UIC John Marshall Journal of Information Technology & Privacy Law

Volume 28
Issue 4 *Journal of Computer & Information Law - Summer 2011*

Article 1

Summer 2011

Bugs for Sale: Legal and Ethical Proprieties of the Market in Software Vulnerabilities, 28 J. Marshall J. Computer & Info. L. 451 (2011)

Taiwo A. Oriola

Recommended Citation

Taiwo A. Oriola, Bugs for Sale: Legal and Ethical Proprieties of the Market in Software Vulnerabilities, 28 J. Marshall J. Computer & Info. L. 451 (2011)

ARTICLES

BUGS FOR SALE: LEGAL AND ETHICAL PROPRIETIES OF THE MARKET IN SOFTWARE VULNERABILITIES

TAIWO A. ORIOLA*

INTRODUCTION

In general terms, software is the programs on which a computer system is run.[1] However, in specific terms, there are two notional definitions of software: the first conceptualizes software as the intangible functional components of a computer, inclusive of computer programs and data intended to be processed by the programs.[2] The second definition clearly excludes data, and conceives software as "a list of commands and instructions for data-processing."[3] Structurally, software typically exists in two forms: the source code and the machine readable binary code.[4] The source code is the program's logical structure, comprising the commands and instructions written in a specific programming language that is intelligible and accessible.[5] The source code then morphs into a machine readable-only form known as the binary or object code to facilitate program execution by the computer.[6] Software could also be de-

[*] The School of Law, University of Ulster, Northland Road, Londonderry, United Kingdom

[1] *See* LESLEY GOURLAY, CHAMBERS GUIDE TO ENGLISH FOR IT AND THE INTERNET 5 (2000).

[2] *See* Sebastian von Engelhardt, *The Economic Properties of Software*, 2008-045 JENA ECON. RES. PAPERS 1, *available at* http://zs.thulb.uni-jena.de/servlets/MCRFileNodeServlet/jportal_derivate_00119979/wp_2008_045.pdf.

[3] *See id.*

[4] *See id.*

[5] *See id.*

[6] Object code is not directly intelligible and must be converted by disassembly before it can be understood by humans. *See* DAVID I. BAINBRIDGE, INTELLECTUAL PROPERTY vii (8th ed. 2010).

scribed in proprietary and non-proprietary terms.[7] For example, open-source software ("OSS") otherwise known as 'free-ware' or 'share-ware' is non-proprietary,[8] whilst commercial software is proprietary, and is typically disseminated via licenses,[9] or made to order as bespoke software.[10] The non-proprietary Linux operating system remains the putative open-source software, whilst the ubiquitous Microsoft Windows, with its currently estimated ninety per cent total market share of client operating systems on the Internet,[11] is the quintessential commercial or proprietary software.[12] Proprietary software is subject of intellectual property rights, and is both patent-eligible[13] and copyrightable.[14]

[7] *See generally* Sapnar Kumar, *Enforcing the GNU GPL*, U. ILL. J.L. TECH. & POL'Y 1, 3 (2006).

[8] Although the non-proprietary open-source software is generally free, according to Kumar, the increasing use of non-proprietary software in proprietary applications forced the advent of "copyleft" license, or the "GNU" "General Public License" designed to protect software users rather than software owners, whose interests are well secured via intellectual property and licensing agreements. *See id.* at 10-11.

[9] These are generally known as software licenses, and end users are obliged to acquiesce to terms of use, which could be enforceable as contractual terms. *See generally* ProCD, Inc. v. Zeidenberg, 86 F.3d 1447 (7th Cir. 1996), (upholding the validity of the terms of a 'shrink-wrap license' for non-commercial use, which Zeidenberg had acquiesced to, prior to downloading the software in question unto his computer. However, in clear breach of the terms of his license, Zeidenberg resold the information on the CD-ROM database software to third-parties. It was held that he had breached the terms of his license and was in breach of contract). *See also* Vernor v. Autodesk, Inc., 621 F.3d 1102, 1111 (9th Cir. 2010), where the Ninth Circuit held, *inter alia*, that ". . .a software user is a licensee rather than an owner of a copy where the copyright owner (1) specifies that the user is granted a license; (2) significantly restricts the user's ability to transfer the software; and (3) imposes notable use restrictions."

[10] According to Newton, the terms of system supply contracts covering software would typically deal with intellectual property rights issues ranging from copyright, database right, to confidential business information. *See generally* Jeremy Newton, *System Supply Contracts, in* COMPUTER LAW: THE LAW AND REGULATION OF INFORMATION TECHNOLOGY 3, 31-33 (Chris Reed & John Angel eds., 6th ed. 2007).

[11] Apple's *Mac* is the closest rival, with approximately 5.19% of the total market share. *See Operating System Market Share*, NETMARKETSHARE (February 2011), http://www.netmarketshare.com/report/aspx?qprid=8&qptimeframe=M&qpsp=145.

[12] *Windows 7 Passes 20% Global Usage Share*, NETMARKETSHARE (April 2010-February 2011), http://www.netmarketshare.com/operating-system-marketshare.aspx?qprid=11&qpcustom=Windows+7&sample=44 ("Windows 7 passed 20% global usage share in December . . .").

[13] Patent Act, 35 U.S.C. § 101 (1952). The United States pioneered software patents and recognizes software as a patent-eligible invention if it meets the requirements of 35 U.S.C. § 101 Patent Act 1952, providing that "[w]hoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title."

[14] In the United States, for example, software or computer program is defined as ". . .a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." *See id.* The same section defines "literary works" as

Although essentially intangible, software is the quintessential informational good,[15] and the fulcrum anchoring the control systems that undergird critical infrastructures, ranging from fuel pipe lines, nuclear plants, electricity grids, mobile telecommunications, personal and industrial computers, to air-traffic control systems.[16] Consequently, the stakes

". . .works. . .expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied." *Id.* Thus, a combined reading of the two provisions above would place software or computer programs as "literary works." *Id.*

[15] There is a recurring debate on whether software is a good or service. In the United States, this discourse invariably takes into cognizance the nature of the particular software in question. Thus, if the software is embodied in tangible media or packaging such as hard disks or magnetic tapes, most courts would have no hesitation in categorizing the software as a good and properly within the ambit of Article 2 of the Uniform Commercial Code (UCC). *See, e.g.,* Advent Systems Ltd. v. Unisys Corp., 925 F.2d 670, 675 (3d Cir. 1991), where the court opined that if software was in a fixed medium, it became a tangible product and recognizable as a good under the UCC. In the same vein, if software was bundled together with hardware in a commercial transaction, it would be deemed as a contract in goods, and subject to the UCC. *See generally* Carl Beasley Ford, Inc. v. Burroughs Corp., 361 F. Supp. 325, 327 (E.D. Pa. 1973), *aff'd*, 493 F.2d 1400 (3d Cir. 1974). However, courts are still split on whether unbundled or stand-alone software would qualify as a good due to its dominant service nature. On the one hand are judicial opinions, such as those from the United States District Court for the Eastern District of New York, which held that a custom or bespoke software, albeit an intangible product, was nevertheless ". . . more readily characterized as 'goods' than 'services'." *See generally* Triangle Underwriters, Inc. v. Honeywell, Inc., 457 F. Supp. 765, 769 (E.D.N.Y. 1978), *modified on other grounds*, 604 F. 2d 737 (2d Cir. 1979). See generally Ditto, *I.Lan Sys., Inc. v. Netscout Serv. Level Corp.*, 183 F. Supp. 2d 328, 332 (D. Mass. 2002), where the court held that, although technically, UCC Article 2 did not apply to software licenses, but "for the time being, the court will assume that it does." *See also Surplus.com, Inc. v. Oracle Corp.*, No. 10 CV 03510, 2010 WL 5419075 (N.D. Ill. Dec. 23, 2010), where the court held that a software development agreement geared toward the acquisition of software was governed by the UCC. However, the court predicated the outcome on the agreement of the parties in *Cinetic DyAG Corp. v. Forte Automation Sys., Inc.*, No. 2:08-cv-11790, 2008 WL 4858005 (E.D. Mich. Nov. 6, 2008), where the court held that the alleged contract between the parties contemplated the provision of services, and as a result, Article 2 of the UCC was inapplicable. *But see Digital Ally, Inc. v. Z3 Tech., LLC*, No. 09-2292-KGS, 2010 WL 3974674 (D. Kan. Sep. 30, 2010) (holding that a pure software license agreement did not involve transfer of title, and consequently, was not a sales of goods for the purposes of Article 2 of U.C.C.). *See* Lorrin Brenan, *Symposium on Approaching E-Commerce Through Uniform Legislation: Understanding the Uniform Computer Information Transactions Act and the Uniform Electronic Transactions Act: Why Article 2 Cannot Apply to Software Transactions*, 38 DUQ. L. REV. 459 (2000) (arguing that software was not a good by any means); *see also* Andrew Rodau, *Computer Software: Does Article 2 of the Uniform Commercial Code Apply?*, 35 EMORY L.J. 853 (1986) (noting that there was no reason why UCC Article 2 should not be extended to software licenses having been extended to leases).

[16] Defective software could lead to the failures of the control systems of critical infrastructures with potentially catastrophic consequences. *See, e.g.*, Daily Mail Reporter, *That's Not in the Plan! Computer Errors Leaves Astronaut Dangling over Ledge of Space Station 220 Miles Above Earth*, MAIL ONLINE (Mar. 1, 2011, 6:22 PM), http://www.

are high to keep software secure due to the high propensity for unscrupulous exploitation of latent vulnerabilities by malicious hackers.[17] This is underscored by the 2010 Symantec Corporation's global survey, which showed that half of critical information infrastructure providers experienced politically motivated cyber attacks in 2010.[18] Some of the attacks were spear-headed by a shadowy group of hackers self-styling as Anonymous, which specifically targeted businesses such as PayPal and Visa, who had allegedly declined to do business with Wikileaks, following the latter's revelations and subsequent publication of politically sensitive Unite States diplomatic cables on various governments from around the world.[19] In similar attacks in April 2011, hackers raided Sony Corporation's ("Sony") online PlayStation gaming network and stole personal data, including credit card details of an estimated seventy-seven million customers,[20] an unprecedented feat that has been ranked ". . .among the

dailymail.co.uk.sciencetech/article-1361752/Computer-error-leaves-astronaut-Stephen-Bown-dangling-ISS-ledge-220-miles-Earth.html (describing an incident wherein, in February 2011, during a routine spacewalk, Stephen Bowen, an astronaut, was left stuck-up two-hundred twenty miles above earth, outside of the International Space Station, when computer errors caused the fifty-eight foot robotic arm he was walking on to stall for nearly half an hour). Computer errors could also be deliberately instigated by malicious hackers targeting software vulnerabilities. *See generally* Jaziar Randianti & Jose J. Gonzalez, *Understanding Hidden Information Security Threats: The Vulnerability Black Market*, HICSS'07, 40TH HAWAII INT'L CONF. ON SYS. SCIENCES 1 (2007), *available at* http://www.computer.org/portal/web/csdl/doi?doc=doi/10.1109/HICSS.2007.583 (noting that the risk of cyber attack to business included failure of control systems, with consequences ranging from "injuries to life, loss of production, environmental damage, damage to reputation [to] loss of licence to operate").

[17] *See* Helen Nissenbaum, *Hackers and the Contested Ontology of Cyberspace*, 6(2) NEW MEDIA & SOC'Y 195, 198-99 (2004), *available at* http://www.nyu.edu/projects/nissenbaum/papers/hackers.pdf (noting that hackers are conventionally associated with dangerous individuals who attack information systems, violate communication networks, spread viruses, and generally propagate chaos in cyberspace). *See also* Aaron J. Burstein, *Amending the ECPA to Enable a Culture of Cybersecurity Research*, 22 HARV. J.L. & TECH. 167, 168-69 (2008) (noting that hackers' attacks on the network systems and information infrastructures are increasing in sophistication).

[18] *See* Symantec, *Symantec 2010 Critical Infrastructure Protection Study: Global Results* (2010), *available at* http://www.symantec.com/content/en/us/about/presskits/Symantec_2010_CIP_Study_Global_Data.pdf.

[19] *See* Ashley Frantz & Atika Shubert, *Wikileaks 'Anonymous' Hackers: 'We will fight'*, CNN U.S. (Dec. 9, 2010), http://articles.cnn.com/2010-12-09/us/hackers.wikileaks_1_julian-assange-arbor-networks-websites?_s=PM:US (reporting that the Anonymous hackers dubbed the attacks, which temporarily shut down PayPal, Visa, and others, as "Operation Payback," that was a protest "against all things people were unable to change using legal means." The group, which had hitherto focused on attacking anti-digital piracy measures felt they had to come to the aid of Wikileaks with whom they believed they shared "the common idea of free information.").

[20] *See* Nick Wingfield et al., *Hacker Raids Sony Videogame Network*, WALL ST. J., (Apr. 27, 2011), http://online.wsj.com/article/SB10001424052748703778104576287362503776534.html.

biggest data thefts of all time."[21] While Sony's compromised gaming network has not been publicly tied to any known or specific vulnerability,[22] software vulnerabilities reputedly account for most of the reported computer or network security problems,[23] and cyber attacks are routinely characterized as "attempts to exploit vulnerabilities in hardware and software."[24]

Software vulnerabilities have been defined as inherent errors or mistakes in the design, specification and programming of software.[25] That programming mistakes are at the core of vulnerabilities is exemplified by Andy Ozment who characterized software vulnerability as "an instance of a mistake in the specification, development, or configuration of software such that its execution can violate the explicit or implicit security policy."[26] Robert A. Martin expatiated on the nature of programming mistakes or errors that could precipitate software bugs or vulnerabilities as follows:

> Programmers know that they make mistakes when writing software, including typos, math errors, incomplete logic, or incorrect use of functions or commands. Sometimes mistakes occur even earlier in the development process, reflecting an oversight in the requirements guiding the design and coding of a particular function or a software program's capability. Mistakes that have security implications become *vulnerabilities*, which hackers can use directly to access protected data, and exposures,

[21] *See Sony: PlayStation Hack: Top Five Data Thefts*, THE TELEGRAPH (Apr. 27, 2011, 2:11 PM), http://www.telegraph.co.uk/technology/sony/8476757/PlayStation-hack-top-five-data-thefts.html.

[22] High value corporations are known to be notoriously reluctant to openly disclose incidents of cyber attacks or report a compromised system mainly for business expediencies. *See, e.g.,* Wingfield, *supra* note 20. Sony Corporation was pilloried for reputedly delaying informing its customers that the hacking of its network was responsible for its online absence. Sony had initially announced that it had shut down the PlayStation network on its own, rather than disclosing that its network had been compromised and shut down by hackers. *See also* William A. Arbaugh, et al., *Windows Vulnerability: A Case Study Analysis*, 33 COMPUTER 52 (Dec. 2000), *available at* http://www.cs.umd.edu/~waa/pubs/Windows_of_Vulnerability.pdf (noting that most organizations are wary of disclosing compromised systems for fear it could adversely affect their business).

[23] *See* BRUCE SCHNEIER, SECRET & LIES: DIGITAL SECURITY IN A NETWORKED WORLD 205 (2004) [hereinafter SECRET & LIES] (noting that most computer security problems emanated from faulty code).

[24] *See* Symantec, *Symantec Internet Security Report* [Trends for July - December 2005], vol. IX, 24 (Mar. 2006).

[25] *See* BRUCE SCHNEIER, SCHNEIER ON SECURITY, 260 (2008) [hereinafter SCHNEIER ON SECURITY]; *See* Rainer Bohme, *Vulnerability Markets: What is the Economic Value of a Zero-Day Exploit?* PROCEEDINGS OF 22C3 1, (Berlin, Germany, Dec. 27-30, 2005), *available at* http://events.ccc.de/congress/2005/fahrplan/attachments/542-Boehme2005_22C3_VulnerabilityMarkets.pdf.

[26] *See* Andy Ozment, Vulnerability Discovery & Software Security 18 (Aug. 31, 2007) (unpublished Ph.D. dissertation, University of Cambridge), *available at* http://www.andyozment.com/papers/ozment_dissertation-print_version.pdf.

> which provide information or capabilities that can function as stepping-stones to direct access.[27]

Thus, if programming mistakes were imminently inevitable, then arguably, it would be infeasible to write a bug-free code and software architecture would be inherently flawed and violable.[28] Notably, the growing antivirus software market was a direct response to the emergence and proliferation of computer malware, viruses, worms and sundry rogue programs, which are preferred tools of choice for malicious hackers in the hunt for and exploitation of "zero-day"' software bugs.[29] *A fortiori*, software vulnerabilities would logically appear assured and inevitable, hence the continuing relevance of software security industry and the burgeoning global software security market,[30] which is exemplified by the European network and information security market that ranks second largest in the world after the United States', with an estimated global value of 10.7 billion Euros in 2007, and a projected estimated global value of 15.5 billion Euros for 2010.[31] Significantly, the global relevance of software security solutions and market would appear inexorably set in a spiral trajectory, given that on average, fourteen new software vulnerabilities are published per day in the National Vulnerability Database ("NVD") by the United States' National Institute of Standard and Technology ("NIST").[32]

However, the budding literature on the economics of software security directly linked software vulnerabilities exclusively to market failure, whilst largely glossing over the influences of inherent software technical

[27] *See* Robert A. Martin, *Managing Vulnerabilities in Networked Systems*, 34 COMPUTER 32 (Nov. 2001), *available at* http://cve.mitre.org/docs/docs-2001/CVEarticle IEEEcomputer.pdf.

[28] *See* SECRET & LIES, *supra* note 23 (noting that it was hard to design and implement bug-free code).

[29] *See* Meiring de Villiers, *Computer Viruses and Civil Liability: A Conceptual Framework*, 40 TORT TRIAL & INS. PRACTICE L.J. 123, 160 (2004), *available at* http://www.law.unsw.edu.au/sites/law.unsw.edu.au/files/pre/f/docs/pubs/unsw_mdevilliers_virus-ii.pdf [hereinafter *Computer Viruses and Civil Liability*] (noting that vulnerabilities facilitated virus attacks, the proliferation of which precipitated the ever growing market in antivirus solutions).

[30] *See* Stephen Flowers, *Harnessing the Hackers: The Emergence and Exploitation of Outlaw Innovation*, 37 RES. POL'Y 177, 181 (noting that the rise of the computer security industry catering to individual, corporate, and government security solutions, was a direct response to the relentless attacks of hackers on computers and network systems, using worms and viruses).

[31] *See* IDC EMEA, *The European Network and Information Security Market: Scenario, Trends, and Challenges*: *A Study for the European Commission, DG Information Society and Media: Final Study Report* (2009), *available at* http://ec.europa.eu/information_society/policy/nis/docs/others_pdf/smart2007005_D_7_1.pdf.

[32] *See* Nat'l Inst. of Standard & Tech., *National Vulnerability Database: Automating Vulnerability Management, Security Measurement, and Compliance Checking*, http://nvd.nist.gov/home.cfm (last visited Oct. 17, 2011).

dynamics, or the underlying design or programming flaws, and other externalities *sans* market failure, and canvassing a panacea bristling with market solutions, but which belies the seeming inevitability of software vulnerabilities.[33] In other words, the economic theories rationalizing software insecurity offer a plausible counterweight to inherent software vulnerabilities fatalism, as exemplified by the works of Rainer Bohme,[34] Ross Anderson et al.,[35] Robert W. Hann et al,[36] and Jaziar Randanti et al.,[37] which drew on the theory of "information asymmetry," amongst other economic theories, to forge an inexorable nexus between software vulnerabilities and market failure.[38]

The pertinent questions therefore are: first, could software vulnerabilities be obviated simply by ameliorating factors responsible for market failure as canvassed by the literature on the economics of software security, drawing on the strength of the theory of information asymmetry, or are vulnerabilities inevitable irrespective of market dynamics and solutions? Second, to what extent is vulnerabilities research or the surreptitious exploitation of software vulnerabilities by hackers tantamount to trespass, and what are the legal implications, if any? Third, to what extent is the peddling of software vulnerabilities valid or enforceable in law? Fourth, what are the implications of software vulnerabilities research for intellectual property rights? Fifth, what is the moral propriety of the market in software vulnerabilities, or should the beneficial effects of vulnerabilities disclosures trump or exculpate the palpable wrongfulness or ethical concerns underpinning the hacking of information systems? Sixth, if software vulnerabilities were inevitable, how best to manage them to ensure the integrity of digital infrastructures?

The paper is divided into seven parts. Part one is the introduction; part two examines the proprieties of information asymmetry and other economic theories inexorably linking software vulnerabilities to market failure; part three discusses vulnerabilities detection research and reviews the boundaries separating professional and malicious hacking; part four discusses the modality and effects of vulnerabilities disclosure; part five analyzes sundry legal issues probing the legality of vulnerabili-

[33] *See, e.g.*, Bohme, *supra* note 25.

[34] *Id.*

[35] *See* Ross Anderson et al., *Incentive and Information Security*, *in* ALGORITHMIC GAME THEORY, 631 (Noam Nisan et. al. eds., 2007).

[36] *See also* Robert W. Hahn & Anne Layne-Farrah, *The Law and Economics of Software Security*, 30 HARV. J.L. & PUB. POL'Y 283, 340-341 (2006).

[37] *See* Jaziar Randiati & Jose J. Gonzalez, *A Preliminary Model of the Vulnerability Black Market*, PROCEEDINGS OF THE 25TH INT'L CONF. OF THE SYS. DYNAMICS SOC'Y,1-30 (Boston, July 29-Aug. 2, 2007), *available at* http://www.systemdynamics.org/conferences/2007/proceed/papers/RADIA352.pdf.

[38] *See generally* George A. Akerlof, *The Market for "Lemons": Quality Uncertainty and the Market Mechanism*, 84 Q.J. ECON. 488, 489 (1970).

ties research and disclosure, which range from cyber trespass, cybercrime, intellectual property rights to the recurring question on whether a liability regime could rein in insecure software? Part six discusses the ethical proprieties of vulnerabilities research and market, whilst part seven concludes the discourse by proffering best practices for software vulnerabilities governance.

II. THEORIZING VULNERABILITIES: THE LIMITS OF MARKET FAILURE.

A bug is a particular kind of failure. It's an emergent property of a system, one that is not desirable. . .Bugs are unique to systems.[39]

In this section, the paper will discuss recent high profile vulnerabilities exploits by malicious hackers within the context of recurring software vulnerabilities imbroglio dictated by the inherent and underlying flaws in software codes. The primary aim of the discourse is to highlight the inherent structural frailty of program designs architecture in juxtaposition with the economic theory underpinning software market dynamics, with a view to validating the proposition that software vulnerabilities determinants are rooted in the inherent technical or programming errors rather than the whims of the market as adumbrated by the literature on the economics of software security. The paper will argue that this clarification is crucial for a proper diagnosis of the anathema that software vulnerability is in order to pave way for the most apposite policy prescriptions for software vulnerabilities governance.

A. Malicious Exploitation of Vulnerabilities: The Scale of the Problem.

On Thursday June 3, 2011, LulzSec, a self-styled underground group of hackers, announced on Twitter that it had hacked into the servers hosting Sony Pictures Entertainment ("Sony Pictures"), and had appropriated and posted online, over one million individuals' personal data comprising customers' passwords, email addresses, home addresses, and dates of birth.[40] According to a report by the Financial Times, the group also claimed that the hacking had been accomplished using a simple process that took advantage of "one of the most primitive and common vulnerabilities."[41] Whilst gloating over their exploits, the group reputedly

[39] *See* Secret & Lies, *supra* note 23, at 205.

[40] *See* Andy Bloxham, *Sony Hack: Private Details of Million People Posted Online*, The Telegraph, (June 3, 2011, 7:51 AM), http://www.telegraph.co.uk/technology/news/8553979/Sony-hack-private-details-of-million-people-posted-online.html.

