Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Crowdsourcing from Hackers: Strategic Coopetition and Governance in Bug Bounty Programs

Jiali Zhou, Kai-Lung Hui

To cite this article:
Jiali Zhou, Kai-Lung Hui (2026) Crowdsourcing from Hackers: Strategic Coopetition and Governance in Bug Bounty Programs. Management Science

Published online in Articles in Advance 07 Aug 2026

. https://doi.org/10.1287/mnsc.2022.02682

https://pubsonline.informs.org/journal/mnsc

MANAGEMENT SCIENCE
Articles in Advance, pp. 1–24
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

Crowdsourcing from Hackers: Strategic Coopetition and Governance in Bug Bounty Programs

Jiali Zhou,[a,*] **Kai-Lung Hui**[b]

[a] Kogod School of Business, American University, Washington, District of Columbia 20016; [b] School of Business and Management, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China

*Corresponding author

jializ@american.edu (J. Zhou), klhui@ust.hk (K.-L. Hui)

https://orcid.org/0000-0002-8678-5966 (J. Zhou), https://orcid.org/0000-0002-7074-1176 (K.-L. Hui)

Received: August 31, 2022
Revised: November 20, 2024; June 30, 2025; April 15, 2026
Accepted: May 30, 2026
Published Online in Articles in Advance: August 7, 2026

https://doi.org/10.1287/mnsc.2022.02682

Abstract. In a bug bounty program (BBP), organizations incentivize hackers—who could otherwise be "enemies"—to report security vulnerabilities by publicly offering monetary rewards. This paper develops an analytical framework to characterize BBPs and their strategic and economic impacts. Strategically, BBPs induce some hackers to self-select into cooperative vulnerability discovery and reporting, diverting them away from attacking (attack diversion) and delegating part of the protection effort to them (protection delegation). We show that (i) BBPs can be beneficial even when hackers are *inefficient* at vulnerability identification, yet more participants might make BBPs *less attractive* to firms. (ii) Although BBPs improve security, they might *encourage* competitive hackers (i.e., attackers) to exert more efforts. (iii) The coopetition dynamics can render a firm's provision of rewards and in-house efforts socially inefficient. Common cybersecurity regulations, such as breach penalties and bounty reward subsidies, may *exacerbate* the inefficiency. We find complementarity between bounty subsidies and breach penalties, suggesting that combining and customizing them according to firm characteristics could enhance BBP efficiency. (iv) Contrary to the conventional wisdom that crowdsourcing rewards increase with participant size, we identify scenarios in which a firm should *reduce* or make *nonmonotonic* adjustments to a bounty reward as participant size increases. (v) Legal safe harbor for security testing or policies that reduce duplicate report submissions can *lower* firm payoffs under BBPs. We draw related implications for research and practice in information security and crowdsourcing.

History: Accepted by Hemant Bhargava, information systems.
Funding: This research is supported in part by the HKSAR General Research Fund Project 16503620.
Supplemental Material: The online appendix is available at https://doi.org/10.1287/mnsc.2022.02682.

Keywords: bug bounty program • information security • coopetition • crowdsourcing • vulnerability market • cybersecurity governance

1. Introduction

Organizations are constantly challenged to protect their assets in cyberspace. Traditional protection measures deter hacking using in-house detective and preventive measures (Gordon and Loeb 2002, Cavusoglu et al. 2009), managed security services (Lee et al. 2013, Cezar et al. 2014, Hui et al. 2019), and government enforcement (Png and Wang 2009). We study burgeoning bug bounty programs (BBPs), which differ from conventional deterrent measures by encouraging hackers—organizations' potential "enemies"—to identify security vulnerabilities in information technology (IT) systems through monetary rewards (i.e., bug bounty rewards). Economic incentives significantly impact information security (Anderson and Moore 2006), but the implications of incentivizing threat actors to enhance security is largely unexplored. This paper explicates the economic, security, and social welfare implications of BBPs.

Consider Google's BBP. In November 2010, Google published a BBP policy that details the monetary rewards for external parties who report security vulnerabilities in its public-facing products, such as Google Search and YouTube. All previously undiscovered vulnerabilities are eligible for the rewards with the amount depending on the expected impact and type of vulnerability. For example, a remote code exploit in Google Search qualifies for a $101,010 reward (Google 2024). In 2025, Google awarded $17 million to more than 700 reporters (Mendez and Göhmann 2024). Online Appendix A details Google's BBP.

