the 2002 version of (5)(A)(i).[119]

If Morris had been prosecuted under the 2002 version of § 1030 he would have had a much better defense; however the version he was prosecuted under in 1990 only required:

> intentionally access[ing] a Federal interest computer without authorization, and by means of one or more instances of such conduct alters, damages, or destroys information in any such Federal interest computer, or prevents authorized use of any such computer or information, and thereby (A) causes

---

[114] 928 F.2d 504 (2d Cir. 1991) (deciding the first case involving an internet worm).

[115] 18 U.S.C. § 1030(a)(5)(A) (1988).

[116] *Morris*, 928 F.2d at 506; *see also The Internet Worm Program*, *supra* note 73 (describing in computer science terms the technical details of the worm's operation).

[117] *Morris*, 928 F.2d at 506. *But see* Standler, *supra* note 16 (arguing that claims by computer scientists that they did not realize how quickly a virus might spread is a spurious argument because the mathematics known to scientists is sufficient to recognize this result).

[118] *See* Spafford, *supra* note 73 (stating there was no code within the worm which would explicitly cause damage).

[119] *But see* Standler, *supra* note 16 (stating that other comments located in the source code indicated Morris's worm behaved as he intended).

> loss to one or more others of a value aggregating $1,000 or more during any one year period . . .[120]

This version of the statute attaches no mens rea requirement to the qualifying elements.[121] It was argued that the intentional mental state modifying the access requirement should be read as applying to the damage element as well, but the court did not accept the argument.[122] Morris may have argued that this made the statute unconstitutional, but a decision in the Ninth Circuit demonstrates that the court would probably not have found that argument persuasive.[123] Even though Morris lacked the mens rea to cause damage under the current version of § 1030(a)(5)(A)(i), he likely would have been liable under (5)(A)(ii) for damage caused recklessly. He would certainly be liable under both (5)(A)(iii) for any damage caused through intentional unauthorized access[124] and (5)(B)(v) for damage affecting a computer used by a government entity for national defense or national security.[125] The difference would have been the applicable level of punishment. 18 U.S.C. § 1030(c)(2)(A) defines a violation of 18 U.S.C. § 1030(a)(5)(A)(iii) as a misdemeanor requiring less than one year of imprisonment for the particular acts committed by Morris. Under 18 U.S.C. § 1030(c)(4)(B) the violation of § 1030(a)(5)(A)(ii) would be a felony subjecting Morris to the possibility of imprisonment up to five years.

---

120 *Morris*, 928 F.2d at 506 (citing 18 U.S.C. § 1030(a)(5)(A)).

121 *See id.* at 509 (stating the court's rational for not applying a mens rea requirement to the damages phrase of the statute was the legislature's failure to specify a scienter requirement within the wording of that phrase—unlike other phrases where a scienter requirement had been specifically included).

122 *See id.*

123 Five years after *Morris*, the Ninth Circuit held that the government did not have to prove intentional damage and that the lack of a mens rea requirement for the damage element did not render the statute unconstitutional. *See* United States v. Sablan, 92 F.3d 865, 869 (9th Cir. 1996).

124 The question under this section of the statute is whether a negligent release of a virus program could constitute *intentional* unauthorized access based solely upon the design of the program code to gain unauthorized access if the actual release was unintended by the writer.

125 18 U.S.C. § 1030(a)(5)(B)(v) (2002).

3. State Computer Crime Statutes

New York, New Jersey, and Pennsylvania use vastly different approaches to the problem of dealing with computer-oriented crime.[126] None of the state statutes outlaw writing malicious computer software. The New York statutes address unauthorized access with its Computer Trespass and Unauthorized Use of a Computer Act.[127] Pennsylvania has a statute barring unlawful use of a computer, which involves unauthorized access with an intent to interrupt normal functioning.[128] New Jersey addresses access only in regards to additional conduct following the unauthorized access including altering or damaging programs, defrauding, or obtaining computer materials or personal identifying information.[129] To deal with crimes specific to computer usage, New Jersey implemented its own computer crime statutes.[130]

---

[126] These three states were chosen as a manageable sampling of the different approaches taken by State legislatures in defining computer crimes.

[127] N.Y. PENAL LAW § 156.05 (2006) ("A person is guilty of unauthorized use of a computer when he or she knowingly uses, causes to be used, or accesses a computer, computer service, or computer network without authorization."); N.Y. PENAL LAW § 156.10 (2006) ("A person is guilty of computer trespass when he or she knowingly uses, causes to be used, or accesses a computer, computer service, or computer network without authorization and: 1. he or she does so with an intent to commit or further the commission of any felony; or 2. he or she thereby knowingly gains access to computer material.").

[128] 18 PA. CONS. STAT. ANN. § 7611(a)(1) (2003) ("A person commits the offense of unlawful use of a computer if he: (1) accesses or exceeds authorization to access, alters, damages or destroys any computer, computer system, computer network, computer software, computer program, computer database, World Wide Web site or telecommunication device or any part thereof with the intent to interrupt the normal functioning of a person or to devise or execute any scheme or artifice to defraud or deceive or control property or services by means of false or fraudulent pretenses, representations or promises.").

[129] N.J. STAT. ANN. § 2C:20-25 (2003) ("A person is guilty of computer criminal activity if the person purposely or knowingly and without authorization, or in excess of authorization: (a) Accesses any data, database, computer storage medium, computer program, computer software, computer equipment, computer, computer system or computer network; (b) Alters, damages or destroys any data, data base, computer, computer storage medium, computer program, computer software, computer system or computer network, or denies, disrupts or impairs computer services, including access to any part of the Internet, that are available to any other user of the computer services; (c) Accesses or attempts to access any data, data base, computer, computer storage medium, computer program, computer software, computer equipment, computer system or computer network for the purpose of executing a scheme to defraud, or to obtain services,

Each of these state statutes demonstrates a slightly different approach to addressing computer crimes involving malicious programs. New York and New Jersey laws take an approach similar to the federal statute by requiring unauthorized access before permitting law enforcement to prosecute the wrongdoer. Only the Pennsylvania statute directly addresses viruses and worms, and goes as far as making their possession illegal.[131] This is a superior approach because it allows law enforcement to intercede before the virus is released and harm is done.[132] This helps prevent innocent computer users from suffering damage and losses, but it still permits the harmful software to be developed.

An important distinction to make when analyzing what can be damaged is the difference between the definition of property in state and federal statutes. New York, New Jersey, and Pennsylvania explicitly define property as anything of value whether tangible or intangible. Pennsylvania specifically identifies computer programs and software as property regardless of its form.[133] New Jersey's inclusion of intangible computer materials as property allows these computer materials to be protected under statutes originally designed for physical property only.[134] This broadening of the property definition allows New Jersey to use established criminal statutes to deal with anti-social actions that are in need of deterrence. It is easier to identify the proscribed criminal behavior when applying it to a particular form of

property, personal identifying information, or money, from the owner of a computer or any third party.").

[130] N.J. STAT. ANN. § 2C:20-23-34 (2004).

[131] N.Y. PENAL LAW §§ 156.05, 156.10, 156.20, 156.30, 156.35 (2006).

[132] A difficult question that needs to be addressed involves what constitutes ownership of the program. Does the code have to be complete or functional for the suspect to be in possession of the program? If the program is not required to be complete or functional, the prohibition on possession collapses into a prohibition on the writing of the code.

[133] 18 PA. CONS. STAT. ANN. § 7601 (2003) ("'Property' [i]ncludes, but is not limited to, financial instruments, computer software and programs in either machine or human readable form, and anything of value, tangible or intangible.").