[41] *See* David Gelles & Joseph Menn, *Sony Suffers Fresh Hacker Attack*, Financial Times (June 3, 2011, 12:56 AM), *available at* http://www.ft.com/cms/s/2/3081e26c-8d6c-11e0-bf0b-00144feab49a.html.

slated and taunted Sony Corporation for the shoddiness of its cyber security preparedness in self-congratulatory jibe and sarcasm:

> We recently broke into SonyPictures.com. . .and compromised over 1,000,000 users' personal information, including passwords, email addresses, home addresses, dates of birth, and all Sony opt-in data associated with their accounts. Among other things, we also compromised all admin details of Sony Pictures. . .along with 75,000 "music codes" and 3.5 million "music coupons". . .Why do you put your faith in a company that allows itself to become open to these attacks?[42]

The irresistible question is if the vulnerability exploited was as primitive and common as LulzSec claimed, why was it not identified and corrected by Sony prior to the hacking incident? This question is especially pertinent given that the June 2011 cyber attack on Sony Pictures was the second within the space of two months, coming on the heels of an earlier attack in April 2011 during which hackers stole personal data of over seventy-seven million customers of Sony PlayStation consoles, with an estimated concomitant financial loss of nine hundred million pounds to Sony.[43]

The answer to the above question arguably lays in the ubiquitous nature of software vulnerability. It typically lays dormant and unseen amidst millions of codes literally hiding in plain sight until it is outed and exploited as LulzSec did. Given what was at stake for Sony, it would no doubt have promptly pre-empted the attack by patching up the vulnerability had they known of its existence, as Sony is no minion. Headquartered in Tokyo, Japan, it is a multinational corporation giant founded in 1946 with a global brand and reach.[44] Surely, the company that facilitated blockbusting Godzilla and the Spider-Man movies,[45] or that invented the Walkman and collaborated with Phillips Consumer Electronics to bring the world the Compact Disk,[46] or whose PlayStation consoles alone are reputedly worth an estimated twenty billion dollars in annual sales,[47] should have the expertise and wherewithal to correct "the most primitive and common vulnerability"?[48]

A fortiori, whilst an inquiry into how Sony ostensibly easily fell foul twice and in quick succession to hackers within the space of two months is proper and legitimate, the obvious answer should be: "It's the vulnera-

[42] *See* Bloxham, *supra* note 40.

[43] *See* TELEGRAPH, *supra* note 21 (noting that the fallout of the PlayStation hacking was estimated at £900 million loss to Sony).

[44] *See* NORIO OHGA, DOING IT OUR WAY: A SONY MEMOIR, 1-4 (2008).

[45] *Id.* at ix.

[46] *Id.* at 41-43.

[47] *Id.* at xv.

[48] *See* Gelles & Menn, *supra* note 41.

bility, stupid,"[49] for no one it seems is immune to cyber attacks as exemplified by similarly high profile victims ranging from the United States' Central Intelligence Agency,[50] Infragad, the Atlanta chapter of the U.S. Federal Bureau of Investigation's affiliate,[51] to the United States Senate.[52] Indeed, cyber attacks are now so pervasive, prevalent, and transcendental that the attacks on Sony would appear routine and symptomatic of a worrying series of escalating attacks on prominent government agencies and corporations in recent times. For example, Lockheed Martin, an aerospace, defense, security, and advanced technology company, and the United States government's main information technology provider, was allegedly hit by an abortive cyber attack in late May 2011.[53] Similarly, in June 2011, Google announced that the private Gmail accounts of "hundreds of senior officials, military types and journalists from America and Asian countries," had apparently been targeted by cyber attackers ostensibly from the eastern Chinese city of Jinan.[54] And in June 2011, the International Monetary Fund ("IMF") allegedly came under cyber attacks by unknown hackers.[55]

Moreover, numerous governments from around the world have reputedly had the control systems of their digital infrastructures directly targeted in what has become known in military parlance as cyber war-

[49] This is a play on the popular phrase from American politics: "It's the economy, stupid", which hacked back to 1992 Bill Clinton's presidential campaign to oust President George Bush senior from office. *See* Richard Alleyne, *Gordon Brown: It's the Economy, Stupid!*, TELEGRAPH (May 23, 2008, 3:58 PM), http://www.telegraph.co.uk/news/politics/byelection/2015038/Gordon-Brown-Its-the-economy-stupid.html (noting that the phrase probably won President Clinton the presidency in 1992).

[50] LulzSec reputedly took credit for taking down CIA website. *See* Tim Bradshaw, *Hackers Claim CIA Website Disruption*, FINANCIAL TIMES (June 16 2011, 9:54 AM), htttp://www.ft.com/cms/s/2e05661de-97f0-11e0-85e9-00144feab49a.html#axzz1PTJPmDiO.

[51] *See* Mathew J. Schwartz, *How LulzSec Hackers Outsmart Security Gurus*, INFORMATION WEEK (June 15, 2011, 11:09 AM), http://www.informationweek.com/news/security/attacks/230700021.

[52] *See id.*

[53] *See US Defence Firm Lockheed Martin Hit by Cyber-Attack*, BBC (May 30, 2011, 7:07 AM), http://www.bbc.co.uk/news/world-us-canada-13587785.

[54] *See Gmail Under Attack, Something Phishy: A Chinese Cyber Attack on a Jumpy America*, ECONOMIST (June 2, 2011), http://www.economist.com/node/18775603.

[55] Analysts believed that the IMF hacking incident was most probably masterminded by a nation state. *See* Peter Apps & Jim Wolf, *Analysis: Who Might be Behind Attempted IMF Data Hacking?*, REUTERS (June 13, 2011), http://www.reuters.com/assets/print?aid=USTRE75C3XO20110613; *See also* Jim Wolf & William Maclean, *IMF Cyber Attack Aimed to Steal Insider Information: Expert,* REUTERS (June 12, 2001, 11:16 AM), http://www.reuters.com/article/2011/06/12/us-imf-cyberattack-idUSTRE75A20720110612 (noting that the cyber attack on the IMF was most probably meant to steal insider information as the race to IMF leadership contest intensified).

fare.[56] For example, in 2007, Estonia was subjected to "a national-level denial-of-service attack" that crippled the nation's Internet, telecommunications and financial networks for a week.[57] Similarly, in the spring of 2011, Pentagon lost twenty-four thousand files to hackers in "one of the largest cyber attacks in United States history," which United States Deputy Defense Secretary William Lynn attributed to "foreign intruders."[58] While stressing the need for appropriate counter measures, the Deputy Defense Secretary noted that, "[i]n the 21st Century, bits and bytes can be as threatening as bullets and bombs."[59] Also in February 2011, William Hague, the United Kingdom's Foreign Secretary, informed a Munich conference that government computers and computers of government's military defense contractors had been targeted by hackers and cybercriminals.[60]

More worryingly, on June 17, 2010, VirusBlokAda, a Russian software security company, unearthed Stuxnet, reputedly the most sophisticated and dangerous computer virus yet, with real capabilities to disrupt "the software that controls pumps, valves, generators and other industrial machines."[61] The Stuxnet virus had an estimated "15,000 lines of code, representing an estimated 10,000 person hours in software development."[62] Stuxnet's unprecedented sophistication and its disproportionate prevalence in Iranian computers informed speculations that it had been specifically designed by an unknown hostile nation state to target and sabotage Iranian uranium-enrichment facility in Nantaz, with

[56] *See generally* SCHNEIER ON SECURITY, *supra* note 25, at 218-26 (describing warfare scenarios and incidents of warfare in cyberspace); *See generally* Muhammad Saleem & Jawad Hassan, *"Cyber Warfare", the Truth in a Real Case*. PROJECT REPORT FOR INFO. SECURITY COURSE, University of Linkoping, Sweden (2009), *available at* http://www.ida.liu.se/~TDDD17/oldprojects/2009/projects/007.pdf (explaining in relative detail the methodology for the cyber attacks on Estonia's digital infrastructure, how similar attacks could be traced and possible defensive measures in cyberwarfare).

[57] *See* ROBERT K. KNAKE, INTERNET GOVERNANCE IN AN AGE OF CYBER INSECURITY 6 (2010), *available at* http://irps.ucsd.edu/assets/001/501278.pdf (noting that cyber warfare capabilities are no longer confined to the laboratory as exemplified by the attacks on the Estonia government, which took the country offline for one week).

[58] *See Pentagon admits suffering major cyber attack*, BBC (July 14, 2011, 07:01 PM), http://www.bbc.co.uk/news/world-us-canada-14157975; Chloe Albanesius, *Pentagon Loses 24K Files in Huge Cyber Attack*, PC MAGAZINE, (July 14, 2011, 3:53 PM), http://www.pcmag.com/article2/0,2817,2388521,00.asp.

[59] *See Pentagon admits suffering major cyber attack, supra* note 58.

[60] *See William Hague: UK is Under Cyber Attack*, BBC (Feb. 4, 2011, 2:30 PM), http://www.bbc.co.uk/news/uk-12371056.

[61] *See* Sharon Weinberger, *Computer Security: Is This The Start of Cyberwarfare?*, 474 NATURE 142 (2011), *available at* http://www.nature.com/news/2011/110608/pdf/474142a.pdf.

[62] *Id.*

the primary aim of stalling Iran's controversial nuclear program.[63] Experts also reasoned that the malware code was designed to specifically alter the speed of delicate centrifuges and cause critical machineries at the nuclear facility to spin out of control.[64] Analysts also believed that the Stuxnet virus, which had exploited four previously unknown or "zero-day" *vulnerabilities* in Microsoft's Windows,[65] offered a veritable template for future cyberware stratagems, and "provided chilling proof that groups or nations could launch a cyberattack against a society's vital infrastructures for water and energy."[66]

However, while delimiting the parameters of cyber warfare and the possible scenarios for military retaliatory response, Steven Bradbury argued that if a foreign power hacked into a government computer to steal sensitive information, it would be an act of espionage, which was not expressly prohibited by the laws and customs of war. However, if the cyber attack occasioned significant physical destruction and loss of life from the concomitant failure of critical infrastructure, such as dams or water supply system, then it would constitute an armed attack under the law of war, and would justify a full military response under Article 51 of the Charter of the United Nations 1945.[67] Bradbury, however, doubted whether the 2007 denial-of-service attacks on Estonia constituted an armed attack or cyber warfare under the laws of war, on the basis of Article 41 of the Charter of the United Nations, which provides that a "complete or partial interruption of economic relations and of rail, sea, air, postal, telegraphic, radio, and other means of communications" would not be a "measure . . . involving armed force."[68]

From the foregoing analyses, it is sacrosanct that software's inherent structural weakness makes it a fodder for vulnerabilities that reputedly account for most of the reported computer or network security problems.[69] This is exemplified by the Stuxnet malware attacks, which

[63] Iranian national news agency confirmed the attack on nuclear facilities. *See* Admin in Cyberwar, *Stuxnet in Iran – Cui Bono?*, J.L. & CYBERWARFARE, (Sept. 29, 2010), *available at* http://www.jlcw.org/2010/09/29/stuxnet-iran-cui-bono/.

[64] *See* Weinberger, *supra* note 61.

[65] *Id.* at 143 (italic is mine for emphasis).

[66] *Id.*

[67] *See* Steven G. Bradbury, *The Developing Legal Framework for Defensive and Offensive Cyber Operations*, 2 HARV. NAT'L SECURITY J. 17 (2011), *available at* http://harvardnsj.org/wp-content/uploads/2011/04/Vol.-2_Bradbury_Final.pdf.

[68] *Id.* (citing Maj. Arie J. Schaap, *Cyber Warfare Operations: Development and Use Under International Law*, AIR FORCE L. REV. 121, 144-45 (2009), *available at* http://www.thefreelibrary.com/Cyber+warfare+operations%3A+development+and+use+under+nternational+law.-a0212035712. Note that the literature on the law of cyber warfare is still largely rudimentary with a lot of grey and untested areas. Its full discourse is beyond the remit of this paper.

[69] *See* SECRET & LIES, *supra* note 23, at 202-05 (detailing incidences of software failures and noting that most computer security problems emanated from faulty code).

as noted above, exploited four previously unknown *vulnerabilities* in Microsoft's Windows,[70] and the exploits of LulzSec, the "hacktivist" group who purportedly breached Sony Pictures' network security via "one of the most primitive and common *vulnerabilities*."[71] The pertinent question therefore is: could software vulnerabilities be obviated? In answering this question, this section will briefly review the mechanics of software architecture that underpins software functionality, and juxtapose it with economics and computer security scholars' market failure rationalization of vulnerabilities, which largely drew on the theory of information asymmetry for authority.[72] The section will then assay whether vulnerabilities should be attributed mainly to inherent mechanical or structural defects or market failure, or an admixture of both phenomena, and then discuss the best policy strategy for reining in software vulnerability.

B. Deconstructing the Mechanics of Software Vulnerabilities.

A networked world may be more convenient, but it is also much more insecure.[73]

A faulty code or bug is the Achilles' heel of computer or network systems security, and one of the weakest links through which networked computers are traditionally breached.[74] Exploitable software vulnerabilities typically range from buffer overflows bugs, kernel flaws, symbolic links, file descriptor attacks, race conditions, file and directory permissions, Trojans, viruses, to social engineering.[75]

The kernel code is the core of operating systems, which is tasked with responsibilities ranging from maintaining communications between software and hardware components to enforcing the overall security model for operating systems. Consequently, any security flaws in the kernel code could endanger the entire operating systems.[76] Another potential source of software vulnerability are symbolic links, which are files that point to or contain a reference to other files or directories. Because programs often change the permission granted to a file, a user or attacker could strategically create symbolic links to trick programs into

[70] *See* Weinberger, *supra* note 61, at 143.

[71] *See* Gelles & Menn, *supra* note 41 (italic is mine for emphasis).

[72] *See, e.g.*, Bohme, *supra* note 25, at 1-5 (ascribing the existence of vulnerabilities to market failure).

[73] *See* Secret & Lies, *supra* note 23, at 176.

[74] *Id.* at 202-05.

[75] For a discussion on exploitable vulnerabilities, *see* Anand Ramdeo, *Software Testing – Penetration Testing*, TestingGeek, (May 4, 2011), http://www.testinggeek.com/software-testing-penetration-testing.

[76] Security flaws that emanate from kernel codes are known as kernel flaws. *See Id.*

modifying or listing critical system files.[77] Yet another exploitable software vulnerability is an attack on file descriptors, which are non-negative integers used by systems to keep track of files rather than using specific filenames. Whenever a privileged program assigns an inappropriate file descriptor, the file could be vulnerable to compromise and attacks.[78] Another software vulnerability exploit is the "race conditions," which could occur whenever a program or process entered into a privileged mode, and a user or attacker managed to compromise the program or process whilst still in its privileged state or mode.[79] Furthermore, file and directory permissions, which control access to both users and process, could through poor or inappropriate permissions, facilitate any number of attacks, ranging from reading, writing or modifying of password files, to the addition of unscrupulous hosts to the list of trusted remote hosts.[80] Yet another threat to networked systems security is a variant of malevolent software known as Trojan horse, a destructive program that masquerades as a beneficial application, but actually contains harmful payload.[81] Arguably the most infamous in the malicious programs category are computer viruses, which are designed to infect and attach to a host program, to execute when the host program is executed, and to continually replicate amongst host programs.[82]

However, the main structural software defects that provide inroads for malevolent programs are buffer overflows, which have been described as "the most common form of security vulnerability," because they work on fairly predictable protocol, which is extremely malleable and modifiable by hackers.[83] According to Meiring de Villiers, "[b]uffers are limited capacity data storage areas in computer memory," and that a buffer overflow usually occurs when "a program attempts to fill a buffer with more data than it was designed to hold."[84] He analogized the resulting chaos with "pouring ten ounces of water into a glass designed to hold eight."[85] While the glass symbolizes the buffer, the water symbolizes data, and the excess data would then overflow into "adjacent memory locations

[77] *Id.*

[78] *Id.*

[79] *Id.*

[80] *Id.*

[81] *See* PETER SZOR, THE ART OF COMPUTER VIRUS RESEARCH AND DEFENSE 663 (2005).

[82] *See* Meiring de Villiers, *Information Security Standards and Liability*, J. INTERNET L. 1, 4 (2010), *available at* http://papers.ssrn.com/sol3/papers/cfm?abstract_id=1477813 [hereinafter *Information Security Standards and Liability*].

[83] *See* SECRET & LIES, *supra* note 23, at 202.

[84] *See* Meiring de Villiers, *Distributed Denial of Service: Law, Technology & Policy*, U. OF NEW S. WALES (UNSW) L. RES. SERIES, Paper No. 2007-3, 1, 22-23 (2007), *available at* http://papers.ssrn.com/sol3/papers.cfm?abstract_id=952177## [hereinafter *Distributed Denial of Service*].

[85] *Id.* at 22.

where it can corrupt existing data, possibly changing instructions and resulting in unintended executions," which could either be harmful or harmless.[86] Significantly, a buffer overflow could provide an inroad for hackers to "remotely inject executable malicious code" such as a denial-of-service attack code into the memory of a target computer or network system.[87] For example, the notorious 1989 Morris-Worm, which was designed by Robert T. Morris, a Cornell University Graduate Student, reportedly used buffer overflow vulnerability in a UNIX program for launching a massive denial-of-service attack on the Internet.[88]

While most overflow bugs could be fixed upon discovery, and while skilled programming could pre-empt buffer overflows bugs,[89] the main challenge is in the increasing complexity of programs, because the more complex or larger a program or code is, the greater the odd for buffer overflows bugs.[90] According to Bruce Schneier, "[i]t's very difficult to guarantee that there are no overflows problems, even if you take time to check," and the larger and more complex the code is, the more likely the odds for bugs or vulnerabilities and prospects for exploitation and attack.[91] Indeed, Schneier's analysis on the ubiquity of software vulnerabilities is reflective of the general consensus amongst computer security scholars, as exemplified by Aaron J. Burstein's assertion that technically, "it is practically impossible to find all potential vulnerabilities in systems as complex as modern computers."[92] Yet, modern computing is continually defined by increasing complexity and sophistications,[93] whilst the human link remains one of the weakest and a potent source of vulnerability in the computing and network systems security chain.[94]

[86] *Id.* (noting that a buffer overflow could abort the application program without causing much harm or damage).

[87] *Id.*

[88] *See generally Information Security Standards and Liability*, *supra* note 82, at 7-8.

[89] According to Meiring de Villiers, most software firms have a "patch and vulnerability management" team, which is especially geared to "proactively prevent the exploitation of software, hardware and human vulnerabilities within the IT network . . ." *See Distributed Denial of Service*, *supra* note 84, at 23-24.

[90] *See* SECRET & LIES, *supra* note 23, at 209.

[91] *Id.* at 209-10.

[92] *See* Burstein, *supra* note 17, at 175.

[93] Examples of the increasing complexity in modern computing range from tablet computers, to the world's latest and fastest computer, "The K Supercomputer," which was built by the Japanese electronics firm Fujitsu. *See* Paul Thompson, *Japan Creates World's Fastest Supercomputer Which is as Quick as One MILLION Desktop PCs*, MAILONLINE (June 21, 2011, 8:17 AM), http://www.dailymail.co.uk/sciencetech/article-2005920/Japan-creates-worlds-fastest-supercomputer-fast-MILLION-desktop-PCs.html?ito=feeds-news xml.

[94] The human factor constitutes one of the weakest links in computer and network security chain, a phenomenon known as "social engineering," which is the technique of using persuasion and/or deception to gain access to, or information about information sys-

A fortiori, it is sacrosanct that vulnerabilities are inherently embedded in software architecture, and cannot be completely eliminated,[95] as exemplified by the Carnegie Mellon University study, which estimated that a thousand lines of code would typically have five to fifteen bugs.[96] The Carnegie Mellon study thus underscores the sheer scale of possible bugs in a typical operating system, which could run into millions of lines of codes, as exemplified by Microsoft Windows 2000, which was estimated at thirty-five to sixty million lines of codes.[97] It is thus axiomatic that bugs are ubiquitous, a veritable explanation for the discovery and publication of an average of ten new software vulnerabilities per day in the National Vulnerability Database (NVD) by the NIST.[98]

Most significantly, vulnerabilities problem is further exacerbated by the fixation of the technical community on interoperability and standardization of programs,[99] driven by the imperatives and expediency of seamless networks and propped-up by anti-trust policy as exemplified by the United States and European anti-trust case against Microsoft.[100] This has precipitated a call for a shift from undue fixation on interoper-

tems via human access, conversation, and other forms of interaction, ranging from telephone, email, to face-to-face engagements. For a discussion of "social engineering" as a critical source of network vulnerability, *see* Ramdeo, *supra* note 75; SECRET & LIES, *supra* note 23, at 266-67 (noting the human element as a critical weak link in the chain of computer and network security, and recalling the testimony of a hacker before Congress that he "was so successful in [the social engineering] line of attack that [he] rarely had to resort to a technical attack").

[95] *See* Randiati & Gonzalez, *supra* note 37, at 5.

[96] Cited in SECRET & LIES, *supra* note 23. *See also* SCHNEIER ON SECURITY, *supra* note 25, at 260 (noting that vulnerabilities are software mistakes, and that "any large software package will have thousands of mistakes").

[97] *See* SECRET & LIES, *supra* note 23, at 210.

[98] *See* NAT'L INST. OF STANDARD & TECH., *supra* note 32.

[99] For example, Council Directive 91/250/EEC, 1991 O.J. (L 122), recital 10 (EC) (as amended by Council Directive 93/98/EEC, 1993 O.J. (L 290)(EC)) [hereinafter EU Software Directive], which defines interoperability as "a logical and, where appropriate, physical interconnection . . . to permit all elements of software and hardware to work with other software and hardware and with users." Most significantly, not only is interoperability of programs favored for seamless Internet and computer connectivity, but it is often mandated by authorities through competition laws in order to prevent abuse of dominant position by a pioneering software vendor. For example, Microsoft incurred heavy fines in Europe because its European competitors' programs were not interoperable with Microsoft programs. *See Opinion of the Advisory Committee on Restrictive Practices and Dominant Positions Given at its 371st Meeting on 22 March 2004 on a Preliminary Draft Decision in Case COMP/C-3/37.792-Microsoft (2007/C26/03), available at* http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:C:2007:026:0004:0004:EN:PDF [hereinafter *Opinion of the Advisory Committee*].

[100] For the Microsoft European anti-trust case, *see Opinion of the Advisory Committee, supra* note 99. *See also* United States v. Microsoft, 65 F. Supp. 2d 1, 15 (D.D.C. 1999); Randal C. Picker, *Cybersecurity: Of Heterogeneity and Autarky*, *in* LAW AND ECONOMICS OF CYBERSECURITY 115, 124 (Mark F. Grady & Francesco Parisi eds., 2006) (noting that inter-

ability to integrating effective and improved security into networked systems architecture.[101] According to Randolph Beard *et al.*, while desirable, software interoperability for operating systems "can create 'holes' in the software for malicious code . . . leading to unpredictable calculation errors and other manifestations of incompatibilities."[102] Moreover, as noted by Robert Knake, the underlying technologies of the Internet "were designed for a closed network in which access was closely controlled and all users were trusted. They were not built and designed for the purposes for which they are now being used."[103] Notably, Jonathan Zittrain also echoes the simplicity that informs Internet design architecture as follows:

> The network's design is intended to allow all data to be treated the same way: it can be sent from anyone to anyone, and it can be in support of any application developed by an outsider.[104]

In other words, in terms of effective security, the traditional structural design for network systems architecture is more suited to intranet than its current predominantly Internet uses. Arguably, the dramatic transition from intranet to Internet demands on software architecture by organizations, firms, and individuals, is driven, *inter alia*, by the quest for operational efficiency, visibility, and accessibility. This is put in clearer perspective by Robert A. Martin as follows:

> In the past, organizations had stand-alone computer systems that interacted only with other internal systems. Only a few systems used tapes and file passing to exchange information with outside systems. This isolation meant that software errors usually had limited impact. The general public was unaware of most errors, crashes, and oversights, which at best caused occasional troubles for an organization's closest business partners.
>
> Today, however, few organizations – whether in the private or government sector – have or build self-contained systems.
>
> . . .
>
> The movement to highly accessible systems, driven by the need to save resources and improve efficiency as well as the reality of having to do more with less, has dramatically increased the impact of mistakes in commercial and open source software.[105]

operability of Windows operating system with other software was one of the key issues in the anti-trust actions brought by the United States and the European Union).

[101] *See* KNAKE, *supra* note 57, at 25-26.

[102] *See* T. Randolph Beard, et al., *Tort Liability for Software Developers: A Law and Economic Perspective*, 27 J. MARSHALL J. COMPUTER & INFO. L. 199, 231 (2009).

[103] *See* KNAKE, *supra* note 57, at 25.

[104] *See* JONATHAN ZITTRAIN, THE FUTURE OF THE INTERNET AND HOW TO STOP IT 32 (2009).

