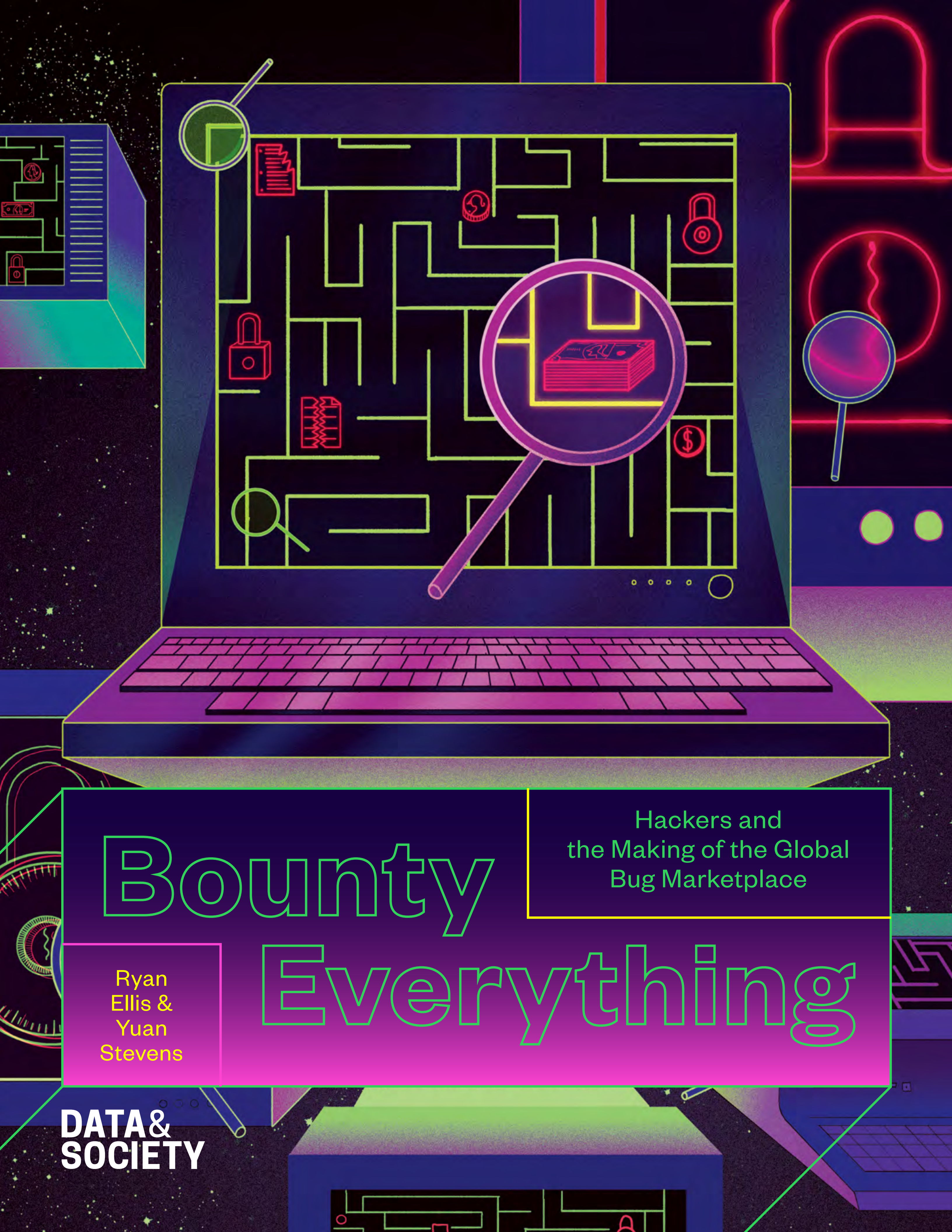


Bounty Everything

Hackers and the Making of the Global Bug Marketplace

Ryan Ellis & Yuan Stevens

DATA& SOCIETY

Contents

Introduction: Bounty Everything

March 2020 was a terrible month for the world, but it was a great month for Zoom. As the COVID-19 pandemic upended nearly every aspect of daily life, Zoom's video conferencing software became ubiquitous and indispensable—elementary school classes, business meetings, birthday parties, church services, and it seemed nearly everything else migrated to Zoom.[1] *The New York Times* March 17 headline captured the state of things well, declaring "We Live in Zoom Now."[2] In a just a few weeks, Zoom surged from roughly 10 million users before the pandemic to more than 200 million.[3] While other tech stocks slumped during the early days of the pandemic, Zoom was a stunning success.[4]

But Zoom's popularity was not without controversy. Its growth was matched by a string of headline-grabbing stories outlining lax security practices.[5] Security researchers revealed previously unknown bugs that could allow malicious attackers

[1] Taylor Lorenz, Erin Griffith, and Mike Isaac, "We Live in Zoom Now," *The New York Times*, March 17, 2020, sec. Style, https://www.nytimes.com/2020/03/17/style/zoom-parties-coronavirus-memes.html

[2] Ibid.

[3] Dain Evans, "How Zoom Became so Popular during Social Distancing," *CNBC*, April 4, 2020, sec. Technology, https://www.cnbc.com/2020/04/03/how-zoom-rose-to-the-top-during-the-coronavirus-pandemic.html; Rupert Neate, "Zoom Booms as Demand for Video-Conferencing Tech Grows," *The Guardian*, March 31, 2020, sec. Technology, https://www.theguardian.com/technology/2020/mar/31/zoom-booms-as-demand-for-video-conferencing-tech-grows-in-coronavirus-outbreak

[4] Zoom's stock would end the year at $337 per share. Carmen Reinicke, "Zoom Video Has Seen Its Stock Spike More than 100% since January as Coronavirus Pushes Millions to Work from Home (ZM)," Markets Insider, March 23, 2020, https://markets.businessinsider.com/news/stocks/zoom-stock-price-surged-coronavirus-pandemic-video-work-from-home-2020-3; Zoom Video Communications, Inc. (ZM)." *Yahoo Finance.* https://finance.yahoo.com/quote/ZM/history?p=ZM.

[5] Natasha Singer and Nicole Perlroth, "Zoom's Security Woes Were No Secret to Business Partners Like Dropbox," *The New York Times*, April 20, 2020, sec. Technology, https://www.nytimes.com/2020/04/20/technology/zoom-security-dropbox-hackers.html

to take control of users' microphones and cameras.[6] As researchers continued to dig, more potentially troubling news came to light. Zoom's public claim that it offered "end-to-end encrypted" communication turned out to be not entirely true.[7] Additionally, an enterprising reporter uncovered that Zoom's iPhone app leaked data to Facebook in ways that were at odds with their stated privacy policy.[8] What's worse, security researchers grumbled that Zoom had long ignored reported bugs and downplayed questions about its security and privacy practices.[9] These revelations were troubling—the FBI issued a stern warning to users; a potential class-action lawsuit was reported; and the New York attorney general began an inquiry into the adequacy of Zoom's security and privacy practices.[10] Just as Zoom was ascending, security and privacy concerns, for a moment, appeared poised to puncture its rise.

Zoom responded quickly. CEO Eric Yuan publicly apologized on CNN and in the pages of the *Wall Street Journal*.[11] He announced a number of changes designed to improve security and privacy and quell the growing concerns.[12] In his public letter, one bullet item stood out: he singled out that Zoom would enhance its "bug bounty" program.[13] Bug bounty programs pay external security researchers who find and report security flaws. These programs are increasingly popular and widespread. Zoom was betting that pouring more resources into its bug bounty program (among other changes) would help quell public concerns that were nagging

[6] Lindsey O'Donnell, "Two Zoom Zero-Day Flaws Uncovered," *Threat Post*, April 1, 2020, https://threatpost.com/two-zoom-zero-day-flaws-uncovered/154337/.

[7] Lily Hay Newman, "The Zoom Privacy Backlash Is Only Getting Started," *Wired*, April 1, 2020, https://www.wired.com/story/zoom-backlash-zero-days/.

[8] Joseph Cox, "Zoom IOS App Sends Data to Facebook Even If You Don't Have a Facebook Account," *Motherboard*, March 26, 2020, https://www.vice.com/en/article/k7e599/zoom-ios-app-sends-data-to-facebook-even-if-you-dont-have-a-facebook-account.

[9] Singer and Perlroth, "Zoom's Security Woes Were No Secret to Business Partners Like Dropbox," *The New York Times*.

[10] Newman, "The Zoom Privacy Backlash Is Only Getting Started," *Wired*; Danny Hakim and Natasha Singer, "New York Attorney General Looks Into Zoom's Privacy Practices," *The New York Times*, March 30, 2020, sec. Technology, https://www.nytimes.com/2020/03/30/technology/new-york-attorney-general-zoom-privacy.html; "FBI Warns of Teleconferencing and Online Classroom Hijacking During COVID-19 Pandemic," Press Release, https://www.fbi.gov/contact-us/field-offices/boston/news/press-releases/fbi-warns-of-teleconferencing-and-online-classroom-hijacking-during-covid-19-pandemic; Kate Cox, "Zoom's Privacy Problems Are Growing as Platform Explodes in Popularity," Ars Technica, March 31, 2020, https://arstechnica.com/tech-policy/2020/03/zooms-privacy-problems-are-growing-as-platform-explodes-in-popularity/.

[11] Kim Lyons, "Zoom CEO Responds to Security and Privacy Concerns: 'We Had Some Missteps,'" *The Verge*, April 5, 2020, https://www.theverge.com/2020/4/5/21208636/zoom-ceo-yuan-security-privacy-concerns.

[12] Eric S. Yuan, "A Message to Our Users," Zoom Blog, April 2, 2020, https://blog.zoom.us/a-message-to-our-users/.

[13] Ibid.

at Zoom. On top of other efforts, getting serious about security for Zoom meant getting serious about its bug bounty program.

Companies, organizations, and even governments now pay rewards to hackers who discover and report bugs—vulnerabilities that undermine security—in their systems.[14] Hackers are people who uncover clever technical solutions and problems through non-obvious means.[15] Paying hackers for bugs was once a radical idea. Up through the early 2010s, most companies and government agencies were far more likely to *threaten* hackers rather than to offer them a reward. But now, attempts to capitalize on hackers' labor—"crowdsourced security," as one leading bug bounty platform puts it—are common.[16] Hundreds of companies and organizations routinely purchase information about flaws in their own systems. Facebook, Google, Microsoft, and Apple each have a bug bounty program. But it is not just a handful of select high-tech companies. United Airlines, Starbucks, the Department of Defense, and countless others also have bug bounty programs.

For many hackers or security researchers, the rise of bug bounty programs provides new benefits and even career paths. Through these programs, hackers interested in security can now look forward to gaining prestige and getting paid rather than primarily worrying about arrest when they discover and disclose security flaws. Bug bounties also provide a unique entry to computer security work. Being the first to find a valid flaw can provide meaningful exposure for a hacker,

[14] This report uses the terms security "bugs", "flaws", and "vulnerabilities" interchangeably. They are weaknesses in information systems or security protocols that could be exploited to crash systems, gain access to sensitive information, or enable other forms of manipulation. See: National Institute of Standards and Technology, "Vulnerability," Computer Security Resource Center, Glossary, n.d., https://csrc.nist.gov/glossary/term/vulnerability; Felivel Camilo, Andrew Meneely, and Meiyappan Nagappan, "Do Bugs Foreshadow Vulnerabilities? A Study of the Chromium Project," in 2015 *IEEE/ACM 12th Working Conference on Mining Software Repositories*, 2015, 269–79, https://doi.org/10.1109/MSR.2015.32.

[15] The term "hacker" is complicated and contested. See: Gabriella Coleman, "Hacker," in *Digital Keywords: A Vocabulary of Information Society and Culture* (Princeton University Press, 2016), 158–72, https://doi.org/10.2307/j.ctvct0023, 163; Leonie Tanczer, "50 Shades of Hacking: How IT and Cybersecurity Industry Actors Perceive Good, Bad, and Former Hackers," SSRN Scholarly Paper (Rochester, NY: Social Science Research Network, September 24, 2019), https://papers.ssrn.com/abstract=3512832. This report uses the terms "hacker," "worker," and "researcher" interchangeably throughout to reflect the fact that participants in our study self-identified as hackers and researchers, and in light of this report's focus on the implications of bug bounty programs as sites of labor and work. For more on the intersections of work in the technical and creative industries, see: Raul Ferrer-Conill, "Playbour and the Gamification of Work: Empowerment, Exploitation and Fun as Labour Dynamics," in *Technologies of Labour and the Politics of Contradicti* on, ed. Paško Bilić, Jaka Primorac, and Bjarki Valtýsson, Dynamics of Virtual Work (Cham: Springer International Publishing, 2018), 193–210, https://doi.org/10.1007/978-3-319-76279-1_11.

[16] Bugcrowd, "Crowdsourced Security Poised for a Breakthrough in 2019," Press Release, n.d., https://www.bugcrowd.com/press-release/crowdsourced-security-poised-for-breakthrough-in-2019/.

serving as valuable professional experience for one's résumé. The effectiveness of these programs has also done much for the public image of the hacker, providing one outlet for a talented, lucrative, and law-abiding technologist. And for the field of computer security, these "bounty" programs are one prominent instance of coordinated vulnerability disclosure—a set of approaches that have the potential to routinize the reporting and disclosure of flaws, improve security, and buffer the risks of legal reprisal.[17] To be sure, software will always contain some bugs.[18] Providing outsiders a clear and safe path to identify and disclose flaws is important both for hackers and public safety.

Like other forms of gig work, bug bounty programs (and platforms) create risks for individual workers: hackers are regularly working long hours for little or even no pay; legal protections are often uncertain or incomplete; and the benefits and opportunities associated with standard forms of employment are largely absent.

But bug bounty programs do more than match hackers and companies looking to buy bugs. They structure and order these interactions: *enclosing* the identification and disclosure of flaws into a new set of market relationships and transactions. This report—based on over 40 interviews and analysis of the history and labor implications of bug bounty programs—provides a window into the working lives of hackers who participate in these programs.[19] Like other forms of gig work, bug bounty programs (and platforms) create risks for individual workers: hackers are regularly working long hours for little or even no pay; legal protections are often uncertain or incomplete; and the benefits and opportunities associated

[17] Tara Swaminatha, "Bug Bounty and Vulnerability Disclosure Programs," *Thomson Reuters Practical Law*, n.d., http://uk.practicallaw.thomsonreuters.com/w-014-4541; Allen D Householder et al., "The CERT Guide to Coordinated Vulnerability Disclosure" (Software Engineering Institute, Carnegie Mellon University, August 2017), https://resources.sei.cmu.edu/asset_files/SpecialReport/2017_003_001_503340.pdf.

[18] OECD, "Encouraging Vulnerability Treatment: Overview for Policy Makers" (Paris: OECD, February 11, 2021), https://doi.org/10.1787/0e2615ba-en, 2.

[19] See Appendix for more information about the methods used for this report.

with standard forms of employment are largely absent.[20] Like other gig workers, hackers are hustling within a world of insecure employment where large firms have enormous power. Bug bounty programs did not start out this way, but over time they have become a legal and economic regime for organizing computer security and maintenance as high-tech piecework.[21] At their core, bug bounty programs rely on vulnerable workers to fix vulnerable systems.

This report considers the implications of bug bounty programs. Part history, part study of political economy, this report identifies the pleasures and hazards that dot the bug bounty market and draws out larger lessons and notes of caution. First, we describe how bug bounty programs work, and how they organize the work of finding and disclosing vulnerabilities. Second, we provide a capsule history of these programs, beginning with Netscape's first "bugs bounty" program in 1995. Then we analyze contemporary bug bounty platforms—the new intermediaries that now structure the vast majority of bounty work—and recount the experiences of the hackers who work in these programs. Finally, we consider how bug bounty programs can be reimagined to better serve the interests of both computer security and the workers that increasingly help maintain our digital world.

> At their core, bug bounty programs rely on vulnerable workers to fix vulnerable systems.

"Bounty" programs are now in vogue—each day another company, government agency, or public department seems to announce with great fanfare a new high-profile bounty program. Organizations have also begun adopting the "bounty" model of paying the crowd to find not just technical flaws, but other kinds of flaws

[20] Gig work at its core transfers risks from organizations onto workers. As Alexandrea J. Ravenelle notes, this is true even for workers that have found success within these markets. Ravenelle, *Hustle and Gig: Struggling and Surviving in the Sharing Economy* (Oakland, CA: University of California Press, 2019), 18. Colin Crouch, *Will the Gig Economy Prevail?* (Medford, MA: Polity, 2019).

[21] Ali Alkhatib, Michael S. Bernstein, and Margaret Levi, "Examining Crowd Work and Gig Work Through The Historical Lens of Piecework," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17: CHI Conference on Human Factors in Computing Systems, Denver Colorado USA: ACM, 2017), 4599-4616, https://doi.org/10.1145/3025453.3025974.

in sociotechnical systems.[22] Efforts are underway to pay people through bounty programs to investigate and report terms of service violations, algorithmic harm, and the spread of disinformation.[23]

Ultimately, if not designed and deployed properly bounty programs might create incentives that undermine the development of secure software and services. Rather than encouraging companies and organizations to invest in security from the get-go, this model can ironically perpetuate a world full of bugs that uses a global pool of insecure workers to prop up a business model centered on rapid iteration and perpetual beta.[24] Such a world creates the ideal conditions for perpetuating racialized labor inequalities in hacking work and for creating forms of predatory inclusion that absorb precarious workers into hacking for a wage in an extractive labor relationship.[25] So far, bounty programs seem willing to integrate a diverse workforce in their practices, but only on terms that deny them the job security and access enjoyed by core security workforces. These inequities go far beyond the difference experienced by temporary and permanent employees at companies such as Google and Apple. The global bug bounty workforce is doing piecework—they are paid for each bug, and the conditions under which a bug is paid vary greatly from one company to the next.[26]

[22] For more on the relationship between the social and the technical particularly in the context of social media platforms, see: Matt Goerzen, Elizabeth Anne Watkins, and Gabrielle Lim, "Entanglements and Exploits: Sociotechnical Security as an Analytic Framework" (9th {USENIX} Workshop on Free and Open Communications on the Internet ({FOCI} 19), Santa Clara, CA, 2019), https://www.usenix.org/conference/foci19/presentation/goerzen.

[23] "Data Abuse Bounty Program | Facebook," https://www.facebook.com/data-abuse; "The Coded Gaze: Unpacking Biases in Algorithms That Perpetuate Inequity," *The Rockefeller Foundation*, https://www.rockefellerfoundation.org/case-study/unpacking-biases-in-algorithms-that-perpetuate-inequity/; Josh Kenway and Camille François, "Bug Bounties for Algorithmic Harms? Lessons from Cybersecurity Vulnerability Disclosure for Algorithimic Harms, Discovery, Disclosure, and Redress," Algorithmic Justice League, 2021 (forthcoming); "Twitter Algorithmic Bias—Bug Bounty Program," HackerOne, https://hackerone.com/twitter-algorithmic-bias.

[24] On the problems of misaligned incentives and cybersecurity, see: Tyler Moore, "The Economics of Cybersecurity: Principles and Policy Options," *International Journal of Critical Infrastructure Protection* 3, no. 3 (December 1, 2010): 103–17, https://doi.org/10.1016/j.ijcip.2010.10.002.

[25] Tressie McMillan Cottom, "Where Platform Capitalism and Racial Capitalism Meet: The Sociology of Race and Racism in the Digital Society," *Sociology of Race and Ethnicity* 6, no. 4 (October 1, 2020): 441–49, https://doi.org/10.1177/2332649220949473; Sareeta Amrute, *Encoding Race, Encoding Class: Indian IT Workers in Berlin*, (Durham, NC: Duke University Press, 2016).

[26] On the differences found among crowdsourced workers see: Lilly Irani, "Difference and Dependence among Digital Workers: The Case of Amazon Mechanical Turk," *South Atlantic Quarterly* 114, no. 1 (January 1, 2015): 225-34, https://doi.org/10.1215/00382876-2831665; retrieved from https://escholarship.org/uc/item/6xk920pj.

The future of bug bounty programs is open: will they be used by companies to hide vulnerabilities and as a way to ship quickly, knowing that a raft of precarious workers will find the bugs for them? Or, will such programs create viable pathways to full time security work, improve response and patching times, clarify terms of payment, and develop stronger legal protections for such workers and hackers alike?[27] As the bounty model is adopted by more and more organizations, this report argues that identifying and addressing the risks faced by precarious workers is not only the right thing to do for workers: it is also a significant long-term investment in improving the security of our digital world.

[27] Notably, this report focuses on the analysis of labor conditions for bug bounty programs drawing on our perspectives as researchers in the US and Canada.

Part I: How Bug Bounty Programs Work

Bug bounty programs transform vulnerability disclosure. These programs take on different forms—some are run by large bounty platforms (see Part III below), others are operated in-house by companies looking to spot bugs in their own software; some accept submissions from any and all hackers, other only accept bugs from invited hackers—but all bug bounty programs compensate independent hackers who find and disclose bugs. At their most basic level, bounty programs pay hackers for flaws.

Bug bounty programs organize hacking as "gig work."[28] Peering inside the world of bug bounties, we see labor conditions that are now familiar: hackers contributing to these programs in some ways resemble Uber drivers, Instacart shoppers, and other workers making their way through the promises, contradictions, pleasures, and conflicts of the gig economy.[29] There are important differences—not all gig work is created equal. Hackers participating in bounty programs are working in a highly-desirable and often high-status field—computer security. They hope—and sometimes do—leverage their bounty work into lucrative payouts and additional

[28] Jamie Woodcock and Mark Graham, *The Gig Economy: A Critical Introduction* (Medford, MA: Polity, 2020).

[29] The gig economy is a broad term that now refers to the reorganization of labor through temporary work arrangements and independent contracting tracked and managed by digital portals or platforms. Ibid, 3.

employment. But, like other gig workers, the hackers that contribute to bounty programs are contingent workers.[30] Vulnerabilities are time-sensitive; workers are only paid when they *are the first* to find and disclose valid, qualifying bugs, regardless of time spent. Under this model, hacking is turned into a form of high-tech piecework.[31] Like other sectors of the gig economy, these workers are not directly employed by bounty programs, but are classified in many places as independent contractors.[32] As such, they do not receive the guaranteed salary, benefits, or legal protections that are afforded to employees.[33]

Before turning to examine the history and tensions of this work, this section provides an overview of bug bounty programs. It addresses a number of initial, general questions, including: What are bug bounty programs? How do bug bounty programs work? What is the financial model that undergirds this market for bugs? And who are the hackers who engage in this work?