The computing industry has been ambivalent about BBPs. Proponents claim that BBPs help firms utilize

crowd wisdom, especially potential hackers, to supplement internal efforts to identify vulnerabilities (Anglin 2020, Winder 2020). However, critics question the economic merits of rewarding outsiders when organizations have more capable internal experts. Some dismiss BBPs as a fad (Oracle.com 2015). Reliance on BBPs might lead a firm to complacency in in-house protection, potentially compromising overall security (Dayan 2022, Harrington 2022). BBPs also introduce new challenges, such as the legality of security testing and balancing the extra effort needed to handle duplicate and invalid submissions.

As shown in Table 1, 9 of the 10 largest IT companies in the United States and 6 of the 10 in China offer BBPs.[1] Companies at the technological frontier, including AI leaders such as Anthropic and OpenAI, have also adopted BBPs. During the COVID-19 outbreak when many organizations relied on digital communications for work-from-home arrangements, the number of major breach incidents associated with firms providing BBPs reduced to eight (nine incidents during 2019) with major breach incidents increasing to 97 for non-BBP firms (compared with 71 incidents during 2019).[2] However, more than 80% of Forbes Global 500 companies had not adopted a BBP by 2021 (Akgul et al. 2023). Even among BBP adopters, many limit the scope to selected systems (known as the program scope; see Online Figure A.1). Evidently, the uncertain effectiveness of BBPs is a major challenge for their development.

To evaluate BBPs, this study addresses the following research questions: Under what circumstances do BBPs provide economic benefits to a firm? What are their impacts on information security and social welfare? How should bug bounty rewards be set under different cyber-environments? We also investigate policies that address common challenges faced by BBP providers, including the legality of security testing and the management of duplicate reports.

We develop a model in which a firm decides whether to offer a BBP to protect its vulnerable system from heterogeneous hackers, who might misuse the vulnerability for economic gains. Our model is similar to previous crowdsourcing analysis in that the firm can announce a reward to induce participants (hackers) to complete a task, namely, identify and report the vulnerability in the firm's system. The unique feature of BBPs, however, is that participants can be a firm's potential enemies. If the bounty reward is not sufficiently attractive, hackers might monetize a vulnerability elsewhere, such as by attacking the system directly. Furthermore, despite the BBP, a firm might still deploy in-house efforts to identify the vulnerability, and this enhances the system's security and leaves fewer opportunities for hackers to find the vulnerability. These features give rise to both cooperation and competition dynamics or coopetition between a firm and external parties, a scenario that the BBP and crowdsourcing literature has not considered.

Our analysis generates interesting insights. First, a BBP benefits a firm when the firm is inefficient in vulnerability identification or when it faces a large proportion of coopetitive hackers (Proposition 1). The latter implies that BBPs can be beneficial even when hackers are inefficient in vulnerability identification. In contrast to the common wisdom that larger crowds enhance crowdsourcing effectiveness, more BBP participants might make a BBP less attractive. Second, a BBP increases overall security but might encourage competitive hackers to exert more efforts (Proposition 2). Third, although a BBP can improve the payoff and security of a firm, the firm's provision of both rewards and in-house effort can be socially inefficient (Proposition 3). Our analysis suggests that widely applied cybersecurity regulations, such as breach penalties and bounty reward subsidies, might further exacerbate that inefficiency. We identify a complementarity between bounty subsidies and breach penalties, suggesting that combining and customizing them according to firm characteristics can increase BBP efficiency (Proposition 4). Fourth, contrary to the conventional wisdom that crowdsourcing rewards increase with participant size, we demonstrate that the coopetition feature of BBPs

Table 1. Bug Bounty Programs Among Large IT Companies in the United States and China

Rank	Company (United States)	Provide BBP?	Launch year	Company (China)	Provide BBP?	Launch year
1	Apple	Yes	2016	China Mobile	No	—
2	Microsoft	Yes	2013	Alibaba	Yes	2013
3	Alphabet (Google)	Yes	2010	Tencent	Yes	2012
4	AT&T	Yes	2012	China Telecom	No	—
5	Amazon	No	2020	China Unicom	No	—
6	Verizon	Yes	2014	JD.com	Yes	2013
7	Walt Disney	No	—	Baidu	Yes	2013
8	Meta (Facebook)	Yes	2011	Xiaomi	Yes	2014
9	Intel	Yes	2017	China Tower	No	—
10	IBM	Yes	2018	Lenovo	Yes	2017

Source. See Online Appendix B.

yields different reward properties. We identify scenarios in which a firm should reduce or make nonmonotonic adjustments to bounty rewards as the number of cooperative hackers increases (Proposition 5).