[134] N.J. STAT. ANN. § 2C:20-1(g) ("'Property' means anything of value, including real estate, tangible and intangible personal property, trade secrets, contract rights, choses in action and other interests in or claims to wealth, admission or transportation tickets, captured or domestic animals, food and drink, electric, gas, steam or other power, financial instruments, information, data, and computer software, in either human readable or computer readable form, copies or originals.").

computer usage. These definitions do not require the software or data to be stored on physical media in order to receive protection.[135]

a) New York's Approach

The New York approach treats computers as a unique form of property different from physical property. The state's cybercrime statutes are modified versions of the physical crimes of larceny, burglary, and criminal tampering, but with allowances made to capture those facets particular to a computer crime.[136] The New York statutes do not seem to directly address malware.[137] Both computer trespass and computer tampering might be interpreted broadly enough to cover a computer virus infection, but the wording of the statute does not specifically cover such an occurrence.[138] It is difficult to know if the wording could be applied broadly enough to encompass virus distribution, and if so how the court could rationalize it, because there is little case law on this issue.

b) New Jersey's Approach

The New Jersey statutes are similar to the federal CFAA and the New York statutes. They are focused on unauthorized access of a computer system for the purpose of causing damage or committing a fraud.[139]

c) Pennsylvania's Approach

In contrast, the Pennsylvania computer crime statute, which was passed in 2002, specifically identifies and outlaws the distribution or possession with intent to distribute of a computer

[135] United States v. Brown, 925 F.2d 1301, 1306–07 (10th Cir. 1991) (stating that, in construing the criminal statute strictly, intellectual property was not a good, ware, or merchandise as contemplated by the National Stolen Property Act, 28 U.S.C. § 2314).

[136] N.Y. PENAL LAW §§ 156.05, 156.10, 156.20, 156.30, 156.35 (2006).

[137] Unlike the 2002 version of the CFAA or the Pennsylvania statute, the New York statutes do not specifically mention computer programs or software as a means of perpetrating a crime. *See id;* 18 PA. CONS. STAT. ANN. § 7616(a).

[138] N.Y. PENAL LAW §§ 156.05, 156.10, 156.20, 156.30, 156.35 (2006).

[139] N.J. STAT. ANN. 2C:20–25 (2003) (Computer-related theft).

program with the capability to disrupt the normal operation of a computer or system.[140] This statute more directly addresses the issue of writing malicious software. It takes into consideration both the intent of the programmer in designing the software as well as the program's capability since the two possibilities are stated in the alternative.[141] In this regard, a program which fails to function properly but is designed with the proscribed purpose, may still result in culpability. While it does not directly outlaw the act of writing virus code, it does prohibit possession if the person has the intent to distribute it.[142] This approach is different from the one implemented by the federal statute, but it may allow better and easier enforcement.

4. Damage Requirements in Computer Crime Statutes and Problems Dealing With Intangible Property

The second element in the federal cyber crime statute is a requirement that a certain amount of damage be done to either a single computer or a number of computers in the aggregate.[143] This creates a number of problems in determining what constitutes damage. The type of damage done to computer systems almost always involves intangible property.[144] Previous federal case law has defined what constitutes damage to intangibles differently.

---

[140] 18 PA. CONS. STAT. ANN. § 7616(a) (2003) ("A person commits an offense if the person intentionally or knowingly sells, gives or otherwise distributes or possesses with the intent to sell, give or distribute computer software or a computer program that is designed or has the capability to: (1) prevent, impede, control, delay or disrupt the normal operation or use of a computer, computer program, computer software, computer system, computer network, computer database, World Wide Web site or telecommunication device; or (2) degrade, disable, damage or destroy the performance of a computer, computer program, computer software, computer system, computer network, computer database, World Wide Web site or telecommunication device or any combination thereof.").

[141] *Id.* (stating the definition as "a computer program that is designed or has the capability to . . .").

[142] An important element not addressed by the statute is whether the program must be functional or capable of causing damage when released. If this were the case, possession would not be illegal until the programming was complete and debugged.

[143] *See* 18 U.S.C. §§ 1030(a)(4), 1030(a)(5)(B)(i) (2002).

[144] In some instances physical characteristics of certain computer components such as the hard drive or flash BIOS can be changed or damaged. *See* SZOR, *supra* note 11, at 305–06.

Since the CFAA does not prevent prosecution under other laws, a conflict can arise between how the court has interpreted damages under the CFAA and under these other laws.

There has been a circuit split in the federal courts since the Supreme Court decided *United States v. Dowling*.[145] This decision concerned whether intangible materials should be treated as property under statutes such as National Stolen Property Act ("NSPA")[146] and the Economic Espionage Act ("EEA").[147] The district court for the Northern District of Illinois in *United States v. Riggs*[148] found no tangibility requirement coming out of the *Dowling* decision, while the Tenth Circuit in *United States v. Brown*[149] held that *Dowling* did distinguish between tangible and intangible property. The Second Circuit followed *Riggs* in *United States v. Farraj* by treating computer materials as property.[150] The court held that "although not tangible in a conventional sense, the stolen property was physically stored on a computer hard drive and could be viewed and printed out with the push of a button."[151]

The importance of the distinction between tangible and intangible lies in defining and measuring the damage caused in computer crimes. There is an inherent difficulty in determining what damage is done to something intangible, and how the cost should be measured. If the determination of damage for meeting

---

[145] 473 U.S. 207 (1985) (holding that the violation of copyrights did not also permit prosecution under the National Stolen Property Act since no property had been stolen).

[146] 28 U.S.C. § 2314 (1994). *See also* Todd H. Flaming, *The National Stolen Property Act and Computer Files: A New Form of Property, A New Form of Theft*, U. CHI. L. SCH. ROUNDTABLE 255, 259–61 (1993) (describing the different directions the 10th Circuit and Northern District of Illinois took in deciding whether something had to be tangible to fall under the NSPA).

[147] 18 U.S.C. § 1831 (1996); Geraldine Szott Moohr, *The Problematic Role of Criminal Law in Regulating Use of Information: The Case of the Economic Espionage Act*, 80 N.C. L. REV. 853, 893 (discussing the treatment by the Supreme Court of trade secrets as property based on natural law).

[148] 739 F. Supp. 414, 421–22 (N.D. Ill. 1990) (distinguishing *Dowling*'s holding regarding intangible property as applied to the NSPA).

[149] 925 F.2d 1301, 1307–08 (10th Cir. 1991) (discussing how *Dowling*'s holding requires a physical component to fall under the NSPA).

[150] United States v. Farraj, 142 F. Supp. 2d 484, 489 (S.D.N.Y. 2001) (applying the holding in *Riggs* and distinguishing *Brown* as a misapplication of the argument in *Dowling*).

[151] *Id*.

the minimum amount for the CFAA is strictly a technical, objective one, it rests almost exclusively on the design of the virus and the intent of its creator to alter or destroy other programs or data on the infected system. Failure to identify an actual injury to a computer program, stored files or data, or to the actual performance of the system should prevent the determination that any measurable harm was done. In the first case, there is no measurable harm because only an actual injury is considered.[152] This does not take into account the time and effort to determine that no harm was done to a computer system. In the second case, damages are a form of restitution in which the injured party is returned to the position he was in before incurring the loss.[153]

The CFAA defines "damage" as "any impairment to the integrity or availability of data, a program, a system, or information."[154] This definition leaves the term ambiguous in its application to the effects caused by the virus code.[155] As was shown in the previous sections, not all infections result in the disabling of a system or program.[156]

In the current legal environment, the federal courts could utilize the holding in *Brown* when interpreting the treatment of damage to property for the NSPA and the statutory definition of damages in the CFAA. The Federal Court for the Northern District of Illinois stated in *Riggs*:

> The problem with Neidorf's argument, however, is that he does not cite, and this court is unable to find, anything in the legislative history of the CFAA

[152] This is similar to the requirement that a plaintiff be able to identify an actual injury that was suffered in order to bring a tort action for compensatory damages before consequential damages can be sought.