[105] *See* Martin, *supra* note 27, at 33.

However, in the context of security, the inherent structural flaws in Internet architecture have been known for years and are typified by the following vulnerabilities in three key Internet protocols:

> [T]he Internet Protocol, which guides data from source to destination across the Internet; the Domain Name System, which translates IP numbers into recognizable Web addresses; and the Border Gateway Protocol, which provides the connection between networks to create the "network of networks".[106]

According to Knake, none of the above-mentioned three protocols "has built-in mechanisms to verify the origin or authenticity of information sent to them, leaving them vulnerable to being spoofed or otherwise manipulated by malicious actors."[107] Although the above problems were identified by a 2003 report, they have not been rectified till date.[108] In fact, as noted earlier, Internet protocols were never designed with security in mind,[109] but rather for the expediency of programs and networks interoperability.[110] Thus it has been suggested that the best way to obviate inherent flaws in networked systems architecture is to create a suite of more secure protocols that sufficiently address all security concerns, without fragmenting the Internet or undermining its independence from state control.[111]

However while it is theoretically possible to create a suite of more secure protocols in relative terms, the main challenge is in simultaneously programming completely bug-free protocols whilst maximizing interoperability of programs, without concomitantly weakening Internet protocols. But due invariably to inherent programming flaws and the simultaneous quest for programs interoperability, it is arguably an impossible challenge.[112] *A fortiori*, computer security is essentially relative, and it would appear that the best anti-vulnerabilities remedial course is to find and apply corrective patch to vulnerabilities before unscrupulous groups or malicious hackers who might use them to attack digital infrastructures do. Whilst the find-and-corrective patch strategy is least reassuring and not a full bulwark against malicious hackers, it arguably underscores the importance of vulnerabilities research as evidenced by

[106] *See* THE WHITE HOUSE, THE NATIONAL STRATEGY TO SECURE CYBERSPACE 30, (2003), *available at* http://www.us-cert.gov/reading_room/cyberspace_strategy.pdf, cited in. KNAKE, *supra* note 57, at 25-26.

[107] *Id.* at 25.

[108] *Id.* (noting that nearly a decade after the security problems were identified, they continue to "plague the Internet").

[109] *See* SECRET & LIES, *supra* note 23, at 176.

[110] *See* Picker, *supra* note 100, at 124.

[111] *See* KNAKE, *supra* note 57, at 26.

[112] The general consensus amongst computer security experts is that it is impossible to write a perfect or flawless or bug-free program. *See* SECRET & LIES, *supra* note 23, at 203.

the increasingly boisterous market in software vulnerabilities.[113] The legality of vulnerabilities research and the ethical implications of the market in vulnerabilities are the subject of further discourse in parts five and six of this paper.[114]

C. Information Asymmetry and Software Insecurity Revisited.

In contradistinction to the technical theory attributing vulnerabilities to inherent programming flaws or mistakes, the economic theory extrapolating the theory of information asymmetry to rationalize software vulnerabilities, firmly ascribes vulnerabilities to market failure by linking programming mistakes to a lack of market incentive for investment in more secure software.[115] However, prior to reviewing the literature on the nexus between software vulnerabilities and the theory of market failure, it is apt to briefly discuss the etymology of information asymmetry, which is an economic theory that provides insights into the decision making patterns of parties in the principal-agent relationship, where one party has more or better information than the other party.[116] Whilst employing the theory of information asymmetry in his seminal work for which he won a Nobel Prize in Economics in 2001,[117] George Akerlof opined that since it was impossible for a used car buyer to differentiate between a good car and a bad car, a used car could not have the same valuation as a new car. In the circumstances, a used but perfectly good car would be undersupplied, undersubscribed, and eventually taken off the market because the seller would not be able to receive the price for "the true value of his car."[118] Consequently, most used cars traded would be "lemons" or bad cars, while used but perfectly good cars might not be traded at all.[119]

[113] For a discussion, *see* Charlie Miller, *The Legitimate Vulnerability Market: Inside the Secretive World of 0-day Exploit Sales*, Indep. Security Evaluators 1 (2007), available at www.securityevaluators.com (noting the brisk trade in vulnerabilities market and describing how vulnerabilities researchers were motivated by monetary gain rather than prestige).

[114] *See infra* parts 5 and 6.

[115] *See* for example, Anderson, *supra* note 35, at 636 (arguing that most commercial software contains preventable designs and implementation flaws and that while vendors are capable of creating more secure software, "the economics of the software industry provide them with little incentive to do so").

[116] *See, e.g.*, Jonathan Lean & Jonathan Tucker, *Information Asymmetry, Small Firm Finance and the Role of the Government*, 1 J. Fin. & Mgmt. in Pub. Services 43, 44-45 (2001), *available at* http://www.cipfa.org.uk/acipfal/download/jour_vol1_no1_c.pdf (discussing how, in the principal-agent relationship, information is neither perfect nor costless, and is often distributed asymmetrically).

[117] *See* Akerlof, *supra* note 38, at 489.

[118] *See id.*

[119] *See id.*

Ross Anderson et al. extrapolated Akerlof's theory to software market and directly linked software vulnerabilities to market failure.[120] According to Anderson, "the software market is a "market for lemons"[121] and that:

> Although vendors are capable of creating more secure software, the economics of the software industry provide them with little incentive to do so. . .Consumers generally reward vendors for adding features, for being first to market, or for being dominant in an existing market. These motivations clash with the goal of writing more secure software, which requires time-consuming testing and a focus on simplicity. . .[W]hy does the potential damage to vendor reputations not motivate them to invest in more secure software?[122]

Anderson's stance is furthermore reinforced by Rainer Bohme, an economics and computer security scholar, who supports the used car and software market analogy, and argues that the bane of computer and network security is explainable by economics and market dynamics and is reducible to market failure.[123] While employing Akerlof's famous lemons market theory, Bohme argues that:

> [s]ecurity is not visible, it's a trust good. Since the buyer is unable to differentiate secure from insecure products. . .the market drops to the level for insecure products. Hence vendors have little incentive to develop sound security technology and rather prefer to invest in more visible gimmicks.[124]

Bohme also employs Garrett Hardin's Commons Theory[125] to rationalize the lack of consumer demand for secure software or private underinvestment in cyber security.[126] He analogizes computer networks with a grazing field held in common by a group of herders as hypothesized in Garrett Hardin's Commons theory.[127] According to Bohme, just as the common pasture is in danger of being overgrazed and depleted in the absence of a rational economic decision limiting the number of cows that

[120] Anderson, *supra* note 35, at 636; Bohme, *supra* note 25.

[121] Bohme, *supra* note 25.

[122] *Id.*

[123] *Id.*

[124] *Id.*

[125] *See* Garrett Hardin, *The Tragedy of the Commons*, 162 SCIENCE 1243, 1243-48 (1968), *available at* http://www.sciencemag.org/content/162/3859/1243.full. The tragedy of the commons theory is premised on the parable of herders who are each entitled to graze their cattle on the same common parcel of land unhindered. However, as each herder benefited by freely adding new cows to the grazing field held in common, its quality progressively diminished due to overgrazing, and the herders collectively suffered the resultant detriments of a depleted pasture. A more rationale economic decision limiting the number of cattle or the number of herders through a buy-off of grazing rights could have saved the common pasture from ruin.

[126] *See* Bohme, *supra* note 25.

[127] Hardin, *supra* note 125, at 1243-48.

individual herders could introduce through a buy-off of individual grazing right for example, the computer network would be in danger of worms and viruses unless the costs of security could be shared amongst all users of network nodes, or all network users invest in network security.[128] However, because not all users of network nodes would be willing to invest in network security, an entire network could get corrupted through the unsecure or weakest node.[129] Bohme further attributes the lack of incentive by individual network users to unilaterally invest in network security to the socialization of both risks and benefits of network nodes amongst users.[130] *A fortiori*, network users are more inclined to savoring the common benefits of the networks than sharing the costs of offsetting the concomitant risks posed by networks insecurity. Thus in the context of Hardin's Commons Theory, individual network users would perceive network nodes as a public good and would prefer to be "free riders" and hope other co-users would "pay in their place."[131] Therefore, if every network user opts to free-ride and hopes that the other network user pays for network security, invariably, few or no network users would pay for security and, consequently, network systems would become vulnerable to attacks due to a weak consumer demand for network security or private underinvestment in cybersecurity.[132]

D. Software Vulnerabilities and the Propriety of Market Failure.

However, whilst plausible, I would contest the propriety of the theories of information asymmetry and the tragedy of the commons for rationalizing the software insecurity and market failure nexus. Whilst the theory of information asymmetry as originally espoused by Akerlof might be well-suited to explaining the dynamics of used car market,[133] and whilst Hardin's Tragedy of the Commons Theory might give clarity

[128] *See* Bohme, *supra* note 25.

[129] *Id.*

[130] *See id.*

[131] *Id.*

[132] The "free-riders" analogy of Garrett Hardin's Commons theory has also been used by a number of computer security and economics scholars to rationalize private underinvestment in network security. *See, e.g.*, Bruce H. Kobayashi, *Private Versus Social Incentives in Cybersecurity: Law and Economics*, *in* Law and Economics of Cybersecurity 13, 21 (Mark F. Grady & Francesco Parisi eds., 2006) (noting that private security expenditures are important due to the centralization of the Internet, and that government intervention might be necessary to shore up private investment in cybersecurity due to underinvestment induced by free-riding and the resultant inadequate level of protection). *See also* Burstein, *supra* note 92, at 176-78 (noting, *inter alia*, that private underinvestment in cybersecurity is due to individual reluctance to invest in security, which could ultimately might benefit others on the network).

[133] *See* Akerlof, *supra* note 38, at 489.

and insight into the herders' folly and lack of foresight into the danger of overgrazing of the common pasture due to overuse, the theories are arguably incongruous and less apposite for rationalizing software insecurity for the reasons set out in the following paragraphs.

First, the theory of information asymmetry as used in the context of software insecurity, *ipso facto* assumes that software vendors are capable of creating "more secure" software, but lack the necessary incentive due to the absence of the market for good software, which is dictated by consumers' preference for bad software or "lemons."[134] The premise for this reasoning is arguably flawed because software security is essentially relative, and "more secure" software, even if achievable, would not necessarily translate into perfect or bug-free software, which is technically required for optimum network security.[135] In fact, as noted earlier, the general consensus amongst computer security experts is that it is impractical to create software that is completely bereft of bugs or vulnerabilities.[136] Thus, even if software vendors were able to create "more secure" software in response to the reputed market incentive, they could by no means guarantee that it would be free from vulnerabilities, which are undoubtedly the Achilles' heel of networked systems and computer security.[137]

Moreover, the claim that a lack of market incentive is stifling the creation of "more secure" software is equally presumptuous as there is indeed a niche market for "more secure" software, and most software vendors do indeed strive to create "more secure" software mainly for reputational and competitiveness reasons. This is exemplified by Bill Gate's famous e-mail message to Microsoft full-time employees on January 15, 2002, noting what damage programming flaws and vulnerabilities could do to the company's reputation, and urging emphasis on better security and trustworthy computing.[138] The following excerpts from the e-mail arguably offer an insight into the company's aspirations and struggles to deliver more secure software:[139]

[134] *See, e.g* Anderson, *supra* note 35, at 636; Bohme, supra note 25.

[135] As noted earlier, most attacks on networks exploit latent software vulnerabilities. *See, e.g.*, SECRET & LIES, *supra* note 23, at 203.

[136] *See, e.g.*, Michael A. Cusumano, *Who is Liable for Bugs and Security Flaws in Software*, 47 COMM. OF THE ACM 25, 26 (2004).

[137] *See generally,* SECRET & LIES, *supra* note 23, at 203.

[138] *See* Bill Gates, *Bill Gates: Trustworthy Computing*, WIRED, (Jan.15, 2002, 5:22 PM), http://www.wired.com/techbiz/media/news/2002/01/49826.

[139] Indeed, Microsoft is not the only vendor to grapple with the perennial problem of software vulnerabilities. According to Randal C. Picker, the "list of infamy" includes prominent names that range from Apache, Apple, BEA Systems, Eudora, GNU, Hewlett Packard, KDE, the Linux kernel, Netscape, Novell, Opera, Sun, to Symantec. For a discussion, *see* Picker, *supra* note 100, at 124.

> Every week there are reports of newly discovered security problems in all kinds of software, from individual applications and services to Windows, Linux, Unix and other platforms. We have done a great job of having teams work around the clock to deliver security fixes for any problems that arise. Our responsiveness has been unmatched—but as an industry leader we can and must do better. Our new design approaches need to dramatically reduce the number of such issues that come up in the software that Microsoft, its partners and its customers create. We need to make it automatic for customers to get the benefits of these fixes. Eventually, our software should be so fundamentally secure that customers never even worry about it. . .As software has become ever more complex, interdependent and interconnected, our reputation as a company has in turn become more vulnerable. Flaws in a single Microsoft product, service or policy not only affect the quality of our platform and services overall, but also our customers' view of us as a company. . . So now, when we face a choice between adding features and resolving security issues, we need to choose security. Our products should emphasize security right out of the box, and we must constantly refine and improve that security as threats evolve. . .[140]

The pertinent question therefore is: does Microsoft, an industry leader with ninety percent of the total market share of client operating systems,[141] need any further market incentives to create "more secure" software? The answer is arguably no, because Microsoft already dominates the market, and if it were technically feasible, it would happily create the most secure software there is on the market in order to irrevocably cement their market dominance.[142] The reality, however, is that it is virtually impossible to create vulnerabilities-free software.[143] This reality is the same even for Apple, which prides itself on its reputedly "more secure" software relative to that of its arch rival and competitor Microsoft.[144] In fact, Apple routinely touts its supposedly inviolable Mac operating systems as exemplified by the following rather boastful assurances:

> Mac OS X doesn't get PC viruses. And its built-in defenses help keep you safe from other malware without the hassle of constant alerts and sweeps. . .With virtually no effort on your part, Mac OS X offers a multilayered system of defenses against viruses and other malicious applica-

[140] *See* Gates, *supra* note 138.

[141] *See*, *Operating System Market Share*, *supra* note 11.

[142] For a discussion, *see* Gates, *supra* note 138.

[143] *See* SECRET & LIES, *supra* note 23, at 203.

[144] The general prevailing perception is that Microsoft's Windows operating suites are more susceptible to malware and viruses than Apple's Mac operating suites. However some analysts have attributed this to the fact that most of the world's computers run on Microsoft's operating systems, and that this monoculture of operating systems makes Microsoft vulnerable and more of a target for hackers than its competitors. For a discussion, *see* Picker, *supra* note 110, at 124; Cusumano, *supra* note 136, at 26.

> tions, or malware. For example, it prevents hackers from harming your programs through a technique called "sandboxing"– restricting what actions programs can perform on your Mac, what files they can access, and what other programs they can launch. Other automatic security features include Library Randomization, which prevents malicious commands from finding their targets, and Execute Disable, which protects the memory in your Mac from attacks.[145]

Notably, there is ample evidence that some of Apple's dedicated customers bought into the Mac OS X operating system's perceived invincibility and were willing to pay a premium for the relatively "more secure" Mac software vis-à-vis the main competition, Microsoft's Windows, which is perennially plagued by malwares.[146] However, recent spates of attacks on Apple's Mac OS X by a "Trojan horse" masquerading as a "Mac Protector" have ostensibly shattered the myth of Mac operating system's invincibility, and demonstrated that no operating systems is completely free from vulnerabilities and the concomitant cyber attacks.[147] And most significantly, the fact that Apple patched a total of twenty-eight documented vulnerabilities in Mac OS X in 2010 belies the company's boastful assurances that its Mac operating system is impervious to viruses.[148] Thus, while Mac's operating system is generally perceived to be "more secure" and less vulnerable to malware relative to its competitors, it is still not completely free from vulnerabilities.[149] While its approximately 5.19 percent share of the total market for operating systems[150] underscores the enduring existence of a niche market for software that is perceived to be relatively "more secure" than rival products, this is contrary to the market failure theory adumbrated by Ross Anderson et al. and

[145] *See* Apple Inc., *OS X. It's What Makes a Mac a Mac*, http://www.apple.com/macosx/security/ (last visited Oct. 21, 2011).

[146] *See* Harry McCracken, *Mac Security Threats: How Vulnerable is Apple?*, TIME (June 2, 2011), http://www.time.com/time/business/article/0,8599,2075218,00.html (noting that most Mac OS X users believe that the operating system is impervious to cyber attacks, and see the perceived invincibility as a major consideration in their choice of computing systems).

[147] *Id.* (noting that following recent attacks on Mac operating system, some Mac users now buy anti-virus software, while others don't see the need to do so). Also, a recent research shows that as many as eight out of ten web browsers (inclusive of Mac operating system) are vulnerable to cyber criminals due to latent vulnerabilities. *See* Christopher Williams, *Eight Out of Ten Web Browsers Vulnerable to Cyber Criminals*, TELEGRAPH, (Feb 18, 2011, 1:00 PM), http://www.telegraph.co.uk/technology/news/8333255/Eight-out-of-10-web-browsers-vulnerable-to-cyber-criminals.html.

[148] *See* Ryan Naraine, *Apple Pugs 28 Mac OS X Security Holes*, ZDNET (June 15, 2010, 3:21 PM), http://www.zdnet.com/blog/security/apple-plugs-28-mac-os-x-security-holes/6707 (noting that hackers could potentially take over Mac users' systems through the said vulnerabilities).

[149] *See id.*

[150] *See Operating System Market Share*, *supra* note 11 (noting that Apple's Mac is the closest rival to Microsoft's Windows in the market for operating systems).

other computer security and economics scholars.[151]

It is pertinent to note that if software were vulnerabilities free and completely inviolable, there would be no need for a parallel market in computer security.[152] *A fortiori*, branding the reluctance of a segment of networked systems' users to invest in anti-malware programs as evidence of market failure for more secure software is nothing short of glossing over "the elephant in the room" and the real reason that anti-malware programs were needed in the first place: software vulnerabilities. Even so, there is ample evidence that individuals, firms, and organizations continually invest in computer security,[153] as exemplified by the burgeoning multi-million dollars proprietary software security industry.[154] For instance, as noted earlier, the projected estimated global value for the European network and information security market in 2010 was 15.5 billion Euros, ranking as the second largest in the world after that of the United States.[155] Even for those pockets of individuals on the network who are unwilling to invest in anti-virus and software security, most mainstream Internet service providers and financial institutions offering Internet banking services have internalized the costs of security, and are now routinely offering free software security to their subscribers and customers.[156] In the United Kingdom for example, all of Virgin Media's broadband packages come with "free top-of-the-range security, including anti-virus, anti-spyware protection and parental controls."[157] Moreover, there is a plethora of non-proprietary but equally relatively effective and popular free anti-virus or software security solutions[158]

[151] *See* Anderson, *supra* note 35, at 631.

[152] *See* IDC EMEA, *supra* note 31 (estimating the global value of the European information security market for the year 2010 at 15.5 billion Euros, the second largest in the world after the United States' information security market).

[153] *Id. See also* Kobayashi, *supra* note 132, at 21 (noting in particular that ". . .although individuals and businesses have made significant investments in cybersecurity," that there was still a concern that cybersecurity could not be left entirely in the hands of the private sector).

[154] In fact, the flourishing software security market is so expansive that there is room enough for fake software security. For example, in June 2011, the FBI shut down a criminal gang who made seventy-two million dollars peddling fake security software. *See* FBI Targets Cyber Security Scammers, BBC (June 23, 2011, 11:14 AM), http://www.bbc.co.uk/news/technology-13887152.

[155] *See* IDC EMEA, *supra* note 31.

[156] In the United Kingdom, for example, Internet service providers such as Virgin Media, and financial institutions such as HSBC, do routinely offer free anti-virus and software security to their customers. For instance, Virgin Media provides free anti-virus to all subscribers. *See Virgin Media Security: Total*, http://shop.virginmedia.com/broadband/broadband-extras/Internet-security.html (last visited Oct. 21, 2011).

[157] *Id.*

[158] *See Computer Viruses and Civil Liability*, *supra* note 29, at 160 (noting that some of the most effective antivirus programs were available free of charge for private users, and

ranging from avast,[159] to AVG anti-virus.[160] *A fortiori*, the absence of effective software security is not a lack of private investment in software security *per se* as adumbrated by some computer security scholars,[161] but rather due to the latent defects or inherent vulnerabilities in software architecture.[162] I would therefore argue that if programmers could not design flawless software,[163] then neither the market, nor reputational incentive, nor the prospects of legal liability for insecure software,[164] could completely obviate the problem of software vulnerabilities or insecurity.

Third, by extrapolating Hardin's Commons Theory to rationalizing the perennial problems of software or network insecurity, there is an implicit assumption in Bohme's analysis that software or computer networks are as much a public good as is the hypothetical grazing commons.[165] If this were so, then the analogy would be at best erroneous because neither software nor computer networks is a public good as such,[166] even with the ubiquitous non-proprietary software in the network systems. Although the Internet originated from the publicly funded research at universities by the government of the United States,[167] it is now being run daily by disparate private for-profit businesses and corpo-

that free antivirus programs such as VirusScan made the Virus Bulletin's 100 Percent Award list and received various honours for their efficacy).

[159] *See* AVAST, http://www.avast.com/free-antivirus-download (last visited Oct. 21, 2011).

[160] *See* AVG, http://free.avg.com/gb-en/homepage (last visited Oct. 21, 2011).

[161] *See, e.g.*, Anderson, *supra* note 35, at 631.

[162] *See, e.g.*, Cusumano, *supra* note 136, at 26 (asserting that software could never be completely free from vulnerabilities).

[163] *E.g.*, Bruce Schneier noted that it was hard to design and implement bug-free code. *See* SECRET & LIES, *supra* note 23, at 203.

[164] Some legal scholars are of the opinion that the introduction of tortuous liability rules for defective software, would lead to a better or error free software. *See, e.g.*, Beard, *supra* note 102, at 231; Carl Almond, *Should Vendors Be Liable for Security Flaws in Software?,* 2009 COMPUTER FRAUD & SECURITY 4-7 (2009) (noting that a regime of legal liability for insecure software could lead to legal entanglements and virtually unenforceable laws without being able to obviate the problem of software vulnerabilities).

[165] *See* Bohme, *supra* note 25, *See also* Hal R. Varian, *System Reliability and Free Riding,* 12 ECON. OF INFO. SECURITY 1-15 (L. Camp & Stephen Lewis eds., 2006).

[166] Although the Internet started out as an academic project at publicly funded United States universities computer science departments in collaboration with some corporate engineers, it was taken mainstream when private investors took it to the stock market. For instance, Yahoo! raised thirty-five million dollars at its 1996 first initial public offering of its shares. *See* ZITTRAIN, *supra* note 104.

[167] *See, e.g.*, Taiwo A. Oriola, *Regulating Unsolicited Commercial Electronic Mail in the United States and the European Union: Challenges and Prospects*, 7 TUL. J. TECH & INTELL. PROP. 113, 114 (2005) (noting that the Internet transmission control protocol (TCP) was standardized at Stanford University in 1973, and later refined at the University of Southern California in January 1978).

rations.[168] For, as Helen Nissenbaum rightly noted, the "commercial marketplace and supporting institution of private property" have hijacked the Internet, and private property has "leached into and became central to all the multiple layerings of the online world, from physical infrastructure upwards."[169] Indeed, the making and ownership of the entire digital infrastructure ranging from personal, organizational, to corporate computers and servers needed for network systems, and the concomitant software are essentially private goods in the hands of publicly quoted multinational corporations giants ranging from Google, Yahoo!, Apple, to Microsoft.[170] Also, the growing global software security market is symptomatic of the increasing private investments in software or network security,[171] a fact, which arguably belies the attribution of software insecurity to free-riding as extrapolated from Hardin's Commons Theory.[172]

A fortiori, I would argue that the theories of information asymmetry and the tragedy of the commons are ill-suited for rationalizing software insecurity. Furthermore, analogizing the software market with the used car market, and comparing software or network systems with grazing commons, is incongruous and no more than attempts to shoehorn what is otherwise an inherent technical failure into market failure dynamics. It would thus appear that the only remedial course for the software vulnerabilities imbroglio is vulnerabilities detection research and corrective patch-up,[173] as vulnerabilities could not be completely eliminated.[174] The third part of this paper will review the literature on the art of hacking or vulnerabilities detection research and how the hacking enterprise is continually fed by the growing legitimate and underground market in software vulnerabilities.