We extend the model to analyze two prominent BBP designs. Whereas it appears natural that a firm benefiting from ethical hacking should prefer legal safe harbors for security testing, we find that a safe harbor term (SHT) may reduce a firm's payoff. We discuss several policies that can help avoid such losses (Proposition 6). We also compare policies to address duplicate report submissions. Interestingly, firms challenged by duplicate report submissions may suffer from policies that more effectively reduce these uninformative submissions, implying that firms should choose duplicate report policies strategically (Proposition 7).

This study makes three important contributions. First, to our knowledge, it is the first formal analysis of a firm's incentive to provide a BBP. We systematically characterize the impact of BBPs on firms' economic payoff, system security, and social welfare by accounting for the novel feature that participants might either attack or protect digital systems, and this has important strategic and social welfare implications. Our study lays the foundation for future research and practice on BBPs and advances the security investment literature by moving beyond conventional deterrence measures to examining how organizations can collaborate with potential adversaries to enhance cybersecurity.

Second, our analysis generates novel insights into multiple BBP policies and practices, and this should be of interest to practitioners and policy makers. For example, we demonstrate that common cybersecurity regulations, such as breach penalties and bounty reward subsidies, can reduce social welfare with BBPs and that popular suggestions, such as safe harbor terms and taxing duplicate reports, might reduce a firm's return from BBPs. We also show that more cooperative participants might make BBPs less attractive and that a firm might need to reduce or adjust bounty rewards nonmonotonically in response to greater BBP participant size. These insights shed new implications for BBP practices and policies because they contradict conventional crowdsourcing and security protection wisdom.

Finally, we extend the crowdsourcing literature by considering the coopetition relationships between a firm and external parties (Brandenburger and Nalebuff 2011, Bouncken et al. 2015). Coopetition exists in many real-world settings,[3] giving rise to insights remarkably different from conventional crowdsourcing wisdoms (e.g., Propositions 1 and 5). Our analysis provides a framework to extend this theoretical literature by explicitly incorporating the possibility for a firm to compete when cooperating with external parties.

2. Related Literature

This study is related to the literature on information security investment. Following Gordon and Loeb (2002), the literature discusses how organizations optimize security investments in various contexts, such as system interdependency (e.g., Hausken 2006), when hackers strategically identify poorly protected systems (e.g., Cremonini and Nizovtsev 2009, Bandyopadhyay et al. 2014), and when organizations share information (Gal-Or and Ghose 2005, Gao et al. 2015). To our knowledge, we are the first to formally examine organization incentives to invest in BBPs that encourage rather than discourage hackers' identification of vulnerabilities.

A BBP's structure is similar to crowdsourcing contests, vulnerability markets, and security outsourcing. Conceptually, it can be considered as a crowdsourcing application because it engages the public to help a firm solve a problem, namely, identifying unknown vulnerabilities (Fryer and Simperl 2017). Studies of crowdsourcing assess the optimal design of a contest (e.g., Chawla et al. 2019), such as award properties (e.g., Terwiesch and Xu 2008, Ales et al. 2017), incentive schemes (e.g., Horton and Chilton 2010), and participation policies (e.g., Jeppesen and Lakhani 2010, Boudreau et al. 2011). We extend the literature by studying the optimal policy design of BBPs and incorporating its unique elements: (i) participants might use an identified vulnerability to hurt the firm; (ii) the firm's in-house security effort influences BBP outcomes as more in-house efforts reduce the chance of paying bounty rewards. In contrast, conventional crowdsourcing always awards the best external contribution regardless of the firm's in-house effort (Terwiesch and Xu 2008, Chawla et al. 2019). (iii) BBPs focus on the uniqueness of the discovered vulnerabilities. This is not the case with conventional crowdsourcing contests, in which quality is often the top consideration.

The literature on vulnerability markets also studies the reward for vulnerability reporting by infomediaries (e.g., iDefense or Zero Day Initiative) and their influences on social welfare and vulnerability diffusion (e.g., Kannan and Telang 2005, Ransbotham et al. 2012, Mitra and Ransbotham 2015). It addresses infomediaries that reward vulnerability reporting and sell vulnerability information to subscribers. Firms with security needs do not offer rewards themselves but subscribe to the infomediary to receive passive vulnerability information. In contrast, BBPs allow firms, such as Alphabet and Meta, to reward vulnerability reporting directly and, thus, address their specific cybersecurity needs. Analysis of a firm's incentives to reward vulnerability reporting instead of a market infomediary's incentive is new.