[153] This approach looks at the damages from an almost contractual point of view where the plaintiff incurred costs to obtain the benefit of correcting any impairment and re-securing the availability of any program or information, but fails to obtain the benefit because it had not been previously impaired or damaged.

[154] 18 U.S.C. § 1030(e)(8) (2002).

[155] There is no indication of what would constitute "impairment" or what aspects of a system's performance are encompassed by the term "integrity."

[156] *See supra* Part I.A.1 (explaining how virus code can be hidden within a program without interfering with the functioning of the program or the computer system in which it is stored).

> which suggests that the statute was intended to be the exclusive law governing computer-related crimes, or that its enactment precludes the application of other criminal statutes to computer-related conduct.[157]

However, there is a contradiction in the application of the CFAA and the NSPA to intangible property in a computer crime if it is treated as incapable of protection under the Stolen Property statute, while any changes to the property are included as damages under the CFAA.

IV. IS A NEW APPROACH TO VIRUSES NEEDED?

A. Does Writing Malware Need to be Criminalized?

In order to have a particular action or result outlawed, there must be strong societal concerns, which outweigh the basic interests in personal freedom.[158] The writing and propagation of malicious software (malware) is anti-social behavior whose harm vastly outweighs any benefits. There are particular actions and mental states that demonstrate the writing and release of computer virus code is anti-social. These particular actions and mental states should be part of the criminal statutes that are used to prosecute this behavior.[159]

The current cybercrime laws approach the threat of malicious software by prohibiting unauthorized access of protected computers and the resulting damage.[160] These laws, however, permit the virus writers to develop and refine their malicious code

---

[157] United States v. Riggs, 739 F. Supp. 414, 423 (N.D. Ill. 1990).

[158] "Liberty has never come from government. Liberty has always come from the subjects of it. The history of liberty is a history of resistance. The history of liberty is a history of limitations of governmental power, not the increase of it." Woodrow T. Wilson Quotes, Proverbia.net, http://en.proverbia.net/citasautor.asp?autor=17780 (last visited Jan. 28, 2008).

[159] *See generally* KADISH & SCHULHOFER, *supra* note 100, at 173–312 (discussing the necessary elements of a criminal statute including actus reus and mens rea).

[160] This approach allows the laws to treat hacking and malicious software in similar manners. However, is allows the threat posed by malicious software to develop to an unacceptable level before permitting law enforcement to deal with the problem.

free of any consequences.[161] Once the working code is distributed, third parties may use the functioning code as they wish. Thus, individuals are prosecuted only if the code is released, infects a protected computer system,[162] causes damage to that computer system, and the infection and damage is reported to the authorities.[163] These requirements make it important that the authorities recognize a virus outbreak immediately and begin to acquire evidence of the crime as soon as possible. In the best-case scenario, authorities can trace back the route of the virus to find the initial source of the code.[164] Ideally the authorities might be able to trace the virus back to a suspect's own computer system and find evidence of the original code on the suspect's computer. To accomplish this, the authorities must obtain warrants in each of the jurisdictions where the virus code was relayed during its spread. A delay or failure in obtaining these warrants can easily prevent the authorities from following the chain all the way back to the source of its initial dissemination due to the loss or destruction of information. Indeed, the initial point of the virus's release may not even be directly connected to the virus's author.[165]

The use of damage as an additional qualification for prosecution raises the question of what constitutes damage.[166] As the previous sections have suggested,[167] not all virus and worm infections result in observable damage to the user's computer system. Additionally, if damage is caused, it often tends to be circumstantial to the propagation of the virus and not designed into

[161] *See infra* Part IV.D.2 (discussing why it is necessary for companies to create viruses).

[162] Under § 1030 almost every computer system is protected because they are connected to the Internet and involved in interstate commerce. 18 U.S.C. § 1030(e)(2)(B) (2002).

[163] Unfortunately, it is too late for the person or business whose computer has become infected and suffered damage. They must now deal with the problem and resulting losses.

[164] This is similar to the methods used in identifying and tracing the spread of a contagious disease.

[165] *See supra* Part I.B (explaining ways a virus author can release his code into the wild).

[166] *See supra* Part III.B.4. "Damage" is defined as, "[i]mpairment of the usefulness or value of person or property." WEBSTER'S II NEW RIVERSIDE UNIVERSITY DICTIONARY 345 (1996).

[167] *See supra* Part I.A.1. (commenting on the effect that the infection has on other software and computer systems).

the actual virus code.[168] When the intent of the virus writer is not evident directly from the code, and the harm that results from its release is circumstantial, proof of the requisite mens rea can be very difficult.

Finally, different nations have different perspectives regarding what type of activity should be allowed or outlawed.[169] There are also different factions within each country that might oppose particular criminal statutes, because of the adverse effect it could have on their particular interests or on the interests of their constituents. [170] This makes implementing adequate international laws difficult if not impossible to achieve.[171] However, by narrowly tailoring criminal statutes, it is more likely that different nations will find a common ground.[172] This is necessary to improve the gathering of evidence, apprehension, prosecution, and extradition of computer criminals.[173] The extra-territorial nature of computer crimes requires the cooperation of judiciaries and police forces across many jurisdictions.[174] A hole anywhere along the line can allow a perpetrator to go free.[175]

---

[168] *See supra* notes 80–87 and accompanying text (noting that viruses and worms can be designed to do different types of damage).

[169] *See* Goodman & Brenner, *supra* note 5, at 170.

[170] *See* Global Internet Liberty Campaign Member Letter on Council of Europe Convention on Cyber-Crime, Version 24.2 (2004), *available at* http://www.gilc.org/privacy/coe-letter-1200.html (listing grievances with proposed EU legislation and listing organizations opposed to its adoption, because of issues with criminalization and liability imposed by the new law).

[171] *See* Goodman & Brenner, *supra* note 5, at 170 (stating that the member nations of the Organization for Economic Co-operation and Development (OECD) were unable to implement uniform laws to deal with computer crimes).

[172] *Id.* at 141 (describing some of the issues which arise in defining cybercrimes).

[173] *Id.* at 142 (identifying difficulties cybercrimes pose for traditional law enforcement). *See also* Mark Richard, Prepared Statement of Mark M. Richard Counselor for Justice Affairs U.S. Mission to the European Union (2005), *available at* http://www.usdoj.gov/criminal/cybercrime/mmrArt29DRstmt041405.pdf (discussing US provisions for data retentions and the need for comparable laws throughout jurisdictions to enable effective enforcement).

[174] *See* Goodman & Brenner, *supra* note 5, at 223.

[175] *See id.* at 141 (explaining how the lack of criminal statutes directed at computer viruses in the Philippines allowed the author responsible for the "Lovebug" virus to avoid both prosecution and extradition).

B. How a New Statute Could Address the Problem

As a possible solution to these difficulties, one alternative is to make it criminal for the average computer user to write and possess virus code.[176] This would provide law enforcement personnel with the tools necessary to prosecute and convict individuals who engage in behavior, which has been identified as undesirable,[177] while not casting a net so wide that innocuous or beneficial behaviors are encompassed.[178] Properly written criminal statutes should help focus the attention and resources of the authorities on actions and behavior, which are a true threat, while avoiding wasted effort on less problematic behavior.[179] This approach shifts the efforts of law enforcement from tracking down culprits after a virus outbreak, to identifying programmers who are writing or have written virus code and placed it in the hands of other computer users. A similar approach is used to track down

[176] In this Note, the "average computer user" is anyone not directly engaged in computer security, research, or cyber-warfare.