[168] *See* Nissenbaum, *supra* note 17 (noting how global "telecommunications corporations took over from government agencies' possession and oversight of the fiber-optic cables, airwaves, and switches". . .and how commercial Internet service providers as well as "cable and phone companies became dominant providers of popular online access.").

[169] *Id.*

[170] In fact, Yahoo! was the first publicly listed Internet ventures, which raised thirty-five million dollars at its 1996 public offering. *See* ZITTRAIN, *supra* note 104, at 32.

[171] As noted earlier, the United States leads the global multi-million software security market, seconded by the European market, which was estimated at 15.5 billion Euros in 2010. *See* IDC EMEA, *supra* note 31.

[172] *See* Bohme, *supra* note 25.

[173] *See* Pu Li & H. Raghav Rao, *An Examination of Private Intermediaries' Role in Software Vulnerabilities Disclosure*, 9 INFO. SYS. FRONTIERS 531 (2007) (noting that there could be no end to the discovery and disclosure of vulnerabilities).

[174] *See* Cusumano, *supra* note 136, at 26 (noting that software could never be completely rid of vulnerabilities).

III. VULNERABILITIES DETECTION RESEARCH AND THE ART OF HACKING.

[t]he good citizen is everything that the hacker is not.[175]

Vulnerabilities typically lie dormant in software systems, and are usually discovered by professional researchers or hackers through detection research.[176] Vulnerabilities detection research is technically known as software penetration testing,[177] a security and quality assurance testing designed to break into a network to demonstrate that it could be done, or simply to detect vulnerabilities or security flaws.[178] Computer security firms often offer penetrating testing services to their clients,[179] therefore, hacking could be especially commissioned by software vendors,[180] or be undertaken *suo motu* by hackers, who might exploit vulnerabilities for criminal ends, or sell vulnerabilities to software vendors or third parties. Vulnerabilities, especially "zero-day" variety, have been described as "hot commodities",[181] and are highly sought after by software vendors in order to design necessary corrective patches for critical programming flaws.[182] Typically, hackers could sell vulnerabilities to software vendors on the black-market,[183] or blackmail vendors with threats of disclosing vulnerabilities, or recklessly publishing vulnerabilities without regards to the prospects that they could fall into the wrong hands of someone who might exploit them to attack computer or network

[175] *See* Nissenbaum, *supra* note 17.

[176] *See* Schneier on Security, *supra* note 25,

[177] *See Id.*

[178] *See Id.; see also* Brad Arkin et al., *Software Penetration Testing*," 3 IEEE Security & Privacy 84-87 (2005) (noting that the essence of software penetration testing was quality assurance and security testing).

[179] *E.g.*, hackers from penetrating testing firm, *Netragard* were recently hired to break into a customer's computer systems. *See* Dan Goodin, *Hacker Pierce Network With Jerry-Rigged Mouse: Mission Impossible Meets Logitech*, Register (June 27, 2011), http://www.theregister.co.uk/2011/06/27/mission_impossible_mouse_attack/.

[180] For instance, Mozilla, the maker of the Firefox browser, started offering bounty for vulnerabilities in 2004 and have paid an average of forty thousand dollars per year since then. Also Google has reputedly paid over fifty thousand dollars as bounties for vulnerabilities discovered from their Chrome browser. *See* Paul Coletti, *Life as a Bug Hunter*, BBC (June 18, 2011), http://www.bbc.co.uk/news/technology-13814395.

[181] *See* Schneier on Security, *supra* note 25.

[182] *See* Stewart Baker, *Vulnerability Comes Cheap?,* INSTAPUNDIT.COM (May, 22 2010, 10:16 AM), http://pajamasmedia.com/instapundit/vulnerability-comes-cheap-theres-an-increasingly-open-market-in-computer-vulnerabilities-%C2%A0-crook/ (noting that there was an increasing open market in computer vulnerabilities, and that while criminals sought them to construct "zero-day" exploits, security firms sought them to improve their detection programs).

[183] *See From Black Market to Free Market*, BusinessWeek (Aug. 22, 2005) http://www.businessweek.com/magazine/content/05_34/b3948022_mz011.htm (noting that even computer security firms were offering to buy vulnerabilities from hackers who bedevilled them).

systems.[184] Invariably, what a hacker does with, or where he sells new vulnerabilities depends on whether they are a good hacker or a bad hacker.[185]

A good hacker is a professional computer security researcher who, *suo motu*, breaks into computer systems or is paid commissioned to break into systems purely for altruistic reasons, and would normally follow responsible disclosure procedures by contacting and either selling the vulnerabilities to the vendors or simply accepting recognition or acknowledgement for vulnerabilities discovery.[186] The wholesome image of the good hacker is put succinctly by Nissenbaum:

> To hack was to find a way, any way that worked, to make something happen, solve the problem, invent the next thrill. There was a bravado associated with being a hacker, an identity worn as a badge of honor.[187]

On the other hand, a bad hacker is a person who typically would breaks into computer systems for malicious reasons ranging from acts of vandalism,[188] theft of sensitive information, to acts of terrorisms.[189] Ironically, notwithstanding the existence of professional good hackers

[184] *See* SCHNEIER ON SECURITY, *supra* note 25; Abhishek Bhuyan, *2010 in Review: The Vulnerability Landscape*, TREND MICRO (Dec. 22, 2010), http://blog.trendmicro.com/2010-in-review-the-vulnerability-landscape/ (noting that criminals routinely take advantage of vulnerabilities to insert malware into computer systems).

[185] *See* SCHNEIER ON SECURITY, *supra* note 25.

[186] *See* Miller, *supra* note 113 (noting that evidence has shown that the best researchers were motivated by monetary gain rather than prestige).

[187] *See* Nissenbaum, *supra* note 17 (noting that hackers were even indirectly sponsored by the United States Defense Advanced Research Projects Agency, the funding agency widely credited for sponsoring the invention of the Internet.).

[188] For example, Igor Blinnikov, hacked into a 20x30 foot advertising video billboard by the Interior Ministry in central Moscow, and replaced contents with pornography clips. Traffic was temporarily halted as distracted motorists and horrified passersby watched the billboard stream pornographic images in broad daylight. Igor was jailed for six years for his little stunt. *See* Daily Mail Reporter, *Russian Jailed for Six Years for Hacking into Advertising Server and Making Electronic Billboard Show Porn to Motorists*, MAIL ONLINE, (Mar. 24, 2011) http://www.dailymail.co.uk/news/article-1369578/Russian-hacker-Igor-Blinnikov-jailed-6-years-porn-billboard.html; *see also* Eugene H. Spafford, *Are Hacker Break-ins Ethical?, in*, COMPUTERS, ETHICS, AND SOCIETY 64, 66-67 (M. David Ermann and Michele S. Shauf eds., 2003) (describing the various motivations and excuses for hacking ranging from boredom to educational).

[189] *See* CLAY WILSON, CONG. RESEARCH SERV., RL32114, COMPUTER ATTACK AND CYBER TERRORISM: VULNERABILITIES AND POLICY ISSUES FOR CONGRESS 1-42 (2003), *available at* http://www.fas.org/irp/crs/RL32114.pdf (noting that hackers do relentlessly search for new vulnerabilities and that Al Qaeda members had reportedly used Internet telephony to communicate with other members and that one Ramzi Yousef, who was serving life sentence for World Trade Center bombing, had used sophisticated encryption to protect his data from law enforcement); *see also* Tara Mythri Raghvan, *In Fear of Cyberterrorism: An Analysis of the Congressional Response*," U. ILL. J.L., TECH. & POL'Y 297, 304-12 (2003) (noting Congressional policy initiatives and legislations passed to combat cyberterrorism following the September 11 2001 terror attacks in the United States).

who hunt for vulnerabilities for purely altruistic reasons, the art of hacking is nevertheless suffused with negative connotations, due to a dramatic and fundamental shift in the social image and moral status of hackers from "the heroes of the computer revolution" to marauding criminals in cyberspace.[190] According to Nissenbaum, this transformation in the traditional image of hackers has alienated and banished professional hackers into the margins with the concomitant loss of their identity, ideas, and ideology.[191] On the other hand, the contemporary image of the hacker as a social misfit unfairly continues to endure and predominate, leading Nissenbaum to opine that "the good citizen is everything that the hacker is not."[192] Notably, the bad boy image of the hacker has been given a fillip by recent spates of high profile attacks on the websites of the CIA, Sony, and the United States Senate, amongst numerous others.[193]

Nevertheless, whether hacking is motivated by pure greed or altruism, it is set to continue due in part to the lucrative market in software vulnerabilities, which do vary, depending on the severity of vulnerabilities on offer.[194] Typically, "zero days" vulnerabilities are the most valuable, because they are new, and previously unknown.[195] For instance, government agents, who reputedly often vie with criminals to offer the most money for the most critical or severe vulnerabilities on the market, were known to have offered as much as one million dollars for a single vulnerability that was adjudged extremely critical and valuable.[196] Furthermore, Mozilla, the makers of Firefox web browser, who were the first

[190] *See generally* Nissenbaum, *supra* note 17.

[191] *See id.*; *see also* Debora Halbert, *Discourses of Danger and the Computer Hacker*, 13 INFO. SOC'Y 361, 362-66 (Dec. 1997) (noting that in the early days, hacking was a benign enterprise framed by "hacker ethics" rooted in the belief that information should be free for all. However, the hacker ethics and counter-culture would soon clash with the economic philosophy underlying the centralization and private ownership of computer networks and the Internet by entrepreneurs. This was to precipitate a rebellion by elements of the hacking community who turned to sabotage and crime).

[192] *See* Nissenbaum, *supra* note 17.

[193] *See generally* the discussions in part two of the paper on various high profile hacking incidents, and specifically, *see* Bradshaw, *supra* note 50.

[194] Severe vulnerabilities tend to induce higher customer loss and more economic damage for software vendors. *See* Rahul Telang & Sunil Wattal, *An Empirical Analysis of the Impact of Software Vulnerability Announcements on Firm Stock Price*, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING 544, 548 (Aug. 2007), *available at* http://www.heinz.cmu.edu/~rtelang/tse_published.pdf.

[195] *See* Jordan Roberson, *Google Attack Spotlights 'Zero-Day' Black Market*, MSNBC (Jan. 28, 2010, 7:34 PM), http://www.msnbc.msn/id/3513312/ns/technology_and_science-security/t/google-attack-spotlights-zero-day-black-market/ (noting that if a vulnerability is not a zero day, it is not valuable at all).

[196] *Id.* One Charlie Miller, a former National Security Agency analyst was allegedly paid fifty thousand dollars by an unspecified United States government contractor for a bug that he found in a version of the Linux operating system.

to establish a bug bounty program in 2004, reputedly paid an average of forty thousand dollars per year to vulnerabilities researchers.[197] Moreover, online auction houses selling vulnerabilities to the highest bidder, such as WabiSabiLabi, have proliferated in recent times.[198] The vulnerabilities market is also amorphous and broadly divisible into two types: legal and illegal or illegitimate markets.[199] Whilst certain security firms such as iDefense and TippingPoint openly pay researchers for vulnerabilities on the budding legal or legitimate market, hackers have been known for years to peddle vulnerabilities on the black market.[200] The following section will examine the legality of vulnerabilities detection research within the contexts of cyber trespass, computer abuse legislations, and intellectual property law. Also, the extent to which the aforesaid legal regimes could hamper vulnerabilities research and both the legitimate and illegitimate market will be analyzed.

IV. MODALITIES FOR AND EFFECTS OF VULNERABILITIES DISCLOSURE.

It is sacrosanct that vulnerabilities detection research is invaluable to computer and network security, as it facilitates the discovery and disclosure of latent 'zero day' or new vulnerabilities that could be exploited by unscrupulous hackers if left uncorrected. Ironically however, other than security concerns, vulnerabilities do not impact software functionality as such, because software quality measurements are essentially framed "in terms of reliability and integrity of the source code," with the primary aim of passing the functionality muster.[201] Indeed, most quality models such, as IS09126, were designed to ensure software functionality, whilst largely glossing over software security issues.[202] In other words, the traditional industry concept of software quality is more attuned to achieving software functionality than software security, even though functionality as a measure of quality does not equal more secure

[197] *See* Coletti, *supra* note 180 (noting that Mozilla's top earner was a German student who purportedly earned thirty thousand dollars from a series of discoveries).

[198] Darrell Dunn, *Vulnerabilities for Sale: What SMEs Need to Know about Online Auction Sites*, PROCESSOR 1, 10 (Feb. 22, 2008), *available at* http://www.processor.com/articles/PDFMagazine/Good/P___3008.PDF?GUID=.

[199] The legal or legitimate market would typically comprise "bug challenges" where companies offer monetary rewards for new vulnerabilities and vulnerability brokers who offer monetary compensation for new vulnerabilities. Suggestions have also been made for "exploit derivatives" and "cyber insurance" as the alternative market for vulnerabilities trading. The illegal market on the other hand typically comprises malicious hackers and criminals. For a discussion, *see* Radianti & Gonzalez, *supra* note 16.

[200] Miller, *supra* note 113.

[201] *See* Telang & Wattal, *supra* note 194, at 548.

[202] *Id.*

or flawless software.[203] Perhaps it is easier for programmers to benchmark or engineer software functionality than software security, which is arguably very relative and often intractable? Nevertheless, programmers are now working on how to better integrate quality and security into software design templates,[204] but given the perennially recurring vulnerabilities problems, designing flawless and vulnerabilities free software is set to be a long term if not impossible challenge.[205]

Paradoxically however, software vulnerabilities are rarely self-evident, and it would take detection and research to uncover or unearth vulnerabilities.[206] In other words, if there were no vulnerabilities detection research and disclosure, there would be no vulnerabilities and the concomitant software security problems.[207] However, some analysts are of the opinion that vulnerabilities disclosure does have dual, if contradictory outcomes in that it is as capable of obstructing as it is of securing computer and network security.[208] According to Peter Swire, vulnerabilities disclosure could impede computer security by facilitating attacks by malicious hackers through the disclosed knowledge of vulnerabilities they would otherwise not have had. On the other hand, vulnerabilities disclosure could also help vendors by teaching them how best to fix critical vulnerabilities.[209]

Irrespective of the unintended consequences, vulnerabilities detection research and disclosure are all but inevitable being that it is standard industry practice to test for weaknesses and potential security flaws in software products at both pre-market[210] and post-market prod-

[203] *See generally* Charles P. Pfleeger, *The Fundamentals of Information Security*, IEEE SOFTWARE 15-17 (Jan. - Feb. 1997).

[204] *See* Telang & Wattal, *supra* note 194, at 548.

[205] *E.g.,* Bruce Schneier noted that it was hard to design and implement bug-free software code. *See* SECRET & LIES, *supra* note 23, at 205.

[206] *See* SCHNEIER, *supra* note 25, at 261.

[207] It would appear logical to assume that if there were no vulnerabilities detection research and disclosure, would be malicious hackers would have no knowledge or means to exploit computer and network systems for criminal ends.

[208] *See generally* Peter P. Swire, *A Model for when Disclosure Helps Security: What is Different about Computer and Network Security?*, *in* LAW AND ECONOMICS OF CYBERSECURITY 29, 30 (Mark F. Grady & Francesco Parisi eds., 2006).

[209] *Id.*

[210] In industry parlance, this is known as "penetrating testing." For a discussion, *see* SCHNEIER ON SECURITY, *supra* note 25, at 261-62. However, penetrating testing cannot completely eliminate all vulnerabilities, and due to the generally perceived haste of vendors to get software products onto the market, there is a belief amongst industry watchers that most vendors prefer to follow the policy of "sell today and fix later", or "I'd rather have it wrong than have it late." Moreover, pre-market product testing for security flaws is said to be very expensive, a factor that could be a disincentive to a vendor from thoroughly testing for security flaws. *See* Telang & Wattal, *supra* note 194, at 544-45.

ucts launch phases.[211] Furthermore, the lucrative (legitimate and illegitimate) market in vulnerabilities is a veritable incentive for malicious and altruistic hackers alike to engage in vulnerabilities detection research following the market launch of software products.[212] Moreover, rival software vendors are known to routinely test and disclose vulnerabilities in rivals' software "as a strategic weapon against competitors" with a view to negatively impacting rivals' stock price and market value.[213]

Whilst vulnerabilities detection research is inevitable, the concomitant disclosure is totally unregulated and highly decentralized.[214] Traditionally, the Computer Emergency Response Team ("CERT")[215] relays vulnerabilities information, which it gets for free from benign vulnerabilities researchers to software vendors for appropriate corrective patch, after which software users would be contacted to upgrade their system.[216] However, for other intermediate vulnerabilities information traders, vulnerabilities disclosure could either be "limited" or "full."[217] A limited disclosure occurs where benign independent security intermediaries report vulnerabilities directly to software vendors so that vendors have sufficient time to release updates and corrective patch for vulnerabilities.[218] On the other hand, a full disclosure occurs where independent security analysts promptly post vulnerabilities to a public listing, such as BUGtrack,[219] forcing vendors to quickly fix the disclosed vulnerabilities[220] in

[211] *See* Martin, *supra* note 27, at 34-35 (noting that post-market launch testing of software is routinely conducted by vendors who employ "vulnerability scanners" or "tools" to monitor the health of software applications by scanning for errors; and that the comparative advantages of using vulnerability scanners, which include testing and comparing "software version information and configuration settings with an internal list of vulnerability data").

[212] Vulnerabilities are very expensive depending on their severity. As noted above, the United States government reputedly offered as high as one million dollars for a very valuable vulnerability. *See* discussion *infra* Part III; *see also* Roberson, *supra* note 195.

[213] *See* Telang & Wattal, *supra* note 194, at 548 (citing the *Wall Street Journal* report that hinted that software vendors were using vulnerabilities disclosure as a strategic weapon against competitors with a view toward negatively affecting their stock price).

[214] *See Id.* at 546 (noting that there were no legal guidelines for the disclosure of vulnerabilities by the discoverers).

[215] The Computer Emergency Response Team Clearing Centre originated from a program funded by the United States Department of Defense under the auspices of Carnegie Mellon University, and continues to exist "as a clearinghouse for information about viruses and other network threats"). *See* ZITTRAIN, *supra* note 104, at 32.

[216] *See* Karthik Kannan & Rahul Telang, *Market for Software Vulnerabilities? Think Again*, 51 MGMT. SCI. 726, 726-27 (May 2005).

[217] *See* Telang & Wattal, supra note 194, at 546.

[218] *See Id.* (noting that the standard industry recommended period within which vendors could patch up disclosed security flaws was thirty days).

[219] *See* BUGTRACK, http://www.bugtrack.net/?gclid=CJyfwoGv86kCFUwc4QoduUh-SYA (last visited Oct. 21, 2011).

order to pre-empt attackers who would not hesitate to exploit the vulnerabilities.[221] According to Rahul Telang et al., the standard industry recommended period within which software vendors could patch up disclosed vulnerabilities or security flaws was within 30 days from the date of initial disclosure.[222] This would no doubt increase the pressure on software vendors, and a very good reason for vendors' reluctance to openly disclose vulnerabilities in their products, and would only do so to pre-empt a third-party from recklessly disclosing vulnerabilities.[223]

Software vendors' guarded stance, with regards to vulnerabilities disclosure, is especially understandable because a reckless or indiscrete vulnerabilities disclosure could lead to unscrupulous exploitation of vulnerabilities that could potentially aggravate the consequential damage and financial loss to software vendors and their clients.[224] The possible harm or damage to software vendors and their customers could range from theft of customers' sensitive data[225] to harm to vendors' reputation, with concomitant depreciation in the market value of vendors' shares and stocks.[226] For example, it is estimated that a software vendor could on average lose 0.63 per cent of its market value, the equivalent of eight-hundred sixty million dollars on the day of vulnerabilities announcement.[227] Also, the annual estimated economic damage caused by the attacks exploiting software vulnerabilities was assessed at sixty billion dollars by a NIST Study.[228] *A fortiori*, given the prospects for real financial and reputational loss for software vendors and their clients, and in light of the popular legal aphorism that "where there is a right there

[220] *See* Telang & Wattal, *supra* note 194.

[221] *See* Sam Ransbotham, *An Empirical Analysis of Exploitation Attempts based on Vulnerabilities in Open source Software*, Ninth Workshop on the Econ. of Info. Security 1, 4 (June 2010), *available at* http://weis2010.econinfosex.org/papers/session6/weis2010_ransbotham.pdf.

[222] *See* Telang & Wattal, supra note 194, at 546.

[223] *See id.* at 548.

[224] *See id.* at 546.

[225] For instance, malicious hackers could target sensitive individual or corporate secrets such as credit cards details and business secrets. For example, a research conducted by security firm McAfee in 2011 revealed that cybercriminals were increasingly targeting and stealing sensitive corporate data, trade secrets, and intellectual property rights to order. *See Hackers Target Business Secrets*, BBC (Mar. 28, 2011, 3:01 AM), http://www.bbc.co.uk/news/technology-12864666.

[226] For a discussion, *see* Telang & Wattal, *supra* note 194, at 548. (noting that vulnerabilities disclosure "adversely and significantly" affected the stock performance of software vendor, and that vulnerabilities disclosure could lead to customer dissatisfaction and the loss of vendors' reputation).

[227] *Id.* at 544-46.

[228] Kannan & Telang, *supra* note 216, at 726.

must be a remedy" (ubi jus ibi remedium),[229] what possible legal rights and redress might stem from vulnerabilities research and disclosures? This paper will attempt to provide an insight into these questions in the following paragraphs.

V. THE LEGALITY OF VULNERABILITIES RESEARCH AND DISCLOSURE.

Although both the government and the software industry actively support and promote cybersecurity research,[230] it is crucial to discuss and analyze possible legal fallouts of vulnerabilities research and disclosure. This is especially so because software and network systems are inherently proprietary, and are invariably potentially protectable by a variety of legal regimes ranging from the tort of trespass, computer misuse legislations, intellectual property law, to a host of legal regimes safeguarding property rights. Thus, vulnerabilities detection research and disclosure are invariably potentially entangled in a web of challenging legal obstacles. This section will analyze and provide insights into the intricate dilemmas posed by these legal regimes, and how they could be safely negotiated by benign professional vulnerabilities researchers and ensure responsible vulnerabilities disclosures, while simultaneously warding off malicious exploitation of vulnerabilities. An insight will also be provided into the diverse legal scenarios and possible cause of action cum legal remedies for software vendors and their customers.

A. Could Vulnerabilities Research be Tantamount to Cyber Trespass?

Starting from the premise that computer and network systems are proprietary in nature, it is logically apposite to consider the legal ramifications of the tort of trespass for vulnerabilities research and disclosures. However, any meaningful academic analysis of the question as to

[229] For a discussion of a case where the Supreme Court invoked the principle to frame a cause of action against the federal government where none ostensibly existed, see Bivens v. Six Unknown Named Agents, 403 U.S. 388 (1971) (Justice Brennan noted that the court would invoke a private right of action for monetary damages in the absence of a federal remedy that could vindicate a constitutional right. The invocation of the right was premised on the principle that for every wrong, there is a remedy).

[230] For example, the United States continually funds cybersecurity research through the National Science Foundation, DARPA, the Department of Homeland Security, and allied agencies. *See* Toward a Safer and More Secure Cyberspace 237-40 (Seymour E. Goodman and Herbert S. Lin eds., 2007) (discussing various government-backed funding programs for cybersecurity research, noting in particular a $175 million cybersecurity research portfolio for the 2007 financial year, which "focused on research and advanced development to prevent, resist, detect, respond to, and/or recover from actions that compromise or threaten to compromise the availability, integrity, or confidentiality of computer-based systems").

whether or not vulnerabilities research could be tantamount to trespass would, of necessity, entail a relatively detailed discussion of key antecedent cases, and the principles underlying the extension of the traditional rules of trespass to chattel, to the intangible and ephemeral electronic space by courts in the United States. Using analogous case law and statutes, this paper will assess the current state of the law of cyber trespass, its applicability or otherwise to vulnerabilities research, and the extent to which it could impact vulnerabilities research and disclosure in the United States.

