Brooklyn Law School

From the Selected Works of Derek Bambauer

March 1, 2010

The Hacker's Aegis

Derek E Bambauer, Brooklyn Law School
Oliver Day

Available at: https://works.bepress.com/derek_bambauer/26/
THE HACKER’S AEGIS

Derek E. Bambauer and Oliver Day

Abstract

Intellectual property law stifles critical research on software security vulnerabilities, placing computer users at risk. Researchers who discover flaws often face IP-based legal threats if they reveal findings to anyone other than the software vendor. This Article argues that the interplay between law and vulnerability data challenges existing scholarship on how intellectual property should regulate information about improvements on protected works, and suggests weakening, not enhancing, IP protections where infringement is difficult to detect, lucrative, and creates significant negative externalities. It proposes a set of three reforms – “patches,” in software terms – to protect security research. Legal reform would create immunity from civil IP liability for researchers who follow “responsible disclosure” rules. Linguistic reform would seek to make the term “hacker” less threatening either by recapturing the term’s original meaning, or abandoning it. Finally, structural reform would ameliorate failures in the market for software vulnerability data by having a trusted third party act as a voluntary clearinghouse. The Article concludes by describing other areas, such as physical security, where reforming how law coordinates IP improvements may be useful.

Contents

I. Introduction: != Bulletproof ............................................................2
II. The Software Security Ecosystem .........................................................6
   A. The Stakes ......................................................................................6
   B. Bug Hunters ..................................................................................15
III. The Vendor’s Arsenal .........................................................................17
   A. Copyright: Breaking the Censor’s Scissors .......................................17
   B. Patent ..........................................................................................18
   C. Trade Secret ..................................................................................22
   D. Trademark ....................................................................................27
   E. Digital Millennium Copyright Act (DMCA) .....................................28
IV. Creating the Aegis .............................................................................34

* Assistant Professor of Law, Brooklyn Law School (Bambauer); Director of Research, StopBadware, Berkman Center for Internet & Society, Harvard Law School (Day). The authors thank Oded Burger, Jennifer Burgomaster, Jelena Kristic, Brad Reid, and Chris Vidiksis for expert research assistance. Thanks for helpful suggestions and discussion are owed to Miriam Baer, Fred Bloom, Mike Carroll, Ed Cheng, Dino Dai Zovi, Erik Dykema, Robin Effron, Jennifer Granick, Brian Lee, Jason Mazzone, Phil Malone, Liberty McAteer, Bill McGeveran, Jessica Silbey, Chris Soghoian, Alexander Sotiroy, Marketa Trimble, Scott Velez, Julie Cromer Young, the IP Colloquium at Brooklyn Law School, and the participants in the Seventh Works in Progress in Intellectual Property Conference. The authors gratefully acknowledge the Dean’s Summer Research Stipend Program, and Joan Wexler, at Brooklyn Law School for financial support. The authors welcome comments at <derek.bambauer@brooklaw.edu> and <oday@fas.harvard.edu>.
I. INTRODUCTION: != BULLETPROOF\textsuperscript{1}

Mike Lynn had done the impossible. He had found a way to crack open the operating system on Cisco’s Internet routers, causing them to run his code.\textsuperscript{2} Routers were Cisco’s most important product – and the backbone of much of the Internet – precisely because they had been legendarily immune to such attacks.\textsuperscript{3} Lynn, though, had discovered their Achilles’s heel. The routers’ vulnerability placed a wide swath of Internet infrastructure at risk.

Lynn, an experienced security researcher with the firm ISS, followed the protocol of “white hat” hackers, who probe for computer software and hardware flaws with the goal of discovering, not exploiting them.\textsuperscript{4} He reported his findings to Cisco, which dutifully issued a patch to correct the bug.\textsuperscript{5} But Cisco – concerned with damaging the invincible image of its products – refused to draw particular attention to the patch, or to press customers to implement it.\textsuperscript{6} Lynn, worried by Cisco’s decision not to

\textsuperscript{1} In programming languages, != means “not equal to”. See, e.g., PYTHON V2.6.4c1 DOCUMENTATION: THE PYTHON STANDARD LIBRARY, at http://docs.python.org/library/stdtypes.html#comparisons (documenting != comparison operator).
\textsuperscript{4} By convention, black hat hackers discover bugs for financial gain or malicious reasons, and gray hat hackers behave either as white hats or black hats, depending on the circumstances.
\textsuperscript{6} Zetter, supra note 2.
publicize the fix, prepared to give a presentation at the Black Hat hacker conference in Las Vegas that would detail the basic concepts of the bug, but would withhold information about how to exploit it.\(^7\)

Cisco objected, fervently. Employing a range of legal theories from intellectual property law, the company convinced a federal judge to issue a restraining order preventing Lynn from giving his presentation.\(^8\) They also forced conference organizers to rip the printed version of Lynn’s slides out of the conference materials, and to turn over CDs with a copy of his slideshow.\(^9\)

This Article argues that conflicts such as the one between Lynn and Cisco conflict are both increasingly common and socially harmful. Intellectual property (IP) law stifles the dissemination of critical research on software security vulnerabilities. We argue that IP law’s incentive effects are superfluous for these bugs, as security research is an exemplar of “peer production” as conceptualized by Yochai Benkler\(^10\), Eric Von Hippel\(^11\), and Eric Raymond\(^12\). Researchers hunt bugs for a variety of reasons: intellectual curiosity, ideology, reputation, and, occasionally, remuneration. For vulnerability research, IP law plays a suppressive rather than a generative function – it blocks or limits whether, and how, hackers share their findings. We argue that, much as researchers hack software to make it behave unexpectedly and thereby serve their purposes, software vendors have hacked IP law, using it for ends unrelated to its original purpose.

Critically, IP law – like the software it protects – malfunctions here. It enables software firms to suppress information about flaws. It presses researchers to avoid legal risks from public disclosure, and to gain financially, by offering their findings on the black market rather than through legitimate channels. Software vulnerability research challenges standard intellectual property scholarship on the regulation of information on improving a protected work or invention. Under current doctrine, someone who possesses information about how to improve a work or invention protected by IP has three options: bargain with the IP owner, seek an improvement patent, or infringe. Contemporary scholarship focuses on tuning patent and copyright law to generate optimal incentives and to coordinate improvements. Thus, Mark Lemley argues that it is unnecessary

\(^8\) Id.
for inventors to capture the full social value of their advances, and that patent law should not set this internalization as a goal. Robert Merges and Richard Nelson analyze the incentive effects of various standards for setting the scope of a patent, as does Edmund Kitch. Wendy Gordon justifies control over improvement information by IP owners as useful in reducing transaction costs, as do William Landes and Richard Posner. Michael Heller and Rebecca Eisenberg worry about the problem of holdout costs when multiple parties must bargain over improvements. Current scholarly wisdom thus presses towards conferring control over improvement information to IP owners.

This Article, in contrast, identifies software security research as a counterexample, where strong controls over improvement information by IP owners are harmful. Security bugs are problematic for three reasons: infringement is difficult to detect, socially harmful due to negative externalities, and lucrative. We argue that information law should be alert to similar situations and that, counterintuitively, such circumstances require a diminution, not an increase, in IP protections. The Article goes on to suggest additional areas where the channeling effect of legal rules over improvement information may be critically important.

This Article is also the first to propose a set of reforms – “patches,” in software terms – to protect socially valuable security research, guide behavior by those searching for vulnerabilities, and channel dissemination of vulnerability data towards legitimate consumers. Other legal scholarship treats intellectual property law as a lost cause. Jennifer Granick argues compellingly against restrictions on vulnerability disclosures, but focuses on IP law solely as a barrier. Peter Swire, in assessing incentives for vulnerability disclosure, notes law’s role as a barrier to a firm’s competitors

---

in creating equivalent software.\textsuperscript{20} Susan Brenner evaluates the Digital Millennium Copyright Act (DMCA) as a form of information censorship.\textsuperscript{21} Bruce Kobayashi argues for more extensive intellectual property protection to drive adoption of cybersecurity.\textsuperscript{22}

Computer scientists are even more pessimistic. Stephen Bono, Aviel Rubin, Adam Stubblefield, and Matthew Green refer to “security through legality” as a “hopelessly flawed methodology.”\textsuperscript{23} Tom Cross views efforts to limit hackers’ investigations as embracing the view that “ignorance makes you safer.”\textsuperscript{24} And Paul Graham, who invented Bayesian spam filtering, views copyright as “a threat to the intellectual freedom [hackers] need to do their job,” which is to reduce the creation and impact of poorly written software code.\textsuperscript{25}

In contrast, this Article seeks to adapt IP law, rather than abandoning it as a tool. The proposed reforms use three methods to accomplish this. First, we argue that the security research community should try to shift the largely negative, threatening set of connotations created by the term “hacker.” If bug hunters cannot reclaim the word’s original meaning, they should cede it and employ an alternative.

Second, a voluntary intermediary – a vulnerability clearinghouse – should be established to coordinate contact between vendors and researchers, to document identified bugs, and to track their evolution. The clearinghouse can address key structural flaws in the market for vulnerability information that impede legitimate transactions and push researchers to sell information illicitly.

Finally, the Article proposes regulating researcher behavior in exchange for a shield from IP law. If hackers follow a prescribed course of conduct during their investigations – roughly tracking the “responsible disclosure” model used in the security community – they should be granted immunity from civil intellectual property liability for that research.

The goal of these reforms is to channel disclosures of vulnerability information in legitimate directions. Threats of legal liability may prompt

\textsuperscript{24} Tom Cross, \textit{Academic Freedom and the Hacker Ethic}, 49 COMM’NS OF ACM 37, 39 (June 2006).
The Hacker’s Aegis

researchers to offer their discoveries on the (lucrative) black market or to withhold research altogether, rather than risking a lawsuit by contacting a vendor or publicizing their findings. The reforms may spur researchers to undertake additional investigations, producing more information about bugs; however, any such benefit (while helpful) is secondary to the useful effects on information distribution.

Part II of this Article describes the ecosystem in which security researchers operate. Part III catalogs the intellectual property tools available to threaten and control hackers, and suggests what doctrinal patches are needed to protect security research. Part IV describes the Article’s three proposed reforms to mitigate IP’s ill effects in this context, and Part V describes how security research unsettles standard analysis of IP regulation of improvements. The Article concludes with observations on how its analysis can be applied outside the realm of computer software.

II. THE SOFTWARE SECURITY ECOSYSTEM

A. The Stakes

Finding security bugs matters. Users face an increasingly hostile Internet environment – one where malware, viruses, identity theft, phishing, and denial of service attacks are ubiquitous. In the United States, hackers took control of over a million personal computers in the last three months of 2009, adding them to the 10 million already infected with rogue code. Security firm Panda Labs tested over 22 million computers and found nearly half (48.35%) infected with malware. The consequences of suffering a hack or an infection can be significant, as the loss of sensitive personal data due to security breaches has become commonplace. A hack at the University of North Carolina School of Medicine exposed the personal data and medical information of 160,000 mammography patients. The Director of the Federal Bureau of Investigation gave up on-line banking

---

26 Ellen Nakashima, China leads the world in hacked computers, McAfee study says, WASH. POST, at http://www.washingtonpost.com/wp-dyn/content/article/2010/02/14/AR2010021403817.html?hpid=sec-tech (Feb. 15, 2010).
The Hacker’s Aegis

after nearly falling for a phishing e-mail that appeared to come from his bank.29 A vulnerability in one of Time Warner Cable’s standard cable modem / wi-fi router units allowed hackers to change the device’s settings and, potentially, intercept data sent through it.30 Viruses can even spread from infected PCs to Web sites.31 Vulnerabilities in software code create the weaknesses that hackers exploit.

Bugs put not only an individual’s information at risk, they create a threat to other Internet users as well. Security flaws present two forms of negative externalities. First, each user whose software is compromised increases the risk to her peers. Computers infected with viruses or malware are often aggregated into botnets that are used to send phishing spam, launch denial of service attacks, or distribute malicious code.32 The harm suffered by the person with the compromised computer is considerably less than the aggregate damage to society and other users. Second, users face greater harm than vendors do, especially overall.33 While precise figures are difficult to ascertain, reliable estimates of the worldwide economic damage caused by viruses range from $13 billion for worms and viruses, and $226 billion for all attacks, in 200334, to $157-192 billion in 200435. Losses to vendors from security breaches, such as from increased support costs, reputational harm, and declines in share price, are also uncertain, but likely considerably smaller.36 Vendors, therefore, have less incentive to fix bugs than is socially optimal.