[177] *See* Marc D. Goodman, *Why the Police Don't Care about Computer Crime*, 10 HARV. J.L. & TECH. 465, 476 (1997); *see also* Alistair Kelman, *The Regulation of Virus Research and the Prosecution for Unlawful Research?*, J. INFO. L. & POL'Y (1997), *available at* http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/1997_3/kelman1 (stating that "virus writing is evil and cannot be justified in any circumstances").

[178] "Should we not be a Socrates, who . . . sought Truth and Wisdom . . . the question that really matters is not how computers can make us wealthy or give us power over others, but how they might make us wise." Meinel, *supra* note 34 (quoting MARK A. LUDWIG, THE GIANT BLACK BOOK OF COMPUTER VIRUSES (Am. Eagle Publ'ns 1995)). The issue of writing new viruses by anti-virus software companies to anticipate future code released into the wild is also considered. *But Cf.*, Public Letter Concerning the Writing of Viruses & How It Does Not Teach About Virus Prevention (2003), http://www.avien.org/publicletter.htm (listing the anti-virus computer professionals who believe colleges and technical schools should not have virus-writing classes as part of their computer science or computer security curriculum).

[179] The current number of malicious code releases, including all forms of malware, is estimated at 6,368 per month. Yury Mashevsky, Malware Evolution: 2005 (2006), http://www.viruslist.com/en/analysis?pubid=178949694. The number of computer viruses released on average day in 1999 was approximately 10 to 15 viruses. Vesselin Bontchev, *Future Trends in Virus Writing* (1994), http://www.people.frisk-software.com/~bontchev/papers/trends.htm [hereinafter *Future Trends in Virus Writing*]. It would be impossible for law enforcement agents to track and prosecute every release of virus code considering most do not properly function, and the ones that do probably do not generate measurable damage.

individuals involved in child pornography.[180] It would permit closing down websites dedicated to disseminating virus code, or at least removing the working virus code content from the site.[181] This would prevent novice virus spreaders[182] from obtaining working code, which should drastically reduce the volume of viruses encountered.[183] Virus writing kits[184] would also fall under this prohibition because there is no legitimate purpose for the existence of such tools. Once the volume of viruses is reduced,[185] it becomes easier to identify and focus on the individuals who do possess the skills necessary to produce working virus code.[186] If

[180] 18 U.S.C. § 2252(a)(4)(B) (2006) ("Any person who knowingly possesses 1 or more books, magazines, periodicals, films, video tapes, or other matter which contain any visual depiction that has been mailed, or has been shipped or transported in interstate or foreign commerce, or which was produced using materials which have been mailed or so shipped or transported, by any means including by computer, if (i) the producing of such visual depiction involves the use of a minor engaging in sexually explicit conduct; and (ii) such visual depiction is of such conduct; shall be punished . . . .").

[181] This is specifically directed at virus code, which could be released or executed without requiring any further actions by a third party. Text including virus code is not be included due to First Amendment free speech issues. It is expected that this would reduce the number of virus outbreaks, because it raises the necessary level of computer sophistication and software ownership above the average computer user. Fred Cohen has shown that there is no clear distinction between text and code because of the ability to convert one into the other through compilers and interpreters. *See* Fred Cohen, *Prevention of Computer Viruses* (1984), http://all.net/books/virus/part3.html. However, not everyone has the necessary software loaded on his or her systems to accomplish this task. *Id.*

[182] One term used to describe the majority of individuals who release viruses is "kode kiddies" because they lack the computer skills to actually do their own programming. *See* Meinel, *supra* note 34. They participate in the destructive behavior by obtaining functional viruses from websites or bulletin boards, or through the use of virus writing kits. *See* Bontchev, *supra* note 179.

[183] A large portion of viruses encountered by A-V groups are the result of this method of virus creation. *See Future Trends in Virus Writing*, *supra* note 179 (stating a large number of viruses are generated through virus kits).

[184] Virus writing kits are programs written by proficient virus programmers that allow novices to construct new viruses by choosing to combine separate virus components or modules. These types of viruses usually do not operate properly, but sometimes the novice gets lucky.

[185] *See* MAXIMUM SECURITY: A HACKER'S GUIDE TO PROTECTING YOUR INTERNET SITE AND NETWORK 328 (4th ed. 2002) [hereinafter MAXIMUM SECURITY] ("[K]it viruses have tended to contribute to the "glut" problem (the sheer weight in numbers), rather than to the "in-the-wild" problem . . . .").

[186] *Id.* ("Some virus writers and their admirers still regard proficiency in assembly language as the hallmark of programming excellence."); c*f. id.* ("[A]ssembly language is

the only notable outcome of allowing the writing of the code is to have it released and cause damage, there is no reason to allow it written in the first place. By moving the prohibited action back from possession of the code to its writing, law enforcement is given a larger window of opportunity to intercede before any harm is done.

Outlawing the writing and possession of working virus code also avoids the issues involved with determining damage. Since prosecution can occur before any computer systems are infected, there is no need to identify what effects constitute damage and to determine how to measure it.

The gathering of evidence also becomes easier if the focus of prosecution shifts to writing and possession, because it localizes the search for evidence down to the computer system of the suspect and any of his accomplices. There is no longer a need to trace a virus outbreak back to a source. This eliminates some of the difficulty in cross-jurisdictional evidence gathering after the virus release. Search warrants become directed at particular locals and individuals, rather than the jurisdiction of each intervening transmission or relay site involved in the virus's spread. This would relieve the need to immediately identify a new virus outbreak in order to preserve the evidence trail.

The difficulty of tracing a virus outbreak back to its source would be eliminated but the difficulty of tracing the source of a posted virus back to the individual who posted it would remain. The virus writer can use similar methods in each case to maintain his anonymity. Multiple relays through numerous disparate jurisdictions can be used to hide the culprit's trail. While this may make identification of the source of the original code much more difficult, it still retains some key advantages over the current approach of tracing an outbreak. Law enforcement could be authorized to investigate the site containing posted virus code, confiscate the computer file containing this virus code, and perhaps quarantine or shut down the site, since possession of the code

not necessarily the language of choice among the current generation of virus writers. Interpreted macro languages (especially Visual Basic for Applications) are generally harder to use than kits, but much easier than assembler.").

would be illegal.[187] This avoids the shortcomings of the results oriented approach, which requires unauthorized access and harm before initiating an investigation by being preemptive of the virus's release.

C. Aspects of the Release of Virus Code Addressed by the Computer Crime Statutes

One of the major issues in writing criminal statutes to prosecute the release of malicious computer code is defining what specific act is criminal and therefore prohibited.[188] If the actual writing of virus code were prohibited, there would be little question of intent because the writing of virus code is not something accomplished accidentally.[189] The mens rea for possession of the virus code could be purposely or knowingly.[190] One concern is that there cannot be strict liability for the act of releasing malware "into the wild."[191] In other words, there cannot be prosecution without intent, however a statute addressing the writing or possession of a computer virus or worm could require only that the person know he has written or possesses a virus or worm as something inherently dangerous.[192] Once again the

---

[187] *See generally* Games v. U.S. Secret Service, 36 F.3d 457 (5th Cir. 1994) (discussing the seizure of computer hardware, software and documents which constitute evidence of federal crimes).

[188] MARCUS D. DUBBER, CRIMINAL LAW: MODEL PENAL CODE 43–48 (Foundation Press 2002) (outlining the required elements of a crime and how they should be used).

[189] *See supra* Part I.A.1, 2 (discussing the unique aspects of virus code which must be purposely included for the program to function as a virus or worm).