The common law tort of trespass to chattel or goods - *trespass de bonis asportatis* - occurs when a person, without any lawful justification, intentionally,[231] directly, and physically interferes with the goods in the lawful possession of another person, either by taking or damaging the goods.[232] The legal academia is unanimous on the view that trespass to goods ought to be actionable *per se*, i.e., the person who is entitled to possession of the goods could sue the trespasser without proof of any actual damage to the goods by the alleged act of trespass.[233] In other words, the trespasser could be liable by merely touching or moving the goods without prior authorization from the person in lawful possession.[234] This view is supported by the English case of *Kirk v. Gregory*,[235] where a woman was held liable in nominal damages for trespass to goods. She had merely moved some jewelry belonging to a recently deceased person from one room in the house to another where it was eventually stolen.[236]

In the United States however, the key elements of the common law tort of trespass to chattel are encapsulated in section 217 of the Restate-

[231] An interference with goods, which is not intentional, would not amount to trespass to goods. *See, e.g.*, Nat'l Coal Bd. v. J.E. Evans Ltd., [1951] 2 All ER 310 (Eng.), where the UK Court of Appeal held that a contractor, whose employee damaged the plaintiff's cable during an excavation, was not liable for trespass because the act was unintentional, and there was no liability for an accidental trespass to goods.

[232] An indirect interference with goods would be more suited to the tort of negligence. For a discussion, see VIVIENNE HARPWOOD, MODERN TORT LAW 365 (7th ed. 2008).

[233] *See* Mary W.S. Wong, *Cyber-Trespass and 'Unauthorized Access' as Legal Mechanisms of Access Control: Lessons from the US Experience*, 15 INT'L J.L. & INFO. TECH. 90, 92-3 (Aug. 2006) (citing the unanimity of views amongst authors of *Salmond and Heuston on the Law of Torts*, *Winfield & Jolowicz on Tort*, and *Street on Torts*, that trespass to goods should be actionable *per se*, without proof of damage). *See also* Shyamkrishna Balganesh, *Common Law Property Metaphors on the Internet: The Real Problem with the Doctrine of Cybertrespass*, 12 MICH. TELECOMM. & TECH. L. REV. 265, 275-76 (2006).

[234] The person entitled to possession may already be in actual possession or entitled to immediate possession. *See e.g.*, Lord Blanesburgh's dictum in William Leitch v. Leydon [1931] A.C. 90, 106 (Eng.), which is often cited for the authority that a mere touching of goods or chattel is actionable *per se*.

[235] Kirk v. Gregory, [1876] 1 Ex D 55 (Eng.).

[236] *Id.*

ment (Second) of Torts, which defines trespass to chattel as: "intentionally. . .dispossessing another of the chattel, or using or intermeddling with a chattel in the possession of another."[237] Also, section 218 further delimits the parameters of liability for trespass to chattel by prescribing the following conditions:

> (a) the person in lawful possession is dispossessed; (b) the intermeddling impaired the chattel's quality, physical condition, and value; (c) the person in possession is deprived of the use of the chattel for a substantial time; or (d) bodily harm is caused to the possessor, or to some person or thing in which the possessor has a legally protected interest.[238]

However, under United States tort law, while a harmless interference with a chattel may still technically be tantamount to trespass, it would not give rise to a legally recoverable claim, even if the act of trespass was intentional.[239] Significantly, the non-availability of nominal damages for harmless trespass to chattel in the United States runs counter to the position in the United Kingdom, as exemplified by *Kirk*[240] and section 3(a), (b) & (c) of the Torts (interference with Goods) Act 1977.[241]

Most significantly however, in the United States, the tort of trespass to chattel has been extrapolated to frame a new cause of action in cyber trespass for unauthorized "intrusions in the form of electronic signals to computer systems connected to the Internet".[242] Significantly, whilst the tort of cyber-trespass is unique to the United States,[243] the general notion of "unauthorized access" to computer and network systems on which the tort of cyber-trespass is premised is by no means unique, and is analogous to the provisions of section 1030(a) (2) & (a) (3) of the United States' Computer Fraud and Abuse Act 1986, which criminalizes know-

[237] *See* RESTATEMENT (SECOND) OF TORTS § 217 (1965).

[238] *See id.* § 218.

[239] *See id.*, comment (e), which provides as follows:

> the interest of a possessor of a chattel in its inviolability, unlike the similar interest of a possessor of land, is not given legal protection by an action for nominal damages for harmless intermeddling with the chattel. In order that an actor who interferes with another's chattel may be liable, his conduct must affect some other and more important interest of the possessor.

[240] *See* Kirk, [1876] 1 Ex D at 55. *See also* Neave v. Neave, [2002] EWHC 784 (Q.B.)(Eng.), where the defendant, who took a number of historic cars belonging to his widowed mother, was held liable for trespass, and ordered to deliver up the cars and pay £3000 damages to his mother. *See* Wong, *supra* note 233, at 95.

[241] Section 3 of The Torts (Interference with Goods) Act generally provides for the types of reliefs or remedies available to claimants suing for wrongful interference with goods. These range from an order for delivery of the goods, consequential damages to damages. *See* The Torts (Interference with Goods) Act, 1977, 32 (Eng.).

[242] *See* Wong, *supra* note 233, at 91 (discussing the uniqueness of the tort of cyber-trespass to the United States, and speculating on the implications of its adoption by other common law countries).

[243] *Id.*

ingly accessing computer without authorization or exceeding authorized access;[244] and the provisions of section 1 of the United Kingdom's Computer Misuse Act 1990, which again penalizes unauthorized access to computer materials.[245]

Significantly, in the United States, the doctrine of trespass to chattel was first applied outside of its traditional and familiar *terra firma* turfs to intangible and ephemeral virtual electronic realm in the case of *Thrifty-Tel v. Bezenek*,[246] where a misuse of telephony via remotely operated electronic signals was held tantamount to trespass to chattel.[247] Two teenagers had used computer software to illegally circumvent the plaintiff, Thrifty-Tel's, access and authorization codes to facilitate the making of over 1,300 illicit free long distance telephone calls.[248] The unauthorized automated telephone calls choked and overburdened the plaintiff's telephony system, culminating in denial of access to some fee-paying subscribers.[249] The plaintiff sued for trespass to property, and the Court found the defendants liable for trespass, holding, *inter alia*, that "[i]n our view, the electronic signals generated by the Bezenek boys' activities were sufficiently tangible to support a trespass cause of action."[250]

In arriving at its decision, the California Court of Appeal in *Thrifty-Tel* drew on Prosser and Keeton's views that trespass to chattel could arise from a mere interference or use that was short of outright dispossession.[251] The Court further noted that the fact that the plaintiff's chattel was intangible was immaterial and no obstacle to finding defendants liable for trespass, and that the 'electronic signals' through which the plaintiff's intangible property was accessed without authorization were 'tangible enough' to ground a case for trespass to chattel.[252] Most significantly, the Court glossed over the traditional requirement of proof of 'actual damage' by plaintiff, a precondition for a cause of action in trespass to chattel under the American tort law,[253] and found the defendant lia-

[244] *See* The Computer Fraud and Abuse Act 1986, 18 U.S.C. §1030 (as amended by the USA Patriot Act 2002 and the Identity Theft Enforcement and Restitution Act 2008).

[245] *See* The Computer Misuse Act, 1990 (Eng.) (as amended by the Police and Justice Act, 2006 (Eng.)). *See also* Ian Walden, *Computer Crime and Information Misuse*, *in* COMPUTER LAW: THE LAW AND INFORMATION OF INFORMATION TECHNOLOGY 553, 564-70 (Chris Reed & John Angel eds., 6th ed. 2007) (discussing the statutory categories of unauthorized access to computer and network systems in the United Kingdom).

[246] *See* Thrifty-Tel v. Bezenek, 46 Cal. App. 4th 1559, (1996).

[247] *Id*. at 1565-66.

[248] *Id*.

[249] *Id*.

[250] *Id*.

[251] *See* PROSSER AND KEETON ON THE LAW OF TORTS (5th ed. 1984).

[252] *See* Thrifty-Tel, 46 Cal. App. 4th at 1567.

[253] *See* RESTATEMENT (SECOND) OF TORTS § 218 (1965).

ble in trespass, despite the Court's categorical findings that the plaintiff did not proffer evidence of any actual losses.[254] In other words, the defendants were held liable for trespass despite the intangibility of the plaintiff's chattel, and the absence of evidential proof of actual physical damage. These significant derogations from the traditional requirements for liability for trespass to chattel in the United States were to pave way for the subsequent judicial extension of the tort of trespass to chattel into cyberspace in the United States.

The first of the cases to apply the principles of the tort of trespass to chattel to the Internet or cyberspace was the 1997 Southern District Court of Ohio case *CompuServe, Inc. v. Cyber-Promotions*, Inc.[255] CompuServe, Inc. ("CompuServe") had received several complaints from its subscribers threatening termination of their subscriptions if unsolicited electronic mails from Cyber Promotions, Inc. ("CyberPromotions") were not stopped.[256] Cyber Promotions specialized in marketing unsolicited electronic mails in bulk, and continued to send unsolicited emails to CompuServe customers by circumventing all anti-spam measures and ignoring repeated demands by CompuServe that Cyber Promotions should cease its unauthorized activities.[257] CompuServe sued Cyber Promotions for trespass to chattel. The Court in *CompuServe* acknowledged the traditional requirements of the tort of trespass to chattel, which mandated the plaintiff to prove actual damage to property, or harm to property, or a diminution in property's condition, quality, or value due to the defendant's actions.[258] The Court then proceeded to find Cyber Promotions liable for trespass on grounds that the following damage occurred to CompuServe property: first, the multiple e-mail messages sent by Cyber Promotions took up a substantial part of disk space on CompuServe's servers, and eroded their processing power, thereby diminishing the value of the servers.[259] Second, the inundation of CompuServe's customers with unsolicited electronic mails by Cyber Promotions led to numerous complaints from the customers and caused damage to the plaintiff's quality of service, reputation and goodwill.[260]

[254] *See* Thrifty-Tel, 46 Cal. App. 4th at 1564-67.

[255] *See* CompuServe, Inc. v. Cyber-Promotions, Inc, 962 F. Supp. 1015 (S.D. Ohio 1997). For a discussion, see Wong, *supra* note 233, at 96.

[256] *See* CompuServe, 962 F. Supp. at 1018-19. For a discussion, *see* Oriola, *supra* note 167, at 118.

[257] *See* CompuServe, 962 F. Supp. at 1019.

[258] *Id.* at 1020-22.

[259] The District Court found that the erosion of the processing power of the plaintiff's servers constituted damage notwithstanding the absence of any physical damage to the servers. *See id.* at 1022.

[260] According to the court, the plaintiff's goodwill was a "legally protected interest" and its loss was a form of damage. *Id.* at 1023. Most significantly, the tort of trespass became a potent weapon of choice in the fight against unsolicited electronic mails in the

While these were not the run-of-the-mill trespass to chattel cases, the *CompuServe* decision re-echoed *Thrifty-Tel*, and underscored the seeming malleability of the tort of trespass for reining in unwanted electronic interference and computer intrusions in cyberspace, as exemplified by subsequent decisions in *eBay, Inc. v. Bidder's Edge, Inc.*,[261] *Register.com, Inc., v. Verio, Inc.*,[262] and *Oyster Software Inc. v. Forms Processing Inc., et al.*,[263] In *Oyster Software*, for example, the defendants, without authorization used robots to trawl the plaintiff's website for meta tags, which were later installed on the defendants' website. The plaintiff sued the defendant for trespass. The United States District Court for the Northern District of California held that in order to establish trespass to computers, a "plaintiff must demonstrate that (1) defendant intentionally and without authorization interfered with plaintiff's possessory interest in the computer system; and (2) defendant's unauthorized use proximately resulted in damage to plaintiff."[264] However, on evidence, the plaintiff only suffered negligible damage from the defendants' activities, and one of the issues for determination before the court was whether a cause of action could still be grounded in trespass on evidence of negligible injury to the plaintiff?[265] The Court answered in the affirmative as follows:

> Oyster concedes that Top Ten's robots placed a 'negligible' load on Oyster's computer system. Oyster asserts that Top Ten's copying of its meta tag is, nonetheless, sufficient to prevail on its trespass claim. The court agrees. . . Therefore, the court declines to dismiss Oyster's trespass claim on the grounds that Oyster has shown only minimal interference

late 1990s across the United States, in cases ranging from Am. Online, Inc. v. IMS, 24 F. Supp. 2d 548 (E.D. Va. 1998); Am. Online, Inc. v. LCGM, Inc., 46 F. Supp. 2d 444 (E.D. Va. 1998); to Am. Online, Inc. v. Prime Data Systems, Inc., 1998 WL 34016692 (E.D. Va. 1998).

[261] eBay, Inc. v. Bidder's Edge, Inc., 100 F. Supp. 2d 1058 (N.D. Cal. 2000). Bidder's Edge's use of robots to cull information on auctions from the website of eBay's without the latter's consent, was held sufficient use of property to justify the finding of trespass and the grant of an injunction, even though the robots only occupied a small percentage of eBay's computers. The court further noted that failure to grant an injunctive relief in the circumstances could potentially culminate in effective denial of access to ebay customers to ebay website.

[262] Register.com, Inc., v. Verio, Inc., 126 F. Supp. 2d 238 (S.D.N.Y. 2000), *aff'd*, 356 F.3d 393. The defendant, Verio, who were a competitor to Register.com, were using the latter's WHOIS server without authorization, to access the latter's customer base and data. The defendant ignored plaintiff's repeated warnings that they should stop their activities. However, despite the absence of any real interference with the plaintiff's server, the court held that the defendant's use of robots to trawl the plaintiff's server was tantamount to trespass because the robots searches caused harm to the server and improperly strained the server's capacity.

[263] Oyster Software Inc. v. Forms Processing Inc., No. C00-0724 JSC, 2001 U.S. Dist. Lexis 225220 (N.D.Cal. Nov. 3, 2001).

[264] *Id.* at 17-20.

[265] *Id.*

because Oyster has presented evidence of 'use' by Top Ten.[266]

However in *Intel Corp. v. Kourosh Kenneth Hamidi*,[267] the California Supreme Court steered clear of the reasoning in *CompuServe* and *Oyster Software* and narrowed the scope of the cause of action in tort for electronic trespass, by predicating tortuous liability on a demonstration either of actual damage or interference with the physical functionality of computer or network systems, or the likelihood of future damage to computer or network systems.[268] Hamidi had transmitted six e-mails, which criticized Intel Corp.'s ("Intel") employment practices to approximately thirty five thousand Intel employees via Intel's intranet despite Intel's objection. Intel conceded that Hamidi's emails did not cause any physical damage or disruption to their computer systems, but contended that the concomitant loss of economic productivity borne out of the time spent by employees on Hamidi's emails was sufficient to ground an action in trespass. The California Supreme Court disagreed with Intel's argument, and held that there could be no tortuous liability where the electronic interference in question "neither damages the recipient computer system nor impairs its function."[269] In arriving at this decision, the Court observed that neither the time Intel's employees spent reading the unsolicited electronic mails, nor the money spent by Intel in trying to block the continuing transmission of Hamidi's emails to its employees, constituted the type of injury necessary to sustain a trespass to chattels claim.[270]

Similarly, in 2006, in an action for summary judgment brought by the plaintiff, the United States Court of Appeals for the Fourth Circuit in *Omega World Travel Inc., v. Mummagraphics, Inc.*,[271] adopted the reasoning in the *Intel* case, and summarily dismissed the defendant's claim for trespass to chattel on grounds that they had failed to establish that they had sustained more than nominal damages following receipt of 11 unsolicited commercial electronic mail messages from the plaintiff, Cruise.com.[272] While noting that the absence of more than nominal injury was fatal to the defendant's case, the Fourth Circuit held further as follows:

> Even if Oklahoma law were to make trespass against chattels available for computer intrusions, Mummagraphics' claim cannot survive summary judgment because the courts that recognize trespass to chattels based upon computer intrusions do not allow "an action for nominal damages for harmless intermeddlings with the chattel." Because Mum-

[266] *Id.* at 19-20.

[267] Intel Corp. v. Hamidi, 30 Cal. 4th 1342 (2003).

[268] *Id.* at 1344-48.

[269] *Id.* at 1345-48.

[270] *Id.*

[271] Omega World Travel Inc., v. Mummagraphics, Inc., 469 F.3d 348 (4th Cir. 2006).

[272] *Id.* at 358-59.

> magraphics failed to submit any evidence that the receipt of eleven commercial email messages placed a meaningful burden on the company's computer systems or even its other resources, summary judgment was appropriate on this counterclaim.[273]

Significantly, although most states have not applied the tort of trespass to chattel to the intangible electronic realm of computing systems in the United States, it is arguable that any future applications would most likely prefer the reasoning in *Intel* to that of *CompuServe*, mainly because the requirement of actual damage by *Intel* is more in conformity with the tenor and expectations of Restatement (Second) of Torts section 218, comment (e), which *inter alia*, prohibits the award of nominal damages "for harmless intermeddling with the chattel".[274] Moreover, the United States' Congress' failure to pass into law a bill entitled *Securely Protect Yourself Against Cyber Trespass Act*[275] arguably denied the burgeoning tort of cyber trespass the much needed fillip to morph into a mainstream cause of action in the United States. Furthermore, the judicial extension of the tort of trespass into the Internet and cyberspace has been variously panned by critics on varying legal grounds, ranging from its incongruity to cyberspace,[276] to concerns that it could undermine the free speech provisions of the First Amendment to the Constitution of the United States.[277]

The pertinent question therefore is: could vulnerabilities research be tantamount to cyber trespass within the spirit of the *Intel* decision[278] and Restatement (Second) of Torts section 218?[279] Answering this question would of necessity entail discussing and analyzing the following variables: first, whether or not software is property; second, whether or not vulnerabilities researchers are professionals or malicious hackers; third, whether or not vulnerabilities researchers have express or implied authorization from software vendors to conduct research; fourth,

[273] *Omega*, 469 F.3d at 359. *See also* Sch. of Visual Arts v. Kuprewicz, 771 N.Y.S.2d 804 (N.Y. Sup. 2003). The Supreme Court of New York followed the decision in *Intel* by finding that on evidence, the defendant's unsolicited emails had caused actual damage to the plaintiff's computer system by draining its hard drive's memory, thereby reducing its functionality.

[274] *See* RESTATEMENT (SECOND) OF TORTS § 218 (1965).

[275] The bill was proposed by the 109th Congress 2005-2006, and it never became law. It was designed "[t]o protect users of the Internet from unknowing transmission of their personally identifiable information through spyware programs, and for other purposes."

[276] *See* Greg Lastowka, *Decoding Cyberproperty*, 40 IND. L. REV. 23, 45-46 (2007) (noting that cyberspace was not analogous to land and therefore the premise for extending the tort of trespass to cyberspace was inherently faulty).

[277] *See* Electronic Frontier Foundation Amicus Brief paras. 28-29, Intel Corp. v. Hamidi 30 Cal. 4th 1342 (2003), *available at* http://w2.eff.org/spam/Intel_v_Hamidi/20000118_eff_amicus.html.

[278] *See* Intel Corp., 30 Cal. 4th at 1342.

[279] *See* RESTATEMENT (SECOND) OF TORTS § 218 (1965).

whether or not vulnerabilities researchers exceeded their authorizations; and fifth, whether or not software vendors suffered actual damage from unauthorized vulnerabilities research. The following paragraphs will discuss these variables in *seriatim*, while keeping the question of 'actual damage' constant in the discourse and analysis.

1. *Cyber trespass: liability scenarios for non-malicious professional vulnerabilities researchers.*

The proposition that non-malicious professional vulnerabilities researchers could be liable for cyber trespass necessarily rests on the following two assumptions: first, that software or the network systems within which it is embedded, albeit intangible, is property. Second, that a vulnerabilities researcher intermeddles or interferes with software or the network systems in which it is embedded, without authorization of software vendors or systems administrators, with consequential actual damage, the latter assumption being necessarily contingent on the former supposition, in line with the relevant provisions of section 217 and section 218 of the Restatement (Second) of Torts on trespass to chattel.[280]

On the first question of whether or not software is property, the answer is undoubtedly in the affirmative, because software is a subject of intellectual property rights,[281] which software vendors usually affirm via licensing agreements that typically retain vendors' intellectual property, and set the conditions of use for software users.[282] However, assuming *arguendo* that the intellectual property-software nexus is too weak to establish vendors' legal proprietary interest in software, the preponderance of judicial views in the United States do support the classification of software as a "good" within the ambit of Article 2 on the sale of

[280] *See generally* RESTATEMENT (SECOND) OF TORTS §§ 217, 218 (1965).

[281] Software ownership is often managed by copyright, patents, and contractual agreements. See for example Article 1, clause 8 of the Constitution of the United States, which empowers the Congress "[t]o promote the progress of science and useful arts, by securing for limited times to authors and inventors the exclusive right to their respective writings and discoveries."

See Patent Act. 35 U.S.C. §101 (1952); Copyright Act, 17 U.S.C. § 101 (1976). For a detailed discourse on the possible impacts of intellectual property rights on vulnerabilities research, see Part V(C), *infra*.

[282] *See* Emily Kuwahara, *Torts v. Contracts: Can Microsoft Be Held Liable to Home Consumers for its Security Flaws?,* 80 S. CAL. L. REV. 997, 1018 (2007) (noting that software is often sold with licensing agreements affirming vendors' intellectual property and setting users' discretionary use of products). *See also* ProCD, Inc. v. Zeidenberg, 86 F.3d 1447, 1450 (7th Cir. 1996) (noting that software licensing were no more than ordinary contracts).

goods under the Uniform Commercial Code ("UCC").[283] Therefore, viewed from the prisms of intellectual property and the judicial interpretation of the provisions of Article 2 of the UCC, software or the network systems within which it is embedded, albeit intangible, is irrefutably a chattel or property and a proper subject matter for the tort of trespass.

Having established the proprietary nature of software, the second pertinent question is how non-malicious professional vulnerabilities may make researchers liable for the tort of cyber trespass under the conditions laid down in both the *Intel* case[284] and Restatement (Second) of Torts Sections 217 and 218.[285] Answering this question would necessarily entail a close examination of the underlying software properties and structures. As noted earlier, software is an intangible functional component of a computer, which is either inclusive of computer programs and data intended to be processed by the programs, or exclusive of data and comprising merely "a list of commands and instructions for data-processing."[286] Also, in structural terms, software typically exists in two forms: the source program (source code), and the machine readable binary code.[287] When testing for security flaws or bugs, software security professionals often use static analysis and dynamic analysis tools.[288] Static analysis tools are used to vet software in both binary and source forms for common implementation bugs such as buffer overflows, while dynamic analysis tools are used to observe a system as it executes and to feed malformed, malicious, and random data into a system's entry points, in order to uncover any underlying faults or vulnerabilities.[289]

Thus, vulnerabilities research, or software security testing,[290]

[283] There is a recurrent debate in the United States as to whether software is a "good" or a "service." While most courts have classified software as a good, as exemplified by Advent Sys. Ltd. v. Unisys Corp., 925 F.2d 670, 676 (3d Cir.1991) (holding that computer software is a good within the meaning of Article 2 of the U.C.C. and that the categorisation facilitated the analyses of implied warranties, consequential damages, and disclaimers of liability); *see also ProCD*, 86 F.3d. at 1450; (classifying software as a service if it was bespoke or customized software especially designed to fit customers' business needs); *See* Micro-Managers, Inc, v. Gregory, 434 N.W.2d 97, 100 (Wis. Ct. App. 1988) (holding, *inter alia*, that the contract for customized software was essentially a service contract based upon terms of contract).

[284] *See generally*, Intel Corp., 30 Cal. 4th 1342 (2003).

[285] *See* RESTATEMENT (SECOND) OF TORTS §§ 217, 218 (1965).

[286] von Engelhardt, *supra* note 2, at 1.

[287] *See also* Arkin et al., *supra* note 178, at 85.