The rise of cloud computing and mobile computing worsens the problem. For example, the popular micro-blogging service Twitter suffered a security breach when a hacker cracked an employee’s Google Gmail

35 mi2g, $290 of malware damage per Windows PC worldwide in 2004; XP Service Pack 2 creates “Haves and Have Nots” as road forks, available at http://www.mi2g.com/cgi/mi2g/frameset.php?pageid=http://www.mi2g.com/cgi/mi2g/press/240804.php.
account, giving him access to business documents stored on Google’s Apps service.\footnote{37} The hacker then forwarded confidential company documents to the Web site TechCrunch, which published them.\footnote{38} He also took over the e-mail accounts of senior executives, and gained access to Twitter employees’ phone records, personal e-mail messages, and credit card data.\footnote{39} Thus, a weakness in a cloud computing application – here, the password-recovery feature of Gmail – caused a cascade of harm to multiple users and to their employer.\footnote{40} Chinese hackers were able to penetrate Google’s security to access accounts of human rights activists by exploiting a security flaw in Microsoft Internet Explorer.\footnote{41} Similarly, in March 2009, a flaw in Google Docs briefly exposed private documents to the public, causing the Electronic Privacy Information Center to file a complaint with the Federal Trade Commission charging Google with a deceptive trade practice.\footnote{42}

As consumers increasingly store data on, and connect to the Internet with, smartphones, vulnerabilities in devices such as Apple’s iPhone\footnote{43}, and operating systems such as Google Android\footnote{44}, put sensitive personal information at risk. The growing move to use phones for payment systems – whether pay-by-SMS or PayPal – makes hacking phones even more attractive.\footnote{45}

Bugs happen. Inevitably, software code is imperfect.\footnote{46} While vendors find and fix some flaws, the demands of the release cycle, and the panoply of configurations and interactions that software encounters when

\footnote{39} Id.
\footnote{40} Miller & Stone, supra note 37.
\footnote{46} VIEGA, supra note 36, 139-44.
deployed by users, ensure that bugs slip through into production code. Some of those bugs create security weaknesses that can be exploited. The research firm Gartner calculates that 75% of security breaches result from software flaws.\(^{47}\) Even large, security-conscious vendors produce vulnerable code. Oracle faces a new automated tool that allows any minimally-skilled computer user to break into the firm’s database software over the Internet.\(^{48}\) In October 2009, Microsoft released a record number of fixes for Patch Tuesday – even though its code base did not yet include the new operating system Windows 7.\(^{49}\) Adobe recently patched a vulnerability in its ubiquitous Acrobat software that allowed hackers to access data on vulnerable computers – nearly a month after the bug was first reported, and code to exploit it became available.\(^{50}\)

Vulnerabilities surface quickly. As Eric Raymond famously observed, “Given enough eyeballs, all bugs are shallow.”\(^{51}\) Some users encounter bugs unexpectedly; others know how to look for them. Hackers are expert in how software fails.\(^{52}\) While they lack inside information about the software, there are more of them than there are engineers on quality assurance teams even at large software firms, and they are highly motivated – by money, by reputation, and even by ideology. In one week in July 2009, for example, outside researchers released information about security flaws in the Linux operating system kernel\(^{53}\), the Mozilla Firefox browser\(^{54}\), and

---

52 See, e.g., Hackers, FRONTLINE, at http://www.pbs.org/wgbh/pages/frontline/shows/hackers/.
the Bluetooth communications protocol.\textsuperscript{55} The Chromium open source project has acknowledged that outside researchers found a number of critical bugs in its browser, and Google has begun offering rewards to hackers who find flaws in it.\textsuperscript{56}

Moreover, outsiders have the advantage of time. A software company’s testers must re-check each new version of a program, and have a limited period of time to inspect the final code before it is released to the public.\textsuperscript{57} Independent researchers can examine the ultimate version at their leisure. In addition, hackers quickly convert information on security flaws into tools for exploiting them. One comprehensive study of vulnerabilities found that over 70\% of bugs had exploit code available by the time the flaw was publicly disclosed.\textsuperscript{58} The incidence of “zero day” bugs – security holes that become public before vendors have software patches ready – is rising sharply.\textsuperscript{59} In short, outside researchers will always find flaws that vendors did not catch, and some of those bugs will have serious security repercussions.

This problem for vendors worsens as software becomes more popular and more complex. Operating system (OS) software, for example, is particularly subject to flaws. An OS must expose key aspects of its internal workings to the software development community, creating the possibility that a bug in an application can wreak havoc on the operating system.\textsuperscript{60} As more developers write applications for the platform, the OS must maintain backwards compatibility (ensuring that programs written for its earlier versions work with the latest one) and must test an increasing number of software interactions and dependencies. Demands from developers, and the need to ensure that older software continues to run properly, can lead OS vendors to tolerate security flaws that could be eliminated at the cost of sacrificing backwards compatibility.

For example, Microsoft maintained a weak, easily cracked password feature (the LAN Manager password hash) in Windows 2000 to avoid breaking numerous third-party applications written for earlier versions of


\textsuperscript{57} JEFF TIAN, SOFTWARE QUALITY ENGINEERING (2005).

\textsuperscript{58} Stefan Frei et al., \textit{Large-Scale Vulnerability Analysis}, ACM SIGCOMM WORKSHOP 131, 135 (2006).

\textsuperscript{59} \textit{Id.} (noting “the number of zero-day exploits is increasing dramatically”).

The trade-off for the Redmond company may have been rational: the benefits of having additional content running on Windows might outweigh the added security risks – to Microsoft -- of LAN Manager hacks. However, for versions of the operating system through Windows 2000, LAN Manager was the default authentication method in certain circumstances. Thus, individuals or companies running Windows would be vulnerable unless they actively took steps to prevent an attack.

The LAN Manager example highlights a critical puzzle: vendors do not always act to fix known weaknesses – or, at least, to fix them promptly. Indeed, companies may learn about vulnerabilities at no cost, as when independent researchers discover and report bugs. Examples abound. Juniper Networks barred one of its researchers from giving a talk at the Black Hat and Defcon conferences about a vulnerability in Automated Teller Machines (ATMs), even though the affected vendor had known of the flaw for nine months. Apple failed to fix three weaknesses in its iCal scheduling software, despite having four months of advance warning from researchers at Core Security. Sun Microsystems left a vulnerability in its Solaris operating system unpatched for over six months, even though it allowed hackers to crash the software via a buffer overflow exploit.

Bruce Schneier, a security expert, posits two reasons for vendors’ lassitude in patching such bugs. First, he notes that harms from vulnerabilities affect vendors far less than customers. Bugs create a negative

---

64 See generally Ashish Arora, Rahul Telang, & Hao Xu, Optimal Policy for Software Vulnerability Disclosure, at http://www.heinz.cmu.edu/research/256full.pdf (arguing vendors patch later than socially optimal).
externality. Furthermore, if customers cannot discern which component of a system is responsible for a problem – for example, detecting whether the flaw was in the operating system, the application, or the data – vendors face reduced reputational or market pressures to improve security.\textsuperscript{70} Second, customers tend to value new features or faster releases over slower, more limited, but more secure software. Added features generate more sales than reduced bugs.

Even if vendors do patch vulnerabilities, they may not call users’ attention to the need to install those fixes.\textsuperscript{71} Despite vendors’ urgings, users cannot always install each new patch. Large-scale users, especially corporations, have limited windows in which they can install patches – typically, they do so several times a year to ensure sufficient time to test the stability of those changes in their environments.\textsuperscript{72} Thus, even if a vendor releases a patch, customers may not have sufficient information to appreciate the relative necessity of applying it immediately – and those who do may be constrained from doing so by the demands of their computing environment. Software companies may also be reluctant to reveal flaws due to fears that disclosure can increase, not decrease, risk.\textsuperscript{73} Describing a bug – even in the documentation available with a patch that remedies it – creates hazards.\textsuperscript{74} Attackers can use the description to reverse-engineer the flaw, and then to create code that exploits it. This highlights the challenge that vendors, and researchers practicing responsible disclosure, face: if they describe flaws with too much precision, hackers can probe the weaknesses, but if they are too general, customers will encounter difficulty taking precautions. Vendors thus prefer to keep vulnerabilities secret, believing this best protects them and their customers while fixes are readied.\textsuperscript{75} To improve security, though, software companies need not only to fix vulnerabilities, but to inform users so they apply those fixes.\textsuperscript{76} This is particularly important since black hat hackers – those who employ

\textsuperscript{70} See generally Viega, supra note 36, at 144.
\textsuperscript{71} See, e.g., Zetter, supra note 2.
\textsuperscript{73} Tom Cross, Academic Freedom and the Hacker Ethic, 49 Comm’ns of ACM 37 (June 2006).
\textsuperscript{75} Id. (describing “silent patching”).
\textsuperscript{76} See Bruce Schneier, The Nonsecurity of Secrecy, 47 COMM’NS OF ACM 120 (Oct. 2004) (stating “We are all less secure if software vendors don’t make their security vulnerabilities public”).
vulnerability data to compromise systems for gain – frequently have access to information about bugs already.\(^\text{77}\)

It may also make economic sense for vendors to underplay bugs. Though users accept that all software has flaws, a vendor may worry about its reputation relative to competitors if it publicizes widely each bug it patches.\(^\text{78}\) The concern is strategic behavior: a competing vendor who keeps vulnerabilities quiet may enjoy an advantage in perception, even if its underlying code is no more secure. Absent information to detect this strategy, users may assume that the number and severity of reported (and even patched) bugs correlates with software quality.\(^\text{79}\) Rational vendors would thus tend to report bugs less frequently, and with less dissemination, than would be socially optimal.

The other key aspect of vendors’ decisions is that fixing vulnerabilities is costly. The cost to fix a bug increases as the software development cycle progresses; once the code is in production and use – when independent researchers typically first get access to it – the cost is greatest.\(^\text{80}\) One study of United Kingdom businesses found that for every dollar spent on software development, a company spent 75 cents on average to remediate security flaws.\(^\text{81}\) Analysts agree that fixing security bugs is expensive, though quantifying those costs with precision is difficult. Vulnerabilities in Web applications may cost as little as $400 per flaw to fix, while a cross-platform vulnerability in software such as Oracle’s can require over one million dollars.\(^\text{82}\) An IDC study found that fixing bugs in applications developed in-house by corporations costs from $5 million to $22 million per year, depending on the organization’s size.\(^\text{83}\)

Software code inevitably has weaknesses that internal testing fails to discover. Outsiders find these flaws in time. Even if they report bugs to the

\(^{77}\) Christiansen, \textit{supra} note 74.


\(^{79}\) Christiansen, \textit{supra} note 74.


vendor, that company may not fix the problems for financial or reputational reasons. This can generate conflicts with outside researchers, whose behavior and motivations are explored in the next section.
B. Bug Hunters

Independent security researchers are a puzzle: they find bugs for free, even when software firms normally pay for this work. Broadly speaking, there are three types of testers: software company employees, consultants, and independent researchers. Employees – generally called Quality Assurance or Quality Engineering – are compensated directly for their work by the software vendor (their employer). Consultants, too, earn remuneration from the vendor by searching for flaws under contract. Independent researchers, though, are neither paid by nor affiliated with the vendor. They might benefit indirectly from their tasks, such as when a tester uses the software herself, works for a firm that does so, or employs testing as a signal of skill or experience. But most hackers have an attenuated relationship at most with the producer of the code they try to break.

Independent researchers test for a variety of reasons: possible future remuneration, intellectual satisfaction, peer recognition, ideological commitment, animus towards a particular vendor, and expectations in a larger community of testers, among others. In short, their incentives are diverse. Their actions represent another example of peer production – of creation outside the standard market economy through a disaggregated, informal process. Scholars such as Jonathan Zittrain, Yochai Benkler, Eric Raymond, and Eric von Hippel have analyzed other instances of peer production, from mapping craters on Mars to Wikipedia to open source software to kitesurfing. Independent bug hunters analyze software for many reasons, but few are linked directly to financial incentives.

Thus, the standard incentives-based model for intellectual property does not apply to hackers. Researchers who find bugs can obtain IP protection for their discoveries rarely, if at all. Vulnerability data consists of insight into how code functions, or fails to do so, and would be

---

84 TIAN, supra note 57.
85 See generally VON HIPPEL, THE SOURCES OF INNOVATION, supra note 11.
90 See NASA, Clickworkers, at http://clickworkers.arc.nasa.gov/.
92 Benkler, supra note 91, at 63-67.
93 VON HIPPEL, DEMOCRATIZING INNOVATION, supra note 89, at 103-104.
94 See, e.g., 17 U.S.C. § 103(a) (denying copyright protection to unauthorized derivative works); see generally Derek E. Bambauer, Faulty Math: The Economics of Legalizing The Grey Album, 54 ALA. L. REV. 345, 348-54 (2008).
unprotectable under copyright as either an idea, or as an unauthorized derivative work of the underlying program. While a patent on the bug might theoretically be available, the time lag for prosecution and the existence of the program as prior art make this possibility largely irrelevant. The incentives generated by IP law do little to spur independent researchers to test code.