The strategic interaction between a firm and BBP participants resembles the interactions between an

outsourcing firm and managed security service providers (Dey et al. 2010, Hui et al. 2012). This literature focuses on addressing inefficiencies because of double moral hazards (Gupta and Zhdanov 2012, Lee et al. 2013, Hui et al. 2019). Their setting is, therefore, a principal–agent problem with hidden actions (Jensen and Meckling 1976). In our context, a firm opens the BBP to the public and pays participants ex post when they find a vulnerability. Our setting is closer to a screening problem (Spence 1973) with the objective of separating different types of users, namely, black-hat and white-hat hackers, and inducing preferred actions, namely, submitting a vulnerability for a bug bounty reward instead of exploiting it.

This study is also related to the literature on hacker behavior, which assesses how hackers respond to stimuli such as law enforcement (Png et al. 2008, Romanosky et al. 2011, Kigerl 2016, Hui et al. 2017), nonfinancial incentives (Leeson and Coyne 2005), hacking technologies (Png and Wang 2009, Ransbotham and Mitra 2009), economic conditions (Png et al. 2008), and vulnerability disclosure (Arora et al. 2006, Ransbotham et al. 2012, Mitra and Ransbotham 2015). Despite the recognition that hackers are neither all black nor all white but sometimes gray (e.g., Gottlieb 1999, Wired 2000, Lacity et al. 2009, Harper et al. 2011), the literature mostly treats them as offenders. We advance this literature by explicitly considering the dual use nature of hackers (Katyal 2001): they can harm or help secure digital assets.

Lastly, a small but rapidly growing literature specifically studies BBPs. Finifter et al. (2013) and Zhao et al. (2015) assess selected BBPs in terms of economic efficiency, vulnerability evolution, and participant behaviors. Other studies examine economic trade-offs related to BBPs, such as relationships between bounty rewards and vulnerability severity (Munaiah and Meneely 2016) and competition among BBPs (Maillart et al. 2017). Zhao et al. (2017) analyze mechanisms that increase BBP effectiveness, including reducing invalid reports and optimizing participant allocation. Zhang et al. (2024) examine how different contextual factors affect firms' bounty rewards and total costs.

Our study differs from this literature in three ways. First, the literature assumes that BBP participants are cooperative, neglecting the possibility that BBPs allow organizations to collaborate with potential enemies. Thus, it does not capture the attack diversion benefit of BBPs (Proposition 1). Second, the literature does not consider firms' in-house vulnerability identification efforts in conjunction with BBPs, and hence, it does not explain why companies such as Alphabet and Meta maintain in-house teams for vulnerability identification despite using BBPs. Our model explicitly captures such in-house incentives, highlighting the coopetition dynamics between a firm and hackers (Propositions 1–7). Third, the literature does not examine the incentive of a firm to offer BBPs. For example, Zhao et al. (2017) and Zhang et al. (2024) assume that the firm has already adopted a BBP. We go one step backward to assume that a BBP is not necessarily beneficial, implying that some settings considered in extant research may not align with the firm's incentive to offer a BBP. More importantly, the literature has not assessed the influence of BBPs on system security and social welfare. Collectively, we contribute to this literature by examining the economic, security, and social welfare implications of a BBP by accounting for its unique feature in enabling providers to coopete with potential enemies.

3. The Model

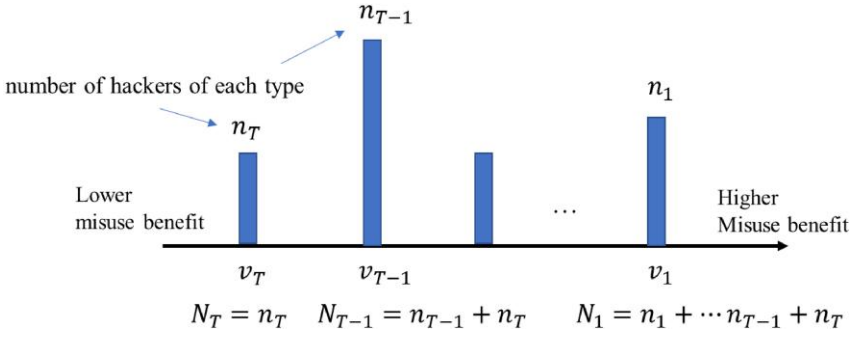
Consider a vulnerability identification model with one firm that invests to uncover a single vulnerability in its IT system.[4] The firm invests effort x_f to search for the vulnerability, for example, by conducting penetration tests and regular security audits or other experiments. It can also offer a public bounty award, r, to reward the first external party that finds and submits the vulnerability. The firm fixes the vulnerability (or implements workarounds) as soon as the vulnerability is identified internally or reported by an external party.[5] It suffers a loss, λ, if the vulnerability is first found and misused by a hacker (e.g., by exploiting or other irresponsible disclosure or use of a vulnerability).

