Texas A&M Law Review

Volume 3 | Issue 3 Article 9

5-2016

All Your IP Are Belong to Us: An Analysis of Intellectual Property Rights as Applied to Malware

Miranda Rodriguez

ALL YOUR IP ARE BELONG TO US: AN ANALYSIS OF INTELLECTUAL PROPERTY RIGHTS AS APPLIED TO MALWARE

*By: Miranda Rodriguez**

ABSTRACT

The cybersecurity and cybercrime industries are tied together in an arms race where both seek out new security vulnerabilities to exploit on offense or to remediate on defense. Malware (malicious software) offers one of the primary weapons pioneering new computer technologies on both sides. However, the average Internet user sees malware at best as an annoyance that is merely the price of surfing the web.

It is clear that cybersecurity is a business and a successful one. The cybersecurity industry maintains copyrights and patents on our cyber defense technologies—antivirus software, firewalls, intrusion prevention systems, and more. There are no federal copyrights and patents on malware, even regarding the cybersecurity industry's creations. From an intellectual property perspective, there is no difference between ordinary software and malicious software. Malware, as offensive software, can and should be protected, just as we protect our defensive software.

TABLE OF CONTENTS

* J.D. Candidate, Texas A&M University School of Law, December 2015; B.S. Computer Science, Texas Tech University, 2008. The Author would like to thank her advising professor, Megan Carpenter, for her guidance during the writing and editing process.

I. INTRODUCTION: JUST A BIT OF MALWARE

Malicious software ("malware") is the bane of any Internet user's daily experience. Entire industries have been born just to combat the spread of malware ever since Creeper, the first documented specimen, appeared in the 1970s. Creeper, though widely accepted as the first computer virus, was not a malicious creation.[1] In 1971, Bob Thomas of BBN Technologies deployed Creeper onto the precursor to the Internet—the Advanced Research Projects Agency Network ("ARPANET")—as an experiment in mobile code.[2] Creeper would move to a system connected to ARPANET, print "I'M THE CREEPER : CATCH ME IF YOU CAN," and then move to the next system.[3] Later, Thomas's partner, Ray Tomlinson, wrote a companion program, Reaper, designed to replicate itself on systems connected to ARPANET in order to find and remove Thomas's Creeper.[4] These two features, mobility and self-replication, would later become the primary indicia of a computer virus.[5]

It is important to note that Creeper and Reaper were not created maliciously. These were experiments in what was a new field of computing. These processes were something novel and had never been done before. Since 1971, this type of software has evolved to the malware we know today—viruses, worms, Trojans, adware, spyware, rootkits, botnets, and more. In the forty-four years since Creeper and Reaper, the malware space has pioneered technologies taking advantage of polymorphic encryption, novel ways to use Internet communication protocols, and manipulation of memory in much the same way as the more benign areas of Computer Science.

But if you look for patents covering Distributed Denial of Service ("DDoS," a common attack vector where an attacker blocks access to a system by using up the system's available resources or bandwidth),[6] you will not find them. Instead, you will find a multitude of patents regarding preventing DDoS.[7] Neither are there patents for buffer overflow (another common attack vector where the attacker causes

1. *See* Richard E. Schantz, *BBN's Network Computing Software Infrastructure and Distributed Applications (1970–1990)*, 28 IEEE ANNALS HIST. COMPUTING, Jan.–Mar. 2006, at 72, 74.

2. *See id.* at 73–74.

3. *First Computer Virus, Creeper, Was No Bug*, DISCOVERY NEWS (Mar. 16, 2011, 4:41 PM), http://news.discovery.com/tech/first-computer-virus-creeper-was-no-bug-110316.htm [http://perma.cc/G79S-3NUS].

4. John F. Shoch & Jon A. Hupp, *The "Worm" Programs—Early Experience With a Distributed Computation*, 25 COMM. ACM, Mar. 1982, at 172, 179.

5. *See What Is a Computer Virus or a Computer Worm?*, KASPERSKY LAB, http://www.kaspersky.com/internet-security-center/threats/viruses-worms [http://perma.cc/6TQV-9FSB] [hereinafter *What Is a Computer Virus?*].

6. *Botnets*, SHADOW SERVER, https://www.shadowserver.org/wiki/pmwiki.php/Information/Botnets (last updated Nov. 2, 2015) [http://perma.cc/PMY3-JG2G].

7. *See, e.g.*, U.S. Patent Nos. 8,359,648 (filed Sept. 1, 2009); 8,634,717 (filed Dec. 8, 2011); 8,886,927 (filed Jan. 14, 2013).

the system to write outside the bounds of allocated memory to corrupt data or execute arbitrary commands),[8] but there are for preventing it.[9] In modern technology, these methods of attack are no longer novel creations. But when DDoS was first used in 1998[10] and Buffer Overflow in 1988,[11] could they have been patented at the time?

Companies who have made their mark securing against malware—McAfee, Symantec, etc.—hold federally registered copyrights for their security software.[12] The same does not exist for rootkits like SpyEye, worms like SQL Slammer, or viruses like Melissa.[13] But could there be a common law copyright applied to these malware samples? Could the authors behind these computer programs federally register a copyright if they developed these programs in the United States?

These questions appeared to have been ignored until 2009 when the ZeuS rootkit was first observed on the Internet and security researchers began to analyze the code.[14] Much to their surprise, ZeuS had a rudimentary license agreement in its documentation.[15] While it did not explicitly retain copyright, it provided the following basic tenets translated into English:

---

8. *Buffer Overflow*, THE OPEN WEB APPLICATION SEC. PROJECT, https://www.owasp.org/index.php/Buffer_Overflow (last updated Sept. 3, 2014) [http://perma.cc/C832-SCK5].

9. *See, e.g.*, U.S. Patent Nos. 6,301,699 (filed Mar. 18, 1999); 8,443,442 (filed Jan. 30, 2007); 9,069,970 (filed Feb. 13, 2013).

10. Dave Dittrich, *DDoS Attack Tool Timeline*, USENIX SECURITY SYMPOSIUM 2000, DDOS — IS THERE REALLY A THREAT?, https://staff.washington.edu/dittrich/talks/sec2000/timeline.html (last updated July 22, 2000) [http://perma.cc/NJ5V-4G4R].

11. Murray Stokely, *3.3 Buffer Overflows*, *in* FREEBSD DEVELOPERS' HANDBOOK 29, 29 (2014), ftp://ftp.freebsd.org/pub/FreeBSD/doc/en_US.ISO8859-1/books/developers-handbook/ l (follow "book.pdf.zip"; then, after download, open "book.pdf" file) [http://perma.cc/6TTR-S3WF].

12. *See, e.g.*, U.S. Copyright Registration Nos. TX0007250609 (registered Apr. 7, 2010) ("McAffee Host Intrusion Prevention 7.0.0"); TX0007703634 (registered June 19, 2012) ("Symantec Endpoint Protection 12.1").

13. These are three examples of infamous malware. SpyEye is part of the growing ZeuS family of rootkits and will be discussed in more detail in Part II, *infra*. First seen in 2002, SQL Slammer exploited a buffer overflow vulnerability in MS SQL Server 2000 giving an attacker remote access to the host machine. *See* Edward Ray, *Malware FAQ: MS-SQL Slammer*, SANS INSTITUTE, http://www.sans.org/security-resources/malwarefaq/ms-sql-exploit.php [http://perma.cc/7LZQ-YXS9]. First seen in 1999, the relatively benign Melissa virus spread at an alarming rate because it would send copies of itself for the first fifty Outlook contacts of the compromised user. Stephen Northcutt, *Intrusion Detection FAQ: What Was the Melissa Virus and What Can We Learn From it?*, SANS INSTITUTE, https://www.sans.org/security-resources/idfaq/what_melissa_teaches_us.php [http://perma.cc/2JSP-LPL9].

14. Andrew Hendry, *Non-Tech Criminals Can Now Rent-a-Botnet*, COMPUTERWORLD (May 15, 2008, 10:37 AM), http://www.computerworld.com.au/article/216322/non-tech_criminals_can_now_rent-a-botnet/.

15. Joel Hruska, *Malware Authors Turn to EULAs to Protect Their Work*, ARS TECHNICA (Apr. 28, 2008, 5:40 PM), http://arstechnica.com/security/2008/04/malware-authors-turn-to-eulas-to-protect-their-work/ [http://perma.cc/BD89-S4HL].