Intellectual property doctrine does have a more subtle, second-order effect on researchers’ behavior, but it affects how they distribute vulnerability data rather than whether they produce it. Vendors’ goals for distribution effects are straightforward: they want to be the sole recipients of bug data. IP law can be deployed to shape with whom hackers share information, and when, regardless of any effects on whether they conduct such research to begin with. Put crudely, vendors frequently employ IP law as a gag order.

This approach is significantly misguided. Researchers have an easy and profitable distribution alternative to sharing data with vendors: they can sell their discoveries on the black market. Organized crime entities, malware operators, and governments pay well for vulnerabilities in important software products, particularly those with no known patch or defense. Selling bug data to the underground is appealing for several reasons. First, it is hard for vendors to detect these transactions: research on bugs takes place in private, as does the exchange of exploits for money. Second, the black market typically pays better for bugs than the legitimate market does. In 2006, anti-virus vendor Trend Micro found that code exploiting a vulnerability sells for $20,000 – 30,000 depending upon how widely used the insecure software is and how reliable the exploit code is, while code that takes over a PC and adds it to a botnet goes for roughly $5000. Finally, the legal risks are, ironically, likely lower in this setting. If what a hacker does to enumerate a flaw is unlawful, she runs less risk by concealing this activity than by announcing it to the vendor, the security

---

96 In 2009, the average time from filing to patent issuance or abandonment was 34.6 months. U.S. Patent & Trademark Office, FY 2009 USPTO Performance and Accountability Report, at http://www.uspto.gov/web/offices/com/annual/2009/nda_02_02.html.
community, or the larger public. Both participants to the illegal transaction have incentives to conceal it, and while they may face criminal penalties if caught, their risk of detection is typically low. The risk is clear: if hackers fear lawsuits for publishing vulnerability discoveries, they can opt to sell their findings on the black market at lower risk and greater reward, placing users at risk.

Thus, intellectual property has but a muted effect on the production of vulnerability data by independent security researchers, but can have a profound effect on its distribution. However, this impact is perverse: rather than pushing bug hunters into sharing information with vendors, it increases the attractiveness of distribution through illicit channels, to consumers who are likely bad actors. The black market is discreet and profitable. Nonetheless, vendors continue to deploy a range of IP-based legal weapons in an attempt to control researchers. The next section examines these tools.

III. THE VENDOR’S ARSENAL

A. Copyright: Breaking the Censor’s Scissors

The Great Firewall of China has holes. Scholars at the University of Michigan found key flaws in part of the firewall, Green Dam-Youth Escort, created by Jinhui Computer System Engineering (JCSE). JCSE built Green Dam to augment China’s formidable Internet censorship apparatus; the Chinese government mandated that all computer manufacturers install – or at least ship – the software on every new computer. Green Dam not only censored users’ Internet access, it created significant security risks. The Michigan researchers found that vulnerabilities in the code could permit malicious Web sites to take control of a user’s computer to steal personal information or to enlist the PC in a botnet.

Jinhui – already under public pressure for helping China’s government censor the Internet – responded with indignation, and a threat. The company’s general manager stated that, “It is not responsible to crack somebody’s software and publish the details, which are commercial secrets, on the Internet. They [the Michigan professors] have infringed the

---

100 See, e.g., 18 U.S.C. § 1030 (penalizing unauthorized access to computer systems with imprisonment and fines).
102 Scott Wolchok, Randy Yao, & J. Alex Halderman, Analysis of the Green Dam Censorware System, at http://www.cse.umich.edu/~jhalderm/pub/gd/ (last updated June 18, 2009).
The Hacker’s Aegis

copyright of our product.”¹⁰⁴ He added that Jinhui planned to sue the researchers.¹⁰⁵ While a suit in the U.S. would likely fail – reverse-engineering is protected as fair use under copyright law¹⁰⁶, and is considered a legitimate practice by trade secret doctrine¹⁰⁷ – and the Michigan team would have little to fear from legal action in China, this response typifies vendors’ reactions to public disclosure of vulnerabilities. Jinhui patched some flaws that the Michigan team found, yet simultaneously threatened them for performing free quality control.¹⁰⁸ To software companies, the perception of security is often more important than security itself. And when that perception is threatened, intellectual property threats are often their first response.

The following sections detail the IP theories that software companies use, and the doctrinal adjustments this Article argues are necessary to protect security research.

B. Patent

Chris Paget was going to give a presentation at the Black Hat conference in 2007 that would show how to clone an RFID (radio frequency identifier) chip – the kind that is used in access cards to control access to buildings, in tags that allow drivers to pay tolls without stopping, and in passports to verify one’s identity.¹⁰⁹ His subject was an RFID sensor made by HID Global; Paget chose the company because it produced the ID cards in the building where his employer, IOActive, was located.¹¹⁰

HID Global objected, strongly. Their letter to IOActive demanded that the research firm not publish information about how to “spoof” HID’s cards – or face legal action for patent infringement.¹¹¹ HID asserted Paget’s cloning would violate two of its patents, which cover an identification system using passive integrated transponders.¹¹² The threat created significant legal risk for Piaget and IOActive: if their cloner was covered by the claims of one or more of HID’s patents, and they proceeded in the face

¹⁰⁵Id.
¹⁰⁶See, e.g., Sony Computer Ent’mt v. Connectix Corp., 203 F.3d 596 (9th Cir. 2000).
¹⁰⁷See, e.g., Chicago Lock v. Fanberg, 676 F.2d 400, 404-05 (9th Cir. 1982).
¹⁰⁸Green Dam breached, supra note 104.
¹¹²Letter from HID Global to IOActive, Feb. 21, 2007 (on file with author).
of the vendor’s warnings, they would be liable for willful patent infringement. Willful infringement subjects a defendant to increased damages – up to three times actual damages – and attorney’s fees.

They decided not to give the offending presentation, and Black Hat staffers tore their prepared materials out of the conference packets. Instead, the researchers gave a generic presentation, with no mention of HID or its technology.

Patent protection is among the most powerful weapons a software vendor can deploy to control its code. Patent law confers a monopoly over making, using, selling, or offering to sell the protected invention. Infringement operates under strict liability: anyone who creates a product, or performs a process, that incorporates all elements listed in a patent’s claims violates the patent owner’s rights. Defenses are scant, and damages are at least a reasonable royalty for use plus interest. There is no fair use in patent law: at most, an “experimental use” exception immunizes use of an invention for “amusement, to satisfy idle curiosity, or for strictly philosophical inquiry.” Any commercial use places an infringer outside this safe harbor.

Thus, patent law incorporates none of the utility calculus present in the copyright or trademark fair use defenses; no matter how

\[113\] 35 U.S.C. § 284 (2008); see generally In re Seagate Tech., 497 F.3d 1360, 1368 (Fed. Cir. 2007) (holding “an award of enhanced damages requires a showing of willful infringement”).


\[117\] 35 U.S.C § 271(a).

\[118\] See, e.g., Bio-Technology General Corp. v. Genentech, 80 F.3d 1553, 1559 (Fed. Cir. 1996).


\[122\] Id.

beneficial a researcher’s findings, if they are obtained in violation of a patent, without authorization, the researcher is liable.

Paget and IOActive made the rational choice to alter their presentation. But society is ill-served by patent’s strict liability in the software security context. RFID tags, for example, are becoming ubiquitous, appearing in subway fare cards, animal identification implants, library books, bicycle race trackers, and shipments of Oxycontin. Vulnerabilities in their operation become particularly worrisome now that RFIDs play a key role in governmental operations such as border control and Department of Defense procurement.

Paget demonstrated the risks: with a few hundred dollars of equipment loaded into his Volvo, he was able to “skim” the serial numbers for six passport cards within an hour of driving along San Francisco’s Fisherman’s Wharf. Despite these problems, researchers cannot lawfully test HID’s RFID chips without authorization, and even if they go ahead without permission, they cannot legally distribute their findings, since doing so would prove infringement. HID thus obtains an effective veto over probing its patented products to assess their security.

---

The scope of the threat to software security research from patent law may change after the U.S. Supreme Court decides *Bilski v. Doll*, a case in which business method patents are at issue that will also likely offer guidance on whether, and when, computer software can be patented.\(^{134}\) Even if the Court should find that such method patents are not directed to patent-eligible subject matter, vendors could still obtain protection for methods implemented via a specific computer, which would meet the Court’s “particular machine” test.\(^{135}\) While the patent problem could become more or less pressing for security researchers, it is virtually certain to persist.

Change could come from revising the Patent Act or re-interpreting the experimental use exception. Legislation to exempt security research from infringement would be in line with prior moves to create exceptions to liability. Congress has previously created exemptions for socially beneficial uses that would otherwise infringe, from protecting doctors against liability for using protected surgical methods\(^{136}\) to allowing prior users of a patented business method to continue employing it\(^{137}\). Creating a narrow exception to patent liability for security research would equally shield helpful activity that would otherwise be subject to injunctive prohibition.\(^{138}\) Unlike the surgical methods exemption, though, any protection for security research would likely need to protect tools specifically adapted to the patented method, since software programs needed to probe a vulnerability could otherwise infringe contributorily.\(^{139}\)

Alternatively, federal courts (in particular the Federal Circuit) could re-interpret the experimental use exemption to cover security research.\(^{140}\) This would necessitate extending immunity to commercial uses of a patent; current doctrine mandates that a defendant’s activity be non-commercial.\(^{141}\) Security research is often commercial, even if indirectly, and thus even widening the ambit of the common-law exemption might not ameliorate the chilling effects. The current experimental use exemption tracks a bright-line divide between commercial and philosophical activity, rather than weighing

---

\(^{134}\) *In re Bilski*, 545 F.3d 943 (Fed. Cir. 2008), *cert. granted*, 556 U.S. __ (U.S. June 1, 2009) (No. 08-964).


\(^{136}\) 35 U.S.C. § 287(c).

\(^{137}\) 35 U.S.C. § 273(b).


\(^{141}\) See generally *CHISUM ON PATENTS* § 16.03[1] (collecting experimental use cases).
the costs and benefits of the infringing acts. Courts may be reluctant to engage in particularized cost-benefit analysis rather than deferring to Congress, given the history of specialized legislative exceptions in patent law.\footnote{142}

Patent law is a potent weapon for vendors who seek to limit creation and dissemination of vulnerability data. While re-interpreting the experimental use exemption to liability could help researchers, it is more likely that Congressional action to establish a security research exception is necessary to overcome patent law’s negative effects.

C. Trade Secret

There may be such a thing as a free lunch – and free laundry, and soft drinks – at schools that use the Blackboard Transaction System (BTS).\footnote{143} BTS lets students use their identification cards to pay for goods and services on campus. Billy Hoffman, a student at Georgia Institute of Technology, and Virgil Griffith, a student at the University of Alabama at Tuscaloosa, discovered fundamental flaws in Blackboard’s system.\footnote{144} For example, BTS did not encrypt the data involved in processing a purchase; instead, the system depended on physical security to prevent access to the data.\footnote{145} However, Hoffman easily bypassed Georgia Tech’s physical restrictions with a low-tech hack: he removed 4 screws holding a locked machine door in place with a “cheap metal knife.”\footnote{146} Doing so gave him access to the devices that controlled the laundry room in which the box was located – and potentially to the rest of the system as well.\footnote{147} Hoffman could now perform a replay attack against BTS.\footnote{148} By monitoring communication over the BTS network during a transaction, Hoffman could duplicate it, giving him an unlimited supply of free laundry cycles and beverages.

\footnote{142} See, e.g., 35 U.S.C. § 271(3)(1) (granting exemption from infringement liability for uses related to developing and submitting data under federal law regulating drugs).
\footnote{145} John R. Hall, Blackboard Transaction System Cease and Desist FAQ, at http://www.yak.net/mirrors/bb-faq.html.
\footnote{147} Acidus, CampusWide Wide Open, available at http://cryptome.org/campuswide.txt. Hoffman published under the pseudonym “Acidus.”
\footnote{148} Id.
Hoffman contacted Blackboard about his findings, but claimed he was “blown off” by the company. He and Griffith subsequently planned to present their research on vulnerabilities in BTS to the Interz0ne computer conference in Atlanta, Georgia. In addition, Hoffman wrote an article covering BTS weaknesses under a pseudonym for the hacker magazine 2600. He closed it by stating, “Hopefully this article will force Blackboard to change to a more secure system.”

It didn’t. But Blackboard did manage to change Hoffman’s proposed talk. The day before Hoffman and Griffith were to present at Interz0ne, Blackboard obtained a temporary injunction from a Georgia state court blocking them from discussing signal traffic on a BTS network, revealing how information was stored in the BTS system or readers, describing how to create compatible readers, releasing Blackboard emulation code or hardware, or claiming they could provide products or services that legitimately could interact with a Blackboard product. The injunction also required the students to remove any such information from their Web sites. Finally, Blackboard sent a letter to Interz0ne’s conference chair stating that the conference could be liable, including criminally, if it permitted Hoffman and Griffith to present their research, or if Interz0ne failed to remove information about BTS from its materials.