[190] It may be argued that this would make all persons whose systems become infected liable, but that would ignore the requirement that need of possession or release being a conscious act by the individual. Since virus infection would not be the result of the owner's willful act, no liability would attach under the law. *See* State v. Baker, 571 P.2d 65 (Kan. App. 2d 1977) (discussing the need for voluntariness to find an actor guilty of a strict liability offence)(citations omitted). In addition, almost every computer user makes every effort to remove malicious code from their system as soon as they are aware of it. This would represent a good faith effort to prevent the further proliferation of the code and demonstrate that any additional infection was involuntary.

[191] *See* Staples v. United States, 511 U.S. 600, 607 (1994) (arguing that the term "strict liability" is really a misnomer, and only the requirement of a guilty mind is eliminated; the defendant must still be aware that he is dealing with something dangerous to the public).

[192] Pennsylvania code already addresses the possession of computer virus code. *See* Part III.B.2.c. The Pennsylvania statute applies an intentional or knowing mens rea to the act

technical sophistication of virus and worm code could easily be used to prove that the person knowingly wrote prohibited code.[193] The scienter requirement that a person knowingly possesses a virus or worm would protect those individuals that become infected with such a program and unwittingly disseminate it to others. Proving this level of mens rea is a minor hurdle for law enforcement to overcome in prosecuting the person who wrote the code or intended to release it because the code would be present on the perpetrator's computer system in a single inactive form rather than as multiple infected files.[194]

Since the person responsible for writing the virus code may not be the person who releases it, statutes should also address the release of such malicious code. In determining what level of mens rea should be associated with each element of a law criminalizing the release of virus programs, one must consider whether releasing the virus must be a purposeful or reckless act in order to rise to the level of a criminal activity.[195] A second question is whether keeping virus code on a system should be considered a negligent act because of the possible harm it may cause[196] or a criminal act

of possession with intent to distribute. *See supra* note 140. A new statute prohibiting the actual writing of virus code could require that the programmer only know that his actions involve an activity that is dangerous to the public. *See Staples*, 511 U.S. at 607 (stating that "as long as the defendant knows that he is dealing with a dangerous device of a character that places him 'in responsible relation to a public danger,' he should be alerted to the probability of strict regulation, and we have assumed that Congress intended to place the burden on the defendant to ascertain at his peril whether [his conduct] comes within the inhibition of the statute.") (citations omitted).

[193] The two unique features that define virus code, namely its ability to replicate itself and its propensity to locate and infect additional computer systems, make it easy to recognize as a dangerous form of computer code. *See supra* Part I.B.1.

[194] This form could be either an executable file containing only the virus code, or a carrier program with the virus code embedded in it. In either case, the virus or worm code would have to be executed to begin spreading. A virus writer would also likely have various versions of source code files of the virus.

[195] *See* DUBBER, *supra* note 188, § 2.02(1) (outlining the minimum requirements of culpability).

[196] Tort law provides for strict liability involving ultra hazardous or abnormally dangerous activities. *See* RESTATEMENT (SECOND) OF TORTS § 519 (1977) (stating that strict liability is applicable to abnormally dangerous activities); *id.* § 520 (examining each of the six factors to be taken into account when determining whether an activity is inherently dangerous). *See also* Sullivan v. Dunham, 161 N.Y. 290 (N.E.2d 1900). This has covered the keeping of wild animals. It is possible viruses could be classified as

because there is no legitimate or beneficial purpose for possessing such code.[197] These questions are important because, negligent behavior is not usually prosecuted as a criminal offense. These actions are not prosecuted as serious crimes because the mental state of the perpetrator has not reached the required level of culpability.[198]

Eliminating the majority of virus code by prohibiting its writing and possession is one way to avoid the issues of defining and determining damage. By attacking the problem before damage can be done, it makes the discussion an academic exercise rather than a practical problem facing investigators and prosecutors. The courts have given the definition of damages as applied in the CFAA a broad interpretation, but that appears to only treat the symptom and not the problem.[199] This broad interpretation just highlights the problem that the legislature and courts have in understanding what effect viruses and worms have on computer systems and the software saved on them. A deeper understanding

inherently dangerous considering their propensity to escape captivity and spread, thereby doing harm—drawing an analogy between the keeping of computer viruses, biological viruses and wild animals.

[197] Possession of certain substances by individuals has been outlawed because of the possible harm they can cause. Having those substances in a person's possession constitutes a criminal offense. *See* CAL. HEALTH & SAFETY CODE § 12305 (2007) ("Every person not in the lawful possession of an explosive who knowingly has any explosive in his possession is guilty of a felony."); *id.* § 12303 ("'Lawful possession of an explosive,' as used in this chapter, means possessing explosives in accordance with the stated purpose and conditions of a valid permit obtained pursuant to the provisions of this part, unless such person is specifically excepted from the permit requirements by the provisions of this part."); N.Y. PENAL LAW § 265.02 (McKinney 2006) ("A person is guilty of criminal possession of a weapon in the third degree when: (2) Such person possesses any explosive or incendiary bomb, bombshell, firearm silencer, machine-gun or any other firearm or weapon simulating a machine-gun and which is adaptable for such use; Criminal possession of a weapon in the third degree is a class D felony.").

[198] Although there are crimes based on negligence such as negligent homicide and child neglect, which are typically treated as felonies, most negligent crimes such as negligent driving are only misdemeanors. *See generally* LAFAVE, *supra* note 99, at 261–71.

[199] *See* United States v. Middleton 231 F.3d 1207, 1213–14 (9th Cir. 2000) (discussing the interpretation of loss and damage in determining that the requirement of $5,000 for prosecution under the CFAA had been met because the definition is sufficiently broad to include items such as the cost of resources to re-secure a computer system as well as any other natural and foreseeable expense to restore items which were damaged).

of the mechanics of these programs would help eliminate the vagueness of this term.

Legislators and law enforcement personnel must work in concert to eliminate cybercrime. This cooperation can help build consensus with other countries in establishing mutual laws for dealing with these cross-border crimes.[200]

D. The Pros and Cons of This Approach

1. Innocent Software

There are legitimate operations occurring on computers, which might fall under the term "compromise of integrity." These operations include automatic updates and patches activated by programs on the user's system.[201] These programs typically function without alerting the user, or only mentioning when an update has been completed successfully.[202] It may be argued that the code functions without the user's knowledge or this loss of control is unwanted.[203] Opponents of the prohibition could argue that this sort of computer activity could result in criminal charges or civil liability against the software manufacturer. An easy counter to this argument is that software users knowingly installed the program on their systems and made the required change of settings so the program would behave in this manner.[204] Alternatively, it could be implied that users want the updates, such as for anti-virus programs and operating systems, even if not expressly notified. In such instances, the access could not be termed unauthorized if the system owner installed the program knowingly, expecting and desiring these updates. Cookies and

---

[200] *See* Goodman & Brenner, *supra* note 5, at 141.

[201] Microsoft has just such an update manager, update.exe. *See* The User Rights that are Required by Update.exe, http://support.microsoft.com/kb/888791 (last visited Dec. 20, 2007).

[202] *See, e.g.*, Windows Server TechCenter, How does Automatic Updates work? (Jan. 21, 2005), http://technet2.microsoft.com/windowsserver/en/library/6d06ca72-d065-45fe-870b-3b5faf60c21d1033.mspx.

[203] *See* Robert Moir, Defining Malware: FAQ (Oct. 1, 2003), http://www.microsoft.com/techne t/security/alerts/info/malware.mspx.

[204] The particular settings that activate automated functions may be set to "on" as a default without ever prompting users, or notifying them of the setting.

commercial spyware might be considered a system compromise, because the user does not know exactly what information is being collected by these programs, when it is occurring, how the computer sends the information out to a receiving system, or to whom the information is sent.[205] A means of circumventing criminal liability for such spyware is by requiring notification to a computer owner that a cookie would be stored on the computer, stating what kind of information would be collected by the spyware, and requiring authorization to place the cookie or spyware on the system.