- The user may not distribute the product in a commercial way.
- The user may not reverse-engineer the bot builder.
- The user may not use the control panel to control other botnets.
- The user may not deliberately share any portion of the code to anti-virus companies.
- The user must pay for any future features or improvements.[16]

Some writers quickly dismissed the agreement as absurd and bizarre.[17] One user on the popular technology news-source "Slashdot" joked, "While it seems silly to imagine Zeus's authors going to the authorities for violations of this EULA, . . . they probably have an extra-judicial means of contract enforcement named Ivan."[18] The opinions were perfectly valid, considering that long-established contracts case law tells us that a contract that arises from an illegal act is not enforceable.[19]

The agreement was a mere curiosity compared to the more interesting support model and pricing structure for ZeuS[20]—a signal to security researchers that malware was no longer the product of a lone computer scientist experimenting with a new technology as with Creeper, but a business unto itself.[21] When ZeuS was transferred to SpyEye, it retained the same licensing agreement and the new owner added features to prevent users from copying the code.[22]

But was it so strange for a software author (malicious or otherwise) to protect his or her invention or creation? This leads to questions regarding whether malware can actually be subject to traditional intellectual property protections that are afforded to other software devel-

16. *Zeus Trojan Family*, DELL SONICWALL SECURITY CENTER (May 26, 2009), https://www.mysonicwall.com/sonicalert/searchresults.aspx?ev=article&id=132 [http://perma.cc/EU2S-VB26] (providing an English translation of the original Russian EULA).

17. Yvette Joy Liebesman, *When Selling Your Personal Name Mark Extends to Selling Your Soul*, 83 TEMP. L. REV. 1, 21 (2010) ("And in a rather bizarre example of an illegal, and thus unenforceable, contract, the creators of the Zeus 'malware' software program 'have added an end-user license agreement to their "product," setting out a bunch of terms controlling how the criminals who buy their products may use it, and threatening dire technological reprisals for violations.'" (footnote omitted) (quoting Cory Doctorow, *Malware Gets a EULA*, BOING BOING, (Apr. 29, 2008, 3:14 AM), http://www.boingboing.net/2008/04/29/malware-gets-a-eula.html [http://perma.cc/CM6C-PMD6])).

18. Kdawson, *EULAs for Malware*, SLASHDOT (Apr. 28, 2008, 10:51 PM), http://entertainment-beta.slashdot.org/story/08/04/29/0057236/eulas-for-malware [http://perma.cc/HZF3-P8UG].

19. *See* Armstrong v. Toler, 24 U.S. 258, 270 (1826).

20. Discussed in more detail in Section II.B, *infra*.

21. Kevin Stevens & Don Jackson, *ZeuS Banking Trojan Report*, DELL SECUREWORKS (Mar. 11, 2010), http://www.secureworks.com/cyber-threat-intelligence/threats/zeus/ [http://perma.cc/W996-URSC].

22. Sean Martin, *The Making of a Cybercrime Market*, CSO (Aug. 11, 2014, 12:56 PM), http://www.csoonline.com/article/2463175/data-protection/the-making-of-a-cybercrime-market.html [http://perma.cc/NM9B-TYTF].

opers. This Comment first examines what malware is and the initial potential barrier to entry: the legality of writing it. The examination then turns to the ZeuS family of rootkits specifically and its copyright and patent prospects. Next, the analysis focuses on the public policy concerns associated with granting or denying intellectual property protections to malware. Finally, this Comment recommends allowing traditional intellectual property to apply to malware, regardless of how it is used.

II. MALWARE IN A NUTSHELL

A. *Definitions*

Although it has various definitions, *malware* commonly means "software designed to interfere with a computer's normal functioning."[23] This can include damaging systems, disrupting functions, stealing data, or otherwise causing some "bad" action.[24] But malware is not just limited to computers. It can also be found on network devices such as switches and routers, mobile devices such as phones and tablets, and on removable media such as thumb drives and CDs.[25] If the device has the ability to hold data, it can be infected with some form of malware.

Malware comes in many forms. Viruses are programs that self-replicate and move from computer to computer.[26] Trojans masquerade as other software or files to execute other malware, enable a remote connection, or exploit some other system vulnerability.[27] Spyware monitors how a system is used or takes data off the system to send to a third party.[28] These are just a few of the different forms malicious software may take. This Comment, however, will focus primarily on rootkits and botnets.

In his work *The Rootkit Arsenal,* Bill Blunden sums up rootkits best as "an uninvited guest that's surprisingly neat, clean, and difficult to unearth."[29] To put it technically, however, rootkits are "a set of binaries, scripts, and configuration files (e.g., a kit) that allows someone covertly to maintain access to a computer so that he can issue com-

23. *Malware*, MERRIAM-WEBSTER, http://www.merriam-webster.com/dictionary/malware [http://perma.cc/EY4G-8EYR].

24. *What Is the Difference: Viruses, Worms, Trojans, and Bots?*, CISCO, http://www.cisco.com/web/about/security/intelligence/virus-worm-diffs.html [http://perma.cc/5NWP-8C4W].

25. *Id.*

26. *See What Is a Computer Virus?*, *supra* note 5.

27. *What Is a Trojan Virus?*, KASPERSKY LAB, http://usa.kaspersky.com/internet-security-center/threats/trojans [http://perma.cc/L6H5-7WUE].

28. *See Types of Spyware*, KASPERSKY LAB, http://usa.kaspersky.com/internet-security-center/threats/adware-pornware-riskware [http://perma.cc/CEH3-MBPS].

29. REV. BILL BLUNDEN, THE ROOTKIT ARSENAL: ESCAPE AND EVASION IN THE DARK CORNERS OF THE SYSTEM 4 (2d ed. 2013).

mands and scavenge data without alerting the system's owner."[30] Rootkits are an end-game type of malware used after something else has already exploited a system and gained access to a target machine.[31]

Rootkits are often paired with other malware. For example, the rootkit might keep a virus hidden or a Trojan might have deployed the rootkit in the first place.[32] The rootkit is also one of the ways that a botnet is established. Botnets are a collection of computers (hosts known as "zombies" or "drones") controlled by a "Command and Control" server ("C&C") that work together to accomplish a task for the botnet owner ("herder").[33] This might be a benign function such as a university in need of a distributed computing power for a large-scale data project[34] or a nefarious function such as a massive DDoS attack.[35]

Though their names are similar, a rootkit is not a malware kit. Malware kits, also known as exploit kits, are prepackaged exploits that require little knowledge on the part of the user to leverage against a target.[36] With some malware kits, the user can customize their malware through configuring a builder in the kit.[37] The kits can include premade landing screens for phishing attempts, executable programs to use as Trojans, or even a customized rootkit.[38] The particular rootkits this Comment uses as an example, the ZeuS family, are built using variants of the original ZeuS malware kit.[39]

B. *The ZeuS Family*

ZeuS was an infamous rootkit in the late 2000s and early 2010s developed by Evgeniy Bogchev, known as "Slavik" in the underground malware market.[40] ZeuS was unique because it was marketed as a

30. *Id.* at 5.

31. *Id.* at 6–7.

32. *See id.* at 13–14.

33. *Botnets*, *supra* note 6.

34. *See* BERKELEY OPEN INFRASTRUCTURE FOR NETWORK COMPUTING (BOINC), http://boinc.berkeley.edu/ [http://perma.cc/63VV-L558] (last modified Sept. 26, 2015, 6:12 AM) [hereinafter BOINC].

35. BLUNDEN, *supra* note 29, at 17.

36. Joshua Cannell, *Tools of the Trade: Exploit Kits*, MALWAREBYTES UNPACKED (Feb. 11, 2013), https://blog.malwarebytes.org/intelligence/2013/02/tools-of-the-trade-exploit-kits/ [http://perma.cc/3TZT-8XZU].

37. *Id.*

38. *Id.*

39. JAMES WYKE, SOPHOSLABS, WHAT IS ZEUS? 2 (2011), http://www.sophos.com/en-us/medialibrary/PDFs/technical%20papers/Sophos%20what%20is%20zeus%20tp.pdf [http://perma.cc/W5FW-XX65].