Blackboard relied on several legal theories to bolster its case for the temporary injunction, including violations of the federal Electronic Communications Protection Act and Computer Fraud and Abuse Act, the Georgia Computer Systems Protection Act, and the Lanham Act, but focused principally on the Georgia Trade Secrets Act. Its complaint repeatedly referred to Hoffman’s hacker background, attempting to balance dire descriptions of the threat from his presentation with qualifications about the accuracy of Hoffman’s claims about BTS security. Blackboard claimed that the presentation risked “massive fraud, security breaches, and

---

149 Manjoo, supra note 146.
150 Acidus, supra note 147.
152 Id. at 2.
153 Cease and Desist Letter Sent to Interz0ne II, at http://www.interz0ne.com/events/interz0ne_cease_order.html (Apr. 11, 2003).
154 The complaint refers to the “Consumer Fraud and Abuse Act,” but cites the statutory sections for the Computer Fraud and Abuse Act, 18 U.S.C. § 1030.
156 Id. at ¶ 5-13, 35.
other harms, threatening both the physical and financial security of college students."^{157}

Tellingly, though, Blackboard stated publicly that it wasn’t “really worried about the security of the system,” but instead was “worried about the reputation of the system.”^{158} Hoffman and Griffith focused attention on BTS’s dependence on physical protection for the system’s security – and on how readily those physical methods could be compromised. Blackboard compared the students’ research to breaking into an ATM, eliding the far greater security protections built into those machines.^{159} The company sought both to minimize the findings, calling Hoffman a mere “vandal,” while also justifying the ban on disclosing information about the vulnerability with warnings about the risk Hoffman created.^{160} The restraining order, and a subsequent confidential settlement, blocked Hoffman and Griffith from presenting at InterzOne, though the ensuing publicity drew attention to the BTS flaws.^{161} Trade secret triumphed over toolboxes.

Trade secret law protects information that has economic value because it is not generally known, and that is the subject of efforts to maintain its secrecy.^{162} Examples include customer lists, Google’s algorithm for building search results,^{163} and the formula for Coca-Cola. Software can qualify for trade secret status, even when the compiled object code is sold to the public.^{164} Trade secrets are protected by injunctions preventing their disclosure when a defendant has obtained them through improper means,^{165} holders can also obtain damages. However, trade secret laws expressly permit the use of reverse-engineering to discover

---

^{157} Id. at ¶ 5.
^{161} Virginia Heffernan, Internet Man of Mystery, N.Y. TIMES, Nov. 23, 2008, at MM 38.
^{165} See, e.g., Rivendell Forest Prods. v. Georgia Pacific, 28 F.3d 1042 (10th Cir. 1994).
^{166} See, e.g., Lamb Weston v. McCain Foods, 941 F.2d 970 (9th Cir. 1991).
protected information; only the acquisition of a secret through improper means creates liability. Where researchers obtain information about security flaws in software or hardware through reverse engineering, their subsequent use and disclosure of that information is beyond the reach of trade secret liability. Accordingly, reform to protect researchers may require only that judges scrutinize trade secret claims more searchingly when reverse engineering is involved.

However, there are at least two complications with trade secret law and software research. First, software vendors often include language in the end-user license agreement governing their software that forbids reverse engineering. While such language would be unlikely to create liability for copyright infringement, since limited reverse engineering typically qualifies as fair use, the contractual obligation might be sufficient to make a software user responsible for maintaining the trade secret. Copyright law faces a similar question when end user license agreements prohibit reverse engineering; even where decompilation would be protected by fair use, such action would create liability for breach of contract. Vendors may be able to circumvent trade secret’s safe harbor via contract.

Second, some researchers may switch sides, working first as an employee or consultant, and then moving to perform independent testing. In this case, the software company may have a plausible claim that the researcher’s work is influenced by her knowledge of the firm’s trade secrets. Mike Lynn’s situation with Cisco exemplifies this problem; Lynn began his work on the flaws in Cisco’s routers while covered by a non-disclosure agreement. Once he resigned his position with ISS, Lynn

169 Chicago Lock Co. v. Fanberg, 676 F.2d 400 (9th Cir. 1982); Data General Corp. v. Digital Computer Controls, 357 A.2d 105 (Del. Ch. 1975); see generally RESTATEMENT OF TORTS § 757 (1939).
170 Georgia’s Trade Secrets Act expressly exempts information acquired from reverse engineering from liability. O.G.C.A. § 10-1-761(1) (stating “Reverse engineering of a trade secret not acquired by misappropriation or independent development shall not be considered improper means”).
171 Microsoft expressly forbids reverse engineering in the license agreement for its Windows XP operating system. § 4, Microsoft Windows XP Home Edition (Retail) End-User License Agreement for Microsoft Software, at http://www.microsoft.com/windowsxp/eula/home.mspx (June 1, 2004).
172 See, e.g., Sony v. Connectix, 203 F.3d 596, 605 (9th Cir. 2000).
174 See, e.g., Atari Games Corp. v. Nintendo of Am., 975 F.2d 832, 843 (Fed. Cir. 1992) (holding reverse engineering can constitute fair use).
transitioned to independent research, but Cisco argued, plausibly, that his work was influenced by exposure to Cisco’s proprietary information.\footnote{Complaint, \textit{Cisco Sys. Inc. v. Lynn}, Case No. C05-03043-JL (N.D. Cal. 2005), at 3-7 (copy on file with author).}

The case for legal reform in the trade secret context is more compelling for reverse-engineering than for researchers who switch sides. Allowing software companies to reify a license agreement into a trade secret claim would confer complete control over research into vulnerabilities in software that involved any decompilation or reverse engineering – which most security testing does.\footnote{\textit{See generally Eldad Eilam, Reversing: Secrets of Reverse Engineering} (2005).} Accepting the terms of a software end user license agreement is generally a prerequisite for using a lawful copy of that program. Researchers thus face a cruel choice: either use an unlawful copy in their research, or agree to terms preventing them from engaging in the activity that necessitates installing the program at all. To mitigate this problem, courts interpreting software license agreements, and state legislatures adopting and modifying trade secret statutes, should reinforce the position that reverse engineering does not constitute improper means.\footnote{Trade secrets statutes typically prohibit acquisition of protected information via improper means. Most states follow the formulation of the Uniform Trade Secrets Act, which defines improper means as including “theft, bribery, misrepresentation, breach or inducement of a breach of duty to maintain secrecy, or espionage through electronic or other means.” §1(1), Uniform Trade Secrets Act (1995), \textit{available at} http://nsi.org/Library/Espionage/usta.htm. States such as California expressly protect reverse engineering as an exception to trade secret liability. \textit{See} Cal. Civ. Code § 3426.1 (stating “Reverse engineering or independent derivation alone shall not be considered improper means”).} In addition, security research should be exempted from trade secret liability, unless the plaintiff can prove improper means.\footnote{\textit{Cf.} 17 U.S.C. § 1201(j) (DMCA security research exemption).} An exemption would shift the burden of proof to the software’s owner, and would continue to protect against breaches of non-disclosure agreements. This proposal accords with the goal of trade secret doctrine, which is to enable the protection of proprietary information that confers a competitive advantage. Security researchers seek not to compete with the software they test, but to improve its resilience and robustness. Further, breach of an end user license agreement should give rise, at most, to a claim for breach of contract. Unlike trade secret claims, contract-based ones rarely justify injunctive relief, and plaintiffs must prove actual damages to recover.\footnote{\textit{See, e.g.,} Doug Rendleman, \textit{The Inadequate Remedy at Law Prerequisite for an Injunction}, 33 FLA. L. REV. 346 (1981).}

Trade secret doctrine is created primarily by state law, although theft of trade secrets can create federal criminal liability.\footnote{18 U.S.C. § 1832.} Thus, ensuring uniform protection for security research requires either action by each state...
to protect reverse engineering, or a federal statute enshrining this shield nationally. While some states already safeguard reverse engineering via statute, researchers may nonetheless face restrictions based on trade secret law. Hoffman and Griffith, for example, were subject to a temporary restraining order based in part on Blackboard’s trade secret claim, even though Georgia’s trade secret statute exempts reverse engineering from liability. Accordingly, even states that protect reverse engineering in theory may need stronger liability shields in practice. This suggests that, if states fail to accord adequate protection to software security research as a legitimate activity under their trade secret laws, Congress may need to pass safe harbor legislation that pre-empts conflicting state statutes. Though trade secret law is historically the province of state regulation, the federal Economic Espionage Act of 1996 criminalizes the theft, copying, distribution, sale, or receipt of unlawfully acquired trade secrets. If federal law can be employed to augment trade secret when necessary, it can (and should) be deployed to limit the doctrine when its effects are pernicious.

D. Trademark

Unsurprisingly, trademark has been the legal doctrine least frequently employed by vendors to control software security research. Researchers are normally careful to note that their work does not bear the imprimatur or approval of a software vendor, and any references to a product or service would likely fall under the nominative use exception to trademark liability. Furthermore, hackers are not offering competing products, or indeed appealing to consumers in a way that could cause confusion about source. Blackboard, for example, included a claim under federal trademark law (the Lanham Act) in its complaint against Hoffman and Griffith over their BTS work. As the Chilling Effects project notes, though, their position was “far-fetched at best,” since Hoffman was neither passing off his work as Blackboard’s, nor implying endorsement by the company. Researchers are at pains both to claim credit for their work, and to demonstrate when a vendor has failed to follow their independent advice.

183 See Complaint, supra note 155, and Order, supra note 151.
186 See 15 U.S.C. § 1125(a) (prohibiting use of a mark that is likely to cause confusion about origin, sponsorship, or approval).
187 Complaint, supra note 155.
Trademark law – at least, federal trademark law under the Lanham Act – likely does not require modification to protect security research. However, exemption from liability does involve thoughtful judicial application of doctrine. For example, the nominative use defense (one species of trademark’s “fair use” defense) should immunize researchers who use a software product’s mark to denominate the code to which their findings apply.\(^{189}\) If hackers are careful in how they describe their work, consumer confusion should be minimal if not non-existent. Similarly, federal dilution law provides express protection for nominative and descriptive fair use (including use to criticize or comment on the mark’s owner), for news commentary employing a mark, and for non-commercial use.\(^{190}\) Courts in different circuits, though, apply the nominative use defense differently.\(^{191}\) Some judges may be receptive to vendor suggestions that the use of their marks implies sponsorship or endorsement of the security researcher’s work, on a likelihood of confusion theory\(^{192}\), or based on a false endorsement claim. Researchers would improve their chances of successfully asserting a nominative fair use defense through steps that reduce the potential to confuse computer consumers, such as through disclaimer statements that expressly negate any connection between the hacker and the software vendor.\(^{193}\)

While trademark law has seen limited use against security researchers, the doctrine’s built-in safeguards suggest that legal reform may not be immediately necessary so long as they are conscientiously applied.

\textit{E. Digital Millennium Copyright Act (DMCA)}

Though there are few court cases applying it to security research, the anti-circumvention provisions of the Digital Millennium Copyright Act (DMCA) recur frequently as a threat employed against hackers. In 2002, Hewlett-Packard (HP) fulminated against SnoSoft’s publishing code that permitted an attacker to gain root (administrator) privileges on HP’s Tru64 Unix operating system via a buffer overflow exploit, characterizing it as a

\(^{189}\) \textit{New Kids on the Block}, 971 F.2d at 308.

\(^{190}\) 15 U.S.C. § 1125(c)(3).

\(^{191}\) Compare \textit{Century 21 Real Estate Corp. v. LendingTree}, 425 F.3d 211, 232 (3rd Cir. 2005) (positioning nominative fair use as affirmative defense) \textit{with Playboy Ent's. v. Welles}, 279 F.3d 796, 801 (9th Cir. 2002) (holding that a nominative use defense simply changes the likelihood of confusion methodology).