Bug bounty programs routinize the disclosure of bugs by hackers for compensation. They are one instance of coordinated vulnerability disclosure, which is the process of finding and disclosing security vulnerabilities to mitigate their potential harm.[34] They are also part of the "defensive market," where bugs are disclosed in order to fix the flaws.[35] Other ways of discovering and disclosing bugs, and other markets,

[30] For a general introduction to the ways in which gig work creates new risks and pressures on workers, see: Ravenelle, *Hustle and Gig*, 1; Woodcock and Graham, *The Gig Economy*; and Nick Srnicek, *Platform Capitalism* (Malden, MA: Pluto, 2017). For a detailed case study of particular forms of gig work, see in particular Alex Rosenblat, *Uberland: How Algorithms are Rewriting the Rules of Work* (Oakland, CA: University of California Press, 2018).

[31] On contemporary gig work as a form of piecework, see Srnicek, *Platform Capitalism*, 72. See also Alkhatib et al, "Examining Crowd Work and Gig Work Through The Historical Lens of Piecework."

[32] Discussions on the classification of gig workers is a point of conflict with significant consequences for both workers and platforms. See Rosenblat, *Uberland*, 8-9, 156; Woodcock and Graham, *The Gig Economy*; Srnicek, *Platform Capitalism*, 75-88

[33] Ibid.

[34] Lorenzo Pupillo, Afonso Ferreira, and Gianluca Varisco, *Software Vulnerability Disclosure in Europe: Technology, Policies and Legal Challenges* (Brussels: Centre for European Policy Studies, 2018), https://www.ceps.eu/wp-content/uploads/2018/06/CEPS%20TFRonSVD%20with%20cover_0.pdf, 5-6, 9. See also ISO/IEC, "ISO/IEC 29147:2014 Information technology-Security techniques-Vulnerability disclosure," 2014, https://www.iso.org/standard/45170.html.

[35] See e.g., Bruce Schneier, "Should U.S. Hackers Fix Cybersecurity Holes or Exploit Them," *The Atlantic*, May 19, 2014, https://www.theatlantic.com/technology/archive/2014/05/should-hackers-fix-cybersecurity-holes-or-exploit-them/371197/.

also exist.[36] Bugs can be discovered during software development by the developers themselves. Bugs can also be discovered through penetration testing, or "pen testing," which are security audits conducted by contracted security firms that simulate real-world attacks and search for unprotected security gaps.[37] Bugs are also disclosed by hackers through vulnerability disclosure programs, which, unlike bug bounty programs, do not pay for flaws. Additionally, particular types of flaws are sold outside of bug bounty programs to governments as well as in what is described as the "offensive market."[38] In the offensive market, bugs are bought and sold not to be fixed, but to be turned into exploits and attacks.[39] Our interviews did not focus on or investigate the offensive market. Unless otherwise specified, the term "the market" in this paper refers to the bug bounty market.

Variation in Bug Bounty Programs: Public or Private, In-House or Platform

All bug bounty programs purchase bugs from hackers. Hackers search out new flaws, write up reports detailing their findings, and submit them through digital

[36] The purchase of bugs by companies offering managed security services are a slightly different element of the defensive market. Here, flaws are bought from hackers in order to support the development of intrusion detection and prevention systems, for example. Trend Micro is a prime example of a managed security service provider that engages in the purchase and sale of flaws. Trend Micro runs a typical bug bounty program Zero Day Initiative (ZDI)—but also uses the information gleaned through this program to inform the development of their threat detection and protection services that they sell to customers. The business model is based on the idea that TrendMicro customers gain knowledge of these flaws—and can receive temporary patches for them—before the flaws are disclosed to the broader public. See: Trend Micro, "Trend Micro's Zero Day Initiative Leads Vulnerability Disclosure Landscape in Independent Research," Trend Micro Press Release, December 3, 2019, https://www.trendmicro.com/en_gb/about/newsroom/press-releases/2019/2019-12-03-trend-micros-zero-day-initiative-leads-vulnerability-disclosure-landscape-in-independent-research.html; and interview with Dustin Childs, 2019.

[37] On penetration testing or red-teaming, see: National Research Council, *Cybersecurity Today and Tomorrow: Pay Now or Pay Later* (Washington, DC: National Academies Press, 2002), 10; Rapid7, "Penetration Testing," n.d., https://www.rapid7.com/fundamentals/penetration-testing/.

[38] See e.g., Lillian Ablon, Martin C. Libicki, and Andrea M. Abler, *Markets for Cybercrime Tools and Stolen Data: Hackers' Bazaar* (Santa Monica, CA: RAND Corporation, 2014), https://www.rand.org/pubs/research_reports/RR610.html; Mailyn Fidler, "Regulating the Zero-Day Vulnerability Trade: A Preliminary Analysis," *I/S: A Journal of Law and Policy for the Information Society* 11, (2015): 405; Jaziar Radianti and Jose J. Gonzalez, "Understanding Hidden Information Security Threats: The Vulnerability Black Market," *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)* (Waikoloa, HI: IEEE, 2007), 156c-156c.

[39] Joshua Kenway, Maho Sugihara, Asaf Zilberfarb, and Pablo Tortolero, "More Sunlight, Fewer Shadows: Guidelines for Establishing and Strengthening Government Vulnerability Disclosure Policies," Cyber Threat Alliance, 2021, https://www.cyberthreatalliance.org/government-handling-of-zero-days-more-sunlight-fewer-shadows/.

interfaces for review and, often they hope, payment. But not all programs are alike. There are variations: some are run "in-house," while others are operated by bounty platforms; many programs are open, accessible to any and all hackers who want to submit, and others are private, open by invitation only.

In-house bounty programs are those run directly by technical staff at a company, which often requires a certain size and technical capability. For example, Apple, Google, and Microsoft each run and manage their own programs with dedicated staff. But organizations can also outsource the management of their bug bounty program to a "platform," such as HackerOne, Bugcrowd, or Intigriti (and others). These platforms are a labor intermediary or emerging type of temporary staffing agency between hackers and organizations.[40] It is free for hackers to sign up for a platform, while companies pay the platforms a flat fee and/or a percentage of all bounties for hosting their bounty program. The vast majority of bounty programs are run through one of two large bug bounty platforms, HackerOne or Bugcrowd (more on these platforms below). Even nominally stand-alone or in-house programs are starting to partner with platforms.[41]

Bounty programs, whether in-house or platform-run, can also be public or private. Public programs are open to all. Any hacker can submit bug reports to these programs. Private programs, however, are only open to those invited to contribute. Most bounty programs are private. For example, 79% of the bounty programs that run on HackerOne's platform were private in 2019 (Bugcrowd reports a similar figure).[42] For many hackers, working in this market means working through a bounty platform and attempting to secure invitations to private programs.

[40] See e.g., Kendra Strauss and Judy Fudge, "Introduction" in *Temporary Work, Agencies and Unfree Labour*, ed. Judy Fudge and Kendra Strauss (UK: Routledge, 2013) 1–25; Niels van Doorn, "Platform Labor: on the Gendered and Racialized Exploitation of Low-Income Service Work in the 'On-Demand' Economy," *Information, Communication & Society* 20, no. 6 (2017): 898–914, doi: 10.1080/1369118X.2017.1294194.

[41] Facebook Bug Bounty, "Launching Payout Collaboration with HackerOne," April 16, 2020, https://www.facebook.com/notes/430340741282490/; Microsoft Security Response Center, "Microsoft Bounty Program Update," April 2, 2019, https://msrc-blog.microsoft.com/2019/04/02/microsoft-bounty-program-updates-faster-bounty-review-faster-payments-and-higher-rewards/.

[42] HackerOne, *The Hacker Powered Security Report: 2019*, 13, https://www.hackerone.com/resources/reporting/the-hacker-powered-security-report-2019; Bugcrowd, *2018 State of Bug Bounty*, 5, https://www.bugcrowd.com/resources/reports/state-of-bug-bounty-2018/.

Surveying the Bug Bounty Market

Publicly available data indicates a large and growing global market for bugs. Definitive aggregate numbers are difficult to pin down.[43] But figures reported by the two largest bug bounty platforms, HackerOne and Bugcrowd, and additional data reported by Facebook, Google, and Microsoft, provide a rough sense of the size and scope of the market. There are hundreds of different bug bounty programs—new programs appear to pop up every day.[44] Thousands of workers sell bugs through these programs. HackerOne counts over 600,000 registered accounts on their platform alone.[45] Bounty programs process hundreds of thousands of bug reports per year. In 2019, programs running on HackerOne's platform received over 30,000 valid bug submissions, with many more submissions reviewed and deemed invalid, either due to falling outside of a program's specifications and scope or determined to be a duplicate report (a report of a known issue).[46] In 2018, Bugcrowd processed over 37,000 submissions.[47] The following year, Facebook's bounty program received 15,000 bug reports, roughly 1,300 of which were eventually deemed valid.[48] These programs pay hackers tens of millions of dollars per year. In 2019, the various bounty programs on the HackerOne platform paid hackers roughly $40 million; during the same period, Facebook paid participants in their program $2.2 million and Google paid $6.5 million in bounties.[49] All of these numbers—the number of programs, the numbers of hackers, the number of submissions, and the dollars paid and earned—are growing annually.

[43] Answers to seemingly simple questions—How many companies offer bounties? How many hackers participate in bounty programs in a given year? What is the size of the market?—are elusive. There is no authoritative list of programs or participants.

[44] HackerOne, *The 2020 Hacker Report: The Survey and Statistics of the Ethical Hackers Community*, (February 2020), https://www.hackerone.com/resources/reporting/the-2020-hacker-report, 4.

[45] Ibid. A large number of registered users does not mean a large number of active users. For example, a study of the freelance platform Upwork found that less than 7% of registered users actually found a job. See: Woodcock and Graham, *The Gig Economy*, 91.

[46] HackerOne, *The 2020 Hacker Report*, 4; HackerOne, *The Hacker Powered Series Report: 2019*, 2.

[47] Bugcrowd, *State of Bug Bounty*, 12.

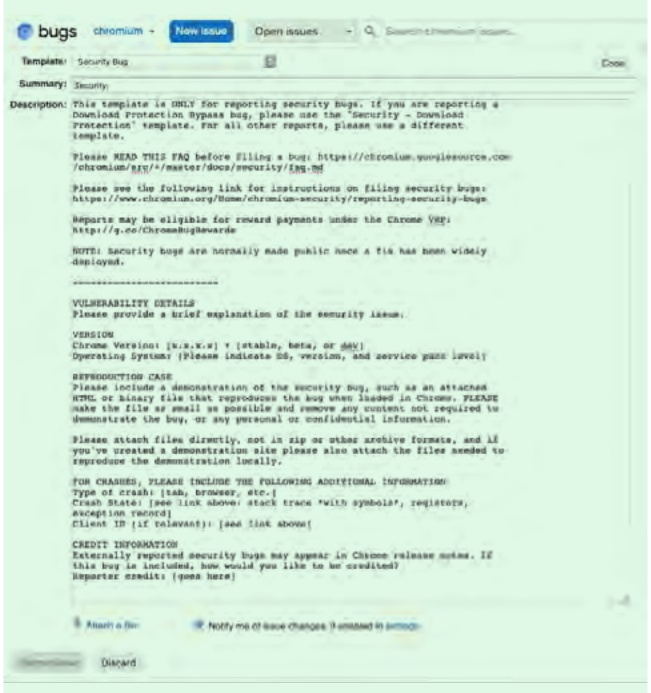
[48] Dan Gurfinkel, "A Look Back at 2019 Bug Bounty Highlights," Facebook, February 7, 2020, https://www.facebook.com/notes/facebook-bug-bounty/a-look-back-at-2019-bug-bounty-highlights.

[49] HackerOne, *The 2020 Hacker Report*, 4; Gurfinkel, "A Look Back at 2019 Bug Bounty Highlights"; Natasha Pabrai, Jan Keller, Jessica Lin, Anna Hupa, and Adam Bacchus, "Vulnerability Reward Program: 2019 Year in Review," *Google Security Blog*, January 28, 2020, https://security.googleblog.com/2020/01/vulnerability-reward-program-2019-year.html.

Bug Reports and Triage Work

Much of the work involved with bug bounty programs is administrative. Hacking, in this context, requires paperwork. Lots of it. Filling out bug reports, submitting them into ticketing queues, replying to queries from middle managers, resubmitting reports with added detail—this is in part what the work of hacking looks like. Along these lines, one of the core tasks of all bug bounty programs—public and private, platform or in-house—is reviewing and triaging incoming reports. Bounty programs employ triage workers to sift through incoming submissions—someone has to review those tens of thousands of bug reports that are submitted. Bug bounty programs facilitate the routine submission and triage of bug reports (see figure 1).[50]

FIGURE 1. SECURITY BUG REPORT TEMPLATE FOR GOOGLE'S CHROMIUM[51]


bugs
chromium
New issue
Open issues
Search chromium issues...
Close
Attach a file
Notify me of issue changes, if enabled in settings
Submit issue
Discard

Template: Security Bug

Summary: Security:

Description:

```
This template is ONLY for reporting security bugs. If you are reporting a
Download Protection Bypass bug, please use the 'Security - Download
Protection' template. For all other reports, please use a different
template.

Please READ THIS FAQ before filing a bug: https://chromium.googlesource.com
/chromium/src/+/master/docs/security/faq.md

Please see the following link for instructions on filing security bugs:
https://www.chromium.org/Home/chromium-security/reporting-security-bugs

Reports may be eligible for reward payments under the Chrome VRP:
http://g.co/ChromeBugRewards

NOTE: Security bugs are normally made public once a fix has been widely
deployed.

---------------------------

VULNERABILITY DETAILS
Please provide a brief explanation of the security issue.

VERSION
Chrome Version: [x.x.x.x] + [stable, beta, or dev]
Operating System: [Please indicate OS, version, and service pack level]

REPRODUCTION CASE
Please include a demonstration of the security bug, such as an attached
HTML or binary file that reproduces the bug when loaded in Chrome. PLEASE
make the file as small as possible and remove any content not required to
demonstrate the bug, or any personal or confidential information.

Please attach files directly, not in zip or other archive formats, and if
you've created a demonstration site please also attach the files needed to
reproduce the demonstration locally.

FOR CRASHES, PLEASE INCLUDE THE FOLLOWING ADDITIONAL INFORMATION
Type of crash: [tab, browser, etc.]
Crash State: [see link above: stack trace *with symbols*, registers,
exception record]
Client ID (if relevant): [see link above]

CREDIT INFORMATION
Externally reported security bugs may appear in Chrome release notes. If
this bug is included, how would you like to be credited?
Reporter credit: [goes here]
```

[50] Describing the act of hacking itself—whether it involves reconnaissance (information-gathering), scanning, or testing—is beyond the scope of this paper.

[51] Available at: https://bugs.chromium.org/p/chromium/issues/entry?template=Security%20Bug or https://perma.cc/UY24-RKX4 (note: must be logged in to a Google account to view).

Bug reports ideally include all relevant information needed to understand the vulnerability—such as the technical context of the person who found the flaw, the steps needed to reproduce the flaw, and the perceived level of impact or criticality of the vulnerability.[52] Using a bug tracking system, triage workers decide if the bug has already been reported or patched. Triage also determines whether the flaw reported falls within a predetermined scope and is therefore deemed "valid" for a payout. For instance, Tesla's bug bounty program provides a list of "targets" that are in-scope—including Tesla's iOS app, a number of identified public facing websites, and a Tesla that you own—while identifying those that are out-of-bounds, including particular websites, for example, feedback.tesla.com. It also provides a list of the types of flaws or issues that are considered to be out-of-scope or non-qualifying (clickjacking, phishing and social engineering attacks, internal IP address disclosure, and others).[53] Finally, triage workers ensure that the reported issue can be reproduced.

There are numerous reasons why a bug report might not qualify for payout, and triage workers have significant power in this decision. The report might be too poorly written or incomplete for the triage team to verify. It may disclose a flaw that's already in the triage pipeline. The report may involve a bug where a patch has been planned but not yet released. The impact of the bug might be unclear. The vulnerability might be an acceptable security risk or may even be a design feature.

Triage happens differently in different programs. In-house programs generally rely on employees to screen and review reports. But, for bug bounty platforms, triage is done by workers several steps removed from the software or service in question. Triage is often managed on bounty platforms by contractors—not full-time employees. Many of these contractors, including some we spoke with, are hackers who moved from submitting bugs to bug bounty programs to screening reports for bug bounty platforms. In all cases, those who do triage reports play a significant role in the decision of whether a bug report is complete, well-written,

[52] See e.g., The Chromium Projects, "Reporting Security Bugs," n.d., https://www.chromium.org/Home/chromium-security/reporting-security-bugs; *HackerOne*, "Introducing Report Templates," HackerOne, September 1, 2016, https://www.hackerone.com/blog/Introducing-Report-Templates; David Sopas, "How to Write a Great Vulnerability Assessment with this Template," *Cobalt.io*, September 29, 2016, https://blog.cobalt.io/how-to-write-a-great-vulnerability-report-ab8654c6290c; Mozilla, "Bug Report Writing Guidelines," *MDN Web Docs*, n.d., https://developer.mozilla.org/en-US/docs/Mozilla/QA/Bug_writing_guidelines, and many others.

[53] BugCrowd, "Tesla," n.d., https://bugcrowd.com/tesla or https://perma.cc/SZF5-DTS6.

or involves a flaw that qualifies as a vulnerability that is worth responding to, patching, and paying for. Disputes over triage, as described in Part IV, are a recurring source of tension for hackers.

Finding Security Flaws

In general terms, a software bug is any function of the code that strays from the programmers' intention. A bug can be as trivial as small graphical glitches or as serious as causing a program to crash.[54] However, bug bounty programs are almost entirely concerned with a particular subset of bugs that are security flaws, which allow for unauthorized data access, disclosure, destruction, or modification.[55]

Security flaws exist in all systems and can be discovered through a wide range of techniques. For instance, numerous hackers interviewed described "cross-site scripting," which involves the unintentional execution of JavaScript by a website. The result can be as harmless as causing the website to display certain text, or as detrimental as collecting and exfiltrating data such as usernames, passwords, or personally identifiable information.

Ultimately, what counts as a valid flaw is open for interpretation. For example, one researcher told us that in his experience, corporations with bug bounty programs are more willing to pay for bug reports that protect aspects of their business model, rather than those that *only* secure user privacy. For him, companies see security issues as those that concern the corporation's business logic or operations, such as unauthorized access to company data, which can directly affect their revenue stream. On the other hand, his view is that companies generally downplay the impact of flaws disclosed in bug reports that involve user privacy, for example—unless the hacker can successfully argue that the leak of private user information has a clear and obvious impact on the corporation's profit goals.

[54] National Institute of Standards and Technology, "Vulnerability," Computer Security Resource Center, Glossary, n.d.,

[55] Carl E. Landwehr, Alan R. Bull, John P. McDermott, and William S. Choi, "A taxonomy of computer program security flaws." ACM Computing Surveys 26 no. 3 (1994): 211-254, doi: 10.1145/185403.185412.

Prices for Flaws

Bug bounty programs often publish their price ranges, including an upper-bound for different types of bugs. The price paid per flaw disclosed can stretch from as low as fifty dollars up to seven figures. On paper, prices are typically organized by type of flaw and its associated severity (the level of impact it has on the organization and users). For example, in figure 2, Apple's bug bounty program shows that the maximum payouts for vulnerabilities or exploits depend on factors such as the system involved and type of attack. In figure 3, Google's Vulnerability Reward Program shows "reward" amounts for categories of flaws dependent on elements including the type of bug found as well as the application in which the flaw was found.

FIGURE 2. APPLE SECURITY BOUNTY—BOUNTY PAYMENT CATEGORIES[56]

BOUNTY CATEGORIES	TOPIC	MAX PAYOUT
iCloud	Unauthorized access to iCloud account data on Apple Servers	$100,000
Device attack via physical access	Lock screen bypass User data extraction	$100,000 $250,000
Device attack via user-installed app	Unauthorized access to sensitive data Kernel code execution CPU side channel attack	$100,000 $150,000 $250,000

In reality, determining the price for a flaw is not as cut and dried as published price lists indicate. There are often significant disagreements between hackers and bounty programs about price. The workers reviewing incoming reports have significant power to determine how much (or if) a hacker is paid. Hackers with more social capital are able to haggle and bargain over price in ways that others cannot (more on this below in part IV).

Platforms and bounty programs use prices strategically. Raising prices is a sure way to draw in more hackers and boost engagement. Some bug bounty programs pay security researchers once the bug report is deemed valid and before the flaw has been patched. Other programs pay people only once the patch is rolled out,

[56] Apple Developer, "Apple Security Bounty" *Apple Developer*, https://developer.apple.com/security-bounty/, version archived on February 24, 2020 online here: https://web.archive.org/web/20200224164045/https://developer.apple.com/security-bounty/.

which results in significant variation in wait-times for payment. Later (in part IV) we will describe the significant discrepancies that people experience when it comes to the timing and amounts they are paid as bug bounty workers.

FIGURE 3 GOOGLE AND ALPHABET VULNERABILITY REWARD PROGRAM (VRP) RULES—REWARD AMOUNTS FOR SECURITY VULNERABILITIES[57]

Rewards for qualifying bugs range from $100 to $31,337. The following table outlines the usual reqards chosen for the most common classes of bugs. To read more about our approach to vulnerabilituy rewards you can read our Bug Hunter University article.