The market has T types of hackers with valuations from misusing the vulnerability, v_t, $t = 1, 2, \ldots, T$, where $v_1 > v_2 > \cdots > v_T \geq 0$. The size of each hacker type t is n_t. The kth hacker of type t exerts effort $x_{t,k}$ to search for the vulnerability, where $k = 1, 2, \ldots, n_t$.[6] The firm and hackers compete to find the vulnerability, and only one of them can win the competition.[7] We assume that the firm and hackers are risk-neutral, and vulnerability patching, submission, and exploitation are costless to focus on strategic interactions between the BBP and vulnerability identification decisions. Figure 1 shows an illustrative hacker distribution. We assume that the distribution of hacker valuations is public information and fixed throughout the game.

We model the game over two periods. In period 1, the firm chooses whether to offer a bug bounty reward and how much it will be. In period 2, the firm and hackers simultaneously choose their effort and make their vulnerability submission/exploitation decisions. We use this game structure because the BBP is offered publicly, but the firm's and hackers' efforts are private information. We assume that the firm and hackers form rational expectations. The solution concept is subgame perfect Nash equilibrium. Figure 2 shows the different scenarios in our setting.[8]

We use superscripts nb and b to indicate scenarios when the BBP is not offered and offered, respectively,

Figure 1. (Color online) Sample Hacker Distribution



and subscripts f and t to indicate the firm and hacker type t, $t = 1, 2, \ldots, T$. In general, the effort of an agent—the firm or a hacker—should increase its chance of finding a vulnerability relative to other agents. One commonly used tractable function that models this competitive dynamic is the Tullock contest success function (Tullock 1980), which is widely applied to analyze patent races (e.g., Loury 1979, Dasgupta and Stiglitz 1980), research and development contests (e.g., Baye and Hoppe 2003), and military conflicts (e.g., Garfinkel and Skaperdas 2007). With the firm's effort x_f and hackers' efforts $x_{t,k}$, the Tullock contest success function defines the probability that the firm or any single hacker finds the vulnerability first:

$$p_i = \frac{x_i}{d + x_f + \sum_{j=1}^{T} \sum_{g=1}^{n_j} x_{j,g}}. \tag{1}$$

Here, i denotes either the firm (when $i = f$) or a hacker (when $i = (t,k)$). The term $d > 0$ captures the case when the vulnerability is not found. We can interpret d as the exogenous difficulty in finding the vulnerability or an exogenous competition from nature.

We complete the model specification by defining the firm's and hackers' cost functions. In general, the cost should increase with effort. We model the firm's cost as $c_f x_f$, and each hacker's cost as $c_h x_{t,k}$, where subscript h categorically indicates the hackers and c_f and c_h are exogenous cost parameters. A linear cost function helps obtain closed-form solutions and is common in Tullock contest settings (e.g., Dixit 1987, Baye and Hoppe 2003) because of analytical tractability. Our key insights are robust to any increasing and convex cost functions as shown in Online Appendix D.2. Cost parameters c_f and c_h measure the firm's and hackers' relative vulnerability identification efficiency (e.g., know-how and experience with security and whether they have access to sophisticated testing tools and knowledge). As is customary in the information security literature, we assume that c_f and c_h are common knowledge (Kannan and Telang 2005, Png and Wang 2009, Kannan et al. 2016).[9]

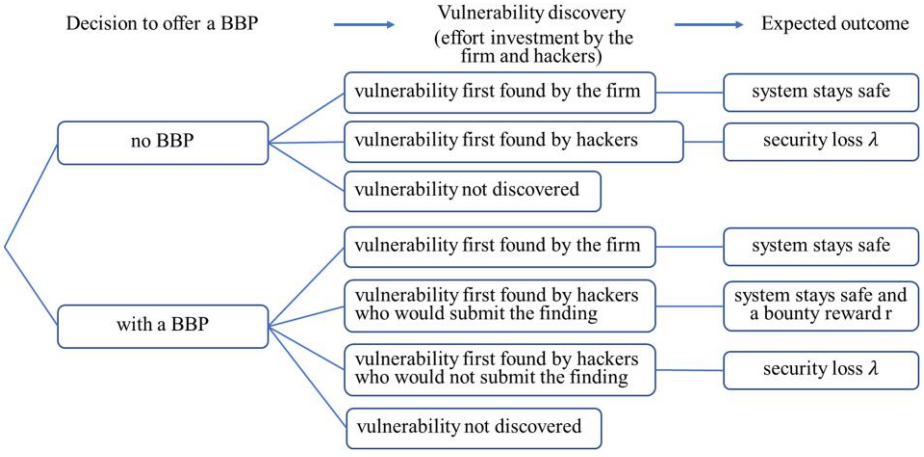
3.1. Equilibrium Without a BBP

The utility of a hacker is its valuation from misusing the vulnerability adjusted for the probability that it finds the vulnerability before all other players, net of its effort cost. Using (1), the net utility of hacker k of type t is[10]