40. David Gilbert, *Gameover for Slavik—The Cybercrime Kingpin Behind the Zeus Malware: Evgeniy Bogachev Unmasked*, INT'L BUS. TIMES (June 3, 2014, 3:21 PM), http://www.ibtimes.co.uk/gameover-slavik-cybercrime-kingpin-behind-zeus-malware-1451095 [http://perma.cc/WP99-B9AT].

malware kit that users could configure to generate their own botnet.[41] While the free version had very few features, a fully functioning kit in 2010 could cost about $4,000.[42] Depending on what other features users wanted, they could pay extra for additional modules to add to their kit.[43] For example, a user could obtain a backdoor functionality through the BackConnect module in order to connect to the infected computer again for $1,500 or a VPN module for $10,000 to connect with a Total Presence Proxy and take full control of the infected machine.[44] Just like innocuous software that a less malicious user might pay for, the licensing of Slavik's product was controlled using a hardware-based system where a user provided a key generated from the user's computer and Slavik would provide a key for that computer in return in order to run the software.[45]

ZeuS in its simplest form can be reduced to the builder, the dropper, the Zbot binary, and the C&C.[46] The builder is the main component of the kit and creates the malware that will infect the victim's machine and connect back to the C&C.[47] In later versions of ZeuS, the user could also configure the malware to use polymorphic encryption, which made it significantly harder to detect.[48] The malware could be dropped on the target machine by another Trojan, or it could function as the dropper itself.[49] If acting as the dropper, the malware would begin to copy itself onto the system.[50] If the malware was being dropped, then it would begin injecting itself into running processes on the machine.[51] The Zbot binary is the core of the malware and is responsible for the major functions such as copying itself, injecting code, scraping data, and communicating with the C&C.[52] The herder issues commands to the compromised machines (the bots) via the C&C.[53] One of the hallmarks that set ZeuS apart from other malicious software was how easy it was to use both the configuration elements

41. *See* WYKE, *supra* note 39, at 2.
42. Stevens & Jackson, *supra* note 21.
43. *Id.*
44. *Id.*
45. Brian Krebs, *SpyEye vs. ZeuS Rivalry*, KREBS ON SECURITY (Apr. 1, 2010), http://krebsonsecurity.com/2010/04/spyeye-vs-zeus-rivalry/ [http://perma.cc/F7QT-75RE].
46. WYKE, *supra* note 39, at 3–7.
47. *Id.* at 3, 6.
48. Stevens & Jackson, *supra* note 21.
49. *See* WYKE, *supra* note 39, at 7.
50. *Id.*
51. *See id.*
52. *Id.* at 6.
53. *Id.* at 5.

in the builder and to set up the C&C.[54] In minutes, a user could have a running C&C to coordinate bots and harvest data.[55]

In 2010, a "new kit on the block" arrived—Aleksandr Panin, also known as "Gribodemon," developed a major competitor to ZeuS called SpyEye.[56] SpyEye also was referred to as the "ZeuS Killer" because not only did the kit perform with similar functionality for a similar price, but it would also search a target machine for ZeuS and eradicate the rootkit.[57] Despite Gribodemon's boasts that SpyEye was superior to ZeuS, the rivalry between Slavik and Gribodemon died down in late 2010 and it appeared that Slavik had transferred the ZeuS code to SpyEye.[58] Gribodemon planned to combine the two malware kits to make an even better rootkit with more modular plug-in options than ZeuS.[59] Just as Slavik had implemented a licensing scheme and rudimentary license agreement in his software, Gribodemon threatened that unauthorized copies of the old ZeuS product had an undocumented backdoor that he would use to add the user's machine to his own botnet.[60]

But just as quickly as the new SpyEye took off, it came crashing back down with the development of Ice IX.[61] Around the same time that SpyEye began integrating ZeuS to build a bigger, better rootkit, the ZeuS source code was leaked in underground malware forums and became fair game to any enterprising malware author.[62] Shortly after the release of the ZeuS code, Ice IX went on the market as a greatly improved version of ZeuS with greater survivability (a key feature of a successful rootkit).[63] Although the Ice IX user interface was perhaps not as flashy and pretty as ZeuS and SpyEye, the much lower cost (the build kit only cost $1,800 compared to ZeuS's $4,000 pricetag) and the advanced features were certainly far more appealing.[64] Security research firm Damballa began to see previous SpyEye customers switch

54. *See* Simon Mullis, *Cybercriminal Intent: How to Build Your Own Botnet in Less than 15 Minutes*, FIREEYE (Aug. 2, 2013), https://www.fireeye.com/blog/executive-perspective/2013/08/cybercriminal-intent-how-to-build-your-own-botnet-in-less-than-15-minutes.html [http://perma.cc/7GSS-DE7R].

55. *Id.*

56. Krebs, *SpyEye vs. ZeuS Rivalry*, *supra* note 45.

57. *Id.*

58. Brian Krebs, *SpyEye v. ZeuS Rivalry Ends in Quiet Merger*, KREBS ON SECURITY (Oct. 24, 2010), http://krebsonsecurity.com/2010/10/spyeye-v-zeus-rivalry-ends-in-quiet-merger/ [http://perma.cc/J7FE-E2NT].

59. *Id.*

60. *Id.*

61. SEAN BODMER, DAMBALLA, SPYEYE BEING KICKED TO THE CURB BY ITS CUSTOMERS? 5–6 (2012), https://www.damballa.com/downloads/r_pubs/RN_SpyEye-Kicked-to-Curb_Bodmer.pdf [http://perma.cc/E7PD-TPPE].

62. Dennis Fisher, *ZeuS Source Code Leaked*, THREATPOST (May 10, 2011, 2:10 PM), http://threatpost.com/zeus-source-code-leaked-051011/75217 [http://perma.cc/8LT4-YXD3].

63. BODMER, *supra* note 61, at 7.

64. *See id.*

over to Ice IX soon after its release.[65] Even if the leaked source code meant that more malware authors could make hybrid forms of the old ZeuS, Ice IX clearly offered something that criminal organizations were willing to pay for.

C. *Malware and its Authors in the Law*

Common definitions, forms, and families aside, there is no clear legal prohibition on creating malware. Black's Law Dictionary provides a legal definition for malicious technology: "[a]ny electronic or mechanical means, [especially] software, used to monitor or gain access to another's computer system without authorization for the purpose of impairing or disabling the system."[66] This mostly coincides with the common meaning discussed earlier—to interfere with the computer's normal functioning.[67] But where is the definition actually used?

The key federal statute on point is 18 U.S.C. § 1030, entitled "Fraud and related activity in connection with computers" (and still commonly known as the Computer Fraud and Abuse Act ("CFAA")). The terms "malicious technology" and "malware" are not used or defined in the CFAA, but what the statute prohibits certainly fits within malware's definition: accessing without authorization, intending to defraud, intentionally causing damage, recklessly causing damage, and extorting money through threat of the aforementioned actions.[68] The statute criminalizes the use of malware by a malicious user who intentionally accesses (or threatens to access) data or systems beyond the scope of the user's authorization or intentionally causes damage (or threatens to cause damage) to that data or system.[69] The statute never mentions writing the software.

Though this particular statute does not specifically target creation and writing, it has been used against authors of malware, including cybersecurity researchers[70] (the "White Hat" hackers who seek to identify and remediate a security vulnerability before it is maliciously exploited),[71] cybercriminals[72] (the "Black Hat" hackers who mali-

65. *Id.* at 5.
66. *Malicious Technology*, BLACK'S LAW DICTIONARY (10th ed. 2014).
67. *Malware*, *supra* note 23.
68. 18 U.S.C. § 1030 (2012).
69. *Id.* § 1030(a).
70. Indictment, United States v. Auernheimer, Crim. No. 11-cr-470 (SDW), 2012 WL 5389142 (D.N.J. Oct. 24, 2012), *vacated for improper venue*, 748 F.3d 525 (3d Cir. 2014) (No. 11-470) [hereinafter *Auernheimer* Indictment] (charging a cybersecurity researcher under CFAA).
71. *What Is a White Hat?*, SECPOINT, https://www.secpoint.com/what-is-a-white-hat.html [http://perma.cc/3HPB-TMZ4].
72. *See* Press Release, Dep't of Justice Office of Pub. Affairs, Cyber Criminal Pleads Guilty to Developing and Distributing Notorious SpyEye Malware (Jan. 28, 2014) [hereinafter Cyber Criminal Pleads Guilty], http://www.justice.gov/opa/pr/cyber-criminal-pleads-guilty-developing-and-distributing-notorious-spyeye-malware

ciously exploit the vulnerability),[73] and those who fall somewhere between the two.[74] U.S. law enforcement recently apprehended cybercriminal and SpyEye author Alexsandr Panin, who pled guilty to "conspiracy to commit wire and bank fraud for his role as the primary developer and distributor of the malicious software."[75] Panin, as a cybercriminal intending to maliciously use SpyEye, is someone you would expect to be caught by the provisions in the CFAA. However, security researcher Andrew Auernheimer was also charged with conspiracy after he discovered a vulnerability in how AT&T handled iPad users' email addresses.[76] Auernheimer published his co-authored proof-of-concept script, Account Slurper, which demonstrated that vulnerability.[77] Auernheimer sought to discover vulnerability and see it remediated as opposed to Panin who sought to exploit vulnerability. In both cases, malware created by a cybercriminal and a security researcher were caught by part (b) of the CFAA: "Whoever conspires to commit or attempts to commit an offense under subsection (a) of this section shall be punished as provided in subsection (c) of this section."[78]

Auernheimer and Panin are only the latest authors to come under fire from CFAA. Later in 2014, *The Guardian* published an article in which several cybersecurity researchers revealed that they had been threatened with CFAA action if they continued the research they were currently performing.[79] Most notable was Metasploit author H.D. Moore, who claimed he had been "warned" that his Internet-wide scanning project would run afoul of CFAA.[80] Despite the threats, H.D. Moore completed his research and, in the process, uncovered serious vulnerabilities in Universal Plug and Play.[81] The article also reported on research that had been abandoned by researcher Zach Lanier when his research team was threatened with CFAA before they could disclose a vulnerability in a device marketed to children.[82] The lack of distinction between a cybercriminal and a cyber-

[http://perma.cc/6XMP-D4NC] (announcing a cybercriminal being charged under CFAA).