[288] *See id.*

[289] This process is also known as "fuzzing." *See id.*

[290] *See* Jianto Pan, *Software Testing*, 18-849b DEPENDABLE EMBEDDED SYS., Carnegie Mellon University (Spring 1999), *available at* http://www.ece.cmu.edu/~koopman/des_s99/sw_testing/ (noting that software testing is conducted for a variety of reasons ranging from debugging, quality assurance, verification and validation or reliability estimation). *See also* Arkin et al., *supra* note 178, at 84-87.

which, *inter alia* involves probing of software programs codes for bugs or security flaws,[291] could theoretically be tantamount to trespass if conducted by non-malicious professional researchers, without prior authorization of software vendors or network administrators, provided the resultant unauthorized research resulted in actual damage to software vendors or their customers as per the decision in *Intel*,[292] and the relevant provisions of the Restatement (Second) of Torts Sections 217 and 218.[293] However, if the non-malicious professional software security researchers are employed or commissioned by software vendors as part of a vulnerability management team, to conduct software penetration testing for security flaws or the existence of vulnerabilities,[294] then they would, without a doubt, be presumed to have the underlying express or implied authorization needed for effective performance of their duty. In the circumstance, such presumed or assumed consent to conduct vulnerability research on vendors' software would negate any notion of cyber trespass. This proposition is arguably supported by the Restatement (Second) of Torts Section 252, which provides that the owner of personal property could create a privilege in the would-be trespasser by granting consent to use the property.[295] Thus, by extrapolation, permission to use property under the Restatement (Second) of Torts Section 252 would, in the context of authorized software security testing for flaws, be tantamount to software vendor's permission to conduct vulnerabilities research, and negate the notion of cyber trespass within the meaning of the relevant provisions of the Restatement (Second) of Torts Sections 217 and 218 on trespass to chattel.[296] *A fortiori*, non-malicious professional vulnerabilities researchers commissioned or employed by software vendors would be

[291] *See* STUART MCCLURE ET.AL., HACKING EXPOSED: NETWORK SECURITY SECRETS & SOLUTIONS, 528-34 (5th ed. 2005) (describing fuzz testing and penetration testing as the two most common security testing approaches; while the former is a form of implementation check involving generation of random and crafted application input from the perspective of a malicious hacker, the latter involves authorized penetration of the physical and logical defences provided by an IT organization, using the same tools and techniques that malicious hackers typically use).

[292] *See generally Intel Corp.,* 30 Cal. 4th at 1342.

[293] *See* RESTATEMENT (SECOND) OF TORTS §§ 217, 218 (1965).

[294] *See* SCHNEIER ON SECURITY, *supra* note 25, at 261-62 (noting that penetrating testing is a standard industry practice). *See also* Meiring de Villiers, *Distributed Denial of Service*, *supra* note 84, at 22-24 (noting that software firms often employ security patch and vulnerability management team, whose duty is to, *inter alia*, pre-empt unscrupulous exploitation of software vulnerabilities, by probing for vulnerabilities in pre and post product market launch phases, and promptly applying corrective patch to any known vulnerabilities).

[295] *See* RESTATEMENT (SECOND) OF TORTS § 252 (1965). *See also* CompuServe, Inc. v. Cyber-Promotions, Inc., 962 F. Supp. 1015 (S.D. Ohio 1997).

[296] *See generally* RESTATEMENT (SECOND) OF TORTS §§ 217 and 218 (1965).

deemed to have the requisite permission, consent or authorization of software vendors, and would be immune from cyber trespass.

However, if professional vulnerabilities researchers exceeded or abused their authority, or were not authorized to test for vulnerabilities in the first place, then on the authority of the *Intel* decision,[297] it could be safely assumed that, given the availability of proof of actual damage, and depending on the circumstances and facts of specific cases, a software vendor could theoretically have a cause of action in cyber trespass against a non-malicious professional vulnerability researcher who exceeded or abused their authority.[298] There are a number of conceivable scenarios in which vulnerabilities researchers could either abuse or exceed their research remit. These could range from conducting non-security related testing on software and over-testing of software for security flaws,[299] to inappropriate disclosure of vulnerabilities information or security flaws.[300]

Moreover, since proof of actual damage is crucial to establishing liability for cyber trespass as per the decision in *Intel*,[301] and the relevant provisions of the Restatement (Second) of Torts Section 218,[302] the pertinent question is: what sorts of incidents might constitute actual damage

[297] *See generally Intel Corp.*, 30 Cal.4th at 1342.

[298] There is a parallel here with the common law concept of trespass *ab initio*, where a person who originally entered a land in exercise of his lawful duty authorized by law, is deemed a trespasser from the beginning, because he abused the power conferred on him by causing damage to the property. There is, however, a limit to the analogy with a vulnerabilities researcher who exceeded his authority in that the initial permission to enter land must be granted by law and not by individuals, who, in the context of vulnerabilities research, would be software vendors or network administrators. For a discussion on the common law tort of trespass *ab initio*, *see McGuire v. United States*, 273 U.S. 95 (1927), where a search warrant was issued to revenue agent officers to enter and search the premises possessed by Mcguire. While executing the warrant, the officers discovered and seized several gallons of intoxicating liquor, which they destroyed without court order or legal authority. The Court held that the officers lost the protection and authority vested in them by the search warrant, and became trespassers *ab initio* by destroying the seized liquor.

[299] *See* Pan, *supra* note 290 (noting the types of software testing there are, i.e. security testing, reliability testing, quality testing, etc., and the fact that testing is potentially endless, and that researchers should know when to stop testing).

[300] Reckless disclosures of vulnerabilities information could land crucial information in the wrong hands and engender unscrupulous exploitations of vulnerabilities, which could potentially cause millions of dollars in damage. *See Fighting The Worms of Mass Destruction: Hooligans Are Trashing Our Online Space. How Can They Be Stopped?,* THE ECONOMIST (Nov. 29, 2003), http://www.economist.com/node/2246018 (noting that writers of computer worms and viruses often exploit known vulnerabilities before software vendors have had opportunity to apply corrective patch).

[301] *See* Intel Corp.*,* 30 Cal.4th at 1342.

[302] *See* RESTATEMENT (SECOND) OF TORTS § 218 cmt. e (1965) (providing as follows: "the interest of a possessor of a chattel in its inviolability, unlike the similar interest of a possessor of land, is not given legal protection by an action for nominal damages for harmless intermeddling with the chattel. In order that an actor who interferes with another's chattel

for the purposes of grounding liability for cyber trespass where non-malicious professional vulnerabilities researchers either conduct unauthorized research or exceed their research remit? It is submitted that actual damage would possibly encompass corruption of, damage to, or impairment of the functions of the software subjected to vulnerabilities testing, or damage to the network systems in which the software is embedded. In the circumstance, possible nominal damage, which would not ground a cause of action in cyber trespass as per the decision in *Intel*[303] and the Restatement (Second) of Torts Section 218,[304] might be in the form of delay in launching software products and any consequential financial loss emanating from unauthorized software security testing.

Admittedly, some of the aforementioned possible scenarios for non-malicious vulnerabilities researchers' liability in cyber trespass could also be tantamount to a breach of underlying terms or conditions of agency or contractual agreement on software security testing, leading to possible cause of action for breach of underlying terms of contract or agency agreement for security testing in a civil court. This possibility is legally feasible by reason of the Computer Fraud and Abuse Act ("CFAA"), 18 U.S.C.A. § 1030 (g), which allows for civil suits if any of the provisions of subclauses (I), (II), (III), (IV), or (V) of subsection (c) (4) (A) (i) of the CFAA are violated.[305] An analogous case where a breach of terms of confidentiality agreement signed by former employees formed the basis of a civil cause of action for exceeding authorized access under CFAA § 1030 (e) (6) was *EF Cultural Travel BV v. Explorica, Inc*.[306] The appellants, Explorica, Inc. ("Explorica"), and several of its employees, were enjoined for alleged violations of the CFAA. In the *EF Cultural Travel BV* case, the appellants, who were ex-employees of the appellees, EF Cultural Travel BV ("EF"), had commissioned and provided confidential information to their IT consultant to design a robotic program, with which appellants scoured for and extracted confidential information on pricing from appellees' website. The appellants thereafter used the information to undercut and compete with the appellees in the market for global tours for high school students, where the appellees had operated

may be liable, his conduct must affect some other and more important interest of the possessor.").

[303] *See* Intel Corp., 30 Cal. 4th at 1342.

[304] *See* RESTATEMENT (SECOND) OF TORTS § 218 (1965).

[305] *See* 18 U.S.C.A. § 1030 (g) (West 2006 & Supp. II 2008) (providing as follows: "Any person who suffers damage or loss by reason of a violation of this section may maintain a civil action against the violator to obtain compensatory damages and injunctive relief or other equitable relief. . ."). The main provisions of § 1030 criminalise the act of intentionally accessing a computer without authorisation or exceeding authorised access. *See generally* 18 U.S.C.A. §1030 (a) (1)-(5) (West 2006 & Supp. II 2008).

[306] *See* EF Cultural Travel BV v. Explorica, Inc., 274 F.3d 577, 582-84 (1st Cir. 2001).

for thirty-five years.[307] The United States Court of Appeals for the First Circuit noted that Philip Gormley, an ex-employee of the appellees, who now worked for the appellants, had directly provided appellees' confidential information to appellants' IT consultant, in clear breach of the voluntary broad confidential agreement prohibiting him from disclosing any information, "which might reasonably be construed to be contrary to the interests of EF."[308] The court further noted that the mining and using of appellees' tours pricing data by the appellants in direct competition with the former was in clear breach of confidential agreement, which clearly negated appellees' interests, and that if the allegation were proven during trial,[309] it would likely prove that Explorica had exceeded whatever authorization it had to navigate around EF's website, and that Explorica had most certainly violated the provisions of CFAA Section 1030 (e) (6).[310] The Court of Appeals then affirmed the District Court's preliminary injunction restraining the appellants from using proprietary data surreptitiously taken from the appellees' website.[311]

Thus, apart from a possible cause of action in cyber trespass as previously adumbrated, software vendors or web administrators arguably have the option to pursue civil litigation against non-malicious vulnerabilities researchers who exceed their authorized access in clear violations of CFAA §§ 1030 (e) (6) and 1030 (g),[312] as amply demonstrated by *EF*

[307] *Id.* at 577-81.

[308] *Id.* at 583.

[309] The appeal before the First Circuit Court of Appeals was in respect to a preliminary injunction granted by the District Court restraining the appellants from using the tours pricing data and other proprietary information pending the determination of the substantive suit for CFAA violations, which was yet to go to trial. The Court of Appeals subsequently affirmed the preliminary injunction albeit on narrower grounds than those exposed by the District Court. *See generally id.* at 578.

[310] *Id.* at 583-84.

[311] *Id.* For a discussion on how breach of contractual and agency terms could violate CFAA and give rise to a civil cause of action, *see* United States v. Czubinski, 106 F.3d 1069 (1st Cir. 1997), where the First Circuit held that Czubinski exceeded his authorized access to his employer's (IRS) computer when he accessed certain files belonging to tax payers within the IRS database, in contravention of his employment contract acknowledging IRS policy, which prohibited accessing the said files, and in violation of 18 U.S.C. § 1030 (a) (4). *See also* Shurgard Storage Centers, Inc. v. Safeguard Self Storage, Inc., 119 F. Supp. 2d 1121 (W.D. Wash. 2000). However, both the contractual and agency bases for civil violations of CFAA have been criticized by academic writers for their overreach. *See* Cyrus Y. Chung, *The Computer Fraud and Abuse Act: How Computer Science Can Help with the Problem of Overbreadth*, 24 HARV. J.L. & TECH. 233, 239-43 (2010) (discussing the legal propriety of the contractual and agency theories for grounding civil liabilities for CFAA violations, and suggesting the use of "the code-based theory" and "a computer security model" interpretation approach to 'without authorization' or 'exceeding authorised access' provisions of CFAA).

[312] *See* 18 U.S.C.A. §.1030 (e) (6) & (g) (West 2006 & Supp. II 2008).

Cultural Travel BV.[313] However, it is extremely doubtful that software vendors would sue because the software industry relies heavily on professional software security researchers, who range from in-house professionals and the not-for-profit Computer Emergency Response Team ("CERT"),[314] to the commercially inclined software security firms like TippingPoint and iDefense, who openly conduct software security testing and habitually broker sales of software vulnerabilities without necessarily seeking prior authorization from software vendors.[315] Furthermore, it is software industry standard practice to test for software security flaws,[316] and professional security researchers are now a putative feature of the largely informal and unregulated supermarket for outsourcing and brokerage of software vulnerabilities.[317] *A fortiori*, the absence of standardized industry practice on vulnerabilities research detection and brokerage, cum the lack of any regulation guaranteeing transparency in vulnerabilities management are no doubt responsible for much of the policy chaos and legal uncertainties underpinning software vulnerabilities governance.

2. *Cyber trespass: liability scenarios for malicious hackers.*

Malicious vulnerabilities researchers or hackers typically exploit known and unknown software vulnerabilities information such as buffer overflows for valuable information or malicious attacks, using a variety of attack systems ranging from denial-of-service attacks, worms, and Trojan horses, to viruses.[318] It is axiomatic that malicious hackers would necessarily have to interfere or intermeddle with software or web server software and associated software packages,[319] without prior authoriza-

[313] *See* EF Cultural Travel BV, 274 F.3d at 582-84.

[314] The Computer Emergency Response Team Clearing Centre originated from a program funded by the United States Department of Defense under the auspices of Carnegie Mellon University, and continues to exist "as a clearinghouse for information about viruses and other network threats." *See* ZITTRAIN, *supra* note 104, at 39. Most countries now have a CERT that monitor computer security. For more information on the Carnegie Mellon University CERT, *see* CERT, http://www.cert.org/ (last visited Oct. 21, 2011). For more information on CERT in the United Kingdom, *see* UKCERT, http://www.ukcert.org.uk/ (last visited Oct. 21, 2011).

[315] *See* Miller, *supra* note 113, at 2. *See also* Li & Rao, *supra* note 173, at 531 (discussing how the roles of commercial and non-commercial private intermediaries (i.e. CERT, iDefense, and TippingPoint) differ in software vulnerabilities disclosure).

[316] *See* SCHNEIER ON SECURITY, *supra* note 25, at 261-62 (noting that penetrating testing is a standard industry practice).

[317] *See* Miller, *supra* note 113, at 2.

[318] *See* Meiring de Villiers, *Information Security Standards and Liability*, *supra* note 82, at 24-26 (noting the variety of malevolent programs used by hackers to access sensitive data and information); *See also Fighting The Worms Of Mass Destruction: Hooligans Are Trashing Our Online Space. How Can They Be Stopped?*, *supra* note 300.

[319] *See* MCCLURE ET. AL., *supra* note 291, at 536.

tion of vendors or network administrators. However, while the unauthorized interference might fulfill the requirement of Restatement (Second) of Torts Section 217,[320] it is legally imperative that there is concomitant actual damage in line with Restatement (Second) of Torts Section 218,[321] and the decision in *Intel*,[322] for a successful cause of action in cyber trespass by software vendors or network administrators as the case might be. Possible actual damage scenarios could comprise corruption of, impairment or damage to software or web servers' software following hacking incidents, and would preclude nominal damage not directly linked to hacking incidents.[323] However, notwithstanding the theoretical possibility of a cause of action in cyber trespass against malicious hackers, software vendors or network administrators might be reluctant to pursue civil litigation due in part to the concomitant adverse publicity that such litigation would surely generate and software vendors or web server administrators are notoriously wary of such publicity.[324]

Significantly, It is also possible for software vendors to sue malicious hackers in civil court for compensatory damages under 18 U.S.C.A. § 1030 (g), which as noted above, allows for compensatory damages if any of the alleged conduct involved the violations of any of the provisions of subclauses (I), (II), (III), (IV), or (V) of subsection (c) (4) (A) (i) of the CFAA.[325] This option might be more attractive than cyber trespass because of the relative ease with which plaintiffs could prove statutory damage emanating directly from malicious hacking. However, as noted earlier, software vendors or network administrators might be discouraged from pursuing trespass litigation or CFAA civil violations by the concomitant adverse publicity and negative connotations of being hacked into.[326] Nevertheless, an order for injunctive relief under 18 U.S.C.A. § 1030 (g) against a habitual hacker could be in order while prosecutors simultaneously pursue a criminal case.[327]

[320] *See* Restatement (Second) of Torts § 217 (1965).

[321] *See* Restatement (Second) of Torts § 218 (1965).

[322] *See* Intel Corp., 30 Cal. 4th at 1342.

[323] For a discussion, *see id.*

[324] *See e.g.*, Walden, *supra* note 245, at 554.

[325] *See* 18 U.S.C. § 1030 (g) (West 2006 & Supp. II 2008).

[326] Multinational commercial establishments are often reluctant to admit software or network security breaches due to the inherent adverse publicity. *See* Walden, *supra* note 245, at 554.

[327] Software vendors or network administrators could seek temporary injunctions to restrain hackers, pending the final determination of the main trespass litigation. In order to secure a temporary injunction, applicants merely have to show on affidavit evidence that there is a real likelihood of irreparable damage for which damages would not be adequate compensation. This would provide the breathing space needed for the pursuit of a criminal case against hackers. *See generally* 18 U.S.C. § 1030 (g) (West 2006 & Supp. II 2008).

B. To What Extent is Vulnerabilities Research a Cyber Crime?

Software vulnerabilities research is crucial for computer and network security, and is actively supported and funded by industry and governments from around the world.[328] For example, Facebook, which operates a bug for bounty program, paid out over forty thousand dollars to vulnerabilities researchers within three weeks of the commencement of the reward-for-bugs program in late 2011.[329] In fact, Facebook now has a dedicated webpage for their "whitehat" program, through which independent, non-malicious hackers are encouraged to research new bugs or vulnerabilities and make responsible vulnerabilities disclosure in exchange for financial rewards.[330] According to Joe Sullivan, Facebook's Chief Security Officer, the bug for bounty program has enabled the company to tap into the expertise of "a whole new and ever expanding set of people across the globe in over sixteen countries from Turkey to Poland who are passionate about Internet security."[331]

Ironically however, vulnerabilities research could also run afoul of key legislation especially designed to thwart cybercriminals and ensure cyber security, privacy rights, and integrity of digital property. It is sacrosanct that cybercrime[332] is a transnational scourge[333] that experts be-

[328] As noted earlier, software firms actively encourage vulnerabilities research and most rely on in-house and independent researchers to plug security flaws in software and network systems. For example, companies like Facebook, Google, and even the American government have openly paid for vulnerabilities information, whilst vulnerabilities brokers are increasingly openly active in the burgeoning vulnerabilities market. *See* Miller, *supra* note 113, at 2. Significantly, the United State's resolve to fund cyber security research was given a legal *imprimatur* by Congress' enactment of The Cyber Security Research and Development Act. The Cyber Security Research and Development Act, 15 U.S.C. § 7401 (2002). The law, *inter alia*, aims at providing sufficient long term research funding for cyber security.

[329] *See* Joe Sullivan, *Update to the Bug Bounty Program,* Facebook (Aug. 29, 2011), https://www.facebook.com/notes/facebook-security/updates-to-the-bug-bounty-program/10150270651335766.

[330] *See* Facebook, *Information for security researchers*, http://www.facebook.com/whitehat/ (last visited Oct. 22, 2011) (listing the names of security researchers who had made "responsible disclosure" of vulnerabilities or security flaws on *Facebook* webpage).

[331] *See* Sullivan, *supra* note 329.

[332] *See* Walden, *supra* note 245, at 554 (categorizing cybercrimes into the following three classes: (a) offenses that may be committed using computers as the instrument of crime; (b) content-related crimes committed using computers and networks as instruments; (i.e., intellectual property theft), and (c) computer integrity crimes, which involve using viruses and malevolent programs to attack the integrity, confidentiality, and availability of computer and communications systems).

[333] *See* Kathleen Ellis, *Cyber Security: Global Risk and Rising Complexity*, Insurance Journal (July 6, 2009), http://www.insurancejournal.com/magazines/features/2009/07/06/158455.htm (describing the intractability of, and inexorable rise in global cybercrimes).

lieve could potentially stifle network connections.[334] This arguably informed the crafting of the 2001 Council of Europe Convention on Cybercrime ("Treaty"),[335] which the United States signed in 2001, and subsequently ratified in 2006.[336] The Treaty stresses international cooperation amongst member states for effective counter cybercrime strategies, and enjoins members to pass key legislations that would, *inter alia*, criminalize illegal access to computers without right,[337] illegal interception of non-public transmission of computer data without right,[338] the damaging, deletion, deterioration, alteration or suppression of computer data without right,[339] the production, sale, procurement for use, import, distribution or otherwise making available of a device, including a computer program, designed or adapted primarily for the purpose of committing offences or dealing in a computer password, access code, or similar data by which the whole or any part of a computer system is capable of being accessed.[340] Without a doubt, the provisions of the Cybercrime Convention cover every conceivable computer-related crime, ranging from crimes requiring the instrumentality of computers and content-related crimes committed using computers, to computer integrity crimes, which are facilitated by malevolent programs such as viruses, Trojans and worms.[341] In the United States, criminal activities of malicious hackers or vulnerabilities researchers are regulated by legislations, including relevant provisions of the CFAA,[342] the Wire Tap Act,[343] and Pen Register or Trap and Trade Devices Act.[344]

[334] *See* Neal K. Katya, *The Dark Side of Private Ordering: The Network/Community Harm of Crime*, *in* THE LAW AND ECONOMICS OF CYBERSECURITY 115, 194-200 (Mark F. Grady & Francesco Parisi eds., 2006) (describing how disruptive cybercrime is to network connections, and how important it was for law enforcement to police cybercrime rather than leaving the task to individual victims who might not have the wherewithal to fend off cybercrimes).

[335] *See* Convention on Cybercrime, Nov. 23, 2001, C.E.T.S. No. 185.

[336] As at October 28, 2010, the United States was the only one of the eleven non-member states of the Council of Europe to have signed and ratified the Convention on Cybercrime, with the treaty coming into force in the United States with effect from January 1, 2007. *See* Council of Europe, Convention on Cybercrime (C.E.T.S. No. 185): Member Signatures' Status as of October, 28, 2010, http://conventions.coe.int/Treaty/Commun/ChercheSig.asp?NT=185&CM=8&DF=28/10/2010&CL=ENG (last visited Oct. 22, 2011).

[337] *See* Convention on Cybercrime art. 2, Nov. 23, 2001, C.E.T.S. No. 185.

[338] *See id.* at art. 3.

[339] *See id.* at art. 4.

[340] *See id.* at art. 6.

[341] *See* Walden, *supra* note 245, at 554 (categorizing computer-related crimes into three parts).

[342] *See* Computer Fraud and Abuse Act (CFAA), 18 U.S.C. § 1030 (1986).

[343] *See* The Wire Tap Act, 18 U.S.C.A. §§ 2510-2522 (1968).

[344] *See* Pen Register or Trap and Trace Device Act, 18 U.S.C. § 3127(3) (2009).

Therefore, the pertinent question is how could vulnerabilities research be tantamount to a cybercrime? As previously noted, vulnerabilities research typically involves researcher and hacker use of static analysis and dynamic analysis tools.[345] Whilst static analysis tools are used to probe the software codes both in source and binary forms for security flaws, dynamic analysis tools are used to observe a computer system as it executes and to feed malformed, malicious, and random data into a system's entry points. This uncovers underlying security flaws of software codes in both source and binary forms.[346] Thus, malicious hackers or vulnerabilities researchers using these techniques to probe for "zero day" or new vulnerabilities, which they subsequently exploited or peddled off to criminal underground hackers, would undoubtedly have committed a cybercrime[347] under the provisions of CFAA § 1030 (a) (4), which, *inter alia*, prohibits "knowingly and with intent to defraud, access a protected computer without authorization, or exceeds authorized access, and by means of such conduct, furthers the intended fraud and obtains anything of value. . ."[348] The section is particularly apposite for malicious hackers who often use means such as passwords, viruses, worms, and related malicious programs to either hunt for new software vulnerabilities or exploit existing software vulnerabilities in computing systems, often without authorization, (or in excess of authority) and with intent to defraud. The fraud element could be evidenced by subsequent black-market or underground sale of new vulnerabilities, direct exploitation of known vulnerabilities, or theft of critical information from hacked networked systems.[349]

This scenario is illustrated in *United States v. Nosal*,[350] where employees used their passwords rather than any known vulnerabilities to access and divulge employer's proprietary information to a third-party.

[345] *See* Arkin, et al., *supra* note 178, at 85.

[346] *Id. See* ALSO McClure et.al., *supra* note 291, at 528-34 (describing how professional hackers could employ the standard tools that malicious hackers use, such as fuzz testing and penetration testing to test for security flaws in networked systems).