2. Legitimate Reasons Not To Prosecute All Makers of Malware

There are legitimate reasons for writing computer viruses even if there are questionable ethical issues involved in doing so.[206] Creators of anti-virus software may need to test their product against a variety of malicious code to see how it performs its task.[207] These software developers can use captured virus code to do this testing,[208] or they can write their own code having the particular characteristics for which they wish to test.[209] Under such circumstances is virus writing criminal behavior? Virus writing can be socially beneficial by creating anti-virus software capable of preventing a particular strand of virus from attacking.[210]

---

[205] *See* SZOR, *supra* note 11, at 38 (discussing spyware).

[206] *See id.* at xxiv, 293 (commenting on the ethical issues involved in the use of virus generating kits even by professional A-V researchers).

[207] One method of testing and certifying anti-virus software involves checking to see if it detects 100% of the viruses on the "InTheWild" watch list. *See* Doctor Web, Updating the Anti-Virus and Virus Databases, http://support.drweb.com/faq/a2 (last visited Nov. 18, 2007).

[208] Often viruses tested for are provided directly to anti-virus software companies by the virus writers themselves, or through virus collection/exchange bulletin boards. *See* Vasselin Bontchev, *Veni Vidi, Vicis*, VIRUS BULL., 10–11 (Oct. 1997), *available at* http://www.people.frisk-software.com/~bontchev/papers/vicis.html.

[209] *See* Meinel, *supra* note 34 (implying a virus researcher tests anti-virus programs using code he has written himself).

[210] Some A-V professionals may feel there is a self perpetuating cycle where virus writers attempt to create a new virus that current software cannot detect, which leads A-V programmers to create new software to detect the new viruses without the virus having been released into the wild. This could be considered a mixed blessing, since the general

This attempt at virus pre-emption could be considered criminal behavior if the act of writing a computer virus is criminalized.[211] Some scholars posit that viruses are sufficiently lifelike so that their study could reveal details about the basic foundations of life itself.[212]

Additionally, anti-virus software companies attempt to keep both their captured and created viruses isolated from connected systems when conducting their tests, but like biological viruses, their nature is to spread. Should an unwanted and unexpected release from a development and test system be treated as a criminal act because the virus accessed protected systems and caused damage? Once the virus is free, its natural course is to infect systems and propagate itself.

3. Free Speech Issues

A major consideration that has prevented the prohibition of virus writing is whether the First Amendment of the Constitution protects computer programs as a writing or expression.[213] Arguments have been made for both sides,[214] but the U.S. Supreme Court has not yet directly addressed protection for computer virus code.[215] The courts have, however, addressed the extent of First

population is not initially exposed to the new code before A-V professionals have an opportunity to analyze and combat it.

[211] Some researchers are of the opinion that there are no acceptable uses for viruses, and their creation alone should be outlawed. *See* Alistair Kelman, *The Regulation of Virus Research and the Prosecution for Unlawful Research?*, JOURNAL OF INFO., LAW, AND TECH. (JILT),
Oct. 31, 1997, *available at* http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/1997_3/kelman1.

[212] It has been proposed that viruses are a new life form and could reveal details about biological evolution in a semi-controlled environment. *See* DR. MARC A. LUDWIG, COMPUTER VIRUSES, ARTIFICIAL LIFE AND EVOLUTION (Am. Eagle Publ'ns 1993) [hereinafter COMPUTER VIRUSES, ARTIFICIAL LIFE AND EVOLUTION].

[213] Sarah Gordon, *Virus Writers: The End of the Innocence*, IBM Research Paper (2000), *available at* http://www.research.ibm.com/antivirus/SciPapers/VB2000SG.htm (citing Tippett inviting congress to outlaw virus writing).

[214] *Id. But cf.* COMPUTER VIRUSES, ARTIFICIAL LIFE AND EVOLUTION, *supra* note 212 (arguing for the value of virus code as a research tool and for philosophical reasons).

[215] *See* Robert Plotkin, *Fighting Keywords: Translating the First Amendment to Protect Software Speech*, 2003 U. ILL. J.L. TECH. & POL'Y 329, 330–31 (2003); Eugene Volokh, *Crime-Facilitating Speech*, 57 STAN. L. REV. 1095, 1103 (2005).

Amendment protections for other types of computer programs.[216] The arguments made by the court in these other cases can be applied to virus code by comparing the technical aspects of this code to the programs the courts have examined when determining whether computer programs are protected forms of speech.[217]

The First Amendment protects the free exchange of ideas.[218] The First Amendment does not protect all speech, but only that which convey ideas, information or messages.[219] Source code is used to communicate complex computer programming concepts between professionals and to students and hobbyists. While almost no one examines binary or hexadecimal code for its expressive content,[220] it can be used to communicate information between programmers.[221]

---

[216] Universal City Studios, Inc. v. Corley, 273 F.3d 429, 445–46 (2d Cir. 2001) (stating that both source code and object code is protected speech due to their ability to convey information even if comprehensible to only a limited audience, just as a novel written in Sanskrit would be protected).

[217] *See*, *e.g.*, *id.* at 449 (discussing the First Amendment protections applicable to programs used for decrypting digital video discs and circumventing copyright protections).

[218] *See* Harte-Hanks Commc'ns, Inc. v. Connaughton, 491 U.S. 657, 686 (1989).

[219] *See* Texas v. Johnson, 491 U.S. 397, 404 (1989) (stating that conduct is only protected under the First and Fourteenth Amendments if it was intended to convey a particular message, and that the message would likely be understood by those that viewed it (citing Spence v. Washington, 418 U.S. 405, 410–11 (1974)); United States v. O'Brien, 391 U.S. 367, 376 (1968) ("This Court has held that when 'speech' and 'nonspeech' elements are combined in the same course of conduct, a sufficiently important governmental interest in regulating the nonspeech element can justify incidental limitations on First Amendment freedoms.").

[220] In fact, very few programmers even bother to learn assembly language or machine code, because the reasons for their preferred use, such as very limited memory resources and inefficient or ineffective compiler programs, are no longer problems faced today. *See* MAXIMUM SECURITY, *supra* note 185, at 328 ("[D]isk space and main memory are no longer expensive, and grossly bloated files are less conspicuous in a Windows environment. Thus, it's become more practical (as well as easier) to write . . . in C++ or Delphi.").

[221] *See Corley*, 273 F.3d at 448 n.19 (identifying information as the protected form of speech most often communicated by computer code); Plotkin, *supra* note 215; Volokh, *supra* note 215, at 1152 (stating that the California Supreme Court acknowledged computer source code is an expressive means used to exchange information between computer professionals that understand how it works, but concluded that in the particular case involving DVD encryption it was not used to comment on a public issue or engage in a public debate, and was only of interest to a select group of enthusiasts).

The courts have decided that computer code is speech deserving First Amendment protection because of its capacity to communicate ideas and information.[222] There may also be messages encoded into viruses or worms that become comprehensible to human beings through the use of a disassembler or debugger program.[223]

The courts, however, have distinguished the expressive or communicative aspects of computer code from its functional aspects.[224] To appreciate this issue, one must understand the general principles involved in getting a computer to perform a given task. Computer software code falls into several different categories. The broadest division is between high level programming code (used by applications programmers) and low-level machine code (executed directly by the hardware).[225] There is also intermediate level code.[226] The higher the level, the more inherently intelligible the code is to humans. The lower the programming level, the more adaptive the code is to machine interpretation.[227] Traditionally, virus coding is done in low or intermediate level programming languages.[228] Computer code may

---

[222] *Corley*, 273 F.3d at 449.