73. *What Is a Black Hat?*, SECPOINT, https://www.secpoint.com/what-is-a-black-hat.html [http://perma.cc/RE6B-Y3WX].

74. *See, e.g.*, United States v. Morris, 928 F.2d 504 (2d Cir. 1991) (blurring the line between cybersecurity researcher and cybercriminal due to the result of the malware).

75. Cyber Criminal Pleads Guilty, *supra* note 72.

76. *Auernheimer* Indictment, *supra* note 70, at ¶¶ 5–7.

77. *Id.* at ¶¶ 7–10.

78. 18 U.S.C. § 1030(b) (2012).

79. Tom Brewster, *US Cybercrime Laws Being Used to Target Security Researchers*, GUARDIAN (May 29, 2014, 11:09 AM), http://www.theguardian.com/technology/2014/may/29/us-cybercrime-laws-security-researchers [http://perma.cc/5GDV-CTVS].

80. *Id.*

81. *Id.*

82. *Id.*

security researcher in CFAA causes concern for cybersecurity experts and leads to a "chilling effect on . . . research."[83]

Considering *United States v. Auernheimer*, the concern is valid. After being charged with both conspiracy and identity fraud for Account Slurper, Aurenheimer was sentenced to forty-one months of imprisonment.[84] Despite elegant arguments from numerous interest groups, cybersecurity experts, and professionals, the court bypassed "a number of complex and novel issues that are of great public importance in our increasingly interconnected age" and vacated Aurenheimer's conviction on a venue error—leaving questions regarding how to handle cybersecurity research undecided.[85]

This is not to say that someone who uses malware to harm or steal should not be punished accordingly. The point of CFAA is to criminalize computer fraud, which is often carried out by malware.[86] Malware liability should be analogous to firearms liability. Those who rob a bank with a gun should be punished for robbing the bank with a deadly weapon. However, under today's laws, gun manufacturers may be subject to traditional products liability—but they are not liable for the crime committed with their product.[87]

It is hard to separate writing malware from maliciously using the malware—this is why the CFAA's conspiracy charge can be so successful. Like a gun, whose purpose may be described as "harm," malware's purpose is often to harm a computer system. To simply say "to harm" in regards to a gun is a gross oversimplification. Harm inflicted in self-defense, for example, is treated far differently than murder. The same can be said of malware—what a security researcher writes is often more of a gray area than what a cybercriminal writes. Aurenheimer used Account Slurper to prove a vulnerability while Bogchev and Panin used their rootkits to commit crimes.

Consider the rootkits and botnets to which this Comment applies intellectual property rights. In *The Rootkit Arsenal*, Blunden examines several different scenarios where rootkit technologies are part of "ordinary" software.[88] Sony's digital rights management software at one point hid files, phoned home to Sony (like a bot to a C&C), and trafficked user data such as media player IDs and the user IP addresses.[89]

---

83. *Id.*

84. United States v. Auernheimer, 748 F.3d 525, 532 (3d Cir. 2014).

85. *Id.* at 532, 535–36.

86. 132 Cong. Rec. 7816 (1986) ("Computer technology has brought us a long way in the past decade. However, computer technology—with all its gains—has left us with a new breed of criminal: the technologically sophisticated criminal who breaks into computerized data files."); *2014 Data Breach Investigations Report*, VERIZON 9 fig.8, http://www.verizonenterprise.com/DBIR/ (follow "Download 2014 DBIR" hyperlink) (showing malware and hacking trend lines representing the prior ten years).

87. *See* 15 U.S.C. §§ 7901–7903 (2012).

88. BLUNDEN, *supra* note 29, at 19–21.

89. *Id.* at 20.

Antivirus vendors have used rootkits to prevent users from deleting or enumerating software files by injecting custom file system filter drivers.[90] Government actors also use the technology in their tools in intelligence operations against other nation states.[91] Blunden put it bluntly: "Asking whether rootkits are inherently good or bad is a ridiculous question. . . . The fact is that rootkit technology is powerful and potentially dangerous."[92]

Like rootkits, botnets inhabit gray areas. Universities can take advantage of botnet technology to create large-scale virtual distributed computer networks such as Berkley's BOINC service.[93] Berkeley project servers function similarly to the C&C in the average botnet.[94] The project servers assign tasks to the volunteer "zombie hosts," who in turn complete the tasks and send the results back.[95] The same type of technology that divides brute force attacks across the botnet herd can also assist researchers in the BOINC projects. Volunteer botnets are not limited to universities. During the height of the WikiLeaks scandal, Anonymous utilized its LOIC botnet software to allow users to voluntarily contribute their computing power to Anonymous-driven DDoS protests of major credit card providers.[96]

CFAA, together with the murky reasons malware authors develop their software, causes substantial confusion over the legality of writing malware. It is difficult to clearly conclude whether writing malware is prohibited. An author or inventor's ability to protect malware does not necessarily hinge on the illegality of the device. Courts have found patents are not void if they are put to illegal purpose, as long as the device "is normally and naturally adapted to a lawful use."[97] However, particularly for cybersecurity researchers, being able to merely write malware without fear of legal action would weigh heavily in the decision-making process of authors or inventors who consider seeking intellectual property protections for their creation.

III. Malware Under Copyright

Because the ZeuS family of rootkits contains a license agreement, this Comment will use the original ZeuS rootkit for copyright analysis. For the purposes of this Section, ZeuS is treated as if it were written in

90. *Id.*
91. *Id.* at 23.
92. *Id.* at 26–27.
93. *See, e.g.*, BOINC, *supra* note 34.
94. *See How BOINC Works*, BOINC, http://boinc.berkeley.edu/wiki/How_BOINC_works [http://perma.cc/C5G9-DQPT] (last updated July 21, 2013, 3:43 AM).
95. *See id.*
96. Elinor Mills, *WikiLeaks Fans Should Think Before They Botnet*, CNET (Dec. 10, 2010, 3:01 PM), http://www.cnet.com/news/wikileaks-fans-should-think-before-they-botnet/ [http://perma.cc/5M6H-SYCD].
97. Koppe v. Burnstingle, 29 F.2d 923, 925 (D.R.I. 1929).

the U.S., as international copyright concerns are not in scope of this Comment.

A. *Ordinary Software's Qualifications for Copyright*

In 1980, Congress amended Title 17 of the United States Code pertaining to copyrights to reflect changes in technology and to extend copyright protection to that technology.[98] At its heart, copyright still requires an "original work[ ] of authorship fixed in [a] tangible medium of expression" that can be perceived in some way, directly or with a machine.[99] Congress' amended language added a definition for computer programs: "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."[100] The test of this language arose in 1983 in *Apple Computer, Inc. v. Franklin Computer Corp.* when Apple sought to enjoin Franklin Computer from further alleged infringement of fourteen of Apple's computer programs.[101] The district court had originally denied the motion to preliminarily enjoin because the court doubted that the object code Apple sought to protect was copyrightable.[102] The Third Circuit Court of Appeals examined copyright protections for computer programs to determine whether there was any difference in how the program appeared to the human reader.[103]

The appellate court in *Apple* looked to the statute and stated that "copyright extends to works in any tangible means of expression 'from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device.'"[104] The court then applied this to the statutory definition of a computer program, concluding that a computer program in its various forms (source code readable by a human or object code readable by a computer) may be perceived either directly or with the aid of a machine and accomplishes its tasks through a set of instructions either directly or indirectly via the computer.[105]

Also at issue in *Apple* was whether programs that were part of an operating system were distinguishable.[106] Franklin Computer argued the programs that were part of the operating system should be excluded from protection because 17 U.S.C. § 102(b) excludes "proce-

98. Deborah F. Buckman, Annotation, *Copyright Protection of Computer Programs*, 180 A.L.R. Fed. 1, § 2[a] (2002) (discussing the findings of the Commission On New Technological Uses of Copyright Works regarding computer programs).
99. 17 U.S.C. § 102 (2012).
100. *Id.* § 101.
101. Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240, 1242 (3d Cir. 1983).
102. *Id.*
103. *Id.* at 1246–47.
104. *Id.* at 1248 (quoting § 102(a) (emphasis omitted)).
105. *Id.*
106. *Id.* at 1249.

dure[s], process[es], . . . [or] method[s] of operation" from copyright.[107] The appellate court countered that, whether the program told the computer "to help prepare an income tax return . . . or to translate a high level language program from source code into its binary language object code form," it was still the same set of instructions that had been expressly referenced in the amended statute.[108] As a result, the copyright analysis for computer programs turns on whether the program includes an original, authored set of instructions fixed in a tangible medium that can be perceived in some way, either directly or with the aid of the computer.