CATEGORY	EXAMPLES	APPLICATIONS THAT PERMIT TAKING OVER A GOOGLE ACCOUNT	OTHER HIGHLY SENSITIVE APPLICATIONS	NORMAL GOOGLE APPLICATIONS	NON-INTEGRATED ACQUISITIONS AND OTHER SANDBOXED OR LOWER PRIORITY APPLICATIONS
VULNERABILITIES GIVING DIRECT ACCESS TO GOOGLE SERVERS					
Remote control execution	Command injection, deseralization bugs,sandbox escapes	$31,337	$31,337	$31,337	$1,337-$5,000
Unrestricted file system or database access	Unsandboxed XXE, SQL injection	$13,337	$13,337	$13,337	$1,337-$5,000
Logic flaw bugs leaking or bypassing significant security controls	Direct object reference, remote user impersonation	$13,337	$7,500	$5,000	$500
VULNERABILITIES GIVING ACCESS TO CLIENT OR AUTHENTICATED SESSION OF THE LOGGED-IN VICTIM					
Execute code on the client	Web: Cross-site scripting Mobile/ Hardware: Code execution	$7,500	$5,000	$3,133.7	$100
Other valid security vulnerabilities	Web: CSRF1 Clickjacking Mobile/ Hardware: Information Leak, privilege escalation	$500-$7,500	$500-$5,000	$500-$3,133.7	$100

[57] Google Application Security, "Google Vulnerability Reward Program (VRP) Rules," *Google Application Security*, https://www.google.com/about/appsecurity/reward-program/.

Who Are Bug Bounty Workers?

Bounty work draws a young and global workforce. Reports from HackerOne and Bugcrowd provide a partial glimpse into the labor market.[58] The majority of hackers are young: 71.5% of participants working on programs hosted by Bugcrowd are under 30; 84% of hackers working on HackerOne supported-programs are under 34, and 48% are under 24.[59] A substantial portion of bounty labor is self-taught and has not completed a college degree; many of the workers are students.[60] For roughly a quarter of the hackers working on Bugcrowd and HackerOne, bug bounty programs are a full-time job.[61] HackerOne's internal figures note that 40% of their hackers spend 20 hours or more per week hunting for bugs.[62] For these participants, the money earned from bounties is used to pay for day-to-day living costs, tuition, and other expenses.[63] Exactly how much these hackers actually earn, however, is an open question. While HackerOne flags the eye-popping six or even seven figures sums earned by some, it acknowledges that the vast majority of hackers earn less than $20k per year.[64] Prior work indicates that a small handful of hackers earns the bulk of bounty payouts.[65]

Bounty work is largely international.[66] The overwhelming majority of hackers who contribute to bug bounties via HackerOne—89%—are based outside the US.[67] India accounts for the largest share of HackerOne contributors: in 2019, 12%

[58] Reports from Google and Facebook's bounty programs mirror those reported by the platforms. See: Gurfinkel, "A Look Back at 2019 Bug Bounty Highlights"; Google Security Team, "Behind the Scenes," Google Bughunter University, https://web.archive.org/web/20210211082718/https://sites.google.com/site/bughunteruniversity/behind-the-scenes

[59] Bugcrowd, *2019 Inside the Mind of a Hacker*, 14; HackerOne, *The Hacker Powered Security Report 2019*, 50.

[60] HackerOne, *The Hacker Powered Security Report 2019*, 18.

[61] Bugcrowd, *2019 Inside the Mind of a Hacker*, 12; HackerOne, *The Hacker Powered Security Report 2019*, 27.

[62] HackerOne, *The 2020 Hacker Report*, 8.

[63] Bugcrowd, *2019 Inside the Mind of a Hacker*, 13.

[64] HackerOne, *The 2020 Hacker Report*, 15.

[65] Ryan Ellis, Keman Huang, Michael Siegel, Katie Moussouris, and James Houghton, "Fixing a Hole: The Labor Market for Bugs," *New Solutions for Cybersecurity*, eds. Howard Shrobe, David L. Shrier, and Alex Pentland (Cambridge, MA: MIT Press, 2018), 129–159.

[66] HackerOne, *The Hacker Powered Security Report: 2019*.

[67] Ibid., 9. Bugcrowd reports comparable figures: 73% of hackers who participate in Bugcrowd programs are from outside the US. Bugcrowd, *Inside the Mind of a Hacker*, 14.

of registered hackers were based in India, and 11% were based in the US, the next largest country by contribution.[68] During the same year, 18% of all reports submitted via HackerOne originated from hackers in India; 11% originated from the US, the next largest country by origin.[69] Companies based in more than 59 different countries have bounty programs on HackerOne's platform.[70] However, companies based in the US make up the majority of all payments. In 2019, US-based companies paid $29 million in bounties via HackerOne—85.9% of all payments processed through the platform.[71] Yet, the majority of these payments—80.9%—are received by hackers outside the US—including India (10.5%), Russia (7.9%), China (6.7%), and over 100 other countries.[72] Bugcrowd's data tells a similar story: 79% of bounty payments originated from the US, but 74% of all payments went to hackers outside the US (34% were paid to hackers in India, the single largest country by receipt).[73] The flow of bounty earnings is clear, going from companies based in the US to mostly young hackers working outside the US, whose experiences we describe in Part IV.

What Motivates Hackers to Contribute to Bounty Programs?

As other studies of hackers have made clear, there is no single motivation or ethic that undergirds all hackers or all hacking.[74] Hackers are not a single identifiable community, but a mix of overlapping yet distinct subgroups.[75]

[68] HackerOne, *The 2020 Hacker Report*, 11.

[69] Ibid., 10.

[70] Ibid.

[71] Ibid. 13.

[72] Ibid. 13, 51. These figures also roughly match previous years. See: HackerOne, *The Hacker Powered Security Report: 2019*, 9-10.

[73] Bugcrowd, *Inside the Mind of a Hacker: 2020*, 2020, https://itmoah.bugcrowd.com/, 9.

[74] Hackers are best understood not as a monolith, but as a constellation of loosely tethered and evolving sub-cultures, each with different (and shifting) members, mores, and rites. On the importance of recognizing the plurality of hacking communities, see E. Gabriella Coleman, *Coding Freedom: The Ethics and Aesthetics of Hacking* (Princeton, NJ: Princeton University Press, 2013) 17-20.

[75] Ibid., 17-20.

The hackers we spoke to range from full-time hackers to brand-new contributors—and everything in between.[76] For many of our interviewees, their contribution to bounty programs changed over time: they would engage in bug bounty work full-time at one point and in a part-time or recreational fashion at other times. Jesse Kinser described her bug bounty work as "fun money," and told us that "I've got my full-time job that pays the bills, keeps the lights on, but my money that I make from bug bounty, I buy all my cool stuff with. So, I buy new computers, yeah. I put a down payment on a Tesla, like, it's all my fun money."[77]

For many others, however, bounty work is not an added luxury: it is the main source of their income, or a crucial employment opportunity. Indeed, some hackers we spoke to were focused on reputation-building and viewed bounty programs as a way to build a career. Some saw bounty programs as an "on-ramp," a way to jump-start a career in information security or a related field—to make some money and connections before moving on to non-bounty work such as doing in-house security for a company or working as a security consultant. One hacker we interviewed saw bug bounty programs as a way to "get spotted" or noticed in a crowded field. Working in India, the hacker observed that the competition for full-time work was fierce. It was hard to stand out in the crowd. For him, bug bounty programs provided an edge—a possible way of getting his résumé noticed.

It was this possibility—using bug bounty programs as a springboard for a career in security—that in part inspired Rohit Guatam and Shifa Cyclewala to create Hacktify Cybersecurity in 2016. Hacktify is a training academy based in Mumbai, India, that provides a bug bounty hunting crash course for eager hackers-in-training. Before founding Hacktify, Guatam worked as a security analyst, eventually becoming a security consultant focused on vulnerability assessment and penetration testing.[78] Yet he saw incredible value in bounty programs. Submitting to bug bounty programs was, in his view, an important part of his professional development. It allowed him to hone skills that could help him in his career. Hacktify teaches people the tools and methods now commonly used for web application flaw-finding.[79] The training

[76] The blending of part-time and full-time work is a common feature of gig work platforms. See: Rosenblat, *Uberland*, 49–72; Ravenelle, *Hustle and Gig*.

[77] Interview with Jesse Kinser, 2019.

[78] Interview with Rohit Gautam and Shifa Cyclewala, 2020.

[79] "Bug Bounty Hunting and Penetration Testing," Hacktify Cyber Security, https://hacktify.thinkific.com/courses/bug-bounty-hunting-and-penetration-testing.

center focuses on the nontechnical aspects of bounty programs as well, instructing interested students in how to write professional bug reports. This initiative is part of a rapidly expanding industry that provides training and certification online for people interested in entering the cybersecurity field.[80] Guatam and Cyclewala see bounty programs not as a destination for their students, but as a starting point. Getting real-world, practical experience in submitting bug reports can help their students land a full-time position in computer security.

Many workers also described the nonmonetary motivations that sparked their work, describing how they weigh monetary concerns against difficulty or technical interest. Some describe hacking as an intellectual challenge, a puzzle to solve. When asked why he contributes to bug bounty programs, hacker Amat Cama stated that "it's the intellectual satisfaction about, you know—just finding a bug, and exploiting a program, and making it do something it wasn't intended to do. The puzzle-solving aspect of it, I think, is pretty satisfying."[81] For another researcher, participating in bug bounty programs allowed him to hone the craft of hacking in a "real-world environment"; by participating he is able to "hone [his] skills and also learn [by] doing things."[82]

Others find it exciting or thrilling to engage in security work. "I think what motivated me early on was the, say, challenge and excitement of finding these vulnerabilities," said hacker and college student Jack Cable.[83] Cable recalled the rush of finding a very serious flaw: "[T]hat was what made me do it at the start. It wasn't too much the money because it was just kind of crazy that people would pay me for doing that."[84] And it's not just excitement. Many hackers see security work as a valuable social contribution. One hacker described that for him, "what you're doing is for good" when you contribute to bug bounty programs.[85] Another hacker, Alyssa

[80] Consider the following training and certification courses: "Hacktify CyberSecurity," n.d., https://hacktify.in/#About; "BBE—Bug Bounty Expert Course from Hacker Associate," Hacker Associate, n.d., https://hackerassociate.com/training-and-certification/bbe-bug-bounty-expert-training/; "Top Bug Bounty Courses Online," Udemy, n.d., https://www.udemy.com/topic/bug-bounty/; "Bug Bounty Training for Beginners: How to Become a Bug Bounty Hunter," InfoSec Insights, December 8, 2020, https://sectigostore.com/blog/bug-bounty-training-for-beginners-how-to-become-a-bug-bounty-hunter/. Bugcrowd even has a "university": "Bugcrowd University," Bugcrowd n.d., https://www.bugcrowd.com/hackers/bugcrowd-university/.

[81] Interview with Amat Cama, 2019.

[82] Interview with Corben Leo, 2019.

[83] Interview with Jack Cable, 2019.

[84] Ibid.

[85] Interview with Tamir Zahavi-Brunner, 2019.

Herrera, described the satisfaction of working to protect the sensitive information of huge amounts of people. For her, the motivation to do bug bounty work is "to help further protect users and help further the standard of security."[86]

Finally, other people do bug bounty work because it provides them with a sense of community, a sense of belonging, and of being known on a small—or at times large—scale. "You know, at the end of the day, a big factor for me is networking," described one hacker, "so this is one major part of it all, I want to meet people."[87] Numerous people we spoke to run blogs or use social media to share their bug bounty work. For one researcher, the hope of his blog is "getting knowledge out there, getting recognized for that, and helping other people learn."[88]

There is a wide range of factors that motivates hackers to do this work. Some view bounties as a sideline to their main source of employment, others rely on it as a full-time job. Some hackers are motivated by money, others by reputation and fame. Others see it as a way to join a community and participate in meaningful and intellectually stimulating, challenging work. Often there is a blend of motivations that animate hackers' sense of why this work is worthwhile. Hackers participating in bug bounty programs are not, then, so different from the collection of contributors that make up large free and open source (F/OSS) projects or other crowdsourced projects: they are people with different experiences, different expectations, and different motivations.[89] Organizations that run bug bounty programs capitalize on this diverse pool, setting these different hackers against one another in competition for bugs, payouts, and prestige.[90]

[86] Interview with Alyssa Herrera, 2019.

[87] Interview with EdOverflow, 2019.

[88] Interview with Jack Cable, 2019.

[89] Weber notes that open source projects are made up of individuals with diverse and at times conflicting motivations and commitments. Steven Weber, *The Success of Open Source* (Cambridge, MA: Harvard University Press, 2004).

[90] Pitting part-time and full-time workers against one another is a staple of other gig work platforms. See Rosenblat, *Uberland*, 53. On the role that oversupply of labor plays in gig work, see Woodcock and Graham, *The Gig Economy*, 91.

Part II: Enclosing Disclosure: Turning Bugs Into Property And Hackers Into Gig Workers

Power and control have always been central to the logic of bounty programs. Despite the rhetoric of accessibility, autonomy, and choice promoted by bounty platforms (described in detail in Part III), bug bounty programs impose top-down controls over hackers and place limits on the circulation of information. Bounty programs are a type of moral ordering—they arrange hackers, code, and organizations into a set of normative relationships. Hackers are workers; bugs are property to be bought and sold; and organizations have the power to define and manage the terms that govern how hackers and flaws circulate.

This ordering is antithetical to how hacking is often conceptualized. Hacking has taken on a particular cultural resonance: it is typically seen as a set of practices that, at their core, "sit in opposition to established ways of doing."[91] Popular accounts position hackers as alternately pranksters, criminals, and technological high-priests.[92] This image—hacker as outsider and antagonist pushing against the status quo whether it be state power, corporatization, or convention—is a familiar but decidedly overstated trope. Hackers have long moved inside spaces of conventional power, creating, founding, and working for some of the largest multinational corporations in the world. To be sure, hackers have always "gone pro." These ostensible outsiders are (and have long been) working firmly inside. While hackers have historically cycled in and out of the worlds of corporate (and more recently) state power, bug bounty programs provide a distinctly different way of integrating hackers into markets and the state: they are involved not only as founders, CTOs, and senior advisors, but now also as gig workers.[93]

The remaking of bugs into property and hackers into gig workers was not inevitable. It was a historically specific reworking of how hackers and software development could—or should—interact. Bug bounty programs were, from their earliest inception, a way of transforming hacking into something that could be controlled and made compatible with then-innovative commercial software development practices. As this history makes plain, bug bounty programs reformulate hacking into a corporate- and government-friendly guise. Information is bottled up and enclosed within these programs and the terms under which some hackers work are scripted by the firms that purchase or facilitate the purchase of bugs. While bounty programs appear to offer hackers an enviable opportunity—getting paid to hack—hackers lose something significant in this translation: the power to define their working lives.

Bug bounty programs first emerged during the mid-1990s as a direct response and counter to other models of organizing flaws, hackers, and code. This was a fertile period for experimentation: free software and (what would soon become known

[91] Luca Follis and Adam Fish, *Hacking States* (Cambridge, MA: MIT Press, 2020), 7; See also Gabriella Coleman, Hacker, Hoaxer, Whistleblower, *Spy: The Many Faces of Anonymous* (New York, NY: Verso Books, 2014).

[92] Follis and Fish, *Hacking States*, 9.

[93] For a larger consideration of how hackers and hacking are adopted, transformed, and repurposed by the state and corporate actors, see Alessandro Delfanti and Johan Söderberg, "Repurposing the Hacker. Three Cycles of Recuperation in the Evolution of Hacking and Capitalism," 2018, https://escholarship.org/uc/item/9c86493g .

as) open source software (F/OSS) were jockeying for position with proprietary software development.[94] At the same time, other hackers were experimenting with different models of handling bugs, including "full disclosure"—the public release of bugs.[95] These quite different practices, F/OSS and full disclosure, shared some important similarities: they were both based on visions of distributed collaboration, self-determination (rather than top-down controls), and, above all, sharing. One of the key innovations of F/OSS approaches was an inversion of traditional notions of property: rather than thinking about software (including bugs) as something to be fenced off and made exclusive through licensing regimes and intellectual property protections, code was taken as something to be accessed, modified, and recirculated again and again.[96] F/OSS hackers created a host of practices, legal documents, and communities structured around and devoted to this understanding of software.[97] Full disclosure offered a different organization of hackers and code. In the full disclosure model, many hackers who found new flaws in software released these bugs publicly without first notifying the vendors, often in order to help improve security, either by alerting vulnerable users or urging those vendors to quickly patch buggy code.[98] Full disclosure was (and remains) a fraught process: public release carries its own security risks—malicious actors can exploit the bugs in the wild. But for proponents of full disclosure, the risks were often worth it. Like F/OSS, full disclosure prized the sharing and circulation of information under terms defined by hackers.[99]

[94] For an overview of the history of F/OSS software, see Coleman, *Coding Freedom*; Weber, *The Success of Open Source*; Christopher M. Kelty, *Two Bits: The Cultural Significance of Free Software* (Durham, NC: Duke University Press, 2008); and Christopher Tozzi, *For Fun and Profit: A History of Free and Open Source Software* (Cambridge, MA: MIT Press, 2017).

[95] For an overview of the different ways in which hackers related to vulnerabilities in the early-1990s, see Matt Goerzen and Gabriella Coleman, "Hacking Security," *Logic*, No. 10 (May 2020).

[96] On F/OSS as a challenge and reconfiguration of liberal notions of property see Coleman, *Coding Freedom*; and Weber, *The Success of Open Source*. In a related vein, Kelty notes that the ability to access and modify code and the underlying structures that govern its production, circulation, and use are at the core of the recursive publics that at once make and are made possible by free software. Kelty, *Two Bits*.

[97] Coleman, *Coding Freedom*; Kelty, *Two Bits*.

[98] Matt Goerzen and Gabriella Coleman, "Wearing Many Hats: The Rise of the Professional Security Hacker," Data & Society Research Institute, 2021 (forthcoming).

[99] This sharing was not without its share of friction. Hackers often looked to full disclosure as a way of building out their reputation and CV. Inevitably, arguments over credit bubbled up as hackers jockeyed for recognition. Goerzen and Coleman, "Wearing Many Hats"; Matt Goerzen and Gabriella Coleman, "Hacking Security," *Logic*, No. 10 (May 2020) https://logicmag.io/security/hacking-security/.

Into this mix, Netscape created the first significant bug bounty program. It adopted some of the ideas associated with F/OSS—distributed work, encouraging rather than scolding hacker—but rejected the radical notion of (communal) property that was overtly central to F/OSS. Pointedly, the company took direct aim at full disclosure. Instead of public circulation of flaws, bugs in Netscape's program were bottled up and controlled under terms defined by its purchasing program. The bug bounty model is therefore a retrenchment of an older, and all-together much more familiar idea of property: bugs were transformed into property to be bought and sold—code and labor fenced off once more through familiar notions of intellectual property and conceptions of work.

Enclosing Hacking: Free and Open Source Software, Full Disclosure, and its Discontents

In 1995, Netscape launched the first high-profile bug bounty program and lit the spark for a counterrevolution.[100] At the time, the success of Netscape was a dominant story in software. Their web browser, Netscape Navigator 1.0, accounted for an estimated 70% to 80% of the web browser market.[101] That summer, Netscape went public and stunned Wall Street with the success of their initial public offering (IPO).[102] On day one, their stock more than doubled, rising from $28 to $58.25 by market close.[103] Investors and the business press were enamored.[104]

[100] Joan E. Rigdon, "Netscape is Putting a Price on the Head of Any Big Bug Found in Web Browser," *The Wall Street Journal*, October 11, 1995; Dow Jones & Company. "Netscape Unveils 'Bounty' Program for Navigator 2.0," *Dow Jones News Service*, October 10, 1995. For an inside account of Netscape's launch and rise, see Jim Clark (with Owen Edwards), *Netscape Time: The Making of the Billion-Dollar Start-up That Took on Microsoft* (New York: St. Martin's, 1999). For a more general overview, see Brian McCullough, *How the Internet Happened: From Netscape to the iPhone* (New York: Liveright, 2018), 8-37.

[101] David A. Kaplan, "Nothing by Net," *Newsweek*, December 25, 1995; Jared Sandberg, "Netscape Acknowledges Encryption Flaw," *Wall Street Journal*, September 20, 1995; Jared Sandberg, "Sun and Netscape are Forming Alliance Against Microsoft on Internet Standard," *The Wall Street Journal*, December 4, 1995.

[102] Scott Reeves, "Netscape's IPO Sends Its Stock into Orbit and Stuns the Market," *Dow Jones News Service*, August 10, 1995.

[103] Clark, *Netscape Time*, 14.

[104] McCullough, *How the Internet Happened*, 36.

In the weeks that followed, however, bad press started to swirl: independent researchers—hackers, students, and other curious onlookers—discovered a series of significant flaws in Navigator.[105] In September, two new computer science graduate students at the University of California, Berkeley—David Wagner and Ian Goldberg—discovered a flaw that undermined the security settings of the browser, specifically the newly added Secure Sockets Layer (SSL).[106] Goldberg has since explained that he was inspired by, of all things, the then-new movie, *Hackers*, to quickly finish documenting the bug and circulate he and Wagner's findings.

At the time, Goldberg was associated with a number of hackers based in San Francisco organized around the Cypherpunks mailing list. The Cypherpunks list collected hackers, geeks, academics, Silicon Valley insiders, and others interested in computers, privacy, and the power of technology.[107] Members of this group congregated both in person and (in much larger numbers) through the alt.cypherpunks listserv. The list contained a range of political viewpoints but, as Maureen Webb's sketch of the group makes clear, the Cypherpunks tended toward libertarian views that were skeptical of government and advocated for technological privacy protections.[108] These were not just idle observers or commentators; the group was devoted to creating technical tools to protect privacy, criticizing existing techniques and software when they fell short, and, above all, sharing code. And so, when Goldberg had written up his and Wagner's crack of Netscape's SSL, he shared it with his friends on the alt.cypherpunks mailing list.[109]

Posting bugs publicly was not unheard of. Hackers within F/OSS communities publicly disclosed bugs for community projects all the time. A central aspect

[105] Susan Moran, "Netscape Security Flaw Bodes Ill for Commerce," *Reuters News*, September 19, 1995; Aaron Zitner, "Netscape Flaw Seen Setback for Business," *Boston Globe*, September 20, 1995; Kevin Maney and Robyn Meredith, "Risky Business on the Internet: Few Feel Safe Making On-Line Transactions," *USA Today*, September 20, 1995; Jared Sandberg, "Netscape Offers Reassurances on Data Safety," *The Wall Street Journal*. September 20, 1995; Jared Sandberg, "Netscape's Internet Software Contains Flaw that Jeopardizes Security of Data," *Wall Street Journal*, September 19, 1995.