$$u_{t,k}^{nb} = \frac{v_t x_{t,k}}{d + x_f + \sum_{j=1}^{T} \sum_{g=1}^{n_j} x_{j,g}} - c_h x_{t,k}. \tag{2}$$

Differentiating (2) with respect to $x_{t,k}$, the first-order condition (FOC) gives an implicit function for any hacker of type t (please refer to Online Appendix E for a

Figure 2. (Color online) Scenarios in Our Setting



detailed derivation):[11]

$$x_{t,k} = X\left(1 - \frac{c_h X}{v_t}\right), \tag{3}$$

where $X \equiv d + x_f + \sum_{j=1}^{T}\sum_{g=1}^{n_j} x_{j,g}$ measures the overall competition in finding the vulnerability. Equation (3) suggests that hackers of the same type make symmetric decisions. We simplify the notation by suppressing the individual hacker index, k. For example, we write x_t instead of $x_{t,k}$ and $n_t x_t$ instead of $\sum_{k=1}^{n_t} x_{t,k}$. We emphasize that each hacker makes an independent decision.

Equation (3) implies $x_t \geq 0$ if and only if $v_t \geq c_h X$ (i.e., the valuation of exploiting the vulnerability must be large enough for a hacker to exert a positive effort). Otherwise, it chooses $x_t = 0$. Because $v_1 > v_2 > \cdots > v_T \geq 0$, there exists some index $B \in [0, T]$ such that $x_t > 0$ for all $t \leq B$ and $x_t = 0$ when $t > B$. If $B = 0$, no hacker exerts a positive effort. Similarly, if $B = T$, all hackers compete to find the vulnerability.

When the competition result is determined by a Tullock function with an increasing and convex cost function, there exists a unique pure strategy equilibrium (see Szidarovszky and Okuguchi 1997, theorem 1). Intuitively, because the effort, x_t, is continuous, if the FOC of (2) is positive for any marginal hacker type t, the n_t hackers of the same type must all expend effort in finding the vulnerability. It then follows that, in equilibrium, all hackers of type $B+1$ and beyond must exert zero effort. The boundary index, B, divides hackers into two groups—competing and noncompeting—in such a pure strategy equilibrium.

After obtaining the number of competing hackers, the firm's payoff function becomes

$$\pi_f^{nb} = -\lambda \sum_{j=1}^{B} \frac{n_j x_j}{X} - c_f x_f. \tag{4}$$

The first term in (4) represents the loss from vulnerability misuse when a competing hacker finds the vulnerability before the firm. By combining the FOC of (4) and those of all hackers with type $t \leq B$, we can compute the equilibrium B.

Lemma 1. *Without a BBP, hackers with type $t \leq B^{nb}$ exert positive effort in finding a vulnerability, in which B^{nb} is the largest index that satisfies the inequality*

$$v_{B^{nb}} > \tau = \frac{\lambda c_h (N_1 - N_{B^{nb}+1})}{c_f + \lambda c_h \sum_{j=1}^{B^{nb}} \frac{n_j}{v_j}}, \tag{5}$$

where $N_j \equiv n_j + n_{j+1} + \cdots + n_T$ represents the number of hackers with misuse benefits lower than or equal to v_j and N_1 is the total number of hackers in the market.

Lemma 1 suggests that, in the baseline setting without a BBP, there exists an endogenous threshold of valuation, τ, which relates positively to type $1, 2, \ldots, B^{nb}$ hackers' valuations and the firm's security loss, λ, from the vulnerability, by which hackers are sorted into potential attackers (i.e., those who would spend effort to find and misuse the vulnerability) and nonattackers (i.e., those who would not exert effort to find the vulnerability). Figure 3 illustrates such an endogenous equilibrium distribution of hackers. Note that τ need not coincide with the valuation of any type of hacker.

After identifying B^{nb}, we can compute the equilibrium strategies of the firm and hackers from the FOC of (4) and (2). Lemma 2 summarizes the equilibrium outcomes.