[347] Hackers often peddle new or "zero day" vulnerabilities to criminal colleagues or directly exploit vulnerabilities for financial or disruptive ends. *See* Brian Bergstein, *Report: Black Market for Computer Vulnerabilities Weaken Web Safety*, INSURANCE JOURNAL (Feb. 13, 2008), http://www.insurancejournal.com/news/national/2008/02/13/87296.htm. *See also* Randianti & Gonzalez, *supra* note 16, at 3-5 (describing the dynamics of the supply and demand sides of vulnerabilities market between malicious hackers suppliers and the work criminal underworld buyers).

[348] Computer Fraud Abuse Act (CFAA), 18 U.S.C. § 1030(a)(4) (1986).

[349] The provisions of 18 U.S.C. § 1030(a)(4) (regulating fraud and related activity in connection with computers) are *in parimateria* with the provisions of Article 6 of the Cybercrime Convention, to which the United States is a party, which oblige Member States to criminalise the supply and possession of a 'device' computer password, access code, or similar data. *See* Convention on Cybercrime art. 6, Nov. 23, 2001, C.E.T.S. No. 185.

[350] *See* United States v. Nosal, 642 F.3d 781 (9th Cir. 2011).

The Court of Appeals for the Ninth Circuit affirmed the indictments of the defendant's co-conspirators before the District Court to the effect that they exceeded their authorized access to their employer's computer system and stole trade secrets and proprietary information, for the benefit of the defendant's business and in violation of 18 U.S.C. § 1030 (a) (4).[351] Other relevant provisions of the CFAA, which ostensibly target malicious malware such as worms, viruses and Trojans that exploit vulnerabilities of a protected computer without authorization, are 18 U.S.C. § 1030 (a) (5) (A) (B) (C), which respectively criminalize the activities of anyone who:

> (A) [K]nowingly causes the transmission of a program, information code, or command, and as a result of such conduct, intentionally causes damage without authorization, to a protected computer,
>
> (B) Intentionally accesses a protected computer without authorization, and as a result of such conduct, recklessly causes damage; or
>
> (C) Intentionally accesses a protected computer without authorization, and as a result of such conduct, causes damage and loss.[352]

Indeed, the legislative history of the CFAA reveals that it was a direct Congressional response to the problems posed by malicious computer hackers whose crimes were not adequately covered by existing criminal statutory framework.[353]

However, the budding practice of direct or indirect outsourcing of software vulnerabilities research to the mass publics, often across national boundaries, as exemplified by the Facebook's bug bounty program, which had volunteer hackers from across sixteen countries hunting for vulnerabilities on its website,[354] could arguably potentially redefine or hamstring the fundamentals of the inherently territorial penal governance of malicious software vulnerabilities research.[355] For instance, Facebook's bug bounty program actively encouraged vulnerabilities researchers to find and report bugs under a "responsible disclosure policy" and guaranteed researchers' immunity from criminal prosecution or civil

[351] Note, however, that the co-conspirators have used their passwords to access their employer's computer system rather than exploiting any known vulnerability to gain access. *See* United States v. Nosal 642 F.3d 781 (9th Cir. 2011); *see also* United States v. Batti, 631 F.3d 371 (4th Cir. 2011), where Batti was convicted of improperly accessing information from a protected computer in violation of 18 U.S.C. § 1030(a)(2)(c) & (c)(2)(b)(iii).

[352] *See generally* Computer Fraud Abuse Act (CFAA), 18 U.S.C. § 1030(a)(4) (1986).

[353] *See* H.R. Rep. No. 98-894, at 4 (1984), *reprinted in* 1984 U.S.C.C.A.N. 3689. *See also* Chung, *supra* note 16, at 237-39.

[354] *See* Sullivan, *supra* note 329.

[355] Whilst there is a transnational convention on cybercrime, not every country is a signatory and its provisions do not apply automatically across national jurisdictions, where national regimes on cyber-related crimes are essentially territorial in nature. *See* Walden, *supra* note 245, at 583-85 (discussing the principles of the territoriality of criminal law and penal sanctions in international criminal law).

liability,[356] under the following terms and conditions:

> If you share details of a security issue with us and give us a reasonable period of time to respond to it before making it public, and in the course of that research made a good faith effort to avoid privacy violations, destruction of data, or interruption or degradation of our service, we will not bring any lawsuit against you or ask law enforcement to investigate you for that research.[357]

Whilst lauding Facebook's bug bounty responsible reporting policy, the Electronic Frontier Foundation highlighted the dilemmas facing volunteer vulnerabilities researchers as follows:

> Well-meaning Internet users are often afraid to tell companies about security flaws they've found – they don't know whether they'll get hearty thanks or slapped with lawsuit or even criminal prosecution.[358]

Thus, an act of hacking which could be tantamount to criminal violations of CFAA could be rewarded by Facebook from five hundred dollars up to five thousand dollars for a very good vulnerability report,[359] provided the hacker followed Facebook's responsible disclosure policy.[360] However, a rogue hacker resident outside of the United States who reneged on Facebook's terms of vulnerability research might still escape prosecution due to the absence of enforceable transnational cybercrime legislation.[361] This underlies the inherent weakness of national cybercrime legislations in combating transnational cybercrimes such as hacking.[362]

Furthermore, whilst most cyber attack counter-measures are often reactive and defensive, a proactive counter measure, which involves the use of honeynet program to sniff for malicious cyber intruders,[363] could ironically violate the penal provisions of the CFAA,[364] the Wiretap

[356] *See* Sullivan, *supra* note 329.

[357] *See* Marcia Hofmann, *Knowledge is Power: Facebook's Exceptional Approach to Vulnerability Disclosure*, ELECTRONIC FRONTIER FOUNDATION (Dec. 17, 2010, 9:46 AM), *available at* http://www.eff.org/deeplinks/2010/12/knowledge-power-facebooks-exceptional-approach.

[358] *See id.*

[359] *See* Sullivan, *supra* note 329.

[360] *See id.*

[361] *See* Walden, *supra* note 245, at 583-85 (discussing the principles of the territoriality of criminal law and penal sanctions in international criminal law).

[362] However, non-resident malicious hackers could still be prosecuted in the United States provided it was possible to have them extradited to the United States.

[363] The honeynet program is a type of high-interaction honeypot that is used primarily to capture information on potential threats to network systems. The program typically interacts with attackers through a network of real computers. *See* The Honeynet Project, *Know Your Enemy: Honeynets: What a Honeynet is, Its Value, Overview of How it Works, and Risk/Issues Involved*, HONEYNET (May 31, 2006), http://old.honeynet.org/paper/honeynet/.

[364] *See generally* Computer Fraud and Abuse Act (CFAA), 18 U.S.C. § 1030 (1986).

Act,[365] as well as the Pen Register or Trap and Trade Devices Act.[366] Honeynet is a type of high-interaction honeypot program set within a network of real computers and designed to surreptitiously track and gather information on potential internal and external threats to a network system.[367] However, sniffing traffic on a network could be considered an interception of electronic communications and be tantamount to a felonious offense punishable by a fine and up to five years imprisonment under the Electronic Communications Privacy Act or Wire Tap Act.[368] Similarly, the Pen Register or Trap and Trace Devices Act generally prohibits the acquisition of non-content information of a communication.[369] According to Richard Salgado, this would include ". . .the source and destination IP address, the port number that handled that communication, and email addresses of the attackers."[370] Significantly, noncompliance with Wire Tap Act is punishable by a fine or one year imprisonment.[371]

Thus, whilst malicious cyber attackers could be criminally liable for exploiting software vulnerabilities, the same law could potentially hamstring the use of proactive counter measures such as a honeynet program to track, ensnare, and gather information on malicious cyber attackers. There is, however, the computer trespasser exception under the USA PATRIOT Act, which allows the government to monitor hackers under the following conditions: first, that the user under surveillance is a trespasser; second, that the communications being monitored are relevant to ongoing investigations; and third, that networks owners' permission be

[365] The Federal Wire Tap Act generally forbids the interception of the content of communications including electronic communications, unless the monitoring is exempted. The Wire Tap Act, 18 U.S.C. §§ 2510-2522 (1968). *See* Richard Salgado, *Legal Issues*, *in* The Honeynet Project, Know Your Enemy: Learning About Security Threats 225, 228 (2d ed. 2004).

[366] *See* Pen Register or Trap and Trace Device Act, 18 U.S.C. § 3127(3).

[367] *See* Salgado, *supra* note 366, at 228.

[368] *See* The Wire Tap Act, 18 U.S.C. § 2511(4) (1968). However, providers of electronic communications services are, under the "provider exception," allowed to intercept communications for the purposes of protecting their rights or property pursuant to 18 U.S.C. § 2511(2)(a)(i). *See also* In Re Pharmatrak, Inc. Privacy Litigation Civ. Act. No. 00-11672-JLT (D. Mass. 2002), where the Court held that defendants did not violate Title I of the ECPA, the Wire Tap Act, because they qualified for the protection of §2511(2)(d) of the ECPA, which permits interception of a communication when it is authorized by one of the participants in the communication, provided the interception is not undertaken for a tortuous or criminal purpose. Defendants were permitted to intercept the communications at issue because (a) the pharmaceutical defendants which participated in them had authorized such interception, and (b) there was no evidence that such interception was done for an improper purpose. *See* Salgado, *supra* note 366, at 228.

[369] *See* Pen Register or Trap and Trace Device Act, 18 U.S.C. §§ 3121-3127 (2009).

[370] *See* Salgado, *supra* note 366, at 237.

[371] *See* Pen Register or Trap and Trace Device Act, 18 U.S.C. § 3121(d) (2009); *see also* Salgado, *supra* note 366, at 238.

secured prior to commencement of monitoring or surveillance.[372] Although the trespasser exception is exclusive to governmental agencies, it may still afford a limited leeway for private firms or agencies using honeynet programs in conjunction with the government or any of its agencies in monitoring cyber threats.[373] *A fortiori*, a general research or trespasser exception for non-governmental vulnerabilities researchers is imperative for effective and proactive counter cyber intrusion strategies, with provisos guaranteeing proper oversight to pre-empt abuse.[374]

C. Could Vulnerabilities Research Impinge on Intellectual Property?

Whilst software is protected by both patent law[375] and copyright statute,[376] the most likely challenging intellectual property related stat-

[372] *See* The Wire Tap Act, 18 U.S.C. § 2511(2)(i) (1968).

[373] *See* Salgado, *supra* note 366, at 236.

[374] *See* Burstein, *supra* note 92, at 184-94 (discussing the impediments posed to cybersecurity research by communications privacy law).

[375] Whilst software is in principle patent eligible, the recurring question is often on the exact parameters for ascertaining software patent eligibility under the Patent Act, which guarantees the grant of a patent to "whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof. . .subject to the conditions and requirements of [the Act]." Patent Act, 35 U.S.C. § 101 (1952). *See*, *e.g.*, In re Bilski, 545 F.3d 943 (Fed. Cir. 2008), (interpreting The Patent Act, 35 U.S.C. § 101 (1952), regarding the patentability of a business method process patent claim for hedging risks in commodity trading, and rejecting the business method process patent claim on grounds that it failed the machine-or-transformation test, the sole criterion for determining process claim patent eligibility). *But see* Bilski v. Kappos, 130 S. Ct. 3218, 3227 (2010) (rejecting the machine-or-transformation test as the sole test for patent eligibility based on the interpretation of the language of The Patent Act, 35 U.S.C. § 101 (1951)). While the Court did not expatiate on what other criteria were, the Court expressly left the door open "For the Federal Circuit's development of other limiting criteria that further the purposes of the Patent Act and are not inconsistent with its text." *See* Bilski v. Kappos, 130 S. Ct. 3218, 3231 (2010). *See also* CyberSource Corp. v. Retails Decisions, Inc., No. 2009-1358 (Fed. Cir. Aug. 16, 2011) (acknowledging that there could be other limiting criteria apart from the machine-or-transformation test, and noting that while software was patent eligible, the patentability bar had definitely gone up). The *Bilski* Court then went on to reject a patent claim for a method and system for detecting fraud in a credit card transaction between a consumer and a merchant over the Internet, as ineligible for patent for being too broad and essentially encompassing any method or system for detecting credit card fraud which utilizes information relating credit card transactions to particular Internet addresses. *See* MARTIN J. ADELMAN ET AL., CASES AND MATERIALS ON PATENT LAW 105-52 (1998) (giving a historical perspective on software patents).

[376] *See* Copyright Act, 17 U.S.C. § 101 (1976), which defines a computer program as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." The same section defines "literary works" as ". . .works. . .expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied." Thus, software or

ute for software vulnerabilities research and disclosure in the United States is the Digital Millennium Copyright Act ("DMCA"),[377] designed to strengthen digital copyright protection.[378] To this end, the DMCA prohibits circumventing access control to technology safeguarding digital copyright such as encryption.[379] Additionally, the DMCA forbids dissemination of devices or technologies that have few secondary commercial uses other than to primarily facilitate circumvention.[380] The DMCA also prohibits the removal or alteration of copyright management information appended to copyright files.[381] Significantly, the DMCA makes a limited exception for encryption research, which is defined as:

> [a]ctivities necessary to identify and analyze flaws and vulnerabilities of encryption technologies applied to copyrighted works, if these activities are conducted to advance the of knowledge in the field of encryption technology or to assist in the development of encryption products. . .[382]

A fortiori, in the context of software vulnerabilities research, the statutory conception of encryption research is particularly apposite. It is sacrosanct for vulnerabilities researchers to, of necessity, "indentify and analyze flaws and vulnerabilities of encryption technologies applied to copyrighted works",[383] of which software is clearly one. However, professional software vulnerabilities researchers in particular, or encryption researchers in general, could avail themselves of the provisions of the encryption research exception from liability provided the following conditions were met: first, that the encryption is conducted in "good faith;" second, that the encrypted copy is lawfully obtained; third, that the act of circumvention is "necessary" for the research; and fourth, that the researcher made a "good faith" effort to obtain authorization from the copy-

computer programs would be defined as literary works pursuant to the definitions in § 101 noted above.

[377] *See* Digital Millennium Copyright Act, 17 U.S.C. §§ 1201-1205 (2000). *See also* Joseph P. Liu, *The DMCA and the Regulation of Scientific Research*, 18 BERKELEY TECH. L.J. 501 (2003) (discussing the limits of research exception under DMCA, and how the negative impacts could be mitigated); Pamela Samuelson, *Anti-circumvention Rules: Threats to Science*, 293 SCIENCE 2028-31 (Sept. 14, 2001) (discussing liability scenarios for scientists who are involved in devising tools for studying encryption, computer security, or reverse engineering of computing technical measures under the DMCA). *Compare* Digital Millennium Copyright Act, 17 U.S.C. §§ 1201-1205 (2000), *with* Council Directive 2001/29/EC, 2001 O.J. (L 167) (EC), on the harmonization of certain aspects of copyright and related rights in the information society. *See, e.g.*, Council Directive 2001/29/EC, 2001 O.J. (L 167) art. 6-7 (EC) (protecting "technological measures" and digital management information).

[378] *See* Liu, *supra* note 377, at 505 (noting that the DMCA was an additional legal support for copyright owners).

[379] *See* Copyright Act, 17 U.S.C. § 1201(a)(1) (1976).

[380] *Id.* at §§ 1201(a)(2) & (b).

[381] *Id.* at § 1202.

[382] *Id.* at § 1201(g).

[383] *Id.*

right owner prior to circumvention.[384]

In ascertaining whether a professional software vulnerabilities researcher or any encryption researcher qualified for the encryption research exception, courts consider factors ranging from how information derived from encryption research is disseminated to whether the researcher "is engaged in a legitimate course of study, employed, or is appropriately trained or experienced, in the field of encryption technology."[385] Furthermore, the DMCA allows encryption researchers to craft or design necessary tools for implementing encryption research and to freely share such tools amongst colleagues "for the purpose of conducting the acts of good faith encryption research."[386] Such flexibility is particularly crucial for professional software vulnerabilities researchers and hackers who typically use static analysis and dynamic analysis tools to respectively probe software codes for security flaws, and to feed malicious malware and random data into systems' entry points to uncover latent vulnerabilities.[387] Thus, in the context of software vulnerabilities research, professional hackers who manufacture, import, offer to the public, provide, or otherwise traffic in any technology, product, service, device, or component that is primarily designed or produced for the purpose of circumventing a technological measure that effectively control access to a copyright protected work could be shielded from liability.[388]

A similar but equally important exception that professional software vulnerabilities researchers could take advantage of is the DMCA provision precluding security researchers from liability for security testing, which is defined as: "accessing a computer, computer system, or computer network, solely for the purpose of good faith testing, investigating, or correcting, asecurity flaw or vulnerability, with the authorization of the owner oroperator of such computer, computer system, or computer network."[389] However, security testing is only permissible to the extent that it does not infringe on relevant provisions of the DMCA, and 18 U.S.C. § 1030 of the CFAA.[390] In ascertaining whether an act of security testing is permissible, courts must consider the following factors: first, whether or not the information derived from the security testing was used solely to promote the security of the owner or operator of such computer, computer system or computer network, or shared directly with the developer of such computer, computer system, or computer network;[391]

[384] *Id.* at §§ 1201(g)(2)(A)-(C).

[385] *Id.* at §§ 1201(g)(3)(A)-(C).

[386] *Id.* at §§ 1201(g)(4)(A)-(B).

[387] *See* Arkin et al., *supra* note 178, at 85.

[388] *See* Copyright Act, 17 U.S.C. § 1201(a)(2) (1976).

[389] *Id.* at § 1201(j)(1).

[390] *Id.* at § 1201(j)(2).

[391] *Id.* at § 1201(j)(3)(A).

and second, whether or not the information derived from the security testing was used or maintained in a manner that does not facilitate infringement under the DMCA or a violation of any applicable law, including a violation of privacy or breach of security.[392] Most significantly, security researchers are further exempted from liability for producing, distributing, or employing technological means for the sole purpose of performing the act of security testing.[393]

It is noteworthy, however, that, but for a special representation made by the encryption research community to the Congress, which urged the research exception prior to the release of the final version and enactment of the DMCA, there would be no encryption research exception in the DMCA at all.[394] Nevertheless, the limited encryption research exception has been criticized for imposing too restrictive operative conditions. These range from the narrow conception of what encryption research entails,[395] to the requirement that researchers must first seek authorization of copyright owners prior to engaging in research,[396] to the ostensible exclusion of non-academic researchers (such as non-affiliated individual researchers or "hobbyists") from the list of qualified encryption researchers,[397] to the restrictive conditions for the publication or dissemination of research information or outcomes.[398]

There is indeed ample evidence that security and encryption researchers are wary of the possible civil and criminal penalties that a violation of any of the restrictive provisions of the DMCA on encryption research, security testing, and reverse engineering of software could engender.[399] Notable amongst such incidents was the much publicized

[392] *Id.* at § 1201(j)(3)(B).

[393] *Id.* at § 1201(j)(4).

[394] *See WIPO Copyright Treaties Implementation Act: Hearings on H.R. 2281 Before the H. S. Comm. on Telecomm., Trade & Consumer Protection*, 104th Cong. (1998), *noted in* Liu, *supra* note 377, at 505-06 (attributing the testimony of the encryption research community to the limited encryption exception in the DMCA).

[395] *E.g.*, the requirement that the act of circumvention be "necessary" for the research was branded as too narrow in that it excluded research that might not be "necessary" but that could be "useful" or "important." *See Id.* at 509-10.

[396] *Id.* at 509-10.

[397] *Id.* (noting that by excluding non-academic and non-affiliated individual encryption researchers or hobbyists, 17 U.S.C. § 1201(g) endorsed a fundamentally mistaken conception of cryptographic science).

[398] *See, e.g.*, Digital Millennium Copyright Act, 17 U.S.C § 1201(g)(3)(A) (providing, *inter alia*, that dissemination of research information may be permissible to the extent that is "reasonably calculated to advance the state of knowledge or development of encryption technology"). *See* also Liu, *supra* note 377, at 505-06.

[399] *See* Fred Von Lohman, Unintended Consequences: Twelve Years Under the DMCA, 21(2010), *available at* http://www.eric.ed.gov/PDFS/ED509862.pdf (detailing and discussing the "chilling effects" of DMCA on encryption research and the reluctance and fears of researchers on the possible effects of DMCA on their work).

event in which Professor J. Alex, Halderman, then a graduate student at Princeton University delayed the publication of the existence of several security vulnerabilities that he found in the CD copy-protection software on dozens of Sony-BMG titles. He delayed disclosing the vulnerabilities for several weeks whilst he sought legal advice from lawyers on how to avoid running afoul of DMCA pitfalls, a measure that left millions of music fans unnecessarily at risk.[400] The fear of prosecution or litigation by vulnerabilities researchers is not entirely unfounded as exemplified by several incidents of actual threats of DMCA lawsuits. For example, in April 2003, the educational software company, Blackboard Inc., obtained a temporary restraining order to stop the presentation of research on security vulnerabilities in its software products at the InterzOne II conference in Atlanta.[401] The said software security vulnerabilities pertained to the Blackboard ID card system used by university campus security systems. However, the students who were scheduled to speak on the vulnerabilities and the conference organizers had no opportunity to challenge the temporary restraining order, which was obtained *ex parte* on the eve of the event.[402]

The Blackboard Inc. case and several others are symptomatic of the way that DMCA has been used to "chill security research", a view that was echoed in October 2002 by the then White House Cyber Security Chief Richard Clarke, who while calling for DMCA reform, was quoted as saying as follows: "I think a lot of people didn't realize that it would have this potential chilling effect on vulnerability research."[403] The real prospects that DMCA could stultify encryption research and constrain the circumstances under which such research are disclosed has precipitated calls for "a broader exemption under the DMCA for encryption research, one that gives maximum freedom to encryption researchers."[404] In fact, the European Union arguably has such a restriction-free encryption research regime in Article 5(1) of the Software Directive, which allows for error correction of computer programs without prior authorization of the copyright owners.[405] It is time the United States reformed the DMCA to accommodate unfettered legitimate software vulnerabilities or encryption research.

[400] *Id.* at 3.

[401] *Id.* at 4.

[402] *Id.*

[403] *Id.*

[404] *See* Liu, *supra* note 377, at 537.

[405] *See* Council Directive 91/250/EEC, 1991 O.J. (L122), art. 5(1)(EC) on the legal protection of computer programs, which provides as follows: "In the absence of specific contractual provisions, the acts referred to in Article 4(a) and (b) shall not require authorisation by the rightholder where they are necessary for the use of the computer program by the lawful acquirer in accordance with its intended purpose, including for error correction."

D. Bugs for Sale: The Legality of Vulnerabilities Market.

It is profitable not to publicly report vulnerability.[406]

As noted earlier in this paper, there is an ongoing legitimate and underground thriving market in software vulnerabilities.[407] The market is largely unregulated, unstructured, and ill-defined, with a real possibility that vulnerabilities information derived from underground market could end up for sale in legal markets and vice-versa, as exemplified by software security firms' routine patronage of black markets for critical vulnerabilities information.[408] Thus, there are different layers and hues to the current software vulnerabilities information market. First, there is the legitimate or legal software vulnerabilities market where brokers such as iDefense and TippingPoint, openly buy and sell software vulnerabilities.[409] Second, there is the underground market managed by malicious hackers, where "zero-day" or critical vulnerabilities information is routinely traded,[410] and where government agents have reputedly offered as much as one million dollars for single vulnerability information adjudged extremely critical and valuable.[411] Third, there is the fledging bug bounty program, an industry initiative that outsources vulnerabilities research to the mass publics often across national boundaries as exemplified by Mozilla and Facebook programs.[412] And fourth, there is the not-for-profit Computer Emergency Response Team (CERT), which acts as a clearing house for vulnerabilities information.[413]

A fortiori, the heady mix of players in the arguably convoluted and chaotic vulnerabilities information marketplace provides a veritable fod-

[406] *See* Bergstein, *supra* note 347 (citing Chris Rouland, chief technology officer, Internet Security Systems, IBM Corporation).

[407] *See* Telang & Wattal, *supra* note 194, at 548 (noting the flourishing market in vulnerabilities and how market value corresponded to the severity of vulnerabilities).

[408] *See From Black Market to Free Market*, *supra* note 183 (noting how even computer security firms routinely patronize black market to buy vulnerabilities from malicious hackers who bedevil them patronize black market).