[106] The account of Wagner and Goldberg's work is drawn from a research interview with Ian Goldberg (2019) and supplemented with contemporaneous sources as indicated below. See also: Zitner, "Netscape Flaw Seen Setback for Business."

[107] Maureen Webb provides an overview of the list, its members, and its shifting politics. Maureen Webb, *Coding Democracy: How Hackers are Disrupting Power, Surveillance, and Authoritarianism* (Cambridge, MA: MIT Press, 2020), 33-52.

[108] Ibid., 34-37.

[109] Ian Goldberg, "Netscape SSL Implementation cracked!" alt.cypherpunks, September 17, 1995, archived version available at: https://cypherpunks.venona.com/date/1995/09/msg01127.html.

of participating in F/OSS projects was collaborative debugging.[110] Users who discovered bugs shared them back with the larger community of users and maintainers. This was more than just something of a personal preference; for those who moved within the F/OSS communities, sharing cut to the core of what made these communities tick.[111] It was a moral imperative. F/OSS hackers were committed to creating spaces and institutions that could protect and enable ongoing sharing and modification, including identifying and disclosing bugs.

These knowledge-sharing practices seeped into the larger hacking world and started to have a direct impact on proprietary software as well. Hackers routinely discovered new flaws in proprietary software. But unlike F/OSS projects, there was often not an obvious way to disclose these bugs to the vendor. Hackers who found and reported bugs to commercial software vendors would often simply be ignored.[112] In the 1990s, some hackers embraced a controversial new way of disclosing bugs in proprietary software: "full disclosure."[113] Turning to new mailing lists like Bugtraq and online forums, hackers posted their findings for the world to see. Rather than reporting a bug privately and discreetly to the contact at a vendor, it would be released to the public.[114]

Full disclosure was an unambiguously anti-establishment act: it challenged the power of companies to control how information about their software circulated, and it smudged the lines between inside and outside, serving as a reminder that hackers were not just tinkering at the edges of proprietary software, but pushing for key changes to how this software worked.[115] At the time, there was a growing resentment and anger toward commercial software development. The public release of bugs was seen by some hackers—including many on the Cypherpunks

[110] The sharing of bugs and bug fixes plays a key role in constituting F/OSS communities. See Kelty, *Two Bits*, 128-131; 236-240.

[111] Kelty, *Two Bits*; Coleman, *Coding Freedom*.

[112] Goerzen and Coleman, "Hacking Security."

[113] On full disclosure, see Bruce Schneier, "Full Disclosure," *Crypto-Gram Newsletter*, November 15, 2001; Bruce Schneier, "Full Disclosure and the Window of Exposure," *Crypto-Gram Newsletter*, September 15, 2000; Bruce Schneier, "Recent Developments in Full Disclosure," *Schneier on Security*, December 6, 2011; Goerzen and Coleman, "Hacking Security."

[114] Goerzen and Coleman, "Wearing Many Hats"; Goerzen and Coleman, "Hacking Security."

[115] These mailing lists often served as a space where different groups—hackers, system administrators, computer scientist, and others—could share practical information and debate the ethics of disclosure (among many other topics). In the process, these mailing lists helped create new security communities that troubled easy categorization. Goerzen and Coleman, "Wearing Many Hats"; Goerzen and Coleman, "Hacking Security".

list—as a way to hold software companies accountable.[116] Mailing lists, conferences, and other venues sprang up to support and provide a platform for full disclosure. Full disclosure was a protest, a middle-finger to software companies that were often seen to be slow to respond to reports of flaws. Rather than reporting bugs and waiting and hoping that a software company might eventually fix the problem, full disclosure attempted to force software companies to react *fast*. Microsoft called full disclosure "information anarchy."[117] But for those that published bugs, full disclosure was a way to push back against closed development ecosystems, demand better products, improve privacy and security, and, crucially, inform the public—they, too, had a right to know about the flaws in the software they were using.

Goldberg's disclosure of the new SSL bug via the Cypherpunks mailing list quickly made waves. Much to Goldberg's surprise, the news spread to the pages of the New York Times just two days after his initial post on alt.cypherpunks.[118] Reporter John Markoff picked up Goldberg's initial post and wrote it up as a story—it appeared on the front page of the Times on September 19, 1995.[119] Goldberg was inundated with interview requests—*the Boston Globe, San Francisco Examiner, Kansas City Star,* NPR, CNN, and others all reached out to follow up and report on the story.[120] Throughout, Goldberg tried to stress the importance of opening up Netscape's code and allowing independent researchers to review and audit their security features—a common concern of the Cypherpunks and a core tenet of the F/OSS community.[121] A day after the *Times* story, Goldberg followed up with another post to alt.cypherpunks. He reviewed the media scrum and the round of interviews that he and Wagner had conducted in the past 24 hours. This time,

[116] This is the tenor of the exchanges that surrounded Goldberg's post on alt.cypherpunks. For example, Chris Wysopal, writing as "Weld Pond," noted roughly a week earlier, in relation to a different disclosure, how postings on the list were driving press attention and, as a result, forcing Netscape to take security more seriously. As he wrote, "The Cypherpunks forced a situation where Net users now have better encryption available to them. I'd say this is a big win." See Weld Pont, "Netscape to patch shareware version," alt.cyperhpunks, September 12, 1995, https://cypherpunks.venona.com/date/1995/09/msg00711.html.

[117] Kevin Poulsen, "Microsoft Reveals Anti-Disclosure Plan," *Security Focus*, November 9, 2001, https://web.archive.org/web/20210206004122/http://www.securityfocus.com/news/281; Scott Culp, "It's Time to End Information Anarchy," *Tech Net*, October 2001, http://www.angelfire.com/ky/microsfot/timeToEnd.html.

[118] Aaron Zitner, "Netscape Flaw Seen Setback for Business," *Boston Globe*, September 20, 1995.

[119] John Markoff, "Security Flaw is Discovered in Software Used in Shopping," *New York Times*, September 19, 1995, A1.

[120] Ian Goldberg, "My Day," alt.cyperhpunks. September 19, 1995, https://cypherpunks.venona.com/date/1995/09/msg01350.html.

[121] Ibid.

Goldberg added a cheeky new addendum to his signature line. Under his name he added a new quote: “So how _did_ Netscape’s stock do today?’”[122]

Netscape pledged to fix the flaw as soon as possible. The public disclosure pressed them into quick action. But the bad news did not stop. Days later yet another Netscape bug was reported via the Cypherpunks list. This time, the *Wall Street Journal* reported on the new flaw, yet again calling into question the integrity of Netscape’s browser.[123] As reports about bugs stacked up, the press openly wondered if the ongoing disclosures of flaws was a sign that Netscape was perhaps destined to fail.

Netscape had to find a way to reconcile their business model with the ongoing disclosure of flaws. They had captured a large share of the browser market, and become a start-up success, by working *fast*.[124] Netscape’s key innovation was speed. As Netscape co-founder Jim Clark argued, “[t]empo, as much as technology, was what Netscape was all about.”[125] They shrunk product cycles, cutting development times down to a then-unheard-of six months, and continually released (and promised to release) new updates and versions, and new features, faster than their competition.[126] Working fast allowed Netscape to capture market share. Working fast allowed them to introduce new features before the competition—namely Microsoft and its then-new browser, Internet Explorer—could catch up.[127] But, working fast also meant that the software would be buggy. As Clark reflected, “at some point you have to say, ‘Let’s stop messing with this! It’s good to go—or good enough.’ If you don’t, you’ll never ship a product. And somebody else sure as hell will.”[128]

Netscape thrived on quick and rapid iteration—a sort of perpetual beta where new versions and updates follow one after another—but that meant that “good

[122] Ibid.

[123] Jared Sandberg, “Netscape Software for Cursing the Internet is Found to Have Another Flaw,” *The Wall Street Journal*, September 25, 1995.

[124] Jim Clark, *Netscape Time*.

[125] Ibid. 206.

[126] Ibid., 60-69, 110, 206.

[127] Mark Tran, “Hacker Takes the Gloss off Netscape’s Floatation Success.” *The Guardian*, Aug. 18, 1995; Molly Baker, “Technology Investors Fall Head Over Heels for Their New Love,” *The Wall Street Journal*, Aug. 10 1995. See also: Clark, *Netscape Time*.

[128] Jim Clark, *Netscape Time*, 153.

enough" code would have to replace "perfect."[129] In this way, bugs were a part of the Netscape's development model. They were part of the cost of moving quickly. But now, in the fall of 1995, press reports charging that the new company's software was defective were becoming a significant problem. They were causing reputational harm. Netscape was gearing up to release its hotly anticipated new browser, Navigator 2.0. It would inevitably contain its share of flaws—and that meant that more bad PR and eroding confidence would likely follow.

Netscape needed a way to balance its rapid development model with the realities of buggy code and negative press attention. On Oct. 10, just a little over two weeks after Goldberg first posted his discovery on alt.cypherpunks, Netscape announced a new strategy: a "bugs bounty" program.[130] Netscape personnel announced that they would buy flaws from researchers who found new bugs in Navigator 2.0 if they reported them directly to Netscape. Participants would earn cash and swag in exchange for vulnerabilities.[131] In the press release announcing the program, Netscape drew directly from the rhetoric of F/OSS, declaring that the new program would harness the "power of the Internet" in order to help improve Navigator.[132] The program would, in the words of Mike Homer, Netscape vice president for marketing, "encourage an extensive, open review of Netscape Navigator 2.0 and… help us to continue to create products of the highest quality."[133] This idea—of open access and distributed participation in development and review—was central to F/OSS. Netscape hoped that the bugs bounty program would funnel bugs away from public disclosure and instead encourage hackers to report them directly—and exclusively—to Netscape. In order to qualify for a bounty award, new flaws had to be reported to Netscape through their bounty program, and not released to the public. The "bugs bounty" program was a novel solution: it would allow Netscape to continue to ship its products with its fair share of flaws, while blunting the negative attention that followed each and every disclosure of a new flaw. As we shall see below, this program would, in effect, not only provide hackers a way to report bugs and improve security, it would also buy silence.

[129] Ibid, 63-64, 153.

[130] Rigdon, "Netscape is Putting a Price on the Head of Any Big Bug Found in Web Browser"; Netscape, "Netscape Announces 'Netscape Bugs Bounty' with Release of Netscape Navigator 2.0 Beta," *PR Newswire*, October 10, 1995.

[131] Ibid.

[132] Netscape, "Netscape Announces 'Netscape Bugs Bounty'."

[133] Italics added, quoted in Netscape, "Netscape Announces 'Netscape Bugs Bounty'."

A Market for Secrets: Hackers, Bugs, and Enclosure

Netscape's story underlines one of the key logics present during the earliest history of bug bounty programs, a logic that continues today: bug bounties offer a way to counter full disclosure, mitigating the costs and unpredictability of the public disclosure of bugs. Put into this context, bug bounty programs are not only a way to invite hackers in, they are also a way of controlling hackers and the flow of bugs—functioning as a type of enclosure.[134] They take an activity previously outside of market relations—the discovery and disclosure of bugs by hackers—and pull it into the logic of the market. Enclosure imposes new rules, structure, and governing logic. Bug bounty programs turn bugs into property. In doing so, they utilize the involvement of crowdsourced labor that is familiar to F/OSS projects but undo the clever inversions to property that F/OSS enacted, and they undermine the public release of bugs that was central to full disclosure. There was an imperative across both of these communities and practices to share new discoveries with the larger community of developers and users. Bug bounty programs, from their very beginning, short-circuited this process: once bugs were turned into property, they were fenced off and circulation was limited—firms, not hackers, controlled how, when, or even if bugs would be made public.

F/OSS and full disclosure each suggested a different ordering of how hackers, code, and organizations might relate to one another. Within these movements and communities, hackers had the power to shape and define the context within which they worked and through which the artifacts they produced circulated. As Gabriella Coleman and Chris Kelty argue in their respective studies of F/OSS communities, these groups collectively worked to shape, define, and refine the conditions under which they accessed technical knowledge.[135] It was these qualities—a conceptualization of rights based on distribution and not exclusion, and commitment to self-determination—that led observers like Steven Weber to

[134] Enclosure has long been used as an analytic lens through which to view the transformation and commodification of digital spaces, practices, and artifacts. See generally, James Boyle, "The Second Enclosure Movement and the Construction of the Public Domain," 2003, https://web.law.duke.edu/pd/papers/boyle.pdf.

[135] Coleman, *Coding Freedom*; Kelty, *Two Bits*.

call F/OSS perhaps “the first and certainly one of the most prominent indigenous political statements of the digital world.”[136] This is what made them interesting and even radical.

> Bug bounty programs, from their very beginning, short-circuited this process: once bugs were turned into property, they were fenced off and circulation was limited—firms, not hackers, controlled how, when, or even if bugs would be made public.

Netscape started something of a counterrevolution. Bug bounty programs allowed for some degree of distributed work, but it stripped out what made F/OSS unique and undermined the political power of full disclosure as a tool of naming and shaming. F/OSS, full disclosure, and bug bounty programs co-existed (and to this day, continue to co-exist) and battled for position. They each presented different visions about how hackers, bugs, and organizations can and should relate. Hackers working within bug bounty programs get paid, but they lose some ability to shape and define the terms under which they work. Within bug bounty programs, hunting for bugs takes on a new set of meanings for those doing it; it is governed by a new set of formal rules guided by assumptions concerning private property; and it remakes the relationship between hackers and software companies. The transformation of bugs into property to be bought and sold tilted power away from hackers and back toward commercial software companies.

Today, bug bounty programs continue to limit public disclosure. When selling a bug, organizations determine how and when information about that bug is or is not released. Hackers must agree not to disclose the details of what they have uncovered until the purchasing program authorizes them to do so—typically after the bug has been fixed and the bounty has been paid. Some programs require nondisclosure agreements (NDAs) in conjunction with bounty submissions. As Sean Roesner, a UK-based bounty veteran, noted, NDAs are typical for all private bounty programs.[137] As Roesner reported, speaking out carries a cost: if you

[136] Weber, *The Success of Open Source*, 7.

[137] Interview with Sean Roesner, 2019.

speak out publicly "you get banned from their program platform… [and they]… sanction you."[138] Public programs, as Jack Cable noted, might not always require NDAs, but they do work to make sure that researchers do not release their bugs publicly before the issue has been fixed.[139] As Cable remarked, "as a baseline, every program will say, 'Don't disclose until it's been fixed'." The programs make it clear to researchers that releasing their bugs publicly will have consequences. As Cable recounted, bounty programs will tell researchers that "'[i]f you [publicly release a bug], you could be removed from our program and… not allowed to submit new reports,' or, 'we reserve the right not to give you a bounty if you don't follow these rules.'"[140] These sorts of threats carry weight. Even if they are not legally binding, like an NDA, they significantly shape researchers' experiences of this market and their decision to work within it.

In more extreme cases, bug bounty programs can be used as a tool to try to prevent disclosure entirely. In 2016, Uber attempted to use its bug bounty program (run through HackerOne's platform) to avoid its legal obligations by concealing a data breach.[141] It paid a pair of hackers $100,000 via bitcoin and provided NDAs that would buy their silence. The secretive scheme, however, was eventually uncovered and led to significant legal penalties for both Uber and the hackers. The following year, DJI, manufacturer of a line of commercial drones, hosted a bug bounty program that also became embroiled in controversy when it appeared the program was being used to hide rather than fix flaws. When a hacker uncovered a potentially serious vulnerability in DJI's products, they were offered a $30,000 bounty—but with a significant series of caveats: they would have to agree to obtain written consent from DJI when disclosing any additional security issues related to their product; they had to agree to not make any "misleading" statements about DJI; and they would have to refrain from "probing" for any additional security issues. When the hacker balked—they saw the terms as an affront to their freedom of speech and their livelihood as a security researcher—

[138] Ibid.

[139] Interview with Jack Cable, 2019.

[140] Ibid.

[141] U.S. Attorney's Office, Northern District of California, "Former Chief Security Officer For Uber Charged With Obstruction Of Justice," August 20, 2020, https://www.justice.gov/usao-ndca/pr/former-chief-security-officer-uber-charged-obstruction-justice.

they faced a more unpleasant offer from DJI: a legal threat.[142] More recently, John Deere, the tractor and farm equipment company, attempted to use NDAs to silence security researchers. A hacker working under the handle "Sick Codes" found flaws in John Deere's app and website that would allow an attacker to access private customer data.[143] The hacker attempted to work with John Deere to publicly release the information after the bugs had been fixed.[144] However, John Deere instead invited the hacker to a hastily created private disclosure program—a program that had only one member: the hacker in question.[145] Under the terms of the program (which was hosted on an unnamed platform), the hacker would have been subject to an NDA, and public release of their work would have been blocked. Sick Codes refused the invitation and went public, bucking the effort to use an NDA to control their work.[146] These three extreme examples point out efforts to use bounty programs (or a private disclosure program) as a type of "catch-and-kill"—a way of trying to bottle up potentially embarrassing or costly information behind legal terms. These efforts are outliers, these are not typical examples of how bug bounty programs work. But they underscore some of the larger power dynamics that follow the transformation of bugs into property.

Enclosure is a complex process. Organizations gain a lot. Disclosure becomes predictable. Hackers gain something too: they are given clear paths to report bugs and they can get recognition for their work. But hackers give up something during this transformation—going pro always has its costs. Hackers working within the market give up some power; full disclosure was—and remains—a powerful tool. Forgoing the ability to circulate knowledge about new, novel hacks, has also forfeited or threatened one of the traditional ways in which hacker communities were built and sustained.

[142] Chris Bing, "How DJI Fumbled Its Bug Bounty Program and Created a PR Nightmare," CyberScoop, November 30, 2017, https://www.cyberscoop.com/dji-bug-bounty-drone-technology-sean-melia-kevin-finisterre/; Sean Gallagher, "Man Gets Threats—Not Bug Bounty—after Finding DJI Customer Data in Public View," Ars Technica, November 17, 2017, https://arstechnica.com/information-technology/2017/11/dji-left-private-keys-for-ssl-cloud-storage-in-public-view-and-exposed-customers/.

[143] Lorenzo Franceschi-Bicchierai, "Bugs Allowed Hackers to Dox John Deere Tractor Owners," April 22, 2021, https://www.vice.com/en/article/4avy8j/bugs-allowed-hackers-to-dox-all-john-deere-owners.

[144] Kevin Kenney and Willie Cade, "Leaky John Deere API's: Serious Food Supply Chain Vulnerabilities Discovered by Sick Codes, Kevin Kenney & Willie Cade." Sick.Codes, April 22, 2021, https://sick.codes/leaky-john-deere-apis-serious-food-supply-chain-vulnerabilities-discovered-by-sick-codes-kevin-kenney-willie-cade/.

[145] Ibid.

[146] Ibid.

‘No More Free Bugs’: Bug Bounty Programs Go Mainstream

Netscape’s experiment was, in their eyes, a success, allowing them to improve their software, maintain their business model, and reverse the flow of negative PR started by the public disclosure of bugs by Goldberg, Wagner, and others. But for many years, their bug bounty program remained an idiosyncratic PR stunt more than the norm of software security. Between 1995 and the early 2000s, the computer security industry lurched through a period of change. Software companies both large and small were not necessarily eager to change their practices and were often openly suspicious of hackers who came to them with security flaws. But as the importance of network technologies grew, and the instances of high-profile and significant security flaws increased, a number of hackers became convinced of the monetary value of their work—and companies started to agree.

Netscape’s bounty program remained an outlier—an important, but still relatively unique case into the early 2000s. Tom Anthony, a hacker and web developer with 20 years of experience, recalled that it took some time for bug bounties to catch on.[147] In the early 2000s, Anthony was using the UK’s second largest domain name provider website when he discovered that one of their webpages had listed domain names twice. Through a trivial error, he was able to take advantage of this flaw to gain control of anyone’s domain, and thought to himself, “Okay, this is big.”[148] He emailed the company and described this serious flaw. They emailed him back, telling him why he was wrong, and he emailed back with screenshots. “They never replied to me, unfortunately. They never replied after that… and I was sort of young, I was in university, I didn’t really realize the seriousness or, if you’re an evil person, the market value of what I had. So I just reported it naively and never heard back.”[149] Stories like these were precisely what the Cypherpunks and full disclosure advocates had railed against a decade earlier: it was as if nothing had changed.

[147] Interview with Tom Anthony, 2019.

[148] Ibid.

[149] Ibid.

Tom's story is not unique. Veteran hacker Cesar Cerrudo described the tactics that software vendors use when they lack a pipeline system for the disclosure of security flaws from outside the company: "When [software vendors] are not mature, they won't answer, or they will answer whenever they want. Sometimes you have to explain to them [the core issues]. Most of the time, they think that you are trying to blackmail them."[150] In his eyes, many software vendors seem to think that what they don't want to know won't hurt them, in order to avoid the bad press and downplay the impact of the vulnerability.

For hackers on the outside looking in, it was still often hard to get their work noticed or acknowledged. Years after Netscape launched its bug bounty program, many companies were still more likely to ignore hackers that had found a new bug than to pay them.

Things would change. Throughout the 2000s, the computer security industry flourished. The dot-com bubble unleashed a flood of money for security companies.[151] Hackers were in demand. Companies like @stake found success by embracing hackers, acquiring the high-profile hacking group L0pht in 2000. Buqtraq—the venerable full disclosure mailing list—was purchased by Symantec in 2002.[152] Elsewhere, companies were ramping up their spending on security, hiring new in-house talent and signing penetration testing contracts with up-and-coming security firms. Companies like iDefense and ZDI looked to hackers as a strategic resource, buying up new bugs in order to boost their managed security services.[153] Hackers were not seen as a liability, but as an asset. The emphasis on security post-9/11, further ramped up interest and spending on security.[154] The market for computer security was awash in cash, but bug bounty programs were still exceedingly rare.

[150] Interview with Cesar Cerrudo, 2019.

[151] Rebecca Slayton and Brian Clarke, "Trusting Infrastructure: The Emergence of Computer Security Incident Response, 1989-2005," *Technology and Culture* 61, no. 1 (2020): 173-206, https://doi.org/10.1353/tech.2020.0036.