[223] *See* SZOR, *supra* note 11, at 24–25 (describing the author's first encounter with a virus through the use of the DEBUG tool). This Note does not address the capacity of the program to display certain messages on the computer screen, or what could be embedded in the computer code, but only what could be comprehended through reading the actual machine code.

[224] *Corley*, 273 F.3d at 450–51 (noting that computer code has both a communicative and a functional aspect, so that both the speech and non-speech elements must be considered in determining the scope of First Amendment protection allotted).

[225] *See* TANENBAUM, *supra* note 31, at 3–7 (defining the programming levels and virtual machine levels of a computer system).

[226] Assembly language can be considered intermediate level programming because it has features of both high and low level code. *Id. See also* TOM SWAN, MASTERING TURBO ASSEMBLER 4 (2d ed. 1995) ("Assembly language programs are also translated to machine code by a program called an *assembler*. Despite this similarity with other languages, assembly language is neither high nor low level; it's sort of stuck in between.").

[227] *See* MANO, *supra* note 32, at 174–75 (describing the different programming categories and how suitable they are for execution by a computer).

[228] *See* MAXIMUM SECURITY, *supra* note 185, at 327 ("Older viruses were often written in assembly language. In fact, it's difficult to write some types of virus in a high-level language, even with the help of an inline assembler. This is an advantage, from the viewpoint of virus victims, in that it takes a certain level of programming expertise to

also be divided into two other categories.[229] The code written by programmers is called source code.[230] The code used to make a computer system perform an operation is called object code or machine code.[231] Source code is converted into object code through the use of a compiler or interpreter.[232] A computer cannot run a source code file directly.[233] A determination whether all of these different incarnations of a computer program are protected as free speech must be made.[234]

The differentiation between the high-level computer languages and machine code makes the application of the First Amendment both more and less difficult. The languages' distinct differences, however, make it much easier to recognize where First Amendment protections should be applied, and how to tailor the laws narrowly to take advantage of those differences. Source code

create even a weak virus (or even to modify an existing virus so as to create a variant)."). These would include Assembly language and Machine language code. *See* SWAN, *supra* note 226, at 4. Now most virus code is written through the use of higher-level language compilers; the exception might be for virus writers who pride themselves on coding in low-level languages. *See* MAXIMUM SECURITY, *supra* note 185, at 328. *But see supra* note 182 and accompanying text.

229 *See* TANENBAUM, *supra* note 31, at 397.

230 *Id.*

231 SWAN, *supra* note 226, at 4 ("Even though it may appear that a computer 'understands' high-level languages such as BASIC, Pascal, or C, all computer programs actually run in *machine language*, the coded bytes that drive the computer's central processing unit (CPU). For this reason, *machine code* is a better term for this lowest of low-level computer languages—the only language the CPU knows.").

232 *See* TANENBAUM, *supra* note 31, at 2. A compiler translates the high-level source code into low-level machine code in a single operation, thereby generating a new file consisting of the low level code. *Id.* An interpreter converts the source code into machine code one instruction at a time. *Id.* As each line is translated, the machine performs the specific instruction. *See id.* (describing the two methods of translating source code into machine instructions).

233 SWAN, *supra* note 226, at 4 ("Because CPUs can't directly execute C and Pascal statements, programs in these and other high level languages must be *compiled* (translated) to machine code before the programs can be used. Similarly, a program written in an interpreted language such as BASIC or LISP must be translated to machine code, although in these cases, the translation happens invisibly while the program runs, usually one statement at a time.").

234 At this point in time, the law has not clearly differentiated between these different types of computer programs. *See, e.g.*, Universal City Studios, Inc. v. Corley, 273 F.3d 429, 445–46 (2d Cir. 2001) (stating that both source code and object code is protected speech due to their ability to convey information even if comprehensible to only a limited audience, just as a novel written in Sanskrit would be protected).

conveys information and ideas between computer programmers and may be the best medium for this communication.[235] Source code is also non-functioning.[236] However, unlike the majority of crime-facilitating speech, which raises First Amendment issues,[237] a compiled virus program in the form of machine code is functioning,[238] and is not comprehensible to the vast majority of human beings.[239] The fact that machine code is functioning means it has both speech and non-speech elements.[240] The presence of the non-speech element would allow the creation of content neutral restrictions, since it could be directed at the function of the program rather than its content or expression.[241] This restriction would have to serve a substantial government interest, such as the protection of persons online or the safe utilization of the Internet.[242] Likewise, the restriction cannot burden substantially more speech than is required.[243]

The machine code's only true purpose is to cause the computer to behave in a particular manner desired by the programmer. While it is possible to communicate programming ideas in the

---

[235] *See* Junger v. Daley, 209 F.3d 481, 485 (6th Cir. 2000) (holding that source code is protected by the First Amendment because it is an expressive means for the exchange of information and ideas about computer programming).

[236] *See supra* note 233 and accompanying text.

[237] *See* Eugene Volokh, *Crime-Facilitating Speech*, 57 STAN. L. REV. 1095, 1097–1103 (2005) (listing a wide range of communications which have some potential for facilitating crime).

[238] *See supra* note 232.

[239] Almost no one, including professional computer programmers, would be able to read a series of ones and zeroes representing the opcodes for a given machine or the data being stored or operated upon as it would be displayed through a core dump of a range of RAM addresses. *See* Universal City Studios, Inc. v. Reimerdes, 111 F. Supp. 2d 294, 326 (S.D.N.Y. 2000).

[240] *See* Universal City Studios, Inc. v. Corley, 273 F.3d 429, 451 (2d Cir. 2001) ("the realities of what code is and what its normal functions are require a First Amendment analysis that treats code as combining nonspeech and speech elements, i.e., functional and expressive elements."); Reimerdes, 111 F. Supp. 2d at 328–29 (stating that computer code does more than express a programmer's concepts, it causes a computer to perform a task, and therefore "has a distinctly functional, non-speech aspect").

[241] *See* Hill v. Colorado, 530 U.S. 703, 720 (2000) (holding that a regulation is content neutral if it does not make reference to the content of the speech); R.A.V. v. City of St. Paul, 505 U.S. 377, 385 (1992) (stating that nonverbal expressive activity can be prohibited because of the action it entails, but not because of the idea it seeks to express).

[242] *See Corley*, 273 F.3d at 454.

[243] *See* Turner Broad. Sys., Inc. v. FCC, 512 U.S. 622, 662 (1994).

virus machine code,[244] it is not a reasonable mode of expression. It is a functioning device like a bomb.[245] Legislation could address the functional aspect by prohibiting people from possessing or posting computer programs capable of secreting themselves within another program or once in residence on a computer capable of locating other computers and copying itself onto such computer without the owner's knowledge or permission.[246]

A third aspect of viruses, unlike other crime-facilitating speech, is that a virus code does not have both a beneficial and harmful use.[247] The specific characteristics of a virus cause it to have only a harmful use.[248] These three aspects should be enough to exempt the actual working virus program from constitutional First Amendment protection for content neutral restrictions.[249]

The prohibition of writing virus source code would be a content based restriction and therefore must serve a compelling state interest.[250] The question is whether the programming falls

---

[244] *Corley*, 273 F.3d at 451.

[245] It is unlikely that one could legitimately argue that the components and wiring patterns used in an operational explosive device are being used to communicate ideas about electrical engineering protected by free speech. Yet, the sequence of opcodes in machine language are protected as communicating computer science ideas. In addition, it is unlikely that someone would argue that textbooks on electrical engineering are not protected even if they might be used to build a bomb, because they convey ideas useful in a wider range of areas than just bomb making. Likewise, source code should be protected, because it conveys useful computer science ideas, even if those ideas could be used to create virus code. *See* Zetter, *supra* note 6 (quoting Peter Tippet, "With a computer virus, the words are the bomb.").