Apple also discussed the limit on computer program copyrights: the *idea/expression dichotomy*.[109] Idea/expression was not a new concept just for computer programs. In 1930, the Second Circuit Court of Appeals grappled with the dichotomy in deciding an infringement suit in *Nichols v. Universal Pictures Corp*.[110] In *Nichols*, playwright Anne Nichols had sued Universal Pictures after the company produced a movie with a similar theme as her "Abie's Irish Rose."[111] Both productions featured a Jewish family and an Irish family whose children fell in love with each other despite the strict religious natures of both families.[112] While certainly Anne Nichols's play had a valid copyright for "Abie's Irish Rose," the court found that her copyright did not extend to "everything that might be drawn from her play."[113] The court further explained that "[a] comedy based upon conflicts between Irish and Jews, into which the marriage of their children enters, is no more susceptible of copyright than the outline of Romeo and Juliet."[114]

While *Nichols* helped define the idea/expression dichotomy, the court conceded that the line between an idea and its expression may seem arbitrary;[115] the court in *Apple* helped draw the line a little clearer with respect to computer programs. The *Apple* court sought to preserve a balance between competition and protection—in this case, balancing protection of Apple Computer's operating system programs with fostering competition with Franklin Computer by allowing them to develop similar programs.[116] The court focused the idea/expression dichotomy inquiry on whether an idea is capable of various modes of expression.[117] If other programs could perform the same function as

107. *Id.* at 1250; § 102(b).
108. *Apple*, 714 F.2d at 1251.
109. *Id.* at 1252–54.
110. Nichols v. Universal Pictures Corp., 45 F.2d 119, 119–20 (2d Cir. 1930).
111. *Id.* at 120.
112. *Id.*
113. *Id.* at 122.
114. *Id.*
115. *Id.*
116. Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240, 1253 (3d Cir. 1983).
117. *Id.*

Apple's, then those programs were all expressions of the idea and copyrightable. However, if there is only one way to do something—an idea which can only be expressed one way—then idea and expression have merged and the program is no longer copyrightable.[118]

Apple's idea/expression dichotomy and merger is not the only test for copyrightability. The Second Circuit Court of Appeals developed the *abstraction-filtration-comparison* test in *Computer Associates International, Inc. v. Altai, Inc.* to help determine if a work has been infringed.[119] In *Altai*, the court used the lessons learned in *Nichols* to abstract the ideas in the computer program from the expressive elements.[120] In the first step, abstraction, the court begins at the code level (the individual instructions protected by statute) and abstracts up, layer by layer, to identify the program's ultimate functions.[121] Once the court has identified levels of abstraction, the next step is to filter out which pieces are ideas and which are expressions.[122] The filtration step is designed to filter out uncopyrightable material such as ideas, elements dictated by efficiency, elements dictated by external factors, and elements taken from the public domain.[123] The last step, comparison, uses what remains after filtration to compare the protectable expressions to the potential offending program.[124]

B. *Ordinary Software's Rights & Limitations Under a Federally Registered Copyright*

The Copyright Act defines the statutory rights provided to federally-registered copyright holders. Section 106(1)–(3) grants the following key rights for computer software: to reproduce, to prepare derivative works, and to distribute copies by sale, transfer, rental, lease, or lending.[125] These rights are subject to limitations,[126] specifically under § 107 "Limitations on exclusive rights: Fair Use" and § 117 "Limitations on exclusive rights: Computer Programs."[127] Section 117 largely focuses on when a user can make copies of the software, such as when it is necessary to run the software or is required to perform maintenance of the host machine.[128] In malware's case, fair use is the more likely defense to copyright prohibitions.

Because idea and expression are so intertwined in software, fair use of the copyrighted materials can extend even to disassembly and

118. *Id.*
119. Comput. Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 706–12 (2d Cir. 1992).
120. *Id.* at 706–07.
121. *Id.* at 707.
122. *Id.*
123. *Id.*
124. *Id.* at 710.
125. 17 U.S.C. § 106(1)–(3) (2012).
126. *See id.* §§ 107–122.
127. *Id.* §§ 107, 117.
128. *Id.* § 117.

decompiling (a process of reverse engineering by transforming machine-readable code to human-readable code).[129] In *Sony Computer Entertainment, Inc. v. Connectix Corp.*, Sony alleged that Connectix had infringed Sony's copyright when Connectix reverse engineered the Sony Playstation BIOS in order to make a video game console emulator.[130] Connectix's reverse engineering included disassembling discrete portions of the Sony BIOS and observing the Sony BIOS as it functioned.[131] The Connectix developers would copy the Sony BIOS into RAM and watch how it interacted with their hardware emulator through use of a debugging program.[132] The court relied on *Sega Enterprises Ltd. v. Accolade, Inc.* to first establish that disassembly was not a prohibited means of fair use: "Where disassembly is the only way to gain access to the ideas and functional elements embodied in a copyrighted computer program and where there is a legitimate reason for seeking such access, disassembly is a fair use of copyrighted work, as a matter of law."[133] Once the court determined disassembly was acceptable, it analyzed the case under the four statutory factors of fair use: (1) purpose and character of the use, (2) nature of the copyrighted work, (3) amount and substantiality of the portion used, and (4) effect of the use upon the potential market.[134]

The first factor, *purpose and character of use*, focuses on "to what extent the new work is transformative."[135] The court found that Connectix's product was not merely a change in form (substituting Sony's product with theirs), but rather "indirect or derivative."[136] The court recognized that Connectix's purpose in reverse engineering the BIOS was to make a product that would be compatible with Sony.[137] The second factor, *nature of the copyrighted work*, looks at what degree of protection the allegedly infringed work requires.[138] The court reiterated its stance from *Sega* that some works are at a distance from the core of copyright protection—a work whose underlying ideas cannot be accessed without copying portions of it is unprotected.[139] Accordingly, the court concluded that the Sony BIOS "lies at a distance from the core because it contains unprotected aspects that cannot be examined without copying."[140] The third factor, *amount and substantial-*

129. *See* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1514, 1520 (9th Cir. 1992).
130. Sony Comput. Entm't, Inc. v. Connectix Corp., 203 F.3d 596, 598–99 (9th Cir. 2000).
131. *Id.* at 601.
132. *Id.*
133. *Id.* at 602 (emphasis omitted).
134. 17 U.S.C. § 107 (2012).
135. *Sony*, 203 F.3d at 606.
136. *Id.* at 607.
137. *Id.*
138. *See id.* at 603.
139. *Id.* at 603–06.
140. *Id.* at 603.

ity of the portion used, examines how much of the original work is used in relation to the original work as a whole.[141] To develop its product, Connectix both disassembled and copied Sony's product multiple times.[142] While in some cases this would weigh heavily against the alleged infringer, it did not against Connectix.[143] The court decided that this factor carried little weight because Connectix's product did not itself contain infringing material from Sony's.[144] The final factor, *effect of the use upon the potential market*, analyzes what impact the alleged infringer's work has on the original work's market.[145] This factor does not seek to protect monopolies.[146] In fact, earlier in the opinion, the court instructed Sony, saying: "If Sony wishes to obtain a lawful monopoly on the functional concepts in its software, it must satisfy the more stringent standards of the patent laws."[147] The court reasoned that Connectix was a competitor and that Sony would see some economic loss due to the competition—but this was not so negative an impact that it would prohibit Connectix's from using the Sony BIOS.[148] Weighing the four factors, the Court found that Connectix's use of the Sony BIOS was indeed fair use.[149]

C. *Copyright Rule Application to the ZeuS Family*

What does this mean, if anything, for ZeuS? The ZeuS rootkit, as software, falls under statutorily defined copyrightable subject matter.[150] The question is whether it can pass the idea/expression test. The underlying idea behind rootkits (and ZeuS is no different) seems to be subverting detection through anti-forensics.[151] Blunden offered several ways this could be accomplished (or expressed): data destruction, data concealment, data transformation, data fabrication, and data source elimination.[152] Moreover, these methods can all be expressed in different ways. For example, ZeuS expressed these ideas by using polymorphic encryption to conceal and transform, as well as by hiding components in registry keys to hide stolen data.[153]

Meanwhile, the idea behind the botnet is to wield the computing power of bots to accomplish the herder's tasks. Botnets are common and there does not seem to be a multitude of ways to express this idea

141. *Id.* at 605–06.
142. *Id.* at 606.
143. *Id.*
144. *Id.*
145. *Id.* at 607.
146. *Id.* at 607–08.
147. *Id.* at 605.
148. *Id.* at 607.
149. *Id.* at 608.
150. 17 U.S.C. § 102 (2012); *see also* H.R. Rep. No. 94-1476, at 51, reprinted in 1976 U.S.S.C.A.N. 5664.
151. *See* BLUNDEN, *supra* note 29, at 13, 35.
152. *Id.* at 46.
153. Stevens & Jackson, *supra* note 42.

in software. Berkeley's BOINC and Anonymous' LOIC accomplish the idea in similar ways—a C&C assigns small, individual tasks to bots and collects the results.[154] Botnets seem to fit better with the abstraction model of copyrightability of software. Botnets require abstracting ideas out to their expression and filtering out the pieces that are in the public domain or are required for the botnet to function.