Lemma 2. *Without a BBP, the equilibrium effort of any hacker of type t is $x_t = X^{nb}\left(1 - \frac{c_h X^{nb}}{v_t}\right)$ when $t \leq B^{nb}$, where $X^{nb} = \frac{\tau}{c_h}$. The effort $x_t = 0$ for all $t > B^{nb}$. The hackers' equilibrium expected payoff when $t \leq B^{nb}$ is*

$$u_t^{nb} = v_t\left(1 - \frac{c_h X^{nb}}{v_t}\right)^2, \tag{6}$$

and zero when $t > B^{nb}$.

The equilibrium effort of the firm is $x_f = X^{nb}(c_h X^{nb} \sum_{j=1}^{B^{nb}} \frac{n_j}{v_j} + 1 - N_1 + N_{B^{nb}+1}) - d$, and its equilibrium expected payoff is

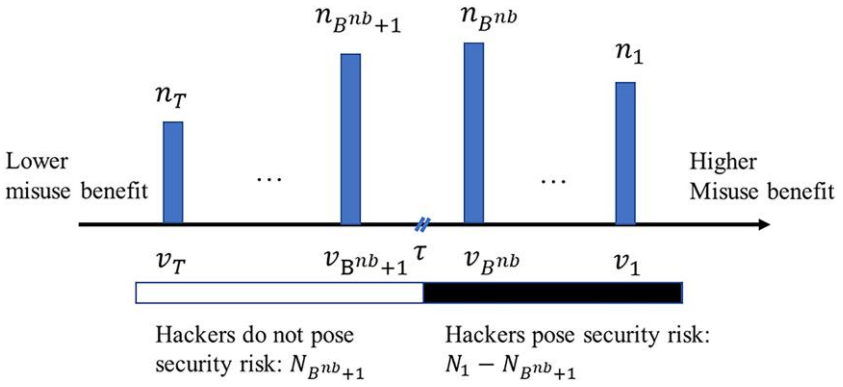
$$\pi_f^{nb} = c_f\left[d - X^{nb}\left(c_h \sum_{j=1}^{B^{nb}} \frac{n_j}{v_j} X^{nb} + 2 - N_1 + N_{B^{nb}+1}\right)\right]. \tag{7}$$

3.2. Equilibrium with a BBP

In addition to exerting effort to compete with hackers in finding the vulnerability, the firm could offer a bug bounty reward, r, to entice a hacker who finds the vulnerability first to submit it to the firm instead of misusing it. The incentive compatibility constraint of any hacker of type t requires that the reward from submission is not less than that from misusing the vulnerability. From (2), this means $\frac{r x_{t,k}}{d + x_f + \sum_{j=1}^{T}\sum_{g=1}^{n_j} x_{j,g}} - c_h x_{t,k} \geq \frac{v_t x_{t,k}}{d + x_f + \sum_{j=1}^{T}\sum_{g=1}^{n_j} x_{j,g}} - c_h x_{t,k}$ or $r \geq v_t$.

From the firm's perspective, by Lemma 1, $r > \tau$ because, when a hacker's valuation is less than τ, the hacker would not exert any effort, making the bug bounty reward unnecessary. Similarly, $r \leq v_1$ because v_1 is the highest valuation that any hacker can obtain. Any r beyond v_1 offers zero marginal benefit to the firm. Also, $r \leq \lambda$ because the firm should not pay an award that is larger than its maximum potential loss. Finally, the firm would not offer a bug bounty reward when even the hacker with the lowest valuation, $v_T > \lambda$ (i.e., the maximum loss it would suffer from exploitation). We assume that $v_T \leq \lambda$ to rule out this trivial case.

Figure 3. (Color online) Hacker Distribution with No BBP



Combining the constraints in r characterized above, the firm needs to choose an r within the range $(\tau, \min\{\lambda, v_1\}]$ in period 1. Denote B^M as the smallest index of hacker type that satisfies $v_{B^M} \le \lambda$ (i.e., the marginal hacker type that the firm is still willing to entice using a bug bounty reward). The firm's problem in period 1 has the following cases:

$$\max_{r\in(\tau,\min\{\lambda,v_1\}]} \pi_f^b(r) = \begin{cases} -\sum_{j<B^{nb}+1}\frac{\lambda n_j x_j}{X} - \sum_{j\ge B^{nb}+1}\frac{r n_j x_j}{X} - c_f x_f \text{ if } r\in(\tau, v_{B^{nb}}) \\ -\sum_{j<B^{nb}}\frac{\lambda n_j x_j}{X} - \sum_{j\ge B^{nb}}\frac{r n_j x_j}{X} - c_f x_f \text{ if } r\in[v_{B^{nb}}, v_{B^{nb}-1}) \\ -\sum_{j<B^{nb}-1}\frac{\lambda n_j x_j}{X} - \sum_{j\ge B^{nb}-1}\frac{r n_j x_j}{X} - c_f x_f \text{ if } r\in[v_{B^{nb}-1}, v_{B^{nb}-2}) \\ \cdots \\ -\sum_{j<B^M}\frac{\lambda n_j x_j}{X} - \sum_{j\ge B^M}\frac{r n_j x_j}{X} - c_f x_f \text{ if } r\in[v_{B^M}, min\{\lambda, v_1\}]. \end{cases} \tag{8}$$