[152] Goerzen and Coleman, "Wearing Many Hats."

[153] For an overview of this period, see Nicole Perlroth, *This is How They Tell Me the World Ends: The Cyber-Weapons Arms Race* (New York: Bloomsbury, 2021).

[154] Perlroth, *This is How They Tell Me the World Ends; Ryan Ellis, Letters, Power Lines, and Other Dangerous Things: The Politics of Infrastructure Security* (Cambridge: MIT Press, 2020).

In 2009, three well-known hackers—Dino Dai Zovi, Charlie Miller, and Alex Sotirov—appeared on stage at an annual security conference, CanSecWest, holding a crude sign.[155] Scrawled across the improvised cardboard placard in block letters was a new mantra: "no more free bugs."[156] For decades, researchers had been identifying and disclosing previously unknown flaws in commercial software and hardware. They were rarely, if ever, paid for their work. As Dai Zovi noted at the time, "reporting vulnerabilities for free without any legal agreements in place is risky volunteer work."[157] In their view, vendors had been "freeloading" off security research for too long or, worse, using legal threats to silence researchers.[158]

Behind the protest stood a more worrisome reality: if companies weren't going to pay for bugs in their systems, someone else certainly would.[159] At this point, an offensive market for bugs was moving out of the shadows. Nation states and their intermediaries were eager to buy previously unknown and undisclosed bugs—zero-days—not to fix them, but instead to use them to craft new exploits and attacks. The growing offensive trade put large software companies and users at risk of being exploited and harmed. Exploits and attacks built on otherwise unknown bugs are difficult to stop. What's more, the offensive market seemed poised to draw the smart hackers that had long been providing this "risky volunteer work" for tech companies away—the Millers, Dai Zovis, and Sotirovs of the world—and entice them to taking their talents and their bugs elsewhere.

[155] Dennis Fisher, "No More Free Bugs for Software Vendors," *Threat Post*, March 23, 2009, https://threatpost.com/no-more-free-bugs-software-vendors-032309/72484/; Security Focus, "No More Bugs for Free, Researchers Say," *Security Focus*, March 24, 2009, https://web.archive.org/web/20210309004144/http://www.securityfocus.com/brief/933 .

[156] Fisher, "No More Free Bugs for Software Vendors."

[157] Dino Dai Zovi, "No More Free Bugs," *Trail of Bits Blog* March 22, 2009, archived version available online: https://web.archive.org/web/20160403215414/http://blog.trailofbits.com/2009/03/22/no-more-free-bugs/.

[158] Ibid.

[159] For an overview of the offensive market and its relation to bug bounty programs, see Perlroth, *This is How They Tell Me the World Ends*.

Miller, Dai Zovi, and Sotirov were clear: if vendors wanted access to their bugs, they were going to have to pay. They viewed hunting for bugs as labor with value that should be compensated. They hoped that the commercialization of vulnerability disclosure, a defensive market, would provide a measure of stability—that it would provide legal protection, recognition, and fair compensation for important work.

Google agreed. In January, 2010, Google piloted its first bug bounty program, a rewards program for contributors to the open source Chromium project.[160] At launch, Google's security team framed these rewards as a way of providing a token of appreciation to those who contributed novel bugs and a way to encourage new researchers to join the project.[161] In November that year, Google built off the success of the initial program and rolled out a bounty program for its web properties.[162] The following year, in the summer of 2011, Facebook announced its own bug bounty program.[163]

> Hackers got paid—some even got paid seemingly fantastic amounts—but the institutionalization of bug bounty programs created new, sometimes unanticipated, risks for workers, organizations, and society.

Netscape's PR move was no longer a one-off, it was becoming institutionalized. These programs embraced the logic that animated Netscape's initial experiment more than 15 years earlier. In order to be eligible for payments, hackers had to, in many cases, agree not to first release their bugs publicly. But these programs were, explicitly or implicitly, responding to the arguments presented in the protest by Miller, Dai Zovi, and Sotirov. Bounty programs offered a way to encourage and engage with hackers that was far more solicitous than what had all too often

[160] "Encouraging More Chromium Security Research," Chromium Blog, January 28, 2010. https://blog.chromium.org/2010/01/encouraging-more-chromium-security.html.

[161] Ibid.

[162] Chris Evans, Neel Mehta, Adam Mein, Matt Morre, and Michal Zalewski, "Rewarding Web Application Security Research," Google Security Blog, November 1, 2010, https://security.googleblog.com/2010/11/rewarding-web-application-security.html.

[163] Chris Brook, "Facebook Launches Bug Bounty Program," *Threat Post*, August 1, 2011, https://threatpost.com/facebook-launches-bug-bounty-program-080111/75500/.

come before. Payments would now replace sternly worded legal threats.[164] These programs sought to provide a clear pathway for hackers to report their bugs, and would provide compensation—even if a nominal amount—for their work. Bugs, as Dai Zovi, Miller, and Sotirov had hoped, would no longer be free.

In the coming years, however, bug bounty programs would be adopted not just by technology companies, but broadly. United Airlines, the Department of Defense, Starbucks—the list grows daily—would, with great fanfare, launch bounty programs. They soon became a familiar security feature for many organizations. Yet this seemingly new golden age of hacking did not turn out like Dai Zovi, Miller, and Sotirov—or even Google, Facebook, and Microsoft—might have hoped. Hackers got paid—some even got paid seemingly fantastic amounts—but the institutionalization of bug bounty programs created new, sometimes unanticipated, risks for workers, organizations, and society.

Turning Hacking into Gig Work: Platforms and the Institutionalization of Bug Bounty Programs

Google and Facebook's resurrection of bug bounties was quickly followed by the rise of bug bounty platforms. The incorporation of Bugcrowd in 2011 and HackerOne in 2012 not only popularized bug bounty programs, but it also transformed them in important and unanticipated ways. Bounty programs outgrew their origins as tools to manage short-run PR snafus and provide protections and recognition for hackers. Instead, they turned into platforms for recruiting, coordinating, and managing labor on a vast scale. These start-ups successfully provided the tools, expertise, technology, and marketing power to help diffuse the bug bounty model beyond select tech companies out into hundreds of different organizations and firms.

[164] The adoption of bounty programs by tech companies during this period, namely Google. Facebook, and Microsoft, was animated by a number of cross-cutting factors, including the ongoing rise of a new cohort inside these companies that agitated for direct engagement with hackers, increasing fears regarding the spread of malicious intrusions and attacks, and antipathy toward what appeared to be an "offensive" market for bugs. See Perlroth, *This is How They Tell Me the World Ends*; Goerzen and Coleman, "Wearing Many Hats."

The platforms drew personnel and expertise from Facebook, Google, and Microsoft (a late but critical addition to bug bounties). Security experts at these companies advised, managed, and, in some cases, ran these new venture capital-backed bug bounty start-ups. Eventually, these platforms spun bug bounty programs off in a different direction: they sold bounty programs to clients as a way to slash costs and manage workers.

> Bounty programs outgrew their origins as tools to manage short-run PR snafus and provide protections and recognition for hackers. Instead, they turned into platforms for recruiting, coordinating, and managing labor on a vast scale.

This was an important shift. Netscape, and, for that matter, Google, Facebook and other initial adopters, used bounties as a way to reward and manage hackers who happened to find bugs in their products. Now, bounty platforms pitched bug bounty programs as a way to limit disclosure as well as outsource and replace *existing* work and workers. They positioned and sold bug bounty programs to companies and public sector organizations as a way to cut labor costs, while maintaining tight control over hackers and the flow of information. Beneath the rhetoric of the "wisdom of the crowd" was a more direct pitch: bug bounty programs can save you money by outsourcing your security work. This was not what Dai Zovi, Miller, or Sotirov necessarily had in mind when they called for "no more free bugs." And it wasn't how Google, Facebook, and other tech companies initially deployed bug bounty programs for that matter. Bug bounty platforms changed the game: they turned security hacking into gig work.

The institutionalization of bug bounty programs, like gig work more generally, is a part of a larger shift in employment and work. Since the 1970s, firms across all industries have turned away from standard forms of employment—jobs that include stable contracts, fixed hours, guaranteed pay, and benefits—and embraced part-time and temporary work as an alternative.[165] These changes are driven by a number of factors, but the possible benefit for companies is clear: casual workers are cheaper than full-time employees (since they do not accrue benefits like healthcare and paid leave) and they are easier to add or subtract from employment rolls (since they typically do not have access to job security or traditional occupational benefits).[166] Rather than directly employing workers, firms now rely on a network of highly competitive subcontractors, sub-subcontractors, franchises, and other arms-length entities to provide services that used to be provided in-house.[167]

These changes cut costs and, in the process, put significant downward pressure on worker wages, benefits, and legal protections.[168] The stakes for workers are clear. When protective labor regulations are insufficient, workers may have the obligations—but not the rights—associated with employees (e.g., accessing labor protection reserved for employees, accessing health benefits, receiving the amount of compensation that an employee would receive).

Outsourcing is a familiar aspect of the computer software and hardware industries—it long pre-dates bug bounty programs. Apple, to take but one example, has long relied on subcontractors to manufacture, assemble, and distribute its products. For example, while Apple directly employees some 60,000 odd workers, it relies on a vast network of 750,000 workers tied to various subcontractors.[169] Computer programming and other IT tasks have frequently been outsourced to offshore subcontractors. In the mid-1980s, Texas Instruments leased a data connection between the US and Bangalore, India, in order to hire Indian programmers for

[165] See David Weil, *The Fissured Workplace: Why Work Became So Bad for So Many and What Can Be Done to Improve It* (Cambridge, MA: Harvard University Press, 2014); Arne L. Kalleberg, *Precarious Lives: Job Insecurity and Well-Being in Rich Democracies* (Medford, MA: Polity, 2018); Ursula Huws, *Labor in the Digital Economy: The Cybertariat Comes of Age* (New York: Monthly Review Press, 2014).

[166] See, generally: Leah F. Vosko, *Temporary work: the gendered rise of a precarious employment relationship* (Toronto: University of Toronto Press, 2000); and Jamie Peck, Nik Theodore, and Kevin Ward, “Constructing markets for temporary labour: employment liberalization and the internationalization of the staffing industry,” *Global Networks 5*, no. 1 (2005): 3-26, doi: 10.1111/j.1471-0374.2005.00105.x.

[167] Weil, *The Fissured Workplace*.

[168] Ibid.

[169] Ibid., 51.

software projects.[170] This type of "body shopping" or "virtual migration" had long been a staple of programming and IT-related projects.[171]

The origin of bug bounties in the crucible of F/OSS software and Netscape press releases doesn't necessarily seem like part of the history of outsourced labor at first blush, but the later popularization of bug bounty programs as a way of conditioning labor reveals just how comfortably they can fit together. Testing software is not simple or cheap. In the early 1990s, it was estimated that roughly half of the labor devoted to developing a working program was spent on testing.[172] A decade later, little had changed. Testing remained resource intensive. By 2002, estimates suggested that debugging, testing, and program verification accounted for between 50% to 75% of total development costs.[173] A substantial market for security testing services, pen test companies, in particular, grew to complement and support in-house testing.[174] Bounty platforms began promoting bug bounties as a cheaper alternative to either in-house work or expensive pen test contracts. Bounty programs manage to cut costs—or promise to cut costs—in two familiar ways: they tap into a global workforce of young and inexpensive labor (see part I); and they replace guaranteed pay and benefits with contingent work. The internationalization of the workforce replaces workers based in higher-wage countries with comparatively lower-cost workers based in countries with a comparatively lower standard of living. At the same time, bounty programs slash benefit costs. Unlike fulltime employees, participants in bug bounty programs do not get paid leave, healthcare, or other guaranteed benefits.[175] Additionally, bug bounty programs rely on a significant amount of

[170] Nick Dyer-Witheford, *Cyber-Proletariat: Global Labour in the Digital Vortex* (Toronto: Between the Lines, 2015), 74.

[171] A. Aneesh, *Virtual Migration: The Programming of Globalization,* (Durham, NC: Duke University Press, 2006); Amrute, *Encoding Race, Encoding Class*.

[172] Gregory Tassey, "The Economic Impacts of Inadequate Infrastructure for Software Testing," Planning Report (National Institute of Standards and Technology, Program Office Strategic Planning and Economic Analysis Group, May 2002), https://www.nist.gov/system/files/documents/director/planning/report02-3.pdf.

[173] Ibid.

[174] Jeff Stone, "HackerOne Thinks its Freelance Hackers Can Conduct Penetration Tests Better than Actual Pentesting Companies," *Cyberscoop*, March 1, 2019, https://www.cyberscoop.com/hackerone-penetration-testing/.

[175] The employment status of gig workers is a recurring source of tension. See: Rosenblat, *Uberland*, 8-9, 156; Woodcock and Graham, *The Gig Economy*; Srnicek, *Platform Capitalism*, 75-88.

uncompensated work (more on this below).[176] Unlike fulltime, in-house employees or pen testers working on contract, hackers contributing to bug bounty programs are contingent workers: they are not paid a predictable wage or salary. Hackers are only paid when—and if—they are the first to find and report a new bug.

> Bug bounty programs buy silence, causing hackers to lose the power to define aspects of their hacking and working lives.

The popularization of bug bounty programs appears to promise a brand-new era of security research and hacking: companies and the government not only tolerate hacking, but now actually encourage and pay hackers. But a closer inspection uncovers a complicated dynamic. Hackers lose something significant in this translation. Bug bounty programs buy silence, causing hackers to lose the power to define aspects of their hacking and working lives. Turning hacking into gig work pulls hackers into a tightly ordered and organized world, where bounty platforms and firms have significant power to dictate the rules that govern their work—and hackers take on significant risks.[177]

[176] Gig work platforms often rely on a significant amount of uncompensated work. For example, see: Ravenelle, *Hustle and Gig*; Rosenblat, *Uberland*.

[177] For Woodcock and Graham, this shift—shifting risks from firms and organizations onto individual workers— defines the gig economy. As they argue, “what all gig economy models have in common is a defining logic that seeks to shift maximal risk and minimal reward onto workers.” Woodcock and Graham, *The Gig Economy*, 141. This transformation is intensified by the growth of new digital platforms, but it has roots in older transformations of capital, the state, and power. See: Ulrich Beck, *Risk Society: Towards a New Modernity*, translated by Mark Ritter (Thousand Oaks, CA: Sage, 1992).

Part III: Power, Risk, And Bug Bounties

Markets are never neutral: they always organize workers, firms, and technologies into particular contingent relationships.[178] Bug bounty programs arrange hackers, vendors, and other central players into a hierarchy. There are real benefits to these programs: bounty programs provide hackers with a pathway to report flaws and earn recognition and income. In an ideal world, bug bounty platforms can incentivize organizations to take seriously vulnerability disclosure, rather than threaten hackers with lawsuits or criminal action. But power is not divided equally. These bounty platforms and the firms that buy bugs have significant power to shape and define this market: they set prices; they act as gatekeepers to private programs and lucrative live events, and they shape the legal risks that hackers face. This power creates tension, hazards, and frustration for workers. It pushes hackers into undertaking significant amounts of uncompensated work; it sometimes places them into legally precarious situations; and it denies hackers a meaningful say over the conditions within which they work. The bundle of promises that bounty programs hold out to workers—low barriers to entry, flexibility, community, and wealth—are qualified by the reality of the market.

[178] Neil Fligstein, *The Architecture of Markets: An Economic Sociology of Twenty-First-Century Capitalist Societies.* (Princeton, NJ: Princeton University Press, 2001); Biao Xiang, *Global "Body Shopping": An Indian Labor System in the Information Technology Industry* (Princeton, NJ: Princeton University Press, 2007).

Hackers confront an economic landscape that is familiar to other gig workers: promises of freedom and autonomy complicated by lopsided power dynamics. That's not to say that hackers do not find pleasure or success in the market—many do. Hackers are not powerless. They navigate this market through a number of useful ad hoc strategies, carving out autonomy and pleasure in the market that, in some ways, is stacked against them. But even for those who have "made it" and report significant benefits from working for bounty programs, their stories pinpoint recurring tensions. Through ad hoc strategies, hackers find ways to challenge and contest the power of firms and organizations that hold significant power over the market for bugs. But for now, many hackers must fend for themselves as individuals within the asymmetrical order of power set out by organizations that run bug bounty programs as well as bounty platforms.

Platforms and (Moral) Visions of Hacking

For most hackers, contributing to bug bounty programs means working with or through a large bounty platform. These platforms connect a mass of workers with companies and organizations that are interested in purchasing bugs. These platforms, like all platforms, are not passive intermediaries: they are powerful players.[179]

Bug bounty platforms proselytize for the benefits of bug bounties and hackers. They do more than provide administrative and technical support for bounty programs—they actively promote a particular image of hackers and bounty programs, an image that conceals as much as it reveals. In the past several years, bug bounty platforms have undertaken extensive marketing campaigns to attract both new clients and an increasingly large pool of hackers. To potential corporate and government clients, platforms advertise hackers as an innovative, agile, and above all low-cost alternative to expensive security workers.[180] Their marketing materials highlight testimonials from key clients, including Motorola,

[179] Despite their protest to the contrary, platforms are always political. See: Tarleton Gillespie, *Custodians of the Internet: Platforms, Content Moderation, and the Hidden Decisions that Shape Social Media* (New Haven: Yale University Press, 2018); Srnicek, *Platform Capitalism*, 47.

[180] HackerOne, *Hacker Powered Security Report: 2019*, 8.

PayPal, Dropbox, Goldman Sachs and others.[181] These accounts recount how these organizations have used hackers to improve their security and their bottom line. The reports are dotted with comments from CEOs, government officials, and others who repeat over and over a simple point: bug bounty programs provide "huge value at a fraction of the cost."[182]

> Hackers confront an economic landscape that is familiar to other gig workers: promises of freedom and autonomy complicated by lopsided power dynamics.

These documents do more, however, than make an economic case for bug bounty programs. They also work to present hacking—and hackers—as something that is manageable, contained, and secure. The marketing materials directly counter the notion of the hacker as outlaw or criminal. The testimonials from Fortune 500 companies and government agencies indirectly make this point, but bounty platforms are more direct as well. In the opening pages of their 2020 annual report, HackerOne begins with a splashy layout featuring headshots of eight smiling hackers.[183] Offset next to their portraits in large block letters is a definition stylized as if to appear straight from the Oxford English Dictionary: "HACK'ER /'ha-ker/ noun. One who enjoys the intellectual challenge of creatively overcoming limitations."[184] The next page presents a self-conscious history of hacking, stating that hacking started out as something "in the darkest underbelly of the internet" only to transition first into a "respectable hobby" and now evolve into a "professional calling." This framing of hacking as something that was once unruly but has now been made respectable is telling. In a decidedly unsubtle way, it is broadcasting to potential clients that hackers (and by extension, bug bounty programs) are nothing to fear. This point is implicitly and explicitly threaded throughout the marketing literature. In each passage that quotes a prominent CEO or government figure, in every section that

[181] Bugcrowd, *Priority One: The State of Crowdsourced Security in 2019*, n.d., 9, https://web.archive.org/web/20200108143022/https://www.bugcrowd.com/resources/reports/priority-one-report/; HackerOne, *Hacker Powered Security Report: 2019*, 27, 30; HackerOne, *The 2020 Hacker Report*, 50.

[182] Amos Elliston, quoted in HackerOne, *Hacker Powered Security Report: 2019*, 43.

[183] HackerOne, *The 2020 Hacker Report: 2020*, 2.

[184] Ibid.

notes that bounty programs are on the rise in what are described as "risk-averse" industries, such as financial services, banking, healthcare, and education, and in the collection of sunny pictures of young, smiling hackers that invariably fill these reports, the same point is made plain: hackers are here to help.[185]

The marketing materials not only target potential clients, they also speak to hackers—constantly trying to recruit new hackers to sign up to work. To this audience, the platforms tell a different story. They promote a fun and youthful type of work, defined by low barriers to entry, flexibility, community, and, above all else, wealth.[186] It is a set of promises that have become a familiar part of the mythos of the gig economy. Digital work platforms routinely offer up the promise of "an idyllic, boss-free future, where workers control their incomes and hours" and position gig work as a "cure-all for the woes of modern society."[187] Bounty platforms stress that anyone can become a hacker, pointing out that most hackers who contribute to bounty programs have little formal experience. They affirm that new hackers can learn on the job—the platforms offer free online training and tutorials for those just starting out—and that bug bounty programs are a good way for curious novices to gain experience and break into the competitive security industry.[188]

Bounty platforms also emphasize hacking as a flexible occupation, one that leaves hackers with independence and autonomy. Hackers can pick and choose when to work or what to work on. As one report notes, working for a platform is an ideal type of part-time work: it leaves hackers with "more time to spend with family, experience other interests, or even work another job."[189] An interview featured in Bugcrowd's marketing materials drives this point home. In response to the question "What was the most extravagant thing you bought with your earnings?" the hacker who goes by the handle TodayIsNew offers up a family-friendly pitch:

> "Bug hunting value lets me stay home with my 2 little girls (3 years and 2 months) and there's nothing more worth" [190]

[185] HackerOne, *The Hacker Powered Security Report: 2019*, 3.

[186] The marketing strategies of these platforms mimics the images and promises offered by Uber. See: Rosenblat, *Uberland*.

[187] Ravenelle, *Hustle and Gig*, 5.

[188] See HackerOne, *The 2020 Hacker Report*, 27-28; HackerOne, *Hacker Powered Security Report: 2019*, 52-51; Bugcrowd, *Inside the Mind of a Hacker*, 1, 10.

[189] HackerOne, *The 2020 Hacker Report*, 15.

[190] Bugcrowd, *Inside the Mind of a Hacker*, 11.