[409] *See* Rainer Boehme, *A Comparison of Market Approaches to Software Vulnerability Disclosure*, Proceedings of ETRICS 5-6 (Mar. 19, 2006), *available at* https://www.is.uni-muenster.de/security/publications/Boehme2006_CompVulnMarkets_ETRICS.pdf. According to Boehme, vulnerability brokers are often referred to as "vulnerability sharing circles," some sort of clubs that are built around independent and mostly private companies who offer money for new vulnerability reports. The clubs do have customer bases comprising software vendors who would get to know what bugs to patch up, and corporate users who would want to protect their systems. Significantly, while honesty is crucial to membership of the clubs, it is difficult to enforce. *See also* Miller, *supra* note 113, at 2.

[410] *See From Black Market to Free Market*, *supra* note 183.

[411] *See* Roberson, *supra* note 195.

[412] *See* Coletti, *supra* note 180 (discussing Mozilla's bounty offerings for vulnerabilities in their Firefox browser); Sullivan, *supra* note 329 (announcing Facebook's bug for bounty program).

[413] *See* Zittrain, *supra* note 104, at 32, 39.

der for unsavory and underhand trading, with concomitant legal dilemmas and questions on the legality of the following vulnerabilities trading events: First, what is the legality of trading in vulnerabilities information obtained in clear breach of the provisions of the CFAA or the provisions of the DMCA for example? And second, could a contract for the sale or supply of stolen or illegally obtained vulnerabilities information be enforceable in law? These legal questions are clearly not far-fetched and are arguably assured by the continuous parallel existence of legal and underground markets in software vulnerabilities information.

Moreover, there are clearly a number of conceivable scenarios where the questioning of the legality of a vulnerability sale transaction would be apposite. Take for example the alleged purchase for one million dollars of extremely valuable vulnerability information by government agents on the black market.[414] Assuming that the agents were duped into buying bogus vulnerability information or one which was less valuable than previously thought, could there, in the circumstances, be a good and enforceable contract breach of which would be remediable in law? The answer would in turn depend on the nature and source of the vulnerability information in question. If for example, the vulnerability information was stolen or illegally obtained in breach of the relevant provisions of CFAA or DMCA, then the answer would clearly be negative, for the contract of sale would be deemed void *ab initio* and unenforceable, being in clear breach of law and public policy.[415] Under these circumstances, the government could be left with potentially no contractual remedy other than to initiate a criminal prosecution against the seller, who in all likelihood might be insolvent at the time of prosecution.[416]

The apparent lack of transparency in the vulnerabilities information marketplace arguably stems from the secrecy shrouding vulnerabilities information, which is a key determinant of the value of software vulnerabilities. Thus, new or "zero-day" vulnerability information is accorded the most value depending on how critical or important it is, while its value would decline dramatically once it becomes public knowledge.[417] For this reason the golden rule in "vulnerability sharing circles"[418] is

[414] *See* Roberson, *supra* note 195.

[415] According to Black's Law Dictionary, a contract is void *ab initio* if right from the start, it seriously offends law or public policy in contradistinction to a contract, which is merely voidable at the election of one of the parties to the contract. *See* BLACK'S LAW DICTIONARY (Bryan A. Garner ed., 4th pocket ed. 2011).

[416] *See* Cundy v Lindsay [1877–78] LR 3 App. Cas. 45965 (Eng) (finding in the English House of Lords *inter alia* that the underlying fraud in the contract of sale of handkerchiefs which were unpaid for, negated the contract for the sale of the handkerchiefs to a third-party, and that it was as if the contract never existed and that "the pretence of a contract was a failure").

[417] *See From Black Market to Free Market*, *supra* note 183.

[418] *See* Boehme, *supra* note 409 (describing the tightly-knit vulnerability brokers).

that "it is profitable not to publicly report vulnerability."[419] Significantly, vulnerabilities secrecy is further reinforced by the "responsible disclosure policy" that software vendors typically insist on,[420] because a loosely controlled public disclosure that leaves vendors little or no time to apply any corrective patch to vulnerabilities would play right into the hands of malicious hackers and be deemed a reckless disclosure.[421]

Nevertheless, there is a good case for discouraging underground markets in software vulnerabilities due to their propensity for perpetuating malicious hacking activities. It is sacrosanct that a flourishing underground vulnerabilities market would only fuel the drive for malicious hackers with the concomitant side effects of reinforcing a vicious circle of crime. Therefore, it would appear that the surest way to stifle black market development in software vulnerabilities is for software security firms, software vendors and authorities to cease patronage of illicit vulnerabilities information market. This measure would delegitimize malicious hackers, pre-empt vulnerabilities laundering, and no doubt facilitates the growth of legal or legitimate vulnerability markets transparent and amenable to legal oversight, a course that could only enhance software security. Most importantly, insurers would find legal vulnerabilities markets more attractive than markets riddled with malicious hackers peddling illicit vulnerabilities.[422]

E. Could a Liability Regime for Insecure Software Curb Software Vulnerabilities?

It is sacrosanct that software vulnerabilities are inevitable and cannot be completely eliminated.[423] However there is a recurring normative question as to whether software developers could make "higher quality

[419] *See* Bergstein, *supra* note 347.

[420] *See* Li & Rao, *supra* note 173, at 532. *See also* Sullivan, *supra* note 329 (stressing responsible disclosure policy as a precondition for the bounty for bug program).

[421] *See also* Li & Rao, *supra* note 173, at 532 (discussing the impact of vulnerabilities disclosure on software vendors' reputation, market share and customer goodwill and the imperative of safeguarding customers against malicious hackers). This is also exemplified by Facebook's responsible vulnerabilities disclosure policy for its bug-for-bounty program. *See* Sullivan, *supra* note 329 (announcing Facebook's bug for bounty program).

[422] Analysts believe that cyber insurance would help ameliorate financial losses caused by frequent computer security breaches. However insurers have traditionally excluded cyber risks policies due to industry general perception that cyber risks are phenomenally high. *See* Jay P. Kesan et al., *The Economic Case for Cyberinsurance*, U. ILL. L. & ECON. WORKING PAPER No. 2, 1-31 (2004), *available at* http://papers.ssrn.com/so13/papers.cfm?abstract_id=577862 (discussing why cyber insurance is the preferred market solution to IT security risks); Rainer Boehme, Cyber-Insurance Revisited, *Workshop on the Economics of Information Security (WEIS)*, Kennedy School of Government, Cambridge, MA, (2005), at 1-15 (noting that cyber insurance could incentivise the construction of more secure systems).

[423] *See* Randiati & Gonzalez, *supra* note 37.

software products that are more secure and that need fewer patches?"[424] The question is often premised on the hypothesis that software developers could actually make their products more secure, but would not do so because customers were unwilling or unable to pay for "increased costs of additional security measures."[425] Significantly, the claim is also the basis of the market failure rationalization of software insecurity by economics and computer security scholars.[426] However, it was previously critically argued in this paper that even if the market was able to deliver a regime of "more secure" software, it would still not completely eradicate the ubiquitous software bugs or vulnerabilities, as this would be technically impossible.[427] Nevertheless, based on the arguably hypothetical premise that software developers could indeed design more secure software if they really wanted to do so; there is a growing body of literature exploring the propriety and prospects of a liability regime on software developers for insecure software, and whether a liability regime could be a panacea for the recurring software vulnerabilities problem?[428]

However, and most significantly, the literature on the propriety, nature, and scope of a possible liability regime on software vendors for insecure software is understandably mixed because of the mutual awareness that not even a liability regime could completely eradicate software vulnerabilities.[429] The ambivalence in literature is epitomized by the mixed reflections of Michael Cusumano, who simultaneously acknowledged the intractability of software vulnerabilities and still argued for judicial oversight of software products that fell short of industry standard:

> The general philosophy held by software customers, the American Arbitration Association, and the U.S. courts seems to be that software is a uniquely complex product that will probably always have some defects. But companies delivering software that exceeds the bounds of common industry practice are vulnerable to penalties. Because some software companies are much better than others atpreventing, detecting, and fix-

[424] *See* WILSON, *supra* note 189, at 19.

[425] *Id*. at 20.

[426] For a discussion, see generally Part II (C) & (D) of this paper for the analyses of the proprieties of the theory of information asymmetry and tragedy of the commons used by computer security and economics scholars to explain the market failure basis of software insecurity. *See also* Rainer Boehme, *Vulnerability Markets: What is the Economic Value of a Zero Day Exploit?*" PROCEEDINGS OF 22C3 1 (Berlin, Germany, Dec. 27-30, 2005), *available at* http://events.ccc.de/congress/2005/fahrplan/attachments/542-Boehme2005_22C3_VulnerabilityMarkets.pdf; Anderson et al., *supra* note 35, at 636.

[427] *See generally* the analyses in Part II (C) & (D) of this paper.

[428] *See* Almond, *supra* note 164, at 4-7 (discussing whether a liability regime on software developers for insecure software could rein in software vulnerabilities, and noting the constraints of licensing agreements and the limits of liability regime).

[429] *See* Cusumano, *supra* note 136, at 26 (noting that even if a liability regime was imposed, software vendors could not realistically be expected to all security flaws and common defects).

> ing defects, it seems to me that many firms can do better and that courts should hold software firms more accountable for what they license or sell. . . As for whether software companies can ever make their products error-free or invulnerable to security flaws, I think the answer is clear: no, they cannot.[430]

The pertinent questions therefore are: how do we begin to benchmark "common industry practice" as the normative standard when it cannot even guarantee software products that are "error-free or invulnerable to security flaws," and how do we ascertain the parameters of legal liability when there is no definable notion of what "more secure" software is? Perhaps a holistic conception of software products that is inclusive of post-launch maintenance services of perennially "detecting, and fixing defects" would be a better benchmark for measuring industry failings and consequential liability? Kevin Pinkney certainly thought so. While ostensibly acknowledging the futility of securing completely error-free software, he argued for a modified liability rule, where software vendors would only be strictly liable for damage caused by security flaws exploited in their software, if they failed to provide corrective patches for the exploited flaws.[431] However, Robert Hahn et al. were opposed to Pinkney's proposal for a limited liability rule, on grounds that it would inevitably lead to litigation, that the corrective patches might not work, or that they might even exacerbate existing flaws or vulnerabilities, and that the limited liability rule could potentially increase the overall social costs.[432] They then concluded by affirming their belief "that strict liability is not justified at this time."[433] Even so, any liability regime, whether limited or strict, would have to take cognizance of the software licensing regime embodying the End-user License Agreement (EULA) that typically excludes liability.[434] I would however argue that in the context of curbing software vulnerabilities, a liability regime, even if feasible, could only have a minimal impact in reducing vulnerabilities incidents. There is absolutely no legal regime that could undo software vulnerabilities, which are inherently technical problems.

[430] *Id.*

[431] *See* Kevin R. Pinkney, *Putting Blame where Blame is Due: Software Manufacturer and Customer Liability for Security-Related Software Failure*, 13 ALB. L.J. SCI. & TECH. 43, 79 (2002) (proposing that software developers could use evidence that they provided corrective patches as a defense in liability-related lawsuits).

[432] *See* Robert W. Hahn & Anne Layne-Farrar, *The Law and Economics of Software Security*, 30 HARV. L.J. & PUB. POL'Y 283, 340-41 (2006).

[433] *Id.*

[434] *See* Almond, *supra* note 164, at 4-7; *See also* Beard et al., *supra* note 102, at 203-210.

VI. ETHICAL PROPRIETIES OF SOFTWARE VULNERABILITIES RESEARCH AND MARKET.

> The question isn't whether it's ethical to do vulnerability research. If someone has the skill to analyze and provide better insights into the problem, the question is whether it is ethical for him not to do vulnerability research.[435]

Certainly, questioning the ethical proprieties of software vulnerabilities research would appear moot because vulnerabilities research is crucial and imperative for software quality assurance,[436] and the continual investments of government and industry in software vulnerabilities research are without a doubt ethically desirable and justifiable.[437] However, malicious vulnerabilities research and the largely unregulated and unstructured market environment in which it thrives are arguably open to, or should be subjected to, ethical scrutiny. For example, there is something unseemly underhand and morally amiss in being able to traffic and launder "zero-day" software vulnerabilities information derived from a malicious hacking operation or incident on the open market,[438] an activity that is clearly analogous to peddling stolen goods on the high street. It would also seem unethical that vulnerabilities disclosures are routinely made in a deliberately reckless manner with a view to negatively impact the reputation or stock market price of a competitor, or in a way that leaves a software developer helplessly ill-prepared for timely corrective patches, and concomitantly accentuating customers' vulnerability to cyber attacks.[439]

Even so, in the field of applied ethics, things are often not as they seem, and acts or omissions, which are ostensibly unethical or amoral, might turn out to be ethically or morally justifiable upon closer analyses and application of relevant ethical principles.[440] But then the pertinent

[435] *See* SCHNEIER ON SECURITY, *supra* note 25, at 261.

[436] *Id.*

[437] Vulnerabilities research is a security and quality assurance testing conducted to uncover vulnerabilities, and it is usually funded by software vendors and governments. *See* Goodin, *supra* note 179 (discussing a hacking incident commissioned by vendors); TOWARD A SAFER AND MORE SECURE CYBERSPACE 237-40 (Seymour E. Goodman & Herbert S. Lin eds., 2007) (discussing various government-sponsored cyber security researches).

[438] *See From Black Market to Free Market*, *supra* note 183 (discussing how frustrated computer security firms were offering to buy vulnerabilities information from the very hackers who bedevil them).

[439] *See* Telang & Wattal, *supra* note 194, at 544-48 (discussing how vulnerability disclosure "adversely and significantly" affected software vendor's stock performance, and led to customer's dissatisfaction).

[440] Even professional ethicists do occasionally disagree on issues especially when serving as expert ethics witnesses, while ironically using the same set of ethical principles to assess similar or identical moral problematic. *See* Taiwo A. Oriola, *The Propriety of Expert Ethics Testimony in the Courtroom: A Discourse*, 6 J. PHIL., SCI. & L. 18 (2006) (noting how

question is: of what use is or what could ethics literature or ethical perspective contribute to the legal and socio-economic analyses of software vulnerabilities governance? The answer simply is that there are ways that the governance or non-governance of software vulnerabilities and tangent digital computing issues could manifestly and irrevocably impact lives and society; these are precisely what the budding field of computer ethics, or cyberethics, is preoccupied with studying and analyzing.[441] *A fortiori*, an alternative ethical perspective to the tried and tested legal and socio-economic ones is both legitimate and apt for a comprehensive treatment of the issues surrounding the reliability, quality, efficacy, and integrity of an integral element of global digital infrastructures: software. Therefore, in this section, I will employ the principle of utilitarianism in an arbitrary sort of way,[442] to analyze the ethical proprieties of illicit vulnerabilities trading and unscrupulous or reckless disclosures of software vulnerabilities information.

Utilitarianism is an ethical and philosophical principle or political morality that is often used to advance or justify policies and actions, which deliberately promote the greatest happiness of the greatest number of people in the society.[443] In essence utilitarianism, which was championed by Jeremy Bentham, and which dominated his thoughts on law, punishment and social reform,[444] demands that "the production of happiness or the reduction of unhappiness should be the standard by which actions are judged right or wrong and by which rules of morality, laws, public policies, and social institutions are to be critically evaluated."[445] However, it is utilitarianism's absolute fixation on achieving a particular result: i.e., the maximization of the happiness of the greatest number of people in the society, which earned it the additional suffix: "consequentialism".[446] According to Will Kymlicka, "consequentialism"

moral philosophers almost always disagree on ethical issues in different fora ranging from the academia, scholarship, to public spaces).

[441] *See* Herman T. Tavani, Ethics and Technology: Controversies, Questions, and Strategies for Ethical Computing 3-4 (3d ed. 2011) (defining cyberethics as "the study of moral, legal, and social issues involving cybertechnology").

[442] The choice of the above mentioned ethical principle is rather arbitrary, random and non-systematic. However, there are several other applicable ethical and political philosophical principles ranging from legal paternalism, the harm principle, libertarianism, communitarianism, multiculturalism, feminism, justice, to redistributive justice, which due to space constraints, cannot all be applied in this instance. For a whole range of other applicable ethical principles, *see* Will Kymlicka, Contemporary Political Philosophy: An Introduction 1-430 (2d ed. 2002).

[443] *Id.* at 10-11 (describing how utilitarianism is both a comprehensive moral theory and a political morality "that seeks to promote happiness, or welfare, or well-being" of everyone in the society).

[444] *See* Jonathan Woolf, *Society,* in Philosophy 197 (David Papineau ed., 2009).

[445] See H.R. West, An Introduction to Mill's Utilitarian Ethics 1 (2004).

[446] See Kymlicka, *supra* note 442, at 11.

compliments utilitarianism perfectly well because it forbids:

> [a]rbitrary moral prohibitions. It demands of anyone who condemns something as morally wrong that they show *who is wronged*, i.e. they must show how someone's life is made worse off. Likewise, consequentialism says that something is morally good only if it makes someone's life better off.[447]

Thus by extrapolation, while some might perceive gambling as morally wrong, by the tacit terms of utilitarianism's consequentialism, gambling would not simply be prohibited unless it adversely affected the interest of the majority in the society.[448] Viewed from the foregoing analysis, it would be perfectly logical and rational to surmise that utilitarianism cares more about happiness maximization than about who pays the social costs of maximizing the happiness of the majority in the society, which undoubtedly is the minority.[449] In other words, *a la* utilitarianism, it is the end (happiness maximization for the majority) that matters, and for as long as this end is realized, the means would be ostensibly justified. It is this apparent glossing over of, and seeming insensitivity to the minority's interest that drew the ire of critics of utilitarianism.[450] According to Jonathan Wolf, "utilitarianism has been criticized for its insensitivity to the distribution of happiness, because the greatest total happiness may be achieved by a policy that has terrible consequences for the minority."[451]

However, the pertinent question is how would utilitarianism perceives malicious software vulnerabilities research, illicit trading in software vulnerabilities, and unscrupulous disclosures of vulnerabilities information? Given that malicious vulnerabilities research or computer hacking is a potent threat to global digital infrastructures with potential concomitant losses of sensitive personal, business, and official data,[452] it is sacrosanct that the happiness or welfare of the majority in the society would be well served by prohibiting malicious software vulnerabilities research. Thus by extrapolation, malicious vulnerabilities research would be unethical and morally wrong because it could engender the loss of sensitive personal, corporate and government data, and consequently detract from the welfare and happiness of the vast majority of the population who have come to depend on computing systems and digital infra-

[447] *Id.*

[448] *Id.* at 10-11. *See also* Taiwo A. Oriola, *Ethical and Legal Analyses of Policy Prohibiting Tobacco Smoking in Enclosed Public Spaces*, J. L., MED. & ETHICS 828, 835 (2009).

[449] *See* Woolf, *supra* note 444 (noting that it is the majority that will ultimately pay the price of maximizing the happiness of the majority in the society).

[450] *Id.*

[451] *Id.*

[452] *See* Li & Rao, *supra* note 173, at 532 (describing the impact of vulnerabilities disclosure on software vendors' reputation, market share and customer goodwill and the imperative of safeguarding customers against malicious hackers).

structures for crucial services, such as commerce, banking, healthcare, and communication.

By the same token, underground or black market in software vulnerabilities cum reckless or unscrupulous disclosures of vulnerabilities information would be unethical and morally wrong and should be prohibited for the following reasons: first, the flourishing underground or black market in "zero-day" or new software vulnerabilities would continue to fuel malicious hacking of computer systems and digital infrastructures, and should therefore be discouraged, prohibited, and shut down by a total boycott both by the industry and government agents, while illicit vulnerabilities traffickers should be duly prosecuted.[453] Second, unscrupulous and cynical disclosures of software vulnerabilities that leave software developers ill-prepared to design and apply corrective patches could expose millions of networked computer users to cyber attacks, and significantly derogate from the happiness of the overwhelming majority of the population.

Significantly, in the context of utilitarianism, branding malicious hacking of computer systems, the black market in software vulnerabilities, and unscrupulous disclosures of vulnerabilities information as unethical and morally inappropriate would be tantamount to sacrificing the interest of the minority malicious computer hackers and illicit vulnerabilities traders for the advancement or maximization of the happiness of the vast majority of the population. However, it is a happy outcome that not even the harshest critics of utilitarianism could fault, because the minority in this instance are not hard done by and their activities are rightly branded amoral and unethical.[454]

However, as unseemly as it may appear, it is indeed tempting to flip the coin and consider the possible utilitarian assessment of the result of a malicious computer hacking incident which maximizes the happiness of the majority of the population? Surely there are conceivable scenarios where unauthorized computer intrusions or malicious hacking and the concomitant irresponsible software vulnerabilities disclosure could be beneficial and maximize the welfare and happiness of the majority of the population? A good hypothetical example could be an unauthorized malicious computer hacking incident, and a subsequent anonymous disclosure by hackers, on an Internet chat forum, of crucial vulnerability in the

[453] Industrial and government agents have reputedly patronized and continue to patronize the black market for illicit zero-day or new software vulnerabilities. *See* Roberson, *supra* note 195 (noting how a government allegedly paid one million dollars for a critical zero day vulnerability that was adjudged extremely valuable).

[454] Interestingly, Eugene H. Spafford also concluded that the activities of malicious hackers were unethical, although he used a deontological assessment, i.e., focusing on whether or not the act of malicious hacking itself is ethical. *See* Spafford, *supra* note 188, at 64-74.

control systems of a local nuclear power station. Assuming further that the vulnerability, albeit irresponsibly disclosed, had alerted authorities just in time to correct a crucial error in the centrifuges of the local nuclear power station and thereby avert an otherwise imminent nuclear disaster. Without a doubt, such an outcome would be happily welcome in the utilitarian context because it maximizes the welfare and happiness of the majority, if not all of the population that live in close proximity to the nuclear power station. But then does the happy ending justify the means? In other words, does the happy ending make the unauthorized computer hacking incident ethically justifiable or morally appropriate? A polemical answer is clearly inevitable in this circumstance, underscoring the limits of utilitarianism, or any ethical principle for that matter, in their application to social problematic, such as malicious computer hacking and unscrupulous software vulnerabilities disclosures.

VII. CONCLUSION: BEST PRACTICES FOR SOFTWARE VULNERABILITIES GOVERNANCE.

Software vulnerabilities are inherent errors or mistakes in software programming designs and arguably the weakest link in digital information architecture. This paper highlights the inevitability of software vulnerabilities in the contexts of the underlying software's technical dynamics and the economic theories of information asymmetry and tragedy of the commons, which the literature on economics and computer security often use to frame the market failure rationalization of software insecurity. Drawing largely on empirical data, this paper joins issues with the market determinism stance of the economics and computer security literature, and argues that the market as it were, offers sufficient incentive for more secure software, as exemplified by the continuous private and public investments in software security. The paper argues further that even assuming that the market lacks the incentive to deliver more secure software, as claimed by the economics and computer security literature, more secure software would not necessarily translate into perfect or bug-free software, which is technically required for optimum network security. The paper concludes that the only panacea to the perennial software vulnerabilities problem is vulnerabilities detection research, and application of timely corrective patches to software vulnerabilities.

The paper then explores the largely unregulated market in software vulnerabilities, where legitimate and underground markets co-exist and where illicit vulnerabilities could easily be laundered and sold on the legal market. The paper notes that the underground market in vulnerabilities is muddling up the vulnerabilities market, and urges that industry and government should cease patronizing the underground market for vulnerabilities, and penalize illicit vulnerabilities trading in order to stem the tide of unauthorized and malicious computer and networked systems hacking and intrusions.

On the premise that software is proprietary, the paper examines the legal and ethical proprieties of vulnerabilities research and disclosure, with special focus on the tort of trespass, CFAA, intellectual property rights, privacy legislation, and the ethical principle of utilitarianism. The paper explores potential legal and ethical obstacles to vulnerabilities research, and analyzes how legitimate research could be conducted professionally around existing laws, and how best to clamp down on malicious vulnerabilities research. The paper specifically argues for favorable research exception under the CFAA and Wire Tap legislations in order to ensure effective vulnerabilities research, the only panacea to the perennial problem of software vulnerabilities. While discussing the limit of utilitarianism as an ethical framework for appraising the ethics of vulnerabilities research, the paper observes that legitimate vulnerabilities research is ethically and morally justifiable, while malicious software vulnerabilities research and disclosure would be unethical and morally reprehensible.