\(^{193}\) \textit{Cf. Century 21 Real Estate}, 425 F.3d 211 at 231 (stating that “a disclaimer must be considered in determining whether the alleged infringer accurately portrayed the relationship that existed between plaintiff and defendant”).
DMCA violation.\footnote{Declan McCullagh, \emph{Security warning draws DMCA threat}, CNET NEWS, at http://news.cnet.com/2100-1023-947325.html (July 30, 2002). A copy of the letter from HP to SnoSoft is available at http://www.politechbot.com/docs/hp.dmca.threat.073002.html.} Though HP invoked the specter of criminal sanctions for SnoSoft’s post\footnote{Id.} the researchers also faced civil liability\footnote{17 U.S.C. § 1203(c).}. HP had known about the vulnerability since 2001 – a different researcher had posted a separate exploit that achieved root access – but had not issued a patch.\footnote{McCullagh, \emph{supra} note 194; the exploit code is available at http://packetstorm.linuxsecurity.com/0101-exploits/tru-64.su.c.} When HP chief executive Carly Fiorina and the company were inundated with complaints from researchers, reporters, and even HP employees, the firm retreated from its threats against Snosoft.\footnote{Kim Zetter, \emph{HP, Bug-Hunters Declare Truce}, PC WORLD, at http://www.pcworld.com/article/103853/hp_bughunters_declare_truce.html (Aug. 9, 2002).} Nonetheless, the incident prompted a number of attendees at the Black Hat conference that year to consider the possibility of reducing vulnerability sharing with vendors\footnote{Declan McCullagh, \emph{HP backs down on DMCA warning}, ZDNET UK, at http://news.zdnet.co.uk/itmanagement/0,1000000308,2120211,00.htm (Aug. 2, 2002) (quoting HP general manager Martin Fink). \footnote{1 J. Alex Halderman, \emph{Analysis of the MediaMax CD3 Copy-Prevention System}, at http://www.cse.umich.edu/~jhalderm/pub/cd3/ (last updated June 21, 2004). \footnote{2 John Borland, \emph{Student faces suit over key to CD locks}, CNET NEWS, at http://news.cnet.com/2100-1025-5089168.html (Oct. 9, 2003) (citing SunnComm CEO Peter Jacobs).} } and HP stated it would forgo legal threats if researchers “reveal security threats using industry standard security practice.”\footnote{Id.; see generally Brij Khurana, \emph{Halderman GS sees copy-protection flaw in new CDs}, DAILY PRINCETONIAN, at http://www.dailyprincetonian.com/2003/10/09/8785/ (Oct. 9, 2003). \footnote{Id.}}

Similarly, in 2003, Princeton graduate student J. Alex Halderman (now a professor at the University of Michigan) analyzed MediaMax CD3, a copy protection scheme for music CDs from SunnComm.\footnote{J. Alex Halderman, \emph{Analysis of the MediaMax CD3 Copy-Prevention System}, at http://www.cse.umich.edu/~jhalderm/pub/cd3/ (last updated June 21, 2004).} SunnComm claimed that the program offered a “verifiable and commendable level of security,” but Halderman found that computer users could evade its restrictions through the simple expedient of holding down the Shift key (thereby disabling Microsoft Windows’ Autorun feature) when loading the CD.\footnote{John Borland, \emph{Student faces suit over key to CD locks}, CNET NEWS, at http://news.cnet.com/2100-1025-5089168.html (Oct. 9, 2003) (citing SunnComm CEO Peter Jacobs).} Doing so kept the disc from loading a device driver that blocked users from copying music. Users who allowed the CD to install the driver software could also disable it using instructions Halderman provided.\footnote{Id.} Halderman’s work had a significant effect – SunnComm’s stock dropped in value by $10 million in the days after its release.\footnote{John Borland, \emph{Student faces suit over key to CD locks}, CNET NEWS, at http://news.cnet.com/2100-1025-5089168.html (Oct. 9, 2003) (citing SunnComm CEO Peter Jacobs).} SunnComm
responded. The company released a statement indicating that it would sue Halderman for violating the DMCA, and would refer the matter to federal law enforcement for possible criminal proceedings. The company specifically cited his paper (published on Halderman’s Web site) as “‘disseminated in a manner which facilitates infringement’ in violation of the DMCA or other applicable law,” calling it potentially “a felony.” The company later acknowledged the potential chilling effect of such lawsuits on academic security research.

Both HP and SunnComm rescinded their threats after a wave of unfavorable publicity. But the threat of suit under the DMCA has impeded research into software vulnerabilities, even by academics at major universities. Halderman and Ed Felten, his advisor, delayed publishing data about security flaws in the Sony BMG copy protection system for CDs for a month while consulting counsel about managing DMCA risk. While they did so, consumers of Sony music CDs remained vulnerable to hackers who could use a flaw in the anti-copying software to surreptitiously install software on their computers. (Felten was familiar with DMCA threats, having faced one from the Secure Digital Music Initiative when he cracked the group’s music watermarking scheme and sought to present his research.

---

206 Id. (quoting SunnComm press release).
at an academic conference.\textsuperscript{212}) Andrew “Bunnie” Huang had two companies – including a self-publishing firm – back out of publishing his book on hacking Microsoft’s Xbox (including analysis of the system’s security) due to fears of DMCA liability.\textsuperscript{213} University of Michigan graduate student Niels Provos moved his research publications out of the U.S. and tried to block American citizens from accessing them due to fears of running afoul of the DMCA and a similar Michigan state statute.\textsuperscript{214} Even White House Office of Cyber Security chief Richard Clarke cited the “potential chilling effect on vulnerability research” in a speech at the Massachusetts Institute of Technology.\textsuperscript{215} The DMCA has proved a potent weapon, enabling software companies to dissuade or limit security researchers, and its power is perhaps best demonstrated by its ability to compel adherence even with infrequent formal legal proceedings.

The DMCA contains statutory exceptions that could shield security researchers from liability, including protections for reverse engineering\textsuperscript{216}, encryption research\textsuperscript{217}, and security testing\textsuperscript{218}. However, the safe harbors are so narrow that they are effectively useless, as the extant case law demonstrates. Of 141 decided cases involving Section 1201 of the DMCA, only one involved a claim of protection under the security testing safe harbor, and in it the safe harbor was held inapplicable.\textsuperscript{219} One case involved an unsuccessful attempt to rely on the encryption research exemption\textsuperscript{220}, and four cases had unsuccessful claims for the reverse engineering safe harbor\textsuperscript{221}. While these results cover only reported, decided cases (and hence may not be a representative sample), the lack of success in using any of the safe harbors – and the infrequency with which they are raised – suggests that the built-in statutory mechanisms are insufficiently protective of security researchers.

\textsuperscript{212} See Reading Between the Lines: Lessons from the SDMI Challenge, at http://www.cs.princeton.edu/sip/sdmi/ (last updated Aug. 15, 2001).
\textsuperscript{216} 17 U.S.C. § 1201(f).
\textsuperscript{217} 17 U.S.C. § 1201(g).
\textsuperscript{218} 17 U.S.C. § 1201(j).
\textsuperscript{219} Universal City Studios v. Reimerdes, 111 F. Supp. 2d 294 (S.D.N.Y. 2000).
\textsuperscript{220} Reimerdes, 111 F. Supp. 2d 294, aff’d sub nom, Universal Studios v. Corley, 273 F.3d 429 (2d Cir. 2001).
\textsuperscript{221} Lexmark Int’l v. Static Control Components, 387 F.3d 522 (6th Cir. 2004).
This conclusion is bolstered by qualitative analysis of the exemptions. To qualify for the security testing safe harbor, a researcher must obtain authorization from the owner of the computer, system, or network involved in testing (which might be particularly challenging for cloud computing research); must not violate the Computer Fraud and Abuse Act; must use findings solely to improve the security of the computer, system, or network's owner, or share them directly with the network, system, or computer’s developer; and must use or maintain the information derived from testing so as not to facilitate DMCA infringement or violate other applicable laws, such as privacy and security ones.222 Similarly, to assess vulnerabilities in software encryption, such as that employed in the Transport Layer Security protocol used to protect e-commerce223, a researcher must lawfully obtain a copy of the software or hardware, seek authorization from the owner of the rights in that technology, and not violate other laws (including the CFAA).224 Moreover, the statute conditions the exemption on the researcher’s qualifications, the way in which they disseminate their findings, and whether the researcher provides the copyright owner with documentation of findings in timely fashion.225 The statutory safe harbors are not only narrow, but also uncertain – it is not always clear what conduct violates laws such as the CFAA, nor what constitutes timely provision of information to copyright owners.226

The DMCA permits users who are adversely affected by its restrictions in their ability to make non-infringing uses of copyrighted works to petition the Librarian of Congress to exempt certain classes of works from the statute’s ambit.227 However, these exemptions expire after three years, and a user who seeks to continue the exemption must petition for renewal from scratch.228 The last three rounds of exemption rulemaking have not resulted in additional protections that could benefit security researchers; the fourth is currently underway.229 The current round of

rulemaking includes one proposal that would protect security research on technological protection measures that utilize security flaws in personal computers and their software; however, it is not clear whether this proposed exemption will be adopted.\textsuperscript{230}

Reform of the DMCA to provide greater protection for software security research is straightforward. As currently written, the statute canonizes one type, or industry structure, for such research.\textsuperscript{231} It requires that, to fall within the safe harbor for security testing, a researcher perform his or her activities with the authorization of the owner or operator of the system or network.\textsuperscript{232} Thus, the DMCA protects research where the investigator operates under contract with the software vendor or cloud computing system operator, but leaves independent researchers vulnerable. This type of research is carried out by corporate security firms such as Verisign iDefense Labs, Defensive Thinking, or Symantec. Individual researchers or smaller companies may have trouble obtaining authorization due to negotiation costs or because software firms simply may not trust them.

To change the DMCA to more broadly protect the activity of security research, rather than simply one organizational form of it, Congress should either amend the relevant statutory subsection, or simply treat the DMCA under a more generalized shield law. To carry out piecemeal reform, the DMCA should focus on the activity of the security researcher, not on purpose or on authorization. (Ironically, the current statutory exemption implicitly recognizes the key role of the distribution of vulnerability information by conditioning the safe harbor in part on whether the researcher shares the data directly with the computer system owner or software developer.\textsuperscript{233}) In particular, we suggest removing the requirement of obtaining authorization from the owner of the computer, system, or network. Overall, though, we believe that more comprehensive reform, which treats the DMCA as one aspect of IP problems facing security researchers, is preferable.

The DMCA illustrates the potency of intellectual property threats to security research by showing how dissemination of information can be chilled even without filing suit. This section has demonstrated the tools

\textsuperscript{230} See Comment of J. Alex Halderman, In the Matter of Exemption to Prohibition on Circumvention of Copyright Protection Systems for Access Control Technologies, at http://www.copyright.gov/1201/2008/comments/halderman-reid.pdf (Dec. 2, 2008).\textsuperscript{231} See generally Tom Cross, Academic Freedom and the Hacker Ethic, 49 COMM'NS OF ACM 37, 39 (June 2006) (noting that vital security tools such as Nmap, NetCat, and OllyDbg were developed by independent researchers).\textsuperscript{232} 17 U.S.C. § 1201(j)(1).\textsuperscript{233} See 17 U.S.C. § 1201(j)(3) (stating one factor to determine whether one qualifies for the exemption is “whether the information derived from the security testing was... shared directly with the developer of such computer, computer system, or computer network”).
available to software vendors to muzzle researchers. The next section
describes how to mitigate this problem.

IV. CREATING THE AEGIS

To protect researchers’ valuable contributions to software security,
and to ensure that vulnerability information remains in legitimate channels
rather than being sold on the black market, we propose three changes: one
legal, one in social norms, and one market-based.

A. Legal Reform

Our proposed legal reform seeks to shape researchers’ behavior by
conditioning a grant of immunity from IP suits on following rules of
conduct. Providing a safe harbor from liability strongly encourages those at
risk to act in ways that remain within the exemption. For example, the
Digital Millennium Copyright Act provides a safe harbor for online service
providers (OSPs) who, upon notice from a copyright holder, disable access
to or remove allegedly infringing content.234 Most large OSPs, as a matter
of course, remove content upon notification without inquiring into the
merits of the alleged infringement claim or into their potential risk
exposure.235 While this approach has generated controversy, it is more
certain and less expensive to stay within the safe harbor.236

Similarly, if software security researchers can avail themselves of
immunity from IP infringement claims by acting in certain ways, it is likely
they will conform their behavior to those requirements. Many researchers
have limited resources and legal acumen, making them risk-averse
regarding litigation and more likely to track the exemption’s mandates.
Legal threats unquestionably influence researchers’ actions, as they learn
from prior controversies. For example, a trio of security experts
demonstrated how to hack smartcard-based electronic parking meters at the

235 See, e.g., Michael Piatek, Tadayoshi Kohno, Arvind Krishnamurthy, Challenges and
Directions for Monitoring P2P File Sharing Networks, available at
http://dmca.cs.washington.edu/dmca_hotsec08.pdf; Jennifer M. Urban & Laura Quilter,
Efficient Process or “Chilling Effects”? Takedown Notices Under Section 512 of the
236 See, e.g., Peter Lattman, Law Professor Wendy Seltzer Takes On the NFL, WSJ LAW
Black Hat security conference in August 2009. The researchers deliberately chose not to contact either the vendor or the owner of the meters they hacked, and asked reporters not to do so, citing the injunction entered against MIT researchers who showed how to obtain free rides on Boston’s subway system in 2008. Hackers learn the relevant law quickly.

Crafting the requirements for immunity is critical in two respects. First, conditions for the exemption will strongly influence how hackers behave – what they do while testing code, and what they do with the resulting information. Second, the safe harbor would deprive software vendors of potent legal tools and remedies. If it immunizes undesirable behavior, it will inflict harm on software firms, and on society generally.