The flexibility of bounty work, in this distillation, is a perfect occupation for those with diverse interests and competing responsibilities. What's more, the work is presented as intellectually challenging—the perfect job for someone who loves a challenge. Intellectual challenge, not drudgery or repetition, is stressed. As HackerOne emphasizes, "[c]uriosity is an enduring quality across the hacker community."[191] The intellectual challenge of hacking, rather than money, HackerOne reports, is the "biggest driver" for platform workers.[192]

Both platforms go further—presenting hacking as a way to contribute to something bigger, a way to serve a larger social good, to give back and improve security for all. HackerOne describes working for the platform as a way to "help solve one of the greatest challenges our society faces today."[193] Hackers as saviors and heroes is a recurring motif. HackerOne's 2019 report repurposes classic Marvel comic book covers—*Fantastic Four, the Invincible Iron Man, The Incredible Hulk,* and others—and inserts hackers into the frames.[194] These images ring the pages and surround an executive summary that trumpets the work that hackers have done via the platforms. In this and other reports, hackers contributing to bug bounty programs are not just working, they are "using their talent and grit to keep us safe."[195]

But behind these various appeals, the core pitch is bald: hacking can make you rich. In 2019, the cover of HackerOne's annual report, *The Hacker Powered Security Report*, featured a single image: a large portrait of hacker @try_to_hack with a striking caption: "first hacker to achieve $1M in bounties."[196] The promotion of bounty work as a path to wealth is a constant note. In a recent annual report, HackerOne is quick to highlight that the million-dollar club is growing; seven hackers have now earned over $1 million through bounty programs.[197] They provide a running tab of the number of hackers who have earned $100,000 and $500,000 in lifetime earnings.[198] The marketing pitch is direct: Hacking can be a lucrative career. As one sub-heading of a recent report tantalizingly remarks, "Across the

[191] HackerOne, *The 2020 Hacker Report*, 34.

[192] Ibid., 35.

[193] *Hacker Powered Security Report: 2019*, 2.

[194] Ibid., 2-3.

[195] HackerOne, *The 2020 Hacker Report*, 49.

[196] HackerOne, *Hacker Powered Security Report: 2019*.

[197] HackerOne, *The 2020 Hacker Report*, 4.

[198] Ibid.

Globe, Hackers are Making Millions."[199] These materials frame this type of wealth as something within reach. HackerOne spotlights the hacker "Cosmin," a hacker who took a practical hacking seminar and learned about bug bounty programs and, after falling in love with hacking, earned over a million dollars in as little as two years.[200] The bones of this story, of fast and easy money to be made out of hacking, repeat across marketing materials.

HackerOne includes a tantalizing table in its 2019 report, listing bug bounty earnings by top hackers compared with annual median earnings of security engineers in different countries.[201] The numbers show a staggering difference: top hackers working in bounty programs can earn over 40 times the income of security engineers in Argentina, over 17 times the amount of engineers in India, and six times the income of engineers in the United States.[202] Bugcrowd paints an idyllic picture of hunting for bugs in lower-wage countries. They state that most hackers live in countries where traditional incomes are less than $25,000 USD per year.[203] They interpret these lower wages to mean that hackers living in these countries can obtain "an easygoing lifestyle that costs less than half of what is considered a median salary in the United States."[204] The image of hacking that is presented is hard to resist: with a little bit of work, a free tutorial here, an online class there, you, too, could hack your way into a life of interesting, fulfilling work that can make you rich.

The seductive image of hacking that is packaged and promoted by bug bounty programs is not a mirage. For some, including some of the hackers we interviewed, these promises do come true. But for many, contributing to bug bounty programs is much different. The promises—flexibility, exciting and meaningful work, low-barriers to entry, and good pay—are more complicated than the glossy marketing copy lets on. The freedom and autonomy promised by this type of work masks a blunt reality: bounty platforms wield enormous power over how hackers work. This power creates recurring friction between hackers and the programs that review, manage, and compensate their work.

[199] Ibid., 14.

[200] Ibid.

[201] Hacker One, *Hacker Powered Security Report: 2019*, 51.

[202] Ibid.

[203] Bugcrowd, *Inside the Mind of a Hacker: 2020*, 2.

[204] Ibid.

Platforms and the Ordering of Work

Bug bounty platforms mirror other "lean platforms"—they extract profits from a significant volume of market transactions without directly employing many of the workers who create value.[205] HackerOne and Bugcrowd are backed by significant venture capital funds. Benchmark Capital, the firm behind Uber's first round of venture capital funding, led HackerOne's Series A funding (and Benchmark partner Bill Gurley holds a seat on HackerOne's Board of Directors).[206] Like all platforms, the core service that bug bounty programs offer is moderation: they help to create, organize, and manage the flow of hackers, money, and information.[207] The platforms act in many ways like a traditional employer—setting wages, outlining terms of work, and supervising labor. Despite this, hackers are not considered employees, but instead are classified as independent contractors. Just as Uber famously is said to be a taxi company that employs no taxi drivers, and Airbnb is a rental property company that owns no property, HackerOne and Bugcrowd are hacking companies that do not actually employ hackers.[208] The classification of hackers as independent contractors is consequential: it keeps labor costs down for the platforms and it limits the ability of hackers to receive guaranteed pay, legal protections, or benefits.[209]

Bounty platforms do more than match hackers and companies looking to buy bugs. They structure and manage these interactions: they set the terms that largely define and govern this work. For companies or organizations looking to start a bounty program, platforms provide a ready, off-the-shelf, set of tools to get started. Managers from the platforms help draft and define program scope,

[205] Srnicek, *Platform Capitalism*, 75-88.

[206] HackerOne, "HackerOne Funding Reaches $110M as Hacker Community Surpasses 500,000," September 8, 2019, https://www.hackerone.com/press-release/hackerone-funding-reaches-110m-hacker-community-surpasses-500000; Bugcrowd, "Bugcrowd Announces Record Growth, Secures $30 Million in Series D Funding," April 9, 2020, https://www.bugcrowd.com/press-release/bugcrowd-announces-record-growth-secures-30-million-in-series-d-funding/; Ron Miller, "HackerOne Get $9M in Series A Funding to Build Bug Tracking Bounty Programs," *TechCrunch*, May 28, 2014, https://techcrunch.com/2014/05/28/hackerone-get-9m-in-series-a-funding-to-build-bug-tracking-bounty-programs/.

[207] See Gillespie, *Custodians of the Internet*.

[208] Ibid., 76.

[209] On debates concerning the classification of gig workers, see: Rosenblat, Uberland, 8-9, 156; Woodcock and Graham, *The Gig Economy; Srnicek, Platform Capitalism*, 75-88.

payment structure, and terms of service. These platforms actively recruit hackers to enroll in bounty programs, and directly or indirectly guide their work. They encourage hackers to focus on particular programs, rate and rank hackers based on the number and quality of their submissions, arrange private invitations for closed programs (picking which hackers to invite and which to leave out), process and distribute payments, and mediate disputes that arise between hackers and companies. Additionally, they run live hacking events, where select hackers are invited to hack systems in person or demonstrate the effectiveness of hacks that have already been devised, often for inflated prize amounts. The platforms design and manage these events—curating the invite list, setting prize amounts, and handling the accommodations for the hackers. Far from being simply a meeting place for hackers and companies, platforms are active and powerful players.

To be sure, bounty platforms deliver tangible benefits for the hackers who participate. They create spaces where hackers can build professional reputations and résumés. They provide opportunities for hackers to earn income doing what they like—hacking. They provide a clear path for reporting bugs, something that has never been a given for security researchers. They also measure organizations' response times and can provide standard templates for vulnerability disclosure policies, providing some predictability and stability in the course of market interactions. These benefits should not be brushed aside. The benefits that hackers take away from these programs in many cases is not an illusion—but the reality is more complicated than it appears.

The lopsided power of platforms manifests in ways big and small. Platforms directly or in consultation with enrolled firms determine what work is accepted—defining quality. They set rates for different ranges for bugs—defining price. They determine who can participate in the most lucrative corners of the market—defining access. And they determine what forms of hacking are authorized—defining the legal protections that govern this work. This creates hazards and risks for hackers, leading to uncertainty, uncompensated work, and, in some cases, legal jeopardy.

Part IV: The Hazards of Contingent Hacking Work

Bug bounty programs and platforms only purchase new, previously unknown and undisclosed bugs. Duplicates, or “dupes” as they are known, are not paid out. Finding a new bug—something that nobody else has found or reported—is all that matters. It is time-consuming and labor-intensive work. For the workers, the stakes are high. Finding new bugs takes time, and there is no guarantee that they will find one. Hackers participating in bug bounty programs often speak about the thrill of finding a new flaw. But workers take on significant risks in this market: there is no certainty that their hard work will be rewarded. Failure is always an option.

Hackers spend hours hunting for new bugs. Vishal (“Vis”) Patel, a hacker based in Gujarat, India, jumped into bug bounty just before completing college.[210] He spent, by his estimation, six months working only on vulnerability disclosure programs (VDPs), programs that do not pay for bugs. Patel used VDPs as a training ground, a place to sharpen his skills before he felt ready to submit bugs to paying programs. After some initial success, he started working on bounty programs full-time. He would divide up his time between reading up on the latest bugs and learning new

[210] Interview with Vishal Patel, 2020.

skills and hunting. His days were fairly regimented, as he recalled: “I used to work more than 12 hours per day… two, three hours specifically for reading [on new methods], and then eight, nine hours for completely finding targets.” Recently, Patel had started a full-time, 9-to-5 job. But he continued to put in time hunting for bugs. His typical day was long: after a quick two-hour break after his day job, he would typically log on and start looking for bugs around 7:30 p.m. His work would carry long into the night; working until 2 or 3 a.m. after a full day of work was not unusual. Sleepless nights were not unheard of.

Diksha Chhabra, a hacker from New Delhi, India, recounted that when she first started participating in bug bounty programs, she set an ambitious goal for herself: submitting five or more new bug reports per day.[211] It is hard work. Chhabra noted, “finding bugs is not that easy task like people think it is.” Going days with little or no sleep was, for her, common. Currently, like Patel, she was juggling hunting for bugs with a full-time job, pushing her bounty work to the nights and weekends.

This intensive work can lead to nothing. For hackers, coming up empty or unwittingly submitting a duplicate—a bug that has already been discovered—is a common hazard. It means hours or days of wasted and uncompensated work. Unlike salaried workers or contract work, hackers participating in bounty programs assume all the risk: they only get paid if and when they find a novel bug. Pouring hours into looking for a novel bug, only to find out that it has already been reported, is typical—and frustrating. Jesse Kinser described it like this: “You can put a ton of time into something and write up this awesome report and hit submit and then… ten minutes later they are going to say, ‘Oh, sorry, somebody has already reported this.’”[212] The picture of bounty programs as something of an idyllic job, a family-friendly way to earn money on the side, on a schedule of your choosing—is complicated by these types of stories of long-hours and uncertain payoffs.

The risk of uncompensated or lost time, to some, was driven or compounded by under-resourced or unscrupulous bounty programs. Kinser and others we spoke with suggested that certain programs that are slow to fix bugs are, in effect, wasting hackers’ time. As she recounted, when you submit a bug you have no idea if it is going to be successful “because it could be a vulnerability they already know

[211] Interview with Diksha Chhabra, 2020.

[212] Interview with Jesse Kinser, 2019.

about, and they just haven't fixed yet, and you don't get paid anything."[213] In her view, duplicates were not a sign that hackers were not doing their best, it was often a sign that the companies themselves were lacking. As she described it, slow patch cycles lead hackers to invest time in dead ends: "some companies are really bad about patching, right? Especially companies that only push out a new update in their product twice a year, it may be six months that that bug sits open… [H]ow many hackers are wasting their time finding that same thing, writing that up? So that's discouraging."[214] Others told stories about bug bounty programs that refused to pay out a submitted bug, only to later turn around and fix the bug without compensating or acknowledging the work of the researchers. These stories were often contrasted with anecdotes about responsible programs—programs that were quick to pay out and quick to credit. But they pointed out a familiar anxiety: hackers being exploited by programs that were not able or willing to invest the time and resources into running a mature bounty program.

Hunting for bugs can be a grind. Jack Cable enjoyed hunting for bugs while also working on his undergraduate degree. But he could not see himself doing it long-term. The work, in the end, was too repetitive. He saw others getting burned out and wanted to avoid that fate.[215] Patel agreed. It was easy to get lost in bounty work. After the initial rush of excitement of finding his first successful bugs, the work could start to become stifling and routine. After a couple of months working full time in bounty programs, it started to seem repetitive, a familiar and deadened-process, "Open Google, open this program, open Burp Suite [testing software] and [then] this, this, this."[216] He noted that breaks for mental health were crucial. He saw others working "14 hours or 15 hours a day" and burnout, depression, and other problems seemed to follow. Setting limits for himself was important. Sahil Ahamad recalled his early experience hunting for bugs full time. Bounty work gobbled up nearly all of his free time. It got to the point where Ahamad noted that there was little time for much else: "I was studying, and doing bug bounty, and eating."[217] Eventually, Ahamad's parents had to provide a timely reminder: "You should go out and play." For Alyssa Herrera, breaks were important. Her best

[213] Ibid.

[214] Ibid. See: Alkhatib et al, "Examining Crowd Work and Gig Work Through The Historical Lens of Piecework."

[215] Interview with Jack Cable, 2019.

[216] Interview with Vishal Patel, 2020.

[217] Interview with Sahil Ahamad, 2020.

days hunting for bugs came after getting a much-needed breather. The work was draining. As she observed, "You kind of get burned out easily… You are more or less sitting on the computer for a couple hours, you're just staring at the screen, and if you haven't found anything it can be frustrating at times, too. It can be a bit stressful."[218]

Duplicates are not the only source of lost time and uncompensated work. The determination by a bounty platform or a bounty program that a bug submission is "out-of-scope" or that it is a trivial issue (and therefore assigned a lower price) also leads to uncompensated or devalued work—and is the source of frequent disputes between hackers and triage workers. Kinser, who also has experience running bug bounty programs, explained triage at a vendor-run program: "When a bug would come in, it was the responsibility of myself and the rest of the product security team [at the company] to go in, read through how the write-up is written by the hacker and go in and try to recreate that in the system and see if it is a legitimate vulnerability or not."[219] After a bug is validated as legitimate and in-scope, bounty platforms or programs assign a price based on the bugs severity. But this process is rarely cut and dried.

Hackers report submitting what they take to be valid bugs, only to have these flaws later deemed by bounty platforms or bounty programs to be out-of-scope or ineligible for payment. Many of the hackers we spoke with talked about the frustration of trying and failing to explain to the worker who was triaging incoming bug reports why a bug report was a valid and significant issue. Tom Anthony described a bug he found in Google's interface that would allow you to track users. In his view, this was a serious flaw. But, after some back and forth, he was told that this was "intended behavior."[220] Bipin Jitiya, a veteran hacker based in Ahmedabad, India, echoed this sentiment. He recalled submitting one of his first bugs only to be told, "This is not a security issue. This is the intentional functionality of the system."[221] Jitiya remarked that "everyone [knew] this is a security bug." Indeed, to his wry amusement, the company eventually went ahead and patched this bug, without acknowledging or compensating his work. He took

[218] Interview with Alyssa Herrera, 2019.

[219] Interview with Jesse Kinser, 2019.

[220] Interview with Tom Anthony, 2019.

[221] Interview with Bipin Jitiya, 2020.

this as a sign that he was right—it was a bug. He could have perhaps forced them to reopen the issue, but he didn't. As a then-new participant he did not feel like he could press his case: "Newbies, people… who are new in [the] field, are like, what can we say?" Relatively new participants, Jitiya, observed, are not likely to make waves. Rather than contesting the issue, they offer up a more conciliatory response and dive back into work. When you are new the field, you are more apt to say, "That's okay, we will find some other bugs," rather than push back against an adverse determination.

Fighting to have your work acknowledged and compensated can take its own toll. Dzmitry Lukyanenka, a hacker from Belarus, described a years-long tug of war that followed a bug submission. He insisted that he had found a real issue only to be told that it was a duplicate and that a fix was already in the works. But, upon later inspection the promised fix did not cover the issues he had found.[222] In other cases, he was told his bug could not be reproduced. After painstaking deliberations, they finally paid him for his bug.[223]

Repeatedly, interviewees spoke of having to explain a bug over and over, drawing out not only technical details but connecting the dots between the flaw and the larger business risks that it might contain. Social capital, as much as technical chops, are needed to thrive in bug bounty programs. Chhabra implicitly highlighted this point, noting that many researchers "know how to exploit the vulnerability but they don't know how to represent it" in a clear report. Drafting reports that can be understood by a wide audience inside a company is a must. Often, as she noted, bug reports are reviewed by nontechnical staff members who have a limited understanding of the bug or technology in question. Teasing out the larger nontechnical implications of a bug in these reports was important: it could sometimes help nudge the price of a bug higher. But such debates can be an exercise in futility that breeds mistrust. Herrera recalled an instance when a fellow researcher was paid what she took to be "an extremely low award" for a bug that "allowed malicious actors to… push their own code to the entire company." After the hacker and others discussed in a public forum why the bug had fetched such a low price, the company effectively banned the hacker and others from

[222] Interview with Dzmitry Lukyanenka, 2019.

[223] Ibid.

contributing to the program in the future.[224] In the view of some of the hackers we spoke with, those doing the triage in this case simply did not understand the impact of what was being reported.

The contingent nature of this type of work creates risks for workers. It is hard to know before they sink hours or days of work into hunting for a bug if they will ever be paid for their efforts. Hunting for bugs is, above all, unpredictable. There is no sense of when or if the next payment might arrive. Kinser reflected on the difficulties that full-time hackers face.[225] In her view, it seemed like tough and risky work: "I don't really know how they pay their bills… you know what I mean? Because it's so unpredictable." Doing freelance security work on short-term contracts was tough, but this seemed far better than getting paid bug-by-bug. With short-term contracts, she remarked, "at least you have a signed contract. Here, [with bug bounty programs] it's like, you have no idea."[226] The payment pipeline for hackers is notoriously spotty. As she observed, you "could have one awesome month and then may not have anything for two months." This sort of work, in her view, is not sustainable: "It's just a little bit too unpredictable, to have a family, and that kind of stuff, and do that full-time, in my opinion. Your next payment could be a long way off; there was no sense of security."[227]

Hackers develop a range of strategies to manage the uncertainty and risks of the market. Some turn and invest more time and effort as individuals into bounty programs, effectively doubling down on their work. In many cases, this takes the form of automation, where hackers create tools for identifying bugs in the scope of a bounty program, or automate submission. Some hackers pair bounty work with other full-time work, balancing the unpredictability of bug bounty programs with a degree of stability. Yet others opt out of the market all together, seeking other spaces to hack. And many turn to leverage bounty programs into a more stable income or job.

EdOverflow, a frequent bug bounty participant, tried to avoid the frustration of duplicate submissions by coming up with a clever methodology that would allow him to shrink the time between discovery and submission. He described an approach

[224] Interview with Alyssa Herrera, 2019.

[225] Interview with Jesse Kinser, 2019.

[226] Ibid.

[227] Ibid.

based on speed: "You learn to adapt... you learn methods [for] how to quickly submit things."[228] He uses standard templates for writing up his bug reports and focuses on being concise and clear. For others, the race to be first leads them to search for complex bugs. Herrera explained that while this takes more work, in her view, avoiding simple bugs—"low hanging fruit"—gives her the best chance to avoid duplicates. As she notes: "the chances of having... duplicate [bugs] are a lot lower if it's... harder to find."[229] Patel looked for newly launched programs, figuring that they would be fertile ground that had not yet been picked over by other hackers. Others looked to see which programs had slow response times and avoided them, figuring that if they were slow to respond they were more likely to accumulate duplicate submissions. Ahamad tested out new programs, before devoting serious time to looking for new bugs. He explained: "Whenever I [see] a new program, I try to report one or two bugs. And I see the response from their team." Ahamad watches their response, looking to see if, in his view, the program was capable, both quick to respond and accurate in pricing the bug. If the response looked reasonable, he would invest time in the program; if not, he did not waste his time. Sahil's strategy—and the comments of others—made plain an important point: not all bounty programs are the same.

Some hackers, however, opt out of the market to avoid the frustrations of bug bounty programs all together. Jorden Wiens, a security expert, found fame through the United Airlines bug bounty program.[230] Wiens earned over 1 million United Airlines miles for his bug submission. He submitted his bug quickly, the night the program went live. In his view, he "got lucky." He is certain that others likely submitted the same bug after he did and received nothing for their work. But, for Wiens, the race to be first—the frustration of duplicate submissions, and the pressure to work fast—was ultimately not worth it. Other outlets for hacking prowess are more satisfying—for instance, Capture the Flag (CTF) competitions. In CTF, where hackers hunt for hidden flags in a controlled environment, Wiens noted, skill—and not just speed—are rewarded.

Select hackers, including a number of our interviewees, push back against the power of platforms and bounty programs and challenge what they see as unfair

[228] Interview with EdOverflow, 2019.

[229] Interview with Alyssa Herrera, 2019.

[230] Interview with Jordan Wiens, 2019.

decisions. This power—the power of platforms and firms to determine what counts as valid bugs and set prices—is complicated by a few factors. While bugs are not transferable, top hackers *are*. The most successful hackers are highly sought after: platforms and programs want them to contribute to *their* bounty programs, they want not just the wisdom of the crowd, but their wisdom. Higher prices might be needed to attract this type of top talent.[231] A number of the high-profile hackers recalled instances where their persistence—arguing and advocating for what they determined to be a fair price—occasionally paid off. One high-profile hacker spoke about how he was able to press the triage team to pass along his bug to the developers for review, even though they had already determined that it was not an issue. Once the developers saw it, they agreed that it was a significant issue and paid out a significant bounty. In these cases, being able to not only explain the technical details but the business impact of the bug was critical. This points to the importance of social capital as well as technical acumen. The ability to be comfortable talking about corporate positioning and speaking the language of business was, in these cases, just as important as the technical diagnoses.

The power of these particular hackers to negotiate prices upward is likely exceptional—for low-level bugs, new entrants, or those without the desired social capital, negotiation is most likely off the table. These ad hoc strategies are important; they indicate that the power of bounty platforms and programs is not absolute. Top talent has in some cases the ability to work outside of normal channels and advocate for themselves. But these strategies and efforts are limited. They achieve individual pockets of resistance and carve out narrow spaces of autonomy, but they do not, it appears, lead to larger changes in how platforms or programs operate. The benefits that these individual efforts secure, then, are just that: individual.