ZeuS and its family of software combined the botnet with the rootkit.[155] ZeuS has copyrightable elements and non-copyrightable elements—the rootkit in its multiple expressions likely qualifies but perhaps not the botnet if its expression has merged with the idea. ZeuS could be copyrighted and, if writing malware was clearly legal, its licensing model enforced. The Ice IXs of the malware world could be considered ZeuS infringers, depending on how much of them was actually based on the leaked version of ZeuS. With a valid copyright, ZeuS would have to show that Ice IX had access to ZeuS's copyrighted material and that there was substantial similarity between the copyrighted work and the alleged infringing work.[156]

Before merging with ZeuS, SpyEye may have been an infringer depending on how Gribodemon developed his "ZeuS Killer." Like Connectix, SpyEye pre-merger was a competitor.[157] Unlike Connectix though, SpyEye was not trying to simply play in the same space—it sought out ZeuS installations on a victim machine and then removed it.[158] The nature and use factor as well as the market impact may weigh heavily against SpyEye. Cybersecurity researchers, who reverse engineer malware as a means to protect users from harm, could be seen under a fair use light. Like SpyEye, Cybersecurity researchers would likely be seeking to impact the malware market by completely removing the rootkit. But like Connectix, researchers could also be viewed as competitors discovering how something works and expressing the idea in their own way. That is how remediation of vulnerabilities is discovered. Cybersecurity researchers would analyze the underlying idea to build better more secure products. Protecting users from harm should fall under the existing fair use factors as the researchers put the malware work to a completely different use.

154. *How BOINC Works*, *supra* note 94; *see also* Deepanker Verma, *LOIC (Low Orbit Ion Cannon)—DOS Attacking Tool*, INFOSEC INST. (Dec. 20, 2011), http://resources.infosecinstitute.com/loic-dos-attacking-tool/ [http://perma.cc/FR85-FEKU].

155. Hamid Binsalleeh et al., *On the Analysis of the Zeus Botnet Crimeware Toolkit*, *in* 2010 8TH INT'L CONF. PRIVACY, SEC., & TRUST 31, 32 (2010), http://www.qcwireless.net/pst_paper_2010/papers/p31-binsalleeh.pdf [http://perma.cc/F3DN-EEHK].

156. N. Coast Indus. v. Jason Maxwell, Inc., 972 F.2d 1031, 1033–34 (9th Cir. 1992).

157. *See* Krebs, *SpyEye vs. ZeuS Rivalry*, *supra* note 45.

158. *Id.*

IV. Malware Under Patent

Similar to the copyright analysis in Part III, this Part will focus on the ZeuS family of rootkits for patent analysis. For the purposes of the analysis, this Part treats ZeuS as if it were written in the U.S., as international patent concerns are outside the scope of this Comment.

A. *Ordinary Software's Qualifications for Patent Post-*Alice

Patents cover inventions of processes, machines, manufactures, or compositions that are novel and non-obvious to a person with ordinary skill in the invention's particular field.[159] Software in particular can be tricky to patent especially after the recent Supreme Court decision in *Alice Corp. Proprietary Ltd. v. CLS Bank Int'l.*

In *Alice*, the court grappled with how to handle abstract ideas presented in computer software.[160] Similar to copyrights, patent protection does not extend to abstract ideas—the analysis requires a court to separate the abstract from the expression.[161] Accordingly, the court in *Alice* employed an abstraction test for patents that resembles the copyright abstraction test. First, it examines the claim to determine if it is describing the abstract idea. Second, it examines the individual elements of the claim alone and in combination to uncover an inventive concept which could transform the nature of the claim into an eligible patentable subject matter.[162]

The claims at issue in *Alice* concerned using a computer to mitigate settlement risk in financial exchanges, and in previous cases the Court held this type of financial data handling was an abstract idea.[163] What remained was to determine whether there was something in the claim which could transform it by "includ[ing] 'additional features' to ensure 'that the [claim] is more than a drafting effort designed to monopolize the [abstract idea].'"[164] The Court found that simply implementing the idea via a computer was no longer enough to transform an abstract idea into patentable subject matter.[165] Because computers are so ubiquitous, implementation via a computer was reduced down to a conventional element and a means of monopolizing an idea.[166] Instead, the claim needs to show that it makes some improvement over the existing process and cannot be simply a means of organizing human activity—it needs to offer something "more."[167]

159. 35 U.S.C. §§ 101–103 (2012).
160. Alice Corp. Pty., Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347, 2351–60 (2014).
161. *Id.* at 2354.
162. *Id.* at 2355.
163. *Id.* at 2352, 2357.
164. *Id.* (quoting Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1297 (2012)).
165. *Id.* at 2358.
166. *Id.* at 2357.
167. *Id.*

In a memorandum to patent examiners, the United States Patent and Trademark Office ("USPTO") further clarified the standard.[168] The USPTO was careful to say that the *Alice* decision did not create new standards for software but that it expanded the abstract idea test to computer-implemented claims.[169] Software, however, is a computer-implemented process; that means software needs to go beyond merely organizing human activity.[170] While some in the patent field feared this was the death knell for software patents, cases since *Alice* have further defined how software can distinguish itself from an abstract idea.[171]

Of these cases examining software patenting, *DDR Holdings, LLC v. Hotels.com, LP*, is notable as it finally finds a piece of software patentable.[172] DDR Holdings was the assignee of two patents that described systems and methods that sought to keep visitors on the original website after the visitors clicked a third party advertisement.[173] Normally, clicking a third party advertisement would take the visitor away to the third party's site.[174] In *DDR Holdings*, the described system would host the third party merchant's information but display it in the host website with the host website's look and feel.[175] The claims in question, rather than citing the use of a generic computer or use of the Internet, "specif[ied] how interactions with the Internet [were] manipulated to yield a desired result."[176] The Court cautioned that patents solving Internet-centric problems were not all eligible for patents.[177] The key, according to the Court, was that "the claims recite[d] an invention that [was] not merely the routine or conventional use of the Internet."[178]

After subject matter, another concern for software may be the obviousness requirement. Courts test for obviousness using four factors: "(1) the scope and content of the prior art; (2) the level of ordinary skill in the art; (3) the differences between the prior art and the claimed invention; and (4) the extent of any objective indicia of nonobviousness."[179] In a recent example of the obviousness test, the Fed-

168. Memorandum from Andrew Hirshfeld, Deputy Comm'r for Patent Examination Policy, United States Patent and Trademark Office, to Patent Examining Corps (June 25, 2014) (on file with author).

169. *Id.* at 1.

170. *Id.* at 3.

171. Bart Eppenauer, *DDR Holdings—Federal Circuit Forges a Sensible Path on Software Patents*, PATENTLY-O (Dec. 14, 2014), http://patentlyo.com/patent/2014/12/holdings-sensible-software.html [http://perma.cc/TKJ4-JS9A].

172. DDR Holdings, LLC v. Hotels.com, LP, 773 F.3d 1245, 1259 (Fed. Cir. 2014).

173. *See id.* at 1248.

174. *Id.*

175. *Id.* at 1248–49.

176. *Id.* at 1258.

177. *Id.*

178. *Id.* at 1259.