Proper behavior by researchers is admittedly a contested issue within the software security community. Proposals vary from advocating full disclosure (publishing vulnerability details immediately upon discovery) to revealing bug data only to vendors. Heated debate is common, and occasionally bleeds into active protests such as Web site hacking. The normative position we adopt, responsible disclosure, represents a middle ground that has won considerable support.

Responsible disclosure requires researchers to notify vendors first on
discovering vulnerabilities, but preserves the possibility of public dissemination to prod software firms to remediate flaws.\textsuperscript{245} We believe that the potential for full public disclosure under our proposal motivates vendors to issue patches and to press customers to install those fixes, while the prohibition on selling vulnerability data to third parties reduces the number of potential attackers while the vulnerability is being remedied.

The hacker’s aegis would set a default presumption that security researchers are acting lawfully, and would require plaintiffs (such as aggrieved software companies) to demonstrate that accused activity falls outside its protections. In return for immunity from civil intellectual property claims, software security researchers would be required to adhere to five rules: tell the vendor first, don’t sell the bug, test on your own system, don’t weaponize, and create a trail.

1. Tell the Vendor First

The first conduct-based rule would require a researcher who discovers a security vulnerability to report it to the vendor of the affected software before publishing any information about the flaw. Failure to report before disclosing information about the bug would bar the researcher from availing herself of the safe harbor, but should not function as evidence of infringement or any other legal liability. Hackers should tell the software producer – the party best positioned to remedy the flaw – first.

The reporting requirement includes two additional components. The first would mandate that researchers use the method of contact described on the home page of a vendor’s Web site; if the vendor failed to include a contact mechanism on their site, the researcher could simply notify customer support or the firm’s general counsel.\textsuperscript{246} Bugs submitted to tech support or to a company’s lawyer are less likely to receive attention, which is why this approach seeks to press vendors to establish a means of gathering vulnerability data. Companies would likely opt to create such mechanisms because failure to do so would let researchers potentially avail themselves of the safe harbor merely by contacting technical support. Support representatives may not be trained to deal properly with security reports, and hence firms lose legal recourse without much corresponding benefit. Researchers would use the designated contact path because immunity depends upon it. Moreover, hackers want bugs to be taken

\begin{footnotesize}
\end{footnotesize}
The second component would mandate a post-reporting delay before the researcher could share vulnerability data publicly. This would provide time for the vendor to assess the new information, contact the researcher, and begin work upon a patch if necessary. Public disclosure before this period would negate the safe harbor. After the initial waiting period passes, the researcher would be free to share the vulnerability data. Researchers interested in blackmail are unlikely to follow the safe harbor’s rules in any case, and public disclosure is an important incentive to compel vendors to take bugs seriously. Trusted intermediaries, such as the Computer Emergency Response Team, disclose vulnerabilities publicly 45 days after they are initially reported, regardless of the status of patches from vendors.

2. Do Not Sell the Bug

The second behavior rule would ban sales of vulnerability data to third parties. Researchers would forfeit the safe harbor if there were sufficient credible evidence that they offered data about the vulnerability to any third party for compensation. (This would permit transactions with the vendor.) By “sufficient credible evidence,” we mean concrete facts, and not conclusory allegations or statements on information and belief by vendors. To defeat immunity based on this factor, a software provider would have to adduce and support facts sufficient to survive a motion to dismiss. While

250 See generally VIEGA, supra note 36, at 153-62.
252 The weight of evidence required to overcome a motion to dismiss is unclear after the Supreme Court’s decisions in Ashcroft v. Iqbal, 556 U.S. __ (2009), and Bell Atlantic Corp.
this might seem to impair programs such as Tipping Point’s Zero Day Initiative, where independent firms pay researchers for reporting security bugs to them, this concern is readily mitigated via private law.\textsuperscript{253} If participating in third-party “bounty hunter” programs makes a researcher ineligible for the public law safe harbor, researchers will either demand that their agreements with these programs indemnify them against legal risks from IP suits, thus effectively reproducing the immunity, or the price that they demand will increase to compensate them for the risk.\textsuperscript{254} The goal of this factor is to discourage researchers from engaging in strategic behavior by marketing their wares to the underground before, or concurrently with, offering them to vendors. While proof of efforts to sell vulnerability data may be difficult to obtain, we believe that the existence of this factor will help to discourage gray hat hackers – those willing to act as black hats or white hats depending on circumstances - from selling their findings.

3. Test On Your Own System

Third, researchers must test for vulnerabilities on their own systems unless they cannot reasonably do so, as with cloud computing. A researcher would lose the safe harbor on a showing of sufficient credible evidence that she tested, or employed the flaw to compromise security on, a system not under her lawful control, unless there is no reasonable alternative to doing so. This factor is intended to push researchers to investigate vulnerabilities on test systems rather than on production code that is in use by others.\textsuperscript{255} Analysis of whether there is a reasonable alternative should be searching; hackers should not interfere with others’ systems lightly. For example, if a software company employs only open source code\textsuperscript{256}, or makes trial versions of its products available\textsuperscript{257}, a researcher would have ready access to the relevant code. Hence, the researcher would have to test on her own system to stay within the safe harbor.

However, the rise of Web services and cloud computing complicates this analysis. In cases where the software is available only from third party systems, such as with Amazon’s Elastic Cloud Compute (EC2)\textsuperscript{258},

\begin{footnotesize}v. Twombly, 550 U.S. 544 (2007). We do not believe that these shifts in the standard will have a material effect on the safe harbor.\textsuperscript{253} See Zero Day Initiative, at http://www.zerodayinitiative.com/.
\textsuperscript{258} Amazon Elastic Cloud Compute (Amazon EC2), at http://aws.amazon.com/ec2/.
\end{footnotesize}
researchers should be permitted to test vulnerabilities hosted on those systems as long as the researcher does not use the system to do more than verify the existence and extent of the vulnerability, and does not cause more than temporary, minor disruption to the operation of the service. While this qualification makes this third factor closer to a standard than a rule for cloud computing, it is necessary to enable research on “software as a service” applications and to protect researchers against claims that their testing harmed or impeded the service.\(^{259}\)

4. Do Not Weaponize

Fourth, the researcher must not publish, without the vendor’s authorization, exploit or proof of concept code that enables attacks against the vulnerability. Researchers who “weaponize” vulnerabilities increase the number of potential attackers. Descriptions of security flaws may allow sophisticated black hats to create programs that leverage bugs, but tools that automatically attack weaknesses allow any user who downloads them to wreak havoc.\(^{260}\) While exploit code may alert system administrators to methods of protecting against vulnerabilities, the risk of attacks from “script kiddies” outweights the gain in safety.\(^{261}\) If a vendor can show sufficient credible evidence that the researcher has published weaponized code, the researcher would forfeit immunity.

5. Create A Trail

Finally, the researcher must create an audit trail for the vulnerability by reporting it to the clearinghouse described below in Section D. To qualify for the safe harbor, the discoverer of a bug must upload a detailed description of the flaw, information on how to reproduce it, any known exploits or proof of concept code, her identifying information, and a copy of any correspondence (such as e-mail) with the vendor. This provides proof that a researcher found and elucidated a bug, and that she provided the vendor with sufficient information to investigate it. Moreover, mandating that researchers supply contact information enables vendors to


communicate with them, and also deters strategic behavior, such as claiming credit for others’ discoveries.

The safe harbor for researchers who follow these five rules should be structured as an exemption from liability and not merely as a defense. The difference between an exemption and a defense can be seen by comparing copyright’s fair use defense to the exemption from liability for third-party speech under Section 230 of the Communications Decency Act. Fair use’s burden falls upon the defendant; the plaintiff need not prove that the alleged infringement was unfair.\textsuperscript{262} By contrast, a plaintiff alleging, for example, that a Web site is responsible for comments posted by third party users must show that the site falls outside the exemption from liability created by Section 230.\textsuperscript{263} An exemption is preferable for three reasons. First, the safe harbor is most relevant, and important, in the initial stages of a dispute between a researcher and a vendor. Few IP lawsuits against hackers go to trial. Mike Lynn settled with Cisco, as Griffith and Hoffman did with Blackboard.\textsuperscript{264} The goal of a suit is not to win in court, but to prevent publication or use of vulnerability data by the researcher, to gain time for the vendor to respond to the bug (both in creating patches and in managing public perception of its product), and to force the researcher to agree to terms favorable to the company in settling the dispute. The first stage of the fight is typically a request for a temporary restraining order or preliminary injunction that constrains the hacker’s options, and conduct.\textsuperscript{265} Often, requests for such an order are handled ex parte, as with Mike Lynn and Billy Hoffman. Thus, whether the researcher has an applicable defense is irrelevant, unless it is sufficiently strong to affect the judge’s perception of whether the plaintiff will succeed on the merits of its claim.\textsuperscript{266} (This is unlikely, given that the vendor controls what evidence and arguments are adduced in the initial hearing.) Once the vendor has the order, the researcher must move to change the status quo.

\textsuperscript{262} See, e.g., \textit{Campbell v. Acuff-Rose Music}, 510 U.S. 569, 574 (1994) (stating it was “uncontested that [defendant’s] song would be an infringement of [plaintiff’s] rights in [copyrighted work] under the Copyright Act… but for a finding of fair use”).

\textsuperscript{263} See, e.g., \textit{Doe v. MySpace}, 528 F.3d 413 (5th Cir. 2008) (finding no liability); \textit{but see Fair Housing Council of San Fernando Valley v. Roommates.com}, 521 F.3d 1157 (9th Cir. 2008) (finding liability where site’s design made it partially responsible for content).


\textsuperscript{266} See, e.g., Complaint, supra note 155.

Second, structuring the safe harbor as a defense would likely not reduce the costs researchers face sufficiently. Hackers would have to muster evidence and legal arguments to support their eligibility for the exemption.\textsuperscript{267} Even hiring counsel can be an expensive proposition for individual researchers.

Finally, altering the allocation of the burden of proof is important to this Article’s larger goals. Given its social utility, security research should be presumptively legitimate, not unlawful. It should be incumbent upon an aggrieved vendor to overcome this presumption of legality, not for researchers to have to validate their activities.

Thus, the legal reform component of the Hacker’s Aegis would establish exemption from IP-based liability for researchers who follow five rules: alert the vendor first; do not offer to sell data to any third party; do not test on systems not under their control unless there is no reasonable alternative; do not weaponize code; and create an audit trail. To break through this exemption, vendors would have to show facts that demonstrate that the researcher has violated one of these five requirements.

\textbf{B. Form for Substance}

Legal reform to shield security researchers from the threat of IP litigation by vendors could follow one of two paths. The first adapts existing doctrinal defenses to cover researchers’ activities. The second initiates new legislation to exempt research from liability. Each approach confers benefits, and faces challenges. Overall, we believe research-specific legislation is the preferable path.

A statute conferring immunity upon a designated class of actors – security researchers – has several advantages.\textsuperscript{268} First, legislation could tailor the exemption to reward helpful behavior while leaving malefactors at risk of liability. Second, actor-specific rules could eliminate strategic behavior by vendors and others alleging infringement. If protections for researchers varied by IP doctrine, aggrieved software companies would seek to frame their claims under the theory with the narrowest protection.\textsuperscript{269} Third, it likely operates more rapidly than an accretion of doctrine-specific exemptions developed from individual cases.

\textsuperscript{267}\textit{Cf. Campbell}, 510 U.S. at 590 (stating in copyright case that “Since fair use is an affirmative defense, its proponent would have difficulty carrying the burden of demonstrating fair use without favorable evidence”).


\textsuperscript{269}See, e.g., Complaint, \textit{supra} note 155 (Blackboard); Complaint, \textit{supra} note 177 (Cisco) (adducing multiple IP theories for liability).
However, employing a statute specific to vulnerability research also has weaknesses. Most importantly, public choice problems make it likely that such a law would be underprotective.\textsuperscript{270} Owners of intellectual property in software are concentrated and relatively powerful.\textsuperscript{271} They have strong financial incentives to maximize IP protection for their code, and would likely oppose, or seek to weaken, a research exemption. Other powerful interests – for example, those whose content is protected by code, such as the movie industry – would likely side with vendors.\textsuperscript{272} By contrast, independent researchers tend to be individuals or small firms with less political sophistication and fewer resources. The situation is analogous to the ecosystem of interests involved in crafting copyright legislation described by Jessica Litman: content owners are politically sophisticated, resourceful, and have significant stakes in the outcome, while users and public interests are weaker, dispersed, and lack an effective representative.\textsuperscript{273} A law protecting vulnerability research might appear (if at all) like the exemptions for security research, encryption research, and reverse engineering under the DMCA, which are so narrow that they have only been advanced in 5 cases since 1998, and never successfully.\textsuperscript{274} Moreover, weak protection might be worse for researchers than none at all, as it would be difficult to argue that their actions should be protected if they fell outside the scope of legislatively-determined permissible behavior. The other option – employing doctrine-specific exceptions to protect security research – also confers benefits. It has the standard virtues of common law adjudication: judges can adapt protections to fit different circumstances, and variation among courts permits helpful experimentation in the scope of protection.\textsuperscript{275} Exceptions such as fair use in copyright law


\textsuperscript{272} See, e.g., Motion Picture Association of America, Content Protection, at http://www.mpaa.org/piracy.asp.