[246] *Corley*, 273 F.3d at 454 (discussing how a restriction is content neutral if it is based solely on the functional capabilities of the program without reference to its content).

[247] *See Are "Good" Computer Viruses Still a Bad Idea*, *supra* note 93 (stating there are no applications that are better accomplished by viruses than by a legitimate and legal form of software).

[248] Computer professionals have not identified a single application where a virus form of code is a good alternative to other standard types of computer programs. *See id*.

[249] *See* Universal City Studios, Inc. v. Reimerdes, 111 F. Supp. 2d 294, 328–29 (S.D.N.Y. 2000) (stating that in addition to conveying the thoughts of the programmer, the code has a distinctly functional non-speech aspect).

[250] The restriction is directed at a particular subject matter (virus programs) that utilizes particular ideas (the capability of the code to replicate itself) and contains specific content (particular sets of instructions that allow the program to write a copy of itself into another program). *See id.* at 327 (" . . . government has no power to restrict expression because of its message, its ideas, its subject matter, or its content . . . ." (citation omitted)).

within constitutionally proscribable content.[251] Under a First Amendment analysis to determine whether the government could pass constitutional legislation prohibiting the writing, possession, and distribution of virus code, the government would have to demonstrate a compelling state interest that is achieved by the least restrictive means.[252] The damage previously caused and the future amount threatened by these virus programs should be sufficient to demonstrate a compelling state interest.[253] The restriction can be extremely narrowly tailored because of the unique functional aspects of virus code. The features that make a program a virus, its ability to replicate and append or insert itself into another program without the knowledge or authorization of the user can be used to precisely define the content specific restrictions.

One possible way to circumvent the problem of how to allow the communication of ideas involving virus source code without allowing free access to working viruses is to restrict access to functioning code rather than completely prohibiting it. The writing of viruses could become a licensed activity limited to professionals and requiring the oversight of the federal government or an independent organization.[254] In this manner writing virus code without proper licensing could be added to the list of computer crimes, rather than criminalizing the writing of any virus code.[255] This would result in only the regulation of the specific content rather than an outright prohibition, thereby avoiding the censorship

[251] *See* R.A.V. v. City of St. Paul, 505 U.S. 377, 383 (1992) (". . . areas of speech can, consistently with the First Amendment, be regulated because of their constitutionally proscribable content. . .").

[252] *See* Sable Commc'ns of California, Inc. v. FCC, 492 U.S. 115, 126 (1989) ("[T]he government may regulate the content of constitutionally protected speech in order to promote a compelling interest if it chooses the least restrictive means to further the articulated interest.").

[253] *See* Standler, *supra* note 16; Junger v. Daley, 209 F.3d 481, 485 (6th Cir. 2000) ("The government must show the harms are real, and not merely conjectural, and the regulation will alleviate the harm in a real and material way.").

[254] Many dangerous activities require specific licenses to legally engage in the activity (e.g. driving a motor vehicle, ownership of firearms, practice of medicine or law, storage of dangerous or illegal substances for legitimate purposes, use of nuclear energy, etc.).

[255] Standler, *supra* note 16 (commenting on the failure of legislators to require licensing of computer programmers in a manner similar to other professional such as physicians or engineers, or the restriction of certain computer programming in a manner similar to the licensing of the production or distribution of pharmaceuticals).

of ideas. The creation and possession of virus writing kits could also be outlawed, because there is no legitimate purpose for such possession.[256]

CONCLUSION

After analyzing the technical features of malicious software, the arguments for and against its uses and effects on society, and the legislative approaches taken by state and federal governments to curtail the propagation of malicious software, Alistair Kelman appears correct in stating there is no good reason to allow for viruses.[257] This is supported by statements made by Vesselin Bontchev that there is no good application for viruses which could not be better performed by standard (non-self-replicating) software,[258] and Dr. Tippett[259] that virus writing should be outlawed.[260]

Some statutes do not directly address the issue of viruses and malicious software.[261] However, those statutes that do address viruses do not go far enough. While they outlaw the distribution of viruses and provide sanctions for damage that results from such distribution, they do not address the writing of virus code. The writing of virus code is a very specialized act, and has an inherent intent to cause mischief.[262] The writing and possession of such code should be criminalized with suitable exceptions for specific professionals in place. Virus code should be classified as inherently dangerous due to its harmful nature, and the lack of any socially beneficial facet should proscribe its place in regular

---

[256] This is not unlike federal laws prohibiting drug paraphernalia. Federal laws prohibiting drug paraphernalia prohibit the instrumentality of a crime even though the material would be harmless without the presence of the illegal drug. *See* 21 U.S.C.A. § 863(a)(1) (2000) (making it unlawful to sell or offer for sale drug paraphernalia); § 863(f)(1)(exempting persons authorized by local, state, or federal law from prosecution).

[257] *See* Kelman, *supra* note 177.

[258] *See Are "Good" Computer Viruses Still a Bad Idea*, *supra* note 93.

[259] Dr. Peter Tippett is the Chief Technology Officer at a company that tests antivirus products and sends out reports when new viruses are discovered. *See* Zetter, *supra* note 6.

[260] *See id.*

[261] *See supra* Part II.B.3 and accompanying text.

[262] *See supra* Part I.A. Any intention to secrete a piece of computer code on another person's system should be considered a form of mischief.

society. That is not to say anti-virus professionals, computer science professors, and other suitably qualified individuals and organizations should be prevented from creating, acquiring, accessing, or manipulating such code. But virus code has no place in the hands of the average computer user or even the hands of the average computer professional.

Very little freedom or right of expression would be lost if such acts were outlawed. The virus writing community is very small,[263] and novices create most viruses with the help of virus writing tools. These individuals cannot claim that their viruses are a form of expression, because they lack even the basic comprehension of what they are doing.

Viruses are not inherently evil; Bontchev points out that viruses are technology, and therefore lack any ethical predisposition.[264] The majority of individuals who do write and release viruses are not necessarily bad or evil.[265] There are simply no benefits, which outweigh the dangers and harm caused by viruses or other malicious software in the possession of the general population.

Licensing and oversight by suitable agencies or government departments would allow continued progress by anti-virus and computer security companies and individuals. This scheme would permit researchers to continue their efforts to protect computer users from those individuals and groups who are not dissuaded by the ever-evolving computer crime statutes. It would also leave the door open for research into computer security, counter terrorism and computer warfare; fields where the average person does not tread.

A change in approach from pursuing those who cause virus outbreaks to those who write the viruses would produce a greater return on the time, money, and effort invested by law enforcement

[263] The virus-writing population was placed at no more than 4,500 in 1994. Sarah Gordon, *The Generic Virus Writer* (1994) (unpublished article first presented at the 4th International Virus Bulletin Conference), *available at* http://www.research.ibm.com/antivirus/SciPapers/Gordon/GenericVirusWriter.html (discussing the ethical and demographic make-up of the virus-writing community).

[264] *See Are "Good" Computer Viruses Still a Bad Idea*, *supra* note 93

[265] *See id.*

in preventing and prosecuting computer crimes. Congress has had over twenty years to examine the beneficial aspects, if any, of writing computer worms and viruses. Legislators should take a serious look at statutorily restricting the writing of such computer code. It is an extremely small segment of the population which would be affected and they could find permissible ways of expressing their interests through licensed professionals teaching ethical courses in computer science curriculums. These restrictions could be narrowly tailored and directed at activities, which the government has a legitimate and reasonable interest in controlling. The benefits to everyday computer users and society as a whole must be accorded its due weight in any balancing test, and these benefits clearly outweigh the losses to the virus-writing community.