179. Crocs, Inc. v. Int'l Trade Comm'n, 598 F.3d 1294, 1308 (Fed. Cir. 2010).

eral Circuit Court of Appeals examined the Crocs shoe patent to determine whether it was obvious at the time of invention.[180]

The court used two pieces of *prior art* (a term used to describe previous disclosures such as other patents, published papers, or other things in the public domain[181]) describing other shoes that were similar to Crocs.[182] The claim at issue described a foam strap used to keep the shoe on that both of the previous patents had described as inferior and unsuitable.[183] The court resolved the first three factors quickly—a practitioner with ordinary skill in the art would avoid using the foam straps because previous inventions had shown how inferior they were to other designs, while this patent showed an effective way to use the straps.[184] But as the court pointed out, something does not become patentable because the prior art labeled it inferior.[185] The claim resulted in something that the ordinary art did not predict.[186] The sheer commercial success of Crocs was a clear indicia of non-obviousness and so great that not much other secondary consideration was needed.[187] All four weighed together pushed the Crocs patent beyond some obvious improvement.[188]

B. *Ordinary Software's Rights & Limitations Under a Patent*

In some ways, the rights and limitations for patented software are more straightforward than copyrighted software. Once software gains a patent, the inventor or patent assignee hold "negative" rights similar to copyrights. The holder of the patent has the right to exclude others "from making, using, . . . or selling the invention throughout the United States."[189] This is commonly referred to as a patent monopoly,[190] and it generally lasts for twenty years.[191] The trade-off is that at the end of those twenty years, the invention goes into the public domain and can be used by anyone.[192]

180. *Id.* at 1297.
181. 35 U.S.C. § 102 (2012).
182. *Crocs*, 598 F.3d at 1308.
183. *Id.*
184. *Id.* at 1309–10.
185. *Id.* at 1308.
186. *Id.* at 1310.
187. *Id.* at 1310–11.
188. *Id.* at 1311.
189. 35 U.S.C. § 154 (2012).
190. *See* United States v. Gen. Elec. Co., 272 U.S. 476, 485 (1926) ("But under the patent law the patentee is given by statute a monopoly of making, using and selling the patented article. The extent of his monopoly in the articles sold and in the territory of the United States where sold is not limited in the grant of his patent . . . .").
191. § 154(a)(2).
192. Warden v. City of St. Louis, Mo., 140 F.2d 615, 617 (8th Cir. 1944) ("It must necessarily follow that when that monopoly ceases, the invention, for any and all purposes for which it is adapted, becomes public property and can be used by any one.").

An invention's claims limit the scope of a patent.[193] Unless the patentee creates his or her own definitions in the patent, "the words of a claim are generally given their ordinary and customary meaning as understood by a person of ordinary skill in the art when read in the context of the specification and prosecution history."[194] For example, one of the terms at issue in *In re Papst* was the phrase "virtual files."[195] The patent in question had described a host-native driver that could obtain access to data that was not actually on a device for which the drive was designed.[196] Papst and the camera manufacturers disagreed on where the data in the virtual files was coming from.[197] Papst claimed the data could be derived from the interface device, while the camera manufacturers wanted to limit it to the data device connected to the host.[198] Looking at the claim construction, the court agreed with Papst, stating that nothing in the claims or description had limited the virtual file to data stored on the interface device.[199] The word "virtual" was undefined in the patent itself, but the court looked to how "virtual" had been used in the computer field to come to their conclusion.[200] Defining the term was key to determining whether the computer manufacturers had infringed on Papst's patents.[201] Infringement happens when the accused device contains every claim limitation or its equivalent.[202] In order to make that determination, a clear understanding of what the claim means is needed in the first place.[203]

C. *Patent Rule Application to the ZeuS Family*

It is hard to pinpoint the first rootkit. Rather, it is easier to pinpoint the first time a rootkit was expanded to a different feature—the first BIOS rootkit in 2011,[204] the first industrial espionage rootkit in 2010,[205] and the first Mac rootkit in 2009.[206] Rootkits are still very much a novel field of innovation and highly complex pieces of

193. Thorner v. Sony Comput. Entm't Am., LLC, 669 F.3d 1362, 1367 (Fed. Cir. 2012) ("It is the claims that define the metes and bounds of the patentee's invention.").
194. *Id.* at 1365.
195. *In re* Papst Licensing Dig. Camera Patent Litig., No. 2014-1110, 778 F.3d 1255, 1267 (Fed. Cir. 2015).
196. *Id.* at 1268.
197. *Id.*
198. *Id.*
199. *Id.*
200. *Id.*
201. *Id.*
202. *Id.* at 1261.
203. *Id.*
204. Marco Guiliani, *Mebromi: The First BIOS Rootkit in the Wild*, WEBROOT (Sept. 13, 2011), http://www.webroot.com/blog/2011/09/13/mebromi-the-first-bios-rootkit-in-the-wild/ [http://perma.cc/NZE7-5JLV].
205. Nicolas Falliere, *Stuxnet Introduces the First Known Rootkit for Industrial Control Systems*, SYMANTEC (Aug. 6, 2010), http://www.symantec.com/connect/blogs/stuxnet-introduces-first-known-rootkit-scada-devices [http://perma.cc/T3V5-9GF8].

software. Rootkits manipulate the computers they run on (whether generic or special) in order to conceal their presence, maintain their hold, and transfer their payload.[207] Similar to the DDR Holdings patents, claims for a rootkit would involve situations unique to the computer. A rootkit patent could describe manipulating features of the computer to achieve its desired results such as injecting itself in the existing kernel to intercept processes or overwriting with a new kernel mode driver. ZeuS, however, would have been a barrier to future rootkits gaining a patent. The success and wide availability of the ZeuS family of rootkits creates extensive prior art. ZeuS pioneered a means of modularizing the rootkit and building it into a malware kit that anyone could use to create one. While ZeuS may have been patentable, a future patent for Ice IX would have been very narrow in scope if patentable at all.

A botnet may actually fall into *Alice*'s trap of a computer becoming something long prevalent and traditional. Botnets have been around since the first ill-fated Cornell experiment (dubbed the "Morris Worm") in 1988.[208] Computer science student Robert Morris wanted to create "a network of thousands of computers coordinating with one another and available to carry out further instructions at the direction of their master."[209] This is what we would later call a *botnet*. Since 1988, botnets have become increasingly common and all function in a similar way—zombie hosts communicate with a C&C awaiting instructions for where to put their computing power to use. Botnets would certainly pose a problem with anticipation via prior art (consider, for example, open source projects such as BOINC, whose code is freely downloadable).

While the botnet portion of ZeuS may not be patentable post-*Alice*, the rootkit portion could be. In a patent, the art is disclosed in exchange for the patent-holder's monopoly. The fact that Ice IX built its rootkit largely modeled on the ZeuS code base after it was leaked would not be the major issue. Ice IX instead would need to worry about infringement based on whether its product practices all of ZeuS's claims or some equivalent. SpyEye might not be an infringer. Slavik transferred the code to Gribodemon. While Gribodemon's original invention removed ZeuS, it was not clear that it was originally based on the ZeuS codebase.

206. *See* Thomas Claburn, *Black Hat: Mac OS X Rootkit Debuts*, INFO. WEEK (July 30, 2009, 4:59 PM), http://www.darkreading.com/vulnerabilities-and-threats/black-hat-mac-os-x-rootkit-debuts/d/d-id/1081846 [http://perma.cc/7SJV-BQE7].

207. BLUNDEN, *supra* note 29, at 49.

208. Timothy B. Lee, *How a Grad Student Trying to Build the First Botnet Brought the Internet to Its Knees*, WASH. POST (Nov. 1, 2013), http://www.washingtonpost.com/blogs/the-switch/wp/2013/11/01/how-a-grad-student-trying-to-build-the-first-botnet-brought-the-internet-to-its-knees/ [http://perma.cc/2MXM-4HNJ].

209. *Id.*

V. Conclusion: Who and What are we Protecting?

The ZeuS family of rootkits arguably has copyrightable and patentable elements. But the question remains as to why we would even want to extend intellectual property protection to malware—after all, who and what are we protecting? The discussion will continue to assume that the rootkits were developed in the U.S. as international issues exceed the scope of this Comment.

A. *Protecting Innovation and Creation*

The overriding purpose of patent law is to protect innovation, while copyright law seeks to protect creation.[210] The rights of our innovators and creators are so core to American culture and economy that they can be found rooted soundly in the U.S. Constitution.[211] Malware, despite its dubious nature, has steadily evolved in its technology[212] and has provided valuable research and development. Consider again Creeper and Reaper, the first self-replicating programs and the precursors to the modern-day virus.[213]

While self-replication is an element of identifying viruses, self-replication also has other uses.[214] Bob Thomas at BBN pioneered a technology later used in a wide range of modern applications. For example, self-replication is key in biocomputing and bioinformatics.[215] Self-replication in programming code can be used to mimic the chemical processes in DNA and to map artificial chemical computing models.[216]

Computer scientists have also proposed "good" uses for malware. John Aycock and Alana Maurushat proposed what they dubbed the "Human Rights Worm" to test the extent of Internet censorship in other countries without putting human rights activists at risk.[217] The worm was not created and was instead more of a thought experiment in the issues that "good" malware might face.[218] The Human Rights

210. *See* U.S. Const. art. I, § 8, cl. 8.