\textsuperscript{273} Jessica Litman, Digital Copyright (2001).

\textsuperscript{274} The reverse engineering exemption, 1201(f), was held inapplicable in three cases: Sony Computer Entm’t Am. v. Divineo, Inc., 457 F.Supp.2d 957 (N.D. Cal. 2006), Davidson & Assocs. v. Jung, 422 F.3d 630 (8th Cir. 2005), and Universal City Studios v. Reimerdes, 82 F.Supp.2d 211 (S.D.N.Y. 2000). It was treated favorably in dicta in one case, Lexmark Intern. v. Static Control Components, 387 F.3d 522 (6th Cir. 2004). The encryption research exemption was held inapplicable in two cases: Reimerdes, and Universal City Studios v. Corley, 273 F.3d 429 (2d Cir. 2001) (the Reimerdes appeal).

\textsuperscript{275} See generally Guido Calabresi, A Common Law for the Age of Statutes (1982); Roscoe Pound, The Spirit of the Common Law (1921).
have a rich precedential history that could guide judges in tailoring protection appropriately.\footnote{See generally Pierre Leval, Toward A Fair Use Standard, 103 HARY. L. REV. 1105 (1990).}

Sector-specific rules also suffer drawbacks, though. First, protection for software research would need to measure eligibility in a purposive fashion rather than based on formal characteristics or descriptions of activity. Black hat and white hat hackers perform the same type of research; until they disseminate their findings, only their goals differ.\footnote{See generally Bryan Smith, William Yurcik, & David Doss, Ethical Hacking: The Security Justification, EEI2I (2001), available at ftp://ftp.eng.auburn.edu/pub/avk0002/BE%20Data/PAPERS/Ethical%20hacking/justification.pdf.} If it is not clear what a researcher plans to do with vulnerability information, a court may be risk-averse and block dissemination.\footnote{Daniel Kahneman et al., Anomalies: The Endowment Effect, Loss Aversion, and Status Quo Bias, 5 J. ECON. PERSPECTIVES 193, 199-203 (1991).} Second, judges – especially those unfamiliar with computer technology – may be skeptical of the value of independent security research (rather than that conducted by a vendor), and will likely suffer from information asymmetry, particularly when confronted with ex parte requests for temporary injunctive relief. A vendor’s portrayal of the risks from a rogue teenage hacker may swamp calculations of the greater public interest in salience.\footnote{See generally Derek E. Bambauer, Shopping Badly: Cognitive Biases, Communications, and the Fallacy of the Marketplace of Ideas, 77 COLO. L. REV. 649, 692-94 (2006); RICHARD H. THALER & CASS R. SUNSTEIN, NUDGE 24-26, 33-34 (2009 ed.).} Finally, vendors would likely engage in strategic behavior. Many complaints against researchers allege multiple violations from different IP doctrines: Mike Lynn faced claims for copyright infringement and misappropriation of trade secrets; Billy Hoffman and Virgil Griffith were accused of trade secret and trademark violations. If protections for security research varied by doctrine, software firms would simply recast their claims against hackers in the relevant theory with the least protection. A shield with holes is nearly as ineffective as no shield at all.

Overall, a statute specifically protecting software security research comports best with this Article’s goals. While a shield law faces political challenges, we prefer it as a more effective solution.
C. Changing the Hacker Image

It’s so easy to impress judges with heavily connoted words like “virus”, “pirate”, “terrorist”, “hacker”, and it’s so difficult on the other hand to explain the scientific method and the deep curiosity that makes us analyze how software works and find their flaws.\textsuperscript{280}

The term “hacker” is a loaded one. It connotes not only technical skill, but also a disregard for rules and, at times, a malicious enjoyment in finding flaws and wreaking havoc.\textsuperscript{281} Researchers like being seen as outlaws rather than nerds. However, these normative associations have real drawbacks along with psychological benefits.\textsuperscript{282} Judges, journalists, and the general public may perceive a “hacker” as inherently threatening, and react accordingly.\textsuperscript{283} We propose that the research community attempt to mitigate this semantic problem.

Hackers suffer an inherent disadvantage in how others are likely to perceive their work, in three ways. First, an aggrieved software vendor possesses first mover advantage: the firm is generally the party that frames the dispute for a court by filing a complaint or a request for temporary injunctive relief, which often occurs ex parte. Second, plaintiffs may mix allegations of criminal liability with IP claims, portraying the hackers as vandals or thieves.\textsuperscript{284} Finally, intellectual property itself is a normatively loaded term that confers an advantage on software providers: researchers are seen as meddling, interfering with, or damaging someone else’s valuable possession.\textsuperscript{285}

While the first two of these challenges are hard to remedy, researchers can shift the rhetorical debate surrounding their use of others’ IP. Even real property doctrine permits unauthorized use when there is a compelling reason,\textsuperscript{286} such as necessity or emergency; countervailing


\textsuperscript{282} See generally GEORGE LAKOFF & MARK JOHNSON, \textit{Metaphors We Live By} (1980).

\textsuperscript{283} See generally DOUGLAS THOMAS, \textit{Hacker Culture} 5 (2003) (stating “This is the common perception of today’s hacker – a wily computer criminal”).

\textsuperscript{284} See, e.g., supra notes 160, 194, 205.

\textsuperscript{285} See generally FISHER, supra note 271, at 20-23.


\textsuperscript{287} See, e.g., \textit{Vincent v. Lake Erie Transportation Co.}, 124 N.W. 221 (Minn. 1910).
social need, such as access to social services; or even customary practice. And intellectual property is most commonly viewed, in the U.S., as a utilitarian bargain between creators and the public, where the state confers limited monopoly rights to attain social benefits such as information production and distribution. Those rights are circumscribed by exceptions, such as the nominative use doctrine or privileges for socially beneficial actors, such as public libraries, that safeguard valuable though non-permissive uses. Researchers should therefore emphasize not the potential harm to software companies from vulnerabilities, but the benefits to consumers from fixing those bugs (or, put another way, the risks to consumers if a vendor fails to do so). Property law is most concerned to limit owners’ rights when there are significant externalities involved. Software is a canonical example. To use a real property analogy: researchers should emphasize the interests of the tenants (users) to counteract the claims of the owners. By portraying their work as aligned with users’ needs, hackers can mitigate the power of the property analogy employed by software vendors.

Shifting perceptions is difficult. If security researchers want to alter their public image, two strategies are possible. First, they could seek to reclaim the term “hacker.” Initially, a “hacker” was someone who probed or modified hardware or software to see how it worked, and perhaps to change its functioning. However, the term increasingly connotes one whose activities are illegal, and perhaps malicious (though discerning researchers refer to the latter as “crackers”). To return hacker to its lexical roots requires three things, in ascending order of difficulty: finding a new term for those who invade systems or crack software with ill intent; convincing the security research community to adopt the new term and employ it with some consistency; and convincing others (particularly the media) to follow the new usage pattern. Alternative terms, such as “cracker,” are readily available. However, the research community seems unwilling to shift usage – in part because some like the outlaw image that “hacker” currently

289 See, e.g., State v. Shack, 277 A.2d 369 (N.J. 1971); State v. Schmid, 423
292 See, e.g., New Kids On the Block v. News Am. Publ’g, 971 F.2d 302, 306-08 (9th Cir. 1992) (discussing nominative use).
295 See, e.g., RAYMOND, supra note 88, at 1-18.
The Hacker’s Aegis

provides. Even if vulnerability researchers take up new terms, it is not clear that such a change will spread to the wider public, particularly since “hacker” is evocative.

The second option is for legitimate researchers to abandon “hacker” to the black hats. One way to do this would be to embed the term “hacker” in federal criminal law, such as by defining it in the Computer Fraud and Abuse Act.\textsuperscript{297} This would formalize the equivalence between hacker and black hats. Legitimate researchers would employ a new term to describe those who conform to laws and norms governing software security, and would insist (to the degree they are able) that others refer to them by that moniker. While the adoption challenges described above remain, a new term will lack the cognitive inertia that “hacker” possesses, which may mitigate these issues. Moreover, researchers can seek to shift the analogy that dominates vulnerability analysis. If they portray their role as similar to whistleblowers\textsuperscript{298}, independent testing companies such as Consumers Union\textsuperscript{299}, or watchdogs such as the Center for Science in the Public Interest\textsuperscript{300}, their work is more likely to be treated as legitimate. The term needs to be pithy, appealing, and different from “hacker”; we offer “bug hunter,” “cyber-watchdog,” and “security researcher” as possibilities, but hope others will introduce more catchy options. This shift in perception – drawing a distinct rhetorical line between white hat and black hat researchers – will benefit researchers. As their work appears less threatening, legal measures to restrict its production and dissemination will appear less necessary.

\textbf{D. Freeing Markets}

The market for information about software vulnerabilities is not a well-working one due to high transaction costs, information asymmetry, the risks of strategic behavior, and time pressure. We propose two changes that will ameliorate these issues.

Researchers and vendors are both reluctant participants in transactions involving vulnerability data, in part because of transaction costs. A hacker who discovers a flaw must determine which party to attempt to do business with (for example, deciding between the vendor, the vendor’s

\textsuperscript{297} We thank Scott Velez for this insight.
\textsuperscript{299} See, e.g., Consumers Union, Cutting Surgical Infections, at http://www.stophospitalinfections.org/infection_prevention/ (documenting hospitals’ rates of infections during surgery).
\textsuperscript{300} See, e.g., CSPI, Bayer Ads Misleading Men About Prostate Cancer, Says CSPI, at http://www.cspinet.org/new/200906181.html.
customers, or security consulting firms) and then who to contact (for example, a development team, legal counsel, or management). For their part, vendors must separate legitimate inquiries from attempts at fraud or blackmail. They also must assess whether a vulnerability is a known problem, whether the researcher or others has working code to exploit it, and whether the seller is sufficiently trustworthy to enter into a transaction.

Even once a willing seller locates, and communicates with, a willing buyer, the parties will have difficulty coming to terms due to information asymmetry. There is no price list, or set of criteria, to determine what a bug is worth. Unsurprisingly, vendors tend to value vulnerability information less than researchers do. The market for security flaws is an illiquid one, since transactions are sporadic and often secret. The lack of reference data for pricing means that vendors and researchers may fail to strike deals that would benefit both parties, since they may err (or simply differ) in assessing the data’s value. Moreover, reducing information asymmetry through sharing is challenging. For sellers, presenting their wares to software companies is chancy, not merely because doing so may put them at legal risk, but because sharing findings could destroy their value. This results from Arrow’s paradox: it is hard to demonstrate the value of a security flaw without revealing information sufficient to permit the vendor to remedy it.

Finally, the risk of strategic behavior weakens prospects for a successful bargain. Researchers worry about misappropriation. Arrow’s paradox presents a hard choice: disclose too little, and vendors may not believe the problem is real; disclose too much, and a software company may take the information without compensation. Vendors, in turn, have trouble guaranteeing that a researcher who shares data with them is not also sharing it on the black market. Paying hackers for bugs may also tempt researchers to target a company’s software. Fears about the other party’s behavior effectively decreases the value of a deal for both sides (due to increased risk that the bargain will unravel) and may lead to additional costs from preventive measures.

---

305 Arrow, supra note 301.
Lastly, time pressures shrink the window for vendors and researchers to reach an agreement. Hackers correctly perceive that their vulnerability information has a limited viable lifetime. Other researchers may discover the same weakness and either launch an attack or offer a competing bargain. The vendor may change its code, deliberately or inadvertently remedying the problem. If the hacker has revealed any information to the software company, the firm may be able to reverse-engineer the vulnerability from that limited data, making the research worthless. Firms, too, face time constraints. If one person has found a weakness in their code, others are likely to do so as well. A researcher frustrated by the pace of negotiations may turn to the black market. Finally, vendors require lead time to write, compile, test, and distribute patches. To address a bug, a vendor needs as much time as possible to generate a fix and to get customers to install it.