One European hacker, a frequent participant in the market, talked about how he navigated the insecurity of the market. He used a colorful analogy: “It’s kind of like when you go to the lake and you try to fish”, because you never know if you are going to make a catch. Even when he did find and submit a valid bug, payments were not always quickly forthcoming. He recalled with a laugh waiting two years for a payment. He, like a lot of other hackers, turned to other gigs as a way to

[231] Ellis et al, “Fixing a Hole: The Labor Market for Bugs.”

buffer the lack of stability that hunting for bugs brings in. He pooled his savings, collecting his “money from bug bounties [in order to] have some reserved… which allow us to live maybe few months without having bounties.” Having banked savings from bounties, he felt more secure and able to ride out the days when his proverbial net was empty. He also reinvested his earnings, putting some of his money into apartments that he planned to rent for additional income.

Bounties might return a big payout for people find a major flaw or who do this work full-time, but the inherent instability of the work led many of our interviewees to seek out the stability of full-time, in-house security or programming jobs. This was true even in countries like India, where bounty payouts can often far exceed the monthly salary of programmers. Patel talked about receiving one his first bounty payments, $1000 USD. It was, at the time, a mind-blowing amount of money. As he noted, an entry level programming position at the time paid closer to $300 per month. He recalled with pride taking his family out for dinner to celebrate his earnings. Ahamad recalled a similar experience. As a college student, he received $750 for his first bug. It was a staggering and welcome payout. At the time, his monthly income as a student was around $70 per month. But even these comparatively large sums did not always make up for the inherent uncertainty of the work. Ahamad provided a clear summation of this view: he took a full-time security job with a firm for the stability. He described his thinking: “I have joined full-time work because of the stability… In bug bounty, we get lots of money, but it’s not regular.” The lack of predictability and stability can be a significant problem that conflicts with other goals and priorities. For Ahamad, a more predictable job was crucial: “Getting a home, home loans, you need to have stable income.” Chhabra agreed; bounty work was thrilling and at times lucrative. But it was risky for hackers looking to make a living. In her view, it would not be wise to “make this bug bounty a permanent job for you. [To] every researcher, I [would] recommend [that they] simultaneously do something else or work for a private company.”

But many hackers were not willing or able to leave bounties behind, at least not yet or not completely. A number of hackers turned their bounty experience into a new hustle: they started training others how to hack—producing books, seminars, and other instructional aides to help others get started. Others moved into bounty management, starting to take on jobs doing triage or running a bug bounty program for a vendor. These complementary gigs provided a way to stay close to doing the work they liked—hacking—while adding more stability into the mix.

Working for Free: Capitalizing on Uncompensated Work

The contingent and unpredictable nature of the work creates risk for workers. But this dynamic benefits bounty platforms and their clients: it allows them to keep costs low. While many hackers view their bounty work as a type of apprenticeship, a stepping stone that might help them land a more stable programming or security job, bounty platforms view the abundance of uncompensated labor as a crucial selling point.

Bounty platforms are not the only ones who see bug bounty programs as possible replacements for costly security work. Firms see bug bounty programs as a cheaper—and potentially higher-quality—alternative. Lisa Wiswell, a former Department of Defense (DoD) official and one the key figures behind the design and launch of DoD's bug bounty program dubbed "Hack the Pentagon," helped start DoD's bug bounty program to prove a point: pen test assessments were not worth it.[232] In her view, the pen test assessments that DoD was paying for were "extremely expensive and... almost never actionable or informative." These outside reviews were, in her view, "almost valueless." She wanted to start a bug bounty program to prove that there was a better alternative to the pen tests. As she looked back, she recalled her motivation in pushing for a bug bounty program: "I wanted to prove that the reports that we would get from [a bounty program] would be more meaningful and ultimately more cost-effective."

Hack the Pentagon delivered. In Wiswell's recollection, the pilot program was relatively cheap—it cost $150,000. This included the bounty payments to hackers, fees to HackerOne, and the costs of reviewing the submitted reports. During the first 24 days of the program's pilot, DoD received 138 "actionable" reports. This was, for Wiswell, a stunning success. Looking back, Wiswell still marveled at the success of the initial launch and the quality of the bug submissions. As she put it, the bug submissions "blew [her] mind." Previously, Wiswell observed, DoD had been getting "if we were lucky, maybe close to 10" reports from pen test contracts that

[232] Interview with Lisa Wiswell, 2019.

cost “infinitely more than… $150,000.” At the same time, the pen test contracts provided reports that all too often lacked actionable information—they simply did not have the practical detail that the bug reports submitted through Hack the Pentagon contained. The reason for the difference was easy to spot. With pen test contracts, typically a small number of people participate in the pen test. As Wiswell described: “You have a very small handful of people, maybe two” working on a pen test. With a bug bounty program, you have “a thousand sets of eyes.” Reina Staley, a DoD official who worked closely with the bounty program, echoed Wiswell’s observations.[233] The bounty program, in her view, “was a much greater return on investment” than the previous pen test contracts. With pen tests you are not paying for bugs—you are, as Staley stated, “paying for someone’s time.” With bounty programs, you are paying for bugs. Staley was direct about one of the benefits of this contingent model of work: “Whether hackers spend a hundred hours or a thousand hours on research, the DoD ultimately receives and pays only for quality reports that demonstrate where the weaknesses are.”

For smaller firms, bounty programs are not a way of replacing expensive contracts or expensive in-house workers. These companies and organizations often do not have the budget for this type of security work in the first place. Here, bug bounty programs are not replacing security workers—they are adding them into the organization for the first time. Jesse Kinser also runs a bounty program for a smaller company. She views bug bounty programs as a way of adding security testing in ways that fit her company’s profile:

> “[t]he reason that I love the bounty program: it really helps because we’re a small team, it’s myself that runs this, I’m the director of product security, so it’s just me; and I have one person underneath me that helps with it. So I have a team of hackers by using the bug bounty program to check my product so that I don’t have to hire people to sit in-house and do the security research.” [234]

For Kinser, bug bounty programs are not about cutting costs and replacing workers with a lower-cost alternative. They are a way to add a layer of testing that otherwise would not occur.

[233] Interview with Reina Staley, 2019.

[234] Interview with Jesse Kinser, 2019.

There is a grim irony here. Many hackers view bounty programs as a springboard or on-ramp to a more stable job in security or programming. They use bounty programs as a place to learn new skills, develop a reputation that can help them stand out from other applicants, and make contacts for future career opportunities. Yet, these programs have been packaged and sold to companies as a way to eliminate the very jobs that these hackers are seeking. Platforms might market bug bounties as an extra layer of defense, but they also play up the relatively low costs. It's plausible that organizations may eventually come to rely on this insecure and flexible labor.

From the outside, bounty programs can look like an appealing option for organizations—an easy way to improve security and cut costs. But the reality can be more complicated. Katie Moussouris makes this point plain. Moussouris is the founder and CEO of Luta Security, a firm that specializes in helping organizations and governments facilitate vulnerability disclosure. She is a bounty innovator and expert. She created and designed Microsoft's first bug bounty program—overcoming significant internal opposition—and created a way for hackers to safely report flaws. She also went on to serve as chief policy officer at HackerOne, playing a decisive role in starting the DoD's bug bounty program. Moussouris knows more about the ins and outs of bug bounty programs than nearly anyone.[235] Yet, she is quick to point out that bug bounty programs are not a quick fix for larger organizational failings. Too often, she notes, organizations rush in and adopt a bug bounty program without first doing the necessary work to make sure that they are prepared. For companies that do not have established and strong security practices, the rush of an influx of new bug reports that a bounty program can bring can be overwhelming. It can divert attention and resources away from larger systemic issues. Starting a bug bounty program might look good on the outside, but Moussouris notes that they can wind up doing little more than papering over larger shortcomings.

[235] For an overview of Moussouris' thinking on bug bounty programs and their challenges, see: Andrew Marino, "How the commercialization of bug bounties is creating more vulnerabilities," *The Vergecast*, July 7, 2020, https://www.theverge.com/2020/7/7/21315870/cybersecurity-bug-bounties-commercialization-katie-moussouris-interview-vergecast-podcast; Bri Hand, "Developing Sustainable Vulnerability Management with Katie Moussouris," *Security Nation*, June 9, 2020, https://blog.rapid7.com/2020/06/09/developing-sustainable-vulnerability-management-with-katie-moussouris/.

Access Control: Private Programs, Live Events, and Reputation Scores

Bug bounty platforms and their clients are gate keepers. Despite the rhetoric of open access, empowerment, and low-barriers-to-entry that color bug bounty programs, not all hackers can participate in all aspects of the market. Platforms and clients have significant power to shape who can and cannot access private programs and live events. These are the most lucrative corners of the market—and they are only available to some hackers. The hope and possibility of getting an invite into these profitable and select spaces pushes hackers to undertake a significant amount of uncompensated work, while sometimes leaving them wondering why they are still on the outside looking in.

Hackers want to be invited to private programs. These programs offer a chance to earn significant bounties without facing the competition of the larger crowd. Herrera, the frequent bug bounty participant we spoke with, was clear. She works frequently with closed programs. The benefits of doing so, in her view, were obvious: invitation-only programs "tend to be a lot more lucrative than the public programs because there tend to be a lot less researchers and [they] tend to be a lot more higher paying [than public programs]."[236] Cable agreed. Early in his bug hunting career, he found himself gravitating toward new private programs.[237] In his view, these programs were fertile ground. To Cable, it seemed that "if the program is newer, it's easier to find vulnerabilities."[238] Individual bounties might not always be higher in every closed program, in Cable's view, but there are fewer hackers and less competition for bugs in these spaces.[239]

Exclusive live events are a difficult and sought-after invitation. HackerOne, Bugcrowd, and others stage live hacking events with a curated list of select hackers. Firms contract with bug bounty programs to put on these events and feature their software or systems as exclusive targets. These sponsoring firms put up the

[236] Interview with Alyssa Herrera, 2019.

[237] Interview with Jack Cable, 2019.

[238] Ibid.

[239] Ibid.

money for the event—covering the bounties, the travel and accommodations for the hackers, and a fee paid to the hosting bug bounty platform. The participating hackers are given the details of the target—the software or system they will be focusing on—in advance, often taking several weeks to prepare for the live event. In some cases, participants might have to sign an NDA before looking for flaws agreeing not to publicly disclose details of the event. At the event, they spend a day or more hacking exclusively on the sponsors' products. The sponsoring firm gets the devoted time and energy of a select group of hackers.

These events are part PR/marketing event, part community-building retreat, and part party. Hackers like these events for a lot of reasons. Notably, the bounties paid out during live events are often inflated in order to get top hackers to participate. Hackers get a lot of attention—press availability and interviews are common features—for participating in live events. The exclusive list of hackers that are invited are flown in and given the star treatment. They are provided food, drinks, airfare, and accommodations. Kevin Rosenbaum, a former HackerOne and Bugcrowd employee, saw his job in setting up and managing live events clearly as a way of catering to hackers and giving them the attention they deserve:

> "I pay for their flight, I pay for their hotel, I pay for multiple meals throughout the weekend, if there's something they need, like, let me see if I can get it for you. I try to help them as much as possible... I want to make you feel like they're the celebrity that they are. Because, to me, they are; they are people that have the ability to do great harm, but instead they choose to selflessly help people. Well, not completely selflessly, because they are getting paid..."[240]

These events are not just important to the hackers—they are important branding and sales opportunities for the platforms and clients. Platforms collect fees from sponsors to design and host these events. Platforms want to make a big splash. Budgets easily run into six figures. Without a decent-sized budget, Rosenbaum, notes, you are just wasting hackers' time: "If you're telling 20 hackers that they're going to come only fighting over $100,000, it's not really worth it for anybody... I'm going to blow through that budget. That's the first two hours of the day, gone."[241] A larger budget is needed to attract top talent and drive press attention.

[240] Interview with Kevin Rosenbaum, 2019.

[241] Ibid.

For sponsors, live events drive new bug reports and, perhaps just as important, a higher-profile within the security community.[242]

Hackers view the chance for connection as one of the most significant benefits of participating in a live event. For Cable, getting invitations to live events is a significant incentive—they provided a rich opportunity to meet and connect with other hackers.[243] Rosenbaum sees live events as opportunities to build a rich and diverse community of hackers.[244] They offer a way for a scattered community to connect and interact.[245] He views live events as a chance to make "people across the world feel like they have friends, and I think that that's a beautiful goal."[246] Ideally, in Rosenbaum's view, live events can forge new collaborations between hackers.[247] As he put it, the point of live events is to "make hackers' lives better, and to make them make more money, and to make the world better." These events can and do showcase hacker talent. Rosenbaum tries to use live events to push hackers who might not be as well-known or high-profile into the spotlight and give them a sense of community that might be missing.[248] EdOverflow agreed.[249] Hacking can feel solitary, but the live events allow the community to come into focus and come alive. He saw live events as a way to meet likeminded friends. For some hackers, these events are great opportunities to make career connections with potential employers. Dustin Childs, a manager at ZDI, remarked that companies use live events as a way to spot talent, offering successful hackers a chance to join a company full time.[250] After seeing a few hackers continually rack up significant awards, he told them that "it would be a lot cheaper if we just hired you!"[251] Peter Yaworski, a veteran and successful bug bounty participant, found a full-time job through connections he made at a live event.[252] He recalled getting invited to a live

[242] HackerOne, "HackerOne Live Hacking Events," 2019, https://www.hackerone.com/sites/default/files/2018-06/Live%20Hacking%20Data%20Sheet.pdf.

[243] Interview with Jack Cable, 2019.

[244] Interview with Kevin Rosenbaum, 2019.

[245] Ibid.

[246] Ibid.

[247] Ibid.

[248] Ibid.

[249] Interview with EdOverflow, 2019.

[250] Interview with Dustin Childs, 2019.

[251] Ibid.

[252] Interview with Peter Yaworski, 2019.

event hosted by HackerOne.[253] At the live event, he met the Shopify team.[254] The live event, in his words, "fundamentally led to my job with them."[255] These events, in Yaworski's view were valuable because "you get to meet other security teams and make connections and develop those relationships."[256] These connections are important and powerful: they can lead to friendships, new collaborations, a sense of belonging, and career opportunities.

Access to private programs and live events are not, however, available to everyone. The platforms and their clients pick and choose who to invite. Private programs and live events are enticing for hackers—they work harder and longer hours in the hopes of getting these invites. But how these invites are doled out can be a bit of mystery to hackers, partially based on clear-cut-rules and partially based on other unseen judgments.

Bounty platforms rank and rate hackers based on the frequency and quality of their submissions.[257] These rankings encourage hackers to put in more time working on the platform. Each platform has its own proprietary means of evaluating hackers. For instance, on HackerOne, each hacker is assigned a "Reputation" score.[258] Reputation increases with each new and valid bug submitted. The score is weighted; hackers earn more points for critical and expensive bugs.[259] Submitting bugs that are deemed to be "not applicable" or "duplicate" (already made public) leads to a deduction in a hackers' Reputation score.[260] If your Reputation score dips below a certain threshold, HackerOne can prevent you from submitting new bugs on the

[253] Ibid.

[254] Ibid.

[255] Ibid.

[256] Ibid.

[257] HackerOne has three related main rankings: "Reputation," "Signal," and "Impact." Reputation is calculated based on the number of submissions that are successfully closed out by triage. Successful submissions increase your Reputation score, while duplicate or not applicable submissions can lower your score. Signal is derived from comparing the amount of valid to invalid submissions. Impact is based on the average severity of bug submissions. See: HackerOne, "Signal and Impact," n.d., https://docs.hackerone.com/hackers/signal-and-impact.html; HackerOne, "Introducing Reputation," October 8, 2014, https://www.hackerone.com/blog/introducing-reputation; HackerOne, "Reputation," n.d., https://docs.hackerone.com/hackers/reputation.html; for descriptions of Bugcrowd's metrics, see: Andy White, "How Bugcrowd Sees Vulnerability Disclosure Programs and Points," *Bugcrowd*, May 27, 2021, https://www.bugcrowd.com/blog/how-bugcrowd-sees-vulnerability-disclosure-programs-and-points/.

[258] HackerOne, "Reputation." l

[259] Ibid.

[260] Ibid.

platform within a given time period.[261] This ranking system is important: a high score may lead to invitations into private programs or live events.

How exactly a hacker qualifies for an invitation, however, can still be murky. There is not a clear threshold—a certain Reputation score or rank—that translates into an invite to a closed program or live event. Both HackerOne and Bugcrowd have recently attempted to clarify and make more transparent their processes for doling out invitations to live events. But these processes are still fuzzy. HackerOne notes that invites to live events are based on some combination of three different categories: critical reports, consistency, and community.[262] Criticality is based, for example, on the severity of bugs that a hacker has submitted on the platform (tracking closely to Reputation). Consistency is based on how active a hacker is on the platform and the quality of a hacker's reports.[263] Community is based on a somewhat less concrete foundation.[264] In explaining this category, HackerOne notes that "observed social media engagement" and "professionalism in platform interactions" fit into this category. (Recent code-of-conduct violations also count as a strike against hackers under the "Community" rubric.) BugCrowd's documentation for researchers provides some concrete indicators for when private invites are given (or not), but otherwise uses similarly vague language to explain their process for getting invites.[265] Despite trying to provide more clarity and transparency, the process is still opaque. How these different categories—critical reports, consistency, and community—are set against one another is not clear. On top of this, HackerOne states that people living close to cities where live events occur "will be considered for an invite," making it appear potentially less likely that a person living in a remote or perhaps non-US location would be extended an invite.[266] Additionally, how the different outlined factors within these categories are tabulated (What counts as "social media engagement"?

[261] Ibid.

[262] Luke Tucker, "Live Hacking Events: Stats, Invitations, and What's Next," *HackerOne*, July 15, 2019, https://www.hackerone.com/blog/live-hacking-events-stats-invitations-and-whats-next.

[263] Ibid.

[264] Ibid.

[265] "Viewing and Accepting Program Invitations," Bugcrowd Docs, November 5, 2021, https://docs.bugcrowd.com/researchers/participating-in-program/viewing-invitations/; Michael Hamel, "You've Got Mail!—Receiving Bugcrowd Private Program Invites," *Bugcrowd*, February 1, 2021, https://www.bugcrowd.com/blog/bugcrowd-private-invites/.

[266] Luke Tucker, "Live Hacking Events: Stats, Invitations, and What's Next," *HackerOne*, July 15, 2019.

What counts as "professionalism in platform interactions"?) is not transparent or always obvious to hackers.

Even if the exact criteria are not clear, hackers are encouraged to keep hacking on the platform and to keep submitting bugs. Herrera explained that invitations are typically linked to Reputation scores and hacker ratings: "For companies like HackerOne or Bugcrowd, for example, they tend to weigh on, basically, your reputation, how active you are, your overall scores, because on these platforms you have a profile and you'll get points for every report you make."[267] These scores matter. Kinser—a hacker *and* a bounty program manager—explained how she can select who receives an invite. HackerOne allows programs to define who gets invited: "[y]ou can say 'Invite 50 of your top hackers that have a really high reputation for submitting really good reports'" and HackerOne will send out invites to hackers who fit the supplied criteria.[268] Cable recalls that his increasing Reputation score likely secured him an invite to the DoD's first bounty pilot program: "[I] gradually get a little bit Reputation on the platform and then one of the more interesting things that happened was... I get an email asking me if I want to do Hack the Pentagon."[269]

Legal Risks: "Safe Harbors" and Their Limits

Despite the proliferation and mainstream acceptance of bug bounty programs, hackers can still face legal risks when they contribute to a bounty program.[270] It is legally precarious work. Anti-hacking laws, including, for example, the *Computer Fraud and Abuse Act* (CFAA) and the *Digital Millennium Copyright Act* (DMCA) in the US, create significant civil and criminal risks for hackers. These laws can

[267] Interview with Alyssa Herrera, 2019.

[268] Interview with Jesse Kinser, 2019.

[269] Interview with Jack Cable, 2019.

[270] For a detailed discussion of the legal risks associated with bug bounty programs, see: Amit Elazari Bar On, "Private Ordering Shaping Cybersecurity Policy: The Case of Bug Bounties," in *Rewired: Cybersecurity Governance*, eds. Ryan Ellis and Vivek Mohan (Hoboken, NJ: Wiley, 2019), 231-264.

have consequences for hackers across the globe; particularly since 9/11, courts in the US have been enabled to extend laws like the CFAA beyond the country's own borders in cases where criminal misconduct harms people in the US.[271] Finding new bugs often requires testing and manipulating software and systems in ways that are unexpected and potentially in contravention of not only local but also US laws. The unauthorized access and the circumvention of security controls can, in some cases, put hackers in serious legal jeopardy.

Many bug bounty platforms and programs, along with the Department of Justice (DOJ), have outlined key considerations for drafting disclosure policies, with a focus on including provisions that shield participants from legal liability within certain parameters. For the DOJ, such standard 'safe harbor' language generally includes at least three core elements: (1) a pledge that the organization will not pursue civil action for accidental or good faith policy violations nor initiate a law enforcement complaint; (2) a statement affirming that activities that are undertaken and consistent with the program's policies will be considered "authorized" under the CFAA; and (3) a commitment that if a third party brings a legal action against a hacker that has acted in good faith, the organization will make known that the hacker acted in compliance with program policies.[272] This language is important. HackerOne makes safe harbor language now the default standard for its programs.[273] Bugcrowd has worked to make safe harbors a reality as well.[274] It supports Disclose.io, an open source repository and tool to promote the adoption of legal safe harbors.[275]

These are important and admirable steps—bounty platforms are directly working to protect workers. But they are incomplete. Not all bug bounty programs have

[271] "Prosecuting Computer Crimes," OLE Litigation Series (U.S. Department of Justice, Computer Crime and Intellectual Property Section Criminal Division, n.d.), 115–16, https://www.justice.gov/sites/default/files/criminal-ccips/legacy/2015/01/14/ccmanual.pdf; William K. Kane and Melissa M. Mikail, "Extraterritorial Application of the Computer Fraud and Abuse Act," *The National Law Review*, July 3, 2020, https://www.natlawreview.com/article/extraterritorial-application-computer-fraud-and-abuse-act.

[272] Department of Justice. "A Framework for a Vulnerability Disclosure Program for Online Systems," July, 2017, https://www.justice.gov/criminal-ccips/page/file/983996/download; see also: https://disclose.io/.