211. *See id.*; *see also Statement by President Barack Obama Upon Signing H.R. 1249*, 2011 U.S.C.C.A.N. S6, S6 ("Here in America, our creativity has always set us apart, and in order to continue to grow our economy, we need to encourage that spirit wherever we find it.").

212. *See* Adam Kujawa, *Criminal Innovation in Malware Leaves Antivirus Industry Flagging*, Wired (Oct. 2, 2012), http://www.wired.co.uk/news/archive/2012-10/02/malware-evolution [http://perma.cc/M4Z7-TRS3].

213. Schantz, *supra* note 1, at 74; *First Computer Virus, supra* note 3; Shoch & Hupp, *supra* note 4, at 179.

214. *What Is a Computer Virus?*, *supra* note 5.

215. Lidia Yamamoto et al., *Self-Replicating and Self-Modifying Programs in Fraglets*, 2007 Proceedings Bio-Inspired Models Network, Info., & Computing Systems 159, 159.

216. *Id.* at 160.

217. John Aycock & Alana Maurushat, *"Good" Worms and Human Rights*, 38 ACM SIGCAS Computers & Soc'y 28, 30–31 (2008).

218. *Id.* at 31.

Worm has all of the indicia of malware based on our previous definitions.[219] However, the Human Rights Worm also has the marks of something that could have been patentable when it was proposed in 2008. The invention was certainly useful and novel—Aycock and Maurushat proposed using malware for testing censorship.[220] While the software took an existing abstract idea of the malicious worm, claims could have been constructed around the improvements they suggested that showed an inventive concept beyond the average worm.[221] Aycock and Maurushat proposed malware that improved on the existing worm, malicious or otherwise.[222]

The Human Rights Worm is only one example of the innovation outside the criminal ZeuS family of rootkits. The Welchia worm came in the early 2000s seeking to spread across the Internet and patch the vulnerabilities it found.[223] In the end, Welchia was not successful and caused other problems.[224] But it too had an inventive concept that rose above whatever prior art may have existed for Internet worms.[225] Both the Human Rights Worm and Welchia should have been copyrightable as well. Each of these worms embodies different expressions of the same idea—malware that spreads for the good of the Internet.

B. *Should Illegality or Immorality Matter?*

In intellectual property, trademark law is notoriously able to deny a mark based on a scandalous or immoral nature.[226] For a period of time, there was a morality framework in copyrights and patents. In copyrights, the federal courts let go of the morality bar first in the Fifth Circuit in 1979[227] and in the Ninth Circuit in 1982.[228] In *Mitchell Bros. Film Group v. Cinema Adult Theater*, the court determined that Congress had purposefully extended copyright protection to all creative works regardless of subject matter.[229] As a practical matter, the *Mitchell Bros*. court also did not see how an obscenity or immorality standard could work in copyrights long term, since society's views change and would also effectively "fragment copyright enforcement,

219. *See id.*; *What Is a Computer Virus?*, *supra* note 5.

220. Aycock & Maurushat, *supra* note 217, at 30–31.

221. *Id.* at 31, 33–34.

222. *Id.* at 37.

223. *See* GENE BRANSFIELD, THE WELCHIA WORM 1, 3 (GCIH Practical Series Vol. 3, 2003), http://www.giac.org/paper/gcih/517/welchia-worm/105720 [http://perma.cc/W3CR-NB6V].

224. *See id.* at 27.

225. *See id.* at 3–23.

226. 15 U.S.C. § 1052(a) (2012).

227. Mitchell Bros. Film Grp. v. Cinema Adult Theater, 604 F.2d 852, 861–62 (5th Cir. 1979)

228. Jartech, Inc. v. Clancy, 666 F.2d 403 (9th Cir. 1982).

229. *See Mitchell Bros.*, 604 F.2d at 855.

protecting registered materials in a certain community, while, in effect, authorizing pirating in another locale."[230]

Patents also once had a "moral utility doctrine" that was effectively overruled due to society's changing views.[231] The moral utility doctrine came from Justice Story's formulation of *useful*: something that could not conflict with the sound morals of society.[232] Today, most would say morality has no place in American patent law, though its specter—and Justice Story's interpretation of it—appears in narrow applications such as patenting genetics.[233]

Some scholarship indicates that the beginnings of a morality framework are at least appearing in patents. Since the early 2000s, patents in biotechnology have been increasingly controversial.[234] While international patents were out of scope for this paper, it is important to note that there are bars internationally to patents involving human genetics.[235] During the Supreme Court's review of the patenting of the BRCA1 and BRCA2 genes in *Ass'n for Molecular Pathology v. Myriad Genetics, Inc.*, the parties made arguments for and against patenting that closely resemble this Comment's propositions.[236] On the one hand, proponents wanted to protect innovation.[237] On the other, opponents pointed not only to the law of nature bar in U.S. patent law[238] but also to something intrinsically immoral in patenting human genetics.[239] That same intrinsic implication of something immoral permeates the world of malware. Ultimately, the Supreme Court precluded patenting of naturally occurring DNA by applying the law of nature bar; conversely, synthetically created DNA could be patented.[240] The Court did not rely on morality judgment—only on an application of patent law. The same should be true of malware.

Another argument against patenting human genes also applies to malware: opponents argued that patents could cause monopolies that inhibit access to medication and treatment.[241] This is analogous to

230. *Jartech*, 666 F.2d at 406.

231. *See* Juicy Whip, Inc. v. Orange Bang, Inc., 185 F.3d 1364, 1366–67 (Fed. Cir. 1999).

232. Benjamin D. Enerson, Note, *Protecting Society from Patently Offensive Inventions: The Risk of Reviving the Moral Utility Doctrine*, 89 CORNELL L. REV. 685, 690 (2004) (citing Lowell v. Lewis, 15 F. Cas. 1018, 1019 (C.C. Mass. 1817) (No. 8568)).

233. *Id.* at 692–94.

234. Darryl R. J. Macer, *Inventions, Patents, and Morality*, in BIOTECHNOLOGY, ENCYCLOPEDIA LIFE SUPPORT SYSTEMS (Horst W. Doelle et al. eds., 2001).

235. *Id.*

236. *See* T.C., *Why Are Gene Patents Controversial?*, ECONOMIST (Apr. 18, 2013, 11:50 PM), http://www.economist.com/blogs/economist-explains/2013/04/economist-explains-why-gene-patents-controversial [http://perma.cc/R4N7-JFFH] (discussing Ass'n for Molecular Pathology v. Myriad Genetics, Inc., 133 S. Ct. 2107 (2013)).

237. *Id.*

238. *See* 35 U.S.C. § 101 (2012); Le Roy v. Tatham, 63 U.S. 132 (1859).

239. T.C., *supra* note 236.

240. *Myriad*, 133 S. Ct. at 2116, 2119–20.

241. *See* Macer, *supra* note 234.

malware, in that patents could cause monopolies on security vulnerabilities and limit how the cybersecurity industry protects against these issues. This outcome is unlikely if patent law treats security vulnerability as an abstract idea. How the vulnerability is applied should be what shows the "something more" required for post-*Alice* software patents.

CFAA penalizes access and conspiracy to access without authorization[242] which can be achieved through using malware. It does not, and should not, penalize authoring malware. If writing malware is wrong, then it should be clear that it is wrong. But as it stands today, malware is innovation and is protectable under patent just as it is a creative work protectable under copyright.

C. *White Hats vs. Black Hats*

As previously discussed, CFAA appears to have a chilling effect on both the researchers (the White Hats) and the criminals (the Black Hats).[243] Whether or not some action falls under CFAA does not take into account the author's intent and certainly pays no heed to whether the malware had intellectual property rights.[244] This Comment does not propose altering CFAA in a manner that would give deference to those rights. It does propose continuing to allow intellectual property rights to protect the work.

Cybercriminals are unlikely to seek protection for their work. Like the Slashdot user joked, such criminal activity is more likely to use an equally criminal means of enforcement.[245] Furthermore, a patent would disclose the criminal's method of operation. The criminal activity of unauthorized access to computing resources should be punished.[246] But the law should be able to recognize that *writing* malware is not the same as *using* malware and that the innovation and creations from both White and Black Hats still deserve protection. Punish the person who uses malware for unauthorized access to a system—but protect the work as an addition to our body of technological innovation.

242. 18 U.S.C. § 1030(a)–(b) (2012).
243. *See supra* Section II.C.
244. *See* § 1030(a)–(b).
245. Kdawson, *supra* note 18.
246. *See* § 1030(c).