There are significant structural barriers to market transactions between information suppliers (researchers) and consumers (vendors). To reduce these impediments, we propose that a trusted third party act as a voluntary coordinator or clearinghouse for vulnerability deals. We envision this entity playing three roles. First, it would archive and validate vulnerability data for researchers. This would allow a hacker to claim credit for discovering a bug, and to store diagnostic data and any exploit code. Registries with trusted third parties have been used successfully to overcome the challenges of Arrow’s paradox in other contexts, such as unsolicited manuscripts for television shows and movies. Second, it would maintain contact information for vendors and researchers. The third party might offer anonymous referrals, where the identity of a hacker or vendor is known to the coordinator but not to the other party. This could encourage researchers who are risk-averse to share discoveries through legitimate channels. Finally, the trusted third party could play a reputational role. It could make available data about previous reports and transactions, perhaps in summary form, to help vendors and researchers establish trustworthiness.

A more interventionist role might have the intermediary act as a third-party beneficiary to a non-disclosure agreement between a seller and buyer, allowing the coordinator to enforce bargains and to police

---


adherence through both legal means and through reputational sanctions
(such as disclosing violations publicly).

Existing security organizations, such as the Computer Emergency
Response Team Coordination Center (CERT/CC) at Carnegie Mellon
University’s Software Engineering Institute, are well-positioned to act as
clearinghouses.309 CERT has a strong reputation in the computer security
field and acts as a key channel of information distribution about
vulnerabilities. CERT partners with both private and public-sector entities
in the field, and its Knowledgebase already contains data on thousands of
reported vulnerabilities.310 Other entities, such as the Internet Storm
Center311 and perhaps even government-sponsored organizations such as the
National Vulnerability Database312 might also act as intermediaries.

A benefit of this approach is that both vendors and researchers
would likely utilize a trusted third party system voluntarily, because it
reduces their costs and risks. For example, the combination of the registry
and the reputational metadata would help vendors decide which researchers
are worth the cost of entering into a non-disclosure agreement to further
inspect a claimed vulnerability. This increases the likelihood that this part
of the hacker’s Aegis would be adopted, and used.

Use of a voluntary coordinating intermediary would help reduce
costs that researchers and vendors alike face in exchanging information
about vulnerabilities, making legitimate transactions easier and more likely.

E. Challenges

There are at least two potential challenges that our proposed reforms
might confront. The first is that our legal reforms might be underinclusive.
The second is the risk of strategic behavior based on the legal safe harbor
we propose. In this section, we address each issue.

Our legal proposal contemplates a shield from civil liability under
intellectual property claims or causes of action. However, it does not
encompass other theories of liability – in particular, tort claims, civil claims
under the Computer Fraud and Abuse Act, and criminal prosecution. We
have four reasons for crafting the safe harbor to leave these legal tools
available. First, in analyzing legal threats against security researchers, IP
claims predominate. We view establishing a shield against them as a first
step, but not necessarily a final one. If software companies shift to using
alternative theories such as tort and CFAA claims, with similar chilling

309 CERT Coordination Center (CERT/CC), at http://www.cert.org/certcc.html.
310 CERT Knowledgebase, at http://www.cert.org/kb/.
311 About the Internet Storm Center, at http://isc.sans.org/about.html.
effects, we would advocate expanding the safe harbor to exclude such theories.

Second, tort claims are typically weaker than IP ones. Common law doctrines such as trespass to chattels have largely been displaced by software and Internet-specific statutes. Successful suits under trespass to chattels are relatively rare, and the theory has been questioned by leading courts such as the California Supreme Court. Other tort claims, such as tortious interference with business expectations or prospective economic advantage, typically recapitulate IP claims in slightly different form. In addition, proof of actual damage is required for tort claims such as interference with prospective advantage, and injunctive relief is atypical. These factors reduce the risk from tort theories to software researchers.

Third, the Computer Fraud and Abuse Act contains a built-in limitation on civil liability that offers protection to security researchers. To maintain a cause of action under the CFAA, a plaintiff must demonstrate economic damages of at least $5000 in a one-year period, impairment of a person’s medical treatment, physical injury to a person, threat to public health or safety, damage to a U.S. government computer, or damage affecting 10 or more computers. A researcher testing software on a computer under her control is unlikely to contravene any of these limits, save one. The aspect of CFAA civil liability that may give hackers pause is the ban on damage that affects ten or more computers. This provision, adopted to deal with threats from viruses and worms, may be problematic when researchers probe cloud computing. Services such as Gmail run on multiple servers, and their storage units (such as storage array networks (SANs) or network-attached storage (NAS)) may comprise computers in their own right. If civil CFAA liability interferes with testing of cloud computing security, we propose two fixes. First, the legal safe harbor could modify the CFAA to condition liability on accessing a larger number of computers – perhaps 100. This would maintain liability for virus creators and distributors, but would reduce the threat to researchers. Second – and likely more promising – cloud computing services should be treated as a single computer for purposes of civil CFAA claims. The appeal of cloud

315 See, e.g., Robi v. Five Platters, 838 F.2d 318, 323-24 (9th Cir. 1988) (finding claim preclusion of trademark claim based on prior decision regarding interference with contractual relations).
316 See, e.g., Durassys, Inc. v. Leyba, 992 F.2d 1465 (7th Cir. 1993).
317 18 U.S.C. § 1030(g).
computing is that it appears as a single service or computer to users. Moreover, the number of computers accessed during a cloud computing session is under the control of the service provider. This creates a risk of strategic behavior: providers could ensure that any transaction affected ten or more computers, creating the possibility that any claimed damage would meet the CFAA minimum. Treating cloud computing as a single computer would effectively remove part of the CFAA as a resource for providers, which is why we believe such a change should wait for more evidence of a problem.

Finally, our proposed changes leave security researchers vulnerable to criminal charges, under the Computer Fraud and Abuse Act and other statutes. This is deliberate. We believe that criminal sanctions are necessary to deter strategic behavior by black hat hackers, who may try to fit their activities within the contours of the safe harbor. Criminal law acts to reinforce any gaps within the liability shield. If a hacker’s activity, though protected from civil claims, causes sufficient harm, a software company is likely to be able to convince a prosecutor to file charges. The risk of overdeterrence from criminal law remains, but it is likely no worse than under current circumstances. In addition, prosecutors are more likely to arrive at an objective assessment of whether a hacker’s behavior is beneficial or malicious than the vendor whose software has been targeted. Thus, retaining criminal liability for hacking serves a helpful deterrence function, and should not create additional chilling effects for security researchers.

This Article’s proposals are a first step towards mitigating IP law’s unhelpful channeling effects for software security research. Its proposed legal reforms do not alter tort theories and civil CFAA claims regarding hacking because their built-in doctrinal safeguards should be sufficient. Should this prove incorrect, we propose revisiting the scope of the legal shield to address such risks. And, our proposal retains criminal liability as a necessary deterrent to counter strategic behavior by malicious hackers.

321 But see, e.g., Mark Rasch, German hacker-tool law snare... no one, THE REGISTER, at http://www.theregister.co.uk/2009/06/07/germany_hacker_tool_law/ (June 7, 2009) (arguing German cybercrime law resulted in security companies leaving that country, despite lack of prosecutions under it).
V. Bugs and Other Improvements

The problems that intellectual property law causes for software security research challenge standard scholarship on how intellectual property law should regulate the treatment of improvements on IP-protected works. Patent and copyright law confer strong control over improvements upon IP owners. We argue this is undesirable for software vulnerabilities and other contexts where infringement is difficult to detect, lucrative, and socially harmful.

Vulnerability data is information about how to improve a work of intellectual property (the buggy software). American IP law offers those who hold information on how to improve a protected work three paths: bargain with the owner of the underlying work, seek an improvement patent, or infringe. Copyright bars those who create an unauthorized derivative work from obtaining rights in it. Patent law permits improvers to protect advances, but confers no affirmative right to use the advance without permission from the underlying patent holder. Current scholars tend to agree. Edmund Kitch argues for control over improvements in patent law to avoid wasteful duplication of effort. Wendy Gordon, William Landes, and Richard Posner seek consolidation to reduce transaction costs. Michael Heller and Rebecca Eisenberg fear creation of an anti-commons driven by multiple rightsholders; Mark Lemley and Carl Shapiro worry about patent stacking’s effects on royalty costs and holdup

323 See generally Bambauer, supra note 94, at 370-72.
324 Cantrell v. Wallick, 117 U.S. 689, 695 (1886) (stating “Two patents may be valid when the second is an improvement on the first, in which event, if the second includes the first, neither of the two patentees can lawfully use the invention of the other without the other’s consent”).
325 One who creates an unauthorized derivative work infringes the copyright in the underlying work, and cannot obtain protection for her original expression in the derivative, unless it is clearly separable. 17 U.S.C. § 103(a). See, e.g., Pickett v. Prince, 52 F. Supp. 2d 893 (N.D. Ill. 1999).
326 See supra note 96 and accompanying text.
330 Id.
problems. Both doctrine and academic research tend to view granting control over improvement information to IP owners as beneficial.

Software vendors see the current system as perfectly well-working for security flaws: if a hacker cannot obtain an improvement patent (which is unlikely), she must strike a bargain with the firm that owns the code, or face infringement liability. However, this system creates unacceptable risk of social harm. Hackers have a lucrative and safe alternative to dealing with vendors – they can sell vulnerabilities on the black market. Rewards are generally higher than what software companies pay, and detection of the transaction is unlikely. Worse, though, is that deals with black hats place users of vulnerable software at risk. Their collective risk exposure is greater than the harm, whether pecuniary or reputational, that software firms face. Thus, vendors will underpay for vulnerability information, or use the threat of IP infringement liability to compel hackers to share with them.

The current structure of IP law regarding improvement information – bargain, patent, or infringe – operates on the assumption that infringement is relatively low-cost to society. It also assumes that while infringement is costly to IP owners, they can generally detect violations and pursue remedies. In addition, harm from infringement is typically internalized – the IP owner suffers lost revenue, but third parties are largely unaffected. Thus, copyright or patent holders have efficient incentives to pursue infringement, since they benefit from its mitigation. For software security vulnerabilities, though, detection of violations is challenging, and the societal cost of infringement is significant. One response to this problem is to turn to criminal liability. The combination of public enforcement and significant penalties (particularly imprisonment) can increase deterrence, but detection remains problematic. This Article’s proposal deliberately retains criminal sanctions for hacking in an attempt to deal with black hats, but concedes the low likelihood of obtaining information about illicit transactions.

We argue that, counterintuitively, decreasing the risk of civil liability to hackers ameliorates the distribution problem. Hackers typically face a choice among a forced bargain with vendors, a lawsuit for disseminating findings, and an illegal transaction on the black market. We believe that the safe harbor and vulnerability clearinghouse reforms proposed in this Article give hackers greater bargaining power with vendors, increasing the remuneration available for lawful distribution of data, and also permit sharing information with others to reduce the risk of societal harm. The hacker’s Aegis also smoothes the way for deals between

333 Cf. Lee Anne Fennell, Efficient Trespass: The Case for “Bad Faith” Adverse Possession, 100 NW. U.L. REV. 1037, 1094 (2006) (stating “monitoring costs required to detect infringements and enforce copyright are very large.”).
researchers and vendors through the clearinghouse mechanism and by removing fears of IP liability from the calculus.

IP law deliberately channels information about improvements to IP owners. Even improvement patents rely on the underlying protected invention, placing the two patent owners in a blocking situation that requires bilateral bargaining. This Article’s reforms seek to widen the range of lawful dissemination possibilities. We also argue that IP law – especially when it is based on a utilitarian calculus – must be more sensitive to externality problems. In cases where incentives to infringe, and the concomitant social cost, are high, it may be necessary to reduce, not augment, the weapons available to IP owners to compel bargaining.

VI. CONCLUSION

This Article argues that intellectual property law impedes the dissemination socially valuable research into software security flaws. By reducing the threat of civil IP liability for researchers, the cost of legitimate transactions, and the specter of harm from the term hacker, our proposed reforms would improve software security and decrease users’ risks.

The paper’s conclusions have repercussions beyond software research into security occurs in the physical world as well as the digital one. A graduate student in computer science created a Web application that generates a boarding pass sufficiently realistic to deceive Transportation Security Administration screeners. For his efforts to show the ineffectiveness of airport security, he had his computer seized, was questioned by the FBI, and was denounced by a U.S. Congressman. Cyclists who relied on Kryptonite bike locks were startled when security consultant Chris Brennan showed how to open the locks using a plastic pen. Medeco locks – considered so secure that they are used at the White House and the Pentagon – have been hacked using credit cards and sharp

---

335 The Northwest Airlines Boarding Pass Generator, or, The Emperor Has No Clothes, at http://www.dubfire.net/boarding_pass/.
Security researcher Chris Soghoian published a guide to loopholes and exploits in consumer credit practices that enable attackers to modify their credit reports and obtain loans they could not otherwise qualify for. Locksporters test every lock on the market with tools from custom-made hooks to beer cans. Each of these activities produces valuable information about how safe we really are, and each has been subject to legal threats. It may be necessary to extend the hacker’s Aegis to protect them as well.

Software security research is helpful, and intellectual property law interferes with it. This Article explains how to shield the white hats from the gray suits.

* * *

---