[273] HackerOne. "What is a Responsible Disclosure Policy and Why You Need One," August 30, 2018, https://www.hackerone.com/blog/What-Vulnerability-Disclosure-Policy-and-Why-You-Need-One.

[274] Jason Haddix. "Protecting Hackers (By Default) with Dislcose.io," December 3, 2019, https://www.bugcrowd.com/blog/protecting-hackers-by-default-with-disclose-io/.

[275] See Amit Elazari Bar On, "Standardizing Legal Safe Harbor for Security Researchers," August 2, 2018, https://www.bugcrowd.com/blog/guest-post-standardizing-legal-safe-harbor-for-security-research/. See also https://disclose.io/.

adopted safe harbor language to protect hackers. Programs that lack clear legal terms and scope still put hackers at risk, shifting legal risks onto the hacker for working in this economy.[276] Additionally, even when companies have adopted safe harbor language, the protections are limited. As Bugcrowd points out, safe harbor language inclusions are "band-aids."[277] The DOJ's description of safe harbor provisions gives companies much power to decide what constitutes "authorized" activity.[278] The narrowing of the definition of "exceeds authorized access" by the US Supreme Court in its June 2021 *Van Buren* decision has provided increased—yet limited—clarity regarding the CFAA's applicability to security research.[279] However, until anti-hacking laws in the US and beyond are modified to provide explicit and potentially presumptive carve outs for good faith vulnerability disclosure, hackers can still face significant legal uncertainty and risk when they participate in bug bounty programs.[280]

Bounty platforms and bounty programs also cannot provide complete legal protection even with safe harbor language. Third parties can bring claims against a hacker even if the bounty program has authorized the activity. The legal safe harbor language does not and cannot bind third parties, including law enforcement, and neither can it compel them to abide by the wishes of the bounty program. As Microsoft's bounty terms make clear, third parties may in some cases bring legal action against a hacker over the objections of Microsoft.[281] The bounty program does not and indeed cannot indemnify hackers from third-party complaints.[282] Additionally, for hackers working outside the US, domestic anti-hacking laws and regulations might present legal risks that are not addressed by the typical safe harbor language.[283] Limits on the CFAA and DMCA are helpful (if imperfect),

[276] Elazari Bar On, "Private Ordering Shaping Cybersecurity Policy," See also Elazari Bar On, "Standardizing Legal Safe Harbor for Security Researchers."

[277] Haddix, "Protecting Hackers (By Default) with Dislcose.io."

[278] Yuan Stevens et al., "See Something, Say Something: Coordinating the Disclosure of Security Vulnerabilities in Canada" (Cybersecure Policy Exchange, June 24, 2021), https://www.cybersecurepolicy.ca/vulnerability-disclosure.

[279] Van Buren v. United States, 593 U. S. 19-783 S. Ct. (2021) https://www.supremecourt.gov/opinions/20pdf/19-783_k53l.pdf.

[280] Ibid.

[281] Microsoft Security Response Center, "Microsoft Bounty Legal Safe Harbor," n.d., https://www.microsoft.com/en-us/msrc/bounty-safe-harbor.

[282] Ibid.

[283] For example, China recently announced new restrictions relating to vulnerability research and disclosure. Catalin Cimpanu, "Chinese Government Lays out New Vulnerability Disclosure Rules," *The Record by Recorded Future*, July 14, 2021, https://therecord.media/chinese-government-lays-out-new-vulnerability-disclosure-rules/.

but hackers based in other countries have to try to weigh cross-cutting legal risks from multiple jurisdictions. Even with the adoption of standard safe harbor language, hackers, then, still face some residual risk. The effort to promote and attach legal safe harbors to bug bounty programs is important, but as long as anti-hacking laws in the US and elsewhere fail to protect well-intentioned disclosures of security flaws, hackers still face significant legal risks.

Bug bounty participants navigate these legal risks in different ways. Herrera tries to avoid programs with unclear legal terms. As she notes: "I tend to look more closely at [programs with well-described legal policies rather] than other companies that don't have a safe harbor policy."[284] She was clear: she skipped bug bounty programs with unclear legal terms in order to avoid the "possibility of being legally entrapped."[285] Cable agrees. "If a program doesn't have clear, defined [legal] terms, I might not participate in it."[286] For Cable, more established programs seemed to obviate the legal concerns. As he observed, "there is always… concern [that] legal action [could be] taken."[287] As Cable suggested, legal risks are one of the biggest worries that a bug bounty participant might face. He has heard stories of hackers facing legal threats as part of their involvement in bug bounty programs. But, as he was quick to note, he had not experienced these issues personally. In his view, so long as he stuck with "established" programs, these sorts of issues would "effectively never [happen]."[288]

The Perils and Pleasure of Hacking in the Gig Economy

The stories of the hackers we spoke with give us a window into what is like to hack in the gig economy. Working for bug bounty platforms can be a source of community, pleasure, creativity, and wealth. But there are pressures and tensions as well.

[284] Interview with Alyssa Herrera, 2019.

[285] Ibid.

[286] Interview with Jack Cable, 2019.

[287] Ibid.

[288] Ibid.

Bounty platforms and programs have significant power to shape and define the terms that govern this work. They set and enforce prices for bugs, they control access to the most lucrative aspects of the market, and they shape the legal risks that hackers face as part of their day-to-day work. Hackers working in this market undertake significant uncompensated work. Hours and days spent hunting for a new bug can often lead to a dead-end, a duplicate, or an out-of-scope submission. Hard work all for naught. Getting access to private programs and live events is highly sought after—but getting in is tough. A shifting or opaque invite-criteria can leave hackers guessing. Above all, ranking systems and leaderboards encourage hackers to always keep working. The legal landscape puts hackers in jeopardy. Unclear program terms can lead hackers to retroactive punishment. Even when platforms and bounty programs try to protect hackers by including legal safe harbors, these protections are limited. Until anti-hacking laws are reformed, this work is and remains legally dangerous.

Hackers navigate these currents largely through individual responses (though disclose.io is a powerful exception, offering a meaningful collective response). They find work-arounds that enable them to avoid duplicates. They identify and then avoid programs that appear to be under-resourced. They needle and negotiate over prices. They develop sidelines and complementary gigs to fill in the gaps. And they take steps that they hope will provide legal protection. These efforts are important but largely idiosyncratic. They provide individual moments of autonomy, accommodation, and protection within a market where power is tilted toward platforms and bounty programs. But these ad hoc strategies rarely spill over into larger movements to protect other hackers or mount a larger reconsideration of how the market works.

Conclusion: Rethinking Bug Bounty Programs

Finding, disclosing, and fixing bugs is important infrastructure work. It enables the apps, software, and digital infrastructure of contemporary life to operate and evolve. Bug bounty programs now structure and govern much of this work. This standard is at once visible and invisible. Press releases, news stories, and other accounts document the launch of each new high-profile bounty program; they highlight tales of young hackers making fabulous amounts of money; and they promote lavish branded hacking competitions that take place across the globe, in hotels, conference centers, and ballrooms from Las Vegas to Abu Dhabi. But these flashy accounts only tell part of the story. Many of the details and lived experiences of this market—what it is like for ordinary people to find and sell bugs, the administrative details of how programs operate, and how these programs create new pressures on other related forms of work—still remain largely out of sight. It is a paradox: despite volumes of attention and frequent public praise, this infrastructure work remains largely obscure and poorly understood. This report starts to illuminate this backstage work. It looks behind the curtain to reveal the work and workers involved in finding, reporting, and fixing the near-endless stream of bugs that dot the digital world.

Bug bounty programs started as an on-the-fly solution to a public relations crisis—a way for Netscape to stem the tide of negative headlines that accompanied every new bug that hackers discovered and disclosed. Two decades later, these programs have hardened into a new standard for organizing hackers, software, and organizations. These programs remake hacking into something that is predictable and stable. In the process, alternative models for organizing hacking based on the free circulation of information and self-determination are being traded away. Bugs are bottled up under nondisclosure agreements and terms of service; and hackers are funneled into systems where they are ranked, rated, and and bear the burden of fixing flawed software.

The gig economy holds out a number of promises: flexibility, autonomy, creative work, and wealth. But these dreams are complicated by the realities of bounty work. The hackers we spoke with found much to like in a market for bugs—they had, in one form or another, made it. But even in these interviews, we heard about a more difficult reality. Uncompensated work. Burn out. Barriers to entry. Uneven legal risks.

> Stepping back, it is worth asking, *what are the larger implications of reorganizing infrastructure work based on this type of market?* What are the costs of this particular reordering of hackers, organizations, and code?

Stepping back, it is worth asking, *what are the larger implications of reorganizing infrastructure work based on this type of market?* What are the costs of this particular reordering of hackers, organizations, and code? These programs are tools for outsourcing and offshoring a particular slice of information security work. They provide a vehicle to transform this work into something that is done by a global market of mostly young workers, operating on a contingent basis, and working with inadequate benefits and labor protections. Uncertainty is the rule. They might get paid for their work or they might not. They might get invited to a private program or they might not. They might be legally protected, or they might not. These hackers are entrepreneurs. They stake a claim in the hopes of making a profit. But they work within a system that is designed to shift risks from large organizations onto individual workers. Some will succeed in this system. But many more will not. Either way, the design of the market ensures that the platforms and bounty programs continue to benefit as more and more hackers sign up to take their chances hunting for bugs. The irony is clear: vulnerable workers are enrolled to fix our vulnerable software and systems.

The irony is clear: vulnerable workers are enrolled to fix our vulnerable software and systems.

This market tilts the playing field decisively toward bounty platforms and bounty programs. This not only creates risks for the workers who move within the market—it also creates risks for the public at large. Bounty programs can seem, as Katie Moussouris warns, like a quick fix—a shortcut to improve security. But shifting important digital security work to a contingent model of work—a contingent model of work that sits on top of VC-backed platforms—is risky. What happens if these platforms go bust? If one or both of these companies scaled down or exited the market, what would fill the gap? What's more, outsourcing security work through bounty programs relies on the steady influx of new, low-cost labor. This leads to burn out and, potentially, a loss of goodwill among the hacker community. When that happens it is not just a personal loss—it is a loss for all who rely on the knowledge and expertise of professionals to keep our networks, devices, and software secure.

There is nothing inevitable or natural about this type of market. Other ways of organizing are possible. A larger reckoning might stop and ask if the proliferation of bounty programs actually enables the continued production of buggy code and software. Rather than encouraging companies and organizations to invest in security during development, is the legal and economic regime that organizes low-cost labor into bounty programs simply a way of sustaining a world full of bugs? Decades ago, groups like the Cypherpunks tried to use the discovery and disclosure of bugs as a way to improve what they saw as all-too-often lax security. Companies and organizations learned perhaps the wrong lesson. Rather than invest in security as a core part of software development, they decided to simply buy up the bugs. As a result, bugs continue to proliferate, and an army of low-cost workers are called on to do the important, vital, and risky maintenance work.

There are concrete steps that can help ensure that bounty programs better serve the interests of both computer security and workers. First, bounty programs should be one layer of a larger security posture, not a replacement for a larger organizational effort or full-time security engineering work. At their best, bounty programs can complement ongoing security efforts. Additional eyes, inviting hackers in, can bring fresh perspectives that can otherwise be missed—organizations can draw out larger lessons from bug reports and improve security. But, at their worst, bounty programs can be a way to gut internal security teams and replace them with lower-cost contingent workers. There is a potentially grim irony: workers who turn to bounty programs as a way to build their resume and get a foothold in the field of computer security might find that the very programs they are participating in are erasing the jobs that they seek.

Second, organizations operating bounty programs should invest and commit the necessary resources to run a responsive and effective program. Under-resourced programs can—and do—waste hackers' time and effort. Slow response times, non-payment, and a lack of clear recognition for work done are just a few of the issues that can spring up when an organization has jumped into a bounty program without first devoting the necessary resources to improve baseline security and manage the work associated with running a mature bounty program.

Third, increased transparency is needed for numerous aspects of bug bounty programs. The criteria and performance metrics used to extend invites to private programs and live events could be made more evident to workers. Eliminating or revising NDAs and allowing hackers to talk about the private programs that they contribute to could also increase transparency in the labor market and eliminate the use of bounty programs as vehicles for hiding or covering up flaws.

Fourth, creating review and dispute resolution mechanisms for conflicts over bugs deemed duplicate, out-of-scope, or low severity could provide a counterweight to the mistrust and frustration that triage currently engenders. These mechanisms should be transparent—open to scrutiny—and insulated from the larger business pressures associated with bounty programs and platforms.

Some bounty programs—and some platforms—already embrace some of these elements; but others do not. Addressing these issues would begin to help ensure that bounty programs serve both workers and the larger aims of security. In other words, they would begin to address the power imbalances that characterize the market for flaws.

Looking beyond any single bounty program or platform, larger reforms to anti-hacking laws are also needed. Amending anti-hacking laws to create a secure and sustainable legal environment for this type of work is necessary. Recent developments in US case law as well as efforts by bug bounty platforms and programs to create legal safe harbors, limited though they may be, underline both that positive change is possible and that these changes remain incomplete.

When bounty programs and platforms fail to address the impacts of their working conditions on certain communities such as racialized workers, then these programs are not inclusive but exploitative.

Additionally, many of the problems that hackers working in bounty programs face mirror the challenges that other contingent workers face. Likewise, interventions that address the mismatch between how full-time and contingent workers are treated can help alleviate the imbalances and risks that are part and parcel with gig work. For example, classifying hackers and other gig workers as employees and not independent contractors would open up opportunities for hackers to secure workplace legal protections and benefits; in particular, it might lead bounty platforms to reconsider business models that rely on rapid iteration, operating in perpetual beta mode.

Most importantly, combating racialized labor inequalities will require rethinking how and on what terms different workers are integrated into an organization. Bounty programs have paved the way for a global workforce to enter the security labor market. But these programs can, absent larger organizational changes and commitments to diversity and equity, help to codify a stratified workforce. When bounty programs and platforms fail to address the impacts of their working conditions on certain communities such as racialized workers, then these programs are not inclusive but exploitative. Only larger reforms that take into consideration how racialized labor is extracted by organizations and markets can ensure that bounty programs do not become mechanisms that reproduce and solidify stratification.

Thinking through these issues—how to create bounty programs that benefit workers and improve security—is crucial. Bug bounty programs are being held up as a larger model for how to crowdsource other kinds of vulnerabilities in sociotechnical systems, with Facebook and Twitter, for example, using the bounty model to address data abuses or algorithmic biases.[289] Extending this uptake, organizations are also drawing on the bug bounty model to potentially facilitate what Sasha Costanza-Chock calls "design justice" for communities who stand to be most negatively affected by technical systems.[290] Take the Community Reporting of Algorithmic System Harms (CRASH) project by the Algorithmic Justice League as an example. The initiative, led by Joy Buolamwini, Camille François, and Costanza-Chock, works to "enable broader participation in the creation of more accountable, equitable, and less harmful AI [artificial intelligence] systems."[291] The CRASH project explores the possibility of implementing bug bounty-inspired disclosure systems for the harm that can arise from predictive algorithmic systems, including surveillance, inaccuracy, and biased or discriminatory predictions.[292] People who are the most affected by AI-powered technology are to be meaningfully involved in the entire lifecycle and decision-making processes regarding the use of the technical system.[293] For this project, the definition of a "flaw" is informed by the AI system's impact on the community, rather than focusing on the intention of the system's designer, thereby inviting people to contribute to the design, architecture, and impact of technical systems where they would otherwise be excluded. The AJL's efforts in particular have been marked by very careful consideration of program design like compensation and reporting structure *before* the launch of a prototype harms reporting platform, serving as a role model in many ways to other organizations wishing to deploy such reporting pipelines.[294]

However, without such careful consideration, the bounty model may discourage the broad, public disclosure of reported flaws and harms. Thus far, bug bounty programs have been built on a pool of precarious workers located globally—a

[289] Kenway and François, "Bug Bounties for Algorithmic Harms?"; Collin Greene, "Data Abuse Bounty: Facebook Now Rewards for Reports of Data Abuse," Facebook, April 10, 2018, https://about.fb.com/news/2018/04/data-abuse-bounty/; Rumman Chowdhury, Jutta Williams, "Introducing Twitter's First Algorithmic Bias Bounty Challenge," *Twitter Engineering*, July 30, 2021, https://blog.twitter.com/engineering/en_us/topics/insights/2021/algorithmic-bias-bounty-challenge.

[290] Sasha Costanza-Chock, *Design Justice: Community-Led Practices to Build the Worlds We Need* (Cambridge, MA: The MIT Press, 2020).

[291] "Algorithmic Vulnerability Bounty Project (AVBP)," Algorithmic Justice League, December 9, 2020, https://www.ajl.org/avbp .

[292] Ibid.

[293] "Learn More—The Algorithmic Justice League," n.d., https://www.ajl.org/learn-more.

[294] Joy Buolamwini, Camille François, and Sasha Costanza-Chock, "Happy Hacker Summer Camp Season! A CRASH Project update, from the team at the Algorithmic Justice League," Algorithmic Justice League, July 30, 2021, https://medium.com/@ajlunited/happy-hacker-summer-camp-season-e1f6fdaf7694.

surplus labor force that is included in such programs through extractive and unprotected piecework mechanisms. The people in charge of the disclosure pipeline can easily set the conditions that govern the work of finding and disclosing flaws even while they gatekeep what deserves protection and who gets access to this information.

> Vulnerable workers must not be forced to shoulder the risks when they are asked to do the impossible: fix a broken system.

A well-intentioned effort to include outsiders can curdle and become a way of denying responsibility, but this is not an inevitability. Turning to the wisdom of the crowd without addressing the impacts of this decision on the security of systems and workers can become a way to absolve the institutions that have created flaws from investing in creating secure systems. Vulnerable workers must not be forced to shoulder the risks when they are asked to do the impossible: fix a broken system.

APPENDIX: METHOOLOGICAL OVERVIEW

This report blends interviews, historical research, and analysis of documents—administrative reports and ephemera, marketing materials, annual reports, and more—related to the promotion and operation of bug bounty programs. As part of our work, we conducted 42 semi-structured interviews with current or former bug bounty workers, including hackers who sell bugs, program managers who help design and run bounty programs, and technical staff involved in mitigating bugs after they have been purchased. Interviews served as an ideal way to gain initial insight into people's experiences with bug bounty programs, a phenomenon not yet often studied with qualitative methods from the perspective of the contributor or worker.[295]

The interviews were conducted between January 2019 and February 2021. Interview subjects were drawn from publicly available bug bounty contributor lists and snowball sampling. More specifically, interviewees also included high-profile veterans who have contributed labor to the market for a number of years (with significant success); relatively new participants in this market; people who manage bug bounty programs or work for bug bounty platforms; and individuals who have exited the bug bounty market. Interview subjects included a mix of genders and nationalities, with the majority hailing from the US and Western Europe as well as a handful of workers in India, where a significant portion of the labor market lives.[296] Most interview subjects are based in North America and Western Europe unless otherwise indicated. The study did not include a significant sample of those

[295] There is a growing body of research that examines people's experiences with bug bounty programs using primarily quantitative methods, see for example: Daniel Votipka et al., "Hackers vs. Testers: A Comparison of Software Vulnerability Discovery Processes," in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, 374–91, https://doi.org/10.1109/SP.2018.00003; Matthew Finifter, Devdatta Akhawe, and David Wagner, "An Empirical Study of Vulnerability Rewards Programs," *Proceedings of the 22nd USENIX Security Symposium*, 2013, 273-88; Omer Akgul, "The Hackers' Viewpoint: Exploring Challenges and Benefits of Bug-Bounty Programs," *Workshop for Security Information Workers (WSIW)*, 2020, 1–7.

[296] On the significance of hackers from India specifically and outside the US and Western Europe more generally, see HackerOne, *Hacker-Powered Security Report 2019*, (2019) 12; HackerOne, *The 2020 Hacker Report*; Bugcrowd, *State of Bug Bounty: 2018 Edition*, 13.

who ceased selling their services in the bug bounty market. We also conducted limited observation at multiple hacker/security conferences, including Pwn2Own at CanSecWest in Vancouver, British Columbia, and NorthSec in Montréal, Canada. Interview subjects were given the option to use their name or a pseudonym when appearing in the report. A key theme of the report deals with acknowledging what is often overlooked labor. As such, interviewees were given the opportunity to use their real names at their discretion. Pseudonyms are used when requested.

ACKNOWLEDGMENTS

This work was made possible by the active participation and support of many. The report benefited from the stewardship of a large team at Data & Society Research Institute. In particular, we wish to thank Sareeta Amrute, danah boyd, Patrick Davison, Janet Haven, Charley Johnson, and Matt Goerzen for their feedback and guidance, as well as support from Ania Calderon, Audrey Evans, Molly Laas, Chris Redwood, Anisa Suthayalai, Lily Padula, and the Communications team at Data & Society. We are also grateful to Aiha Nguyen for organizing a workshop at D&S in 2019, providing a rich discussion with David Weil, Alex Rosenblat, and numerous others who informed this work. Additionally, Gabriella Coleman (including the bits-bots-bytes crew), Camille François, Josh Kenway, Wil Robertson, and Peter Yaworski provided crucial notes and suggestions during the drafting process. Expert research assistance was provided by Soraya Alli, Wyatt Hornstein, Madeline O'Hara, and Ava Sasani-Kolori. We also thank Jyotsana Singh for help with transcription. We greatly benefited from the wisdom and generosity of Katie Moussouris, as well as expertise and knowledge of Amit Elazari Bar On and Casey John Ellis. Finally, we wish to thank all of those who kindly shared their stories with us. This report would not be possible without them. Any mistakes are our own.

This material is based upon work supported by the National Science Foundation under Grant Nos. 1915815 and 1935520. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Early work on this project was supported by a research grant from Columbia University's School of International and Public Affairs and the Carnegie Corporation.

Data & Society is an independent nonprofit research institute
that advances new frames for understanding the implications
of data-centric and automated technology. We conduct research and
build the field of actors to ensure that knowledge guides
debate, decision-making, and technical choices.

www.datasociety.net
@datasociety

Cover illustration by Lily Padula | Graphic design by Anisa Suthayalai

January 2022