The first term in each line of (8) is the expected net loss because of hackers misusing the vulnerability. The second term represents the expected net payout of the bug bounty reward when hackers submit the vulnerability. Traversing the cases downward, the first two terms suggest that, as r increases, more hackers are willing to submit the vulnerability to the firm than misusing it. Essentially, the bug bounty reward, r, sorts hackers into three groups with heterogeneous responses (cf. two groups when the BBP is not offered; see Figure 3).

- Cooperative hackers: When $v_T < \tau$, some hackers with low valuations from vulnerability misuse are incentivized by the bug bounty reward to help the firm find the vulnerability. All hackers with valuation $v_t \le \tau$ belong to this group.
- Coopetitive hackers: When $r > v_{B^{nb}}$, some hackers who would otherwise pose a security threat without a BBP now submit the vulnerability. Hackers with moderate misuse benefits, those with valuation $v_t > \tau$ and $v_t \le r$, belong to this group.
- Competitive hackers: When $r < v_1$, some hackers always pose a security threat regardless of whether a BBP is offered. Hackers with high misuse benefits, those with valuation $v_t > r$, belong to this group.

Figure 4 provides an illustrative sorting of hackers in the case with a BBP. We compute the equilibrium using backward induction. The following lemma summarizes the firm's and hackers' behaviors when a BBP is offered.

Lemma 3. *Denote B^O as the smallest index of hackers with $v_{B^O} \le r^*$, where the optimal reward is defined implicitly by $r^* = r(B^{nb}+1)$, $r(B^{nb}+1)$ being the solution of the dynamic programming problem $r(i) = \arg\max_{r\in\{r_i, r(i-1)\}}\pi_f^b(r)$ with the initial condition $r(1) = v_1$, and r_i and $\pi_f^b(r)$ are defined in the Online Appendix.*[12] *The equilibrium effort of any hacker of type $t < B^O$ is $x_t = X^b(1 - \frac{c_h X^b}{v_t})$ and of type $t \ge B^O$ is $x_t = X^b(1 - \frac{c_h X^b}{r^*})$, where $X^b = \frac{(N_1 - N_{B^O})\lambda + N_{B^O} r^*}{c_f + N_{B^O} c_h + \lambda c_h \sum_{j=1}^{B^O-1}\frac{n_j}{v_j}}$. The hackers' equilibrium expected payoff is*

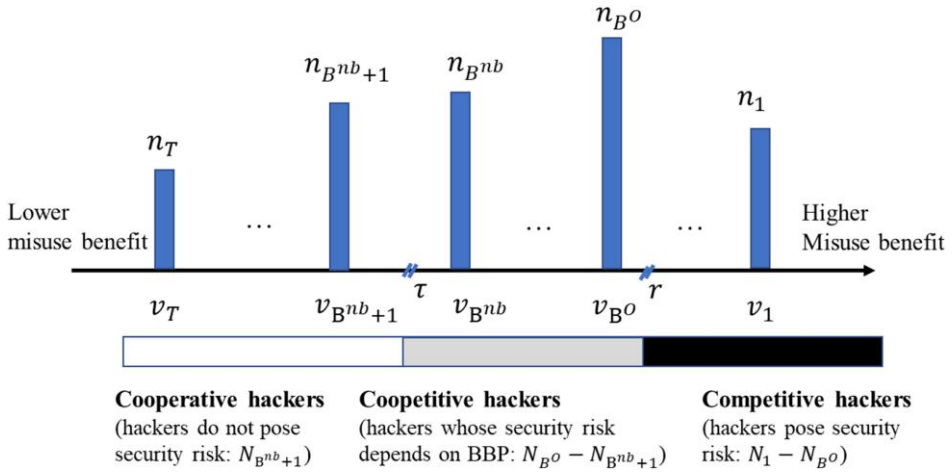
$$u_t^b = \begin{cases} v_t\left(1 - \frac{c_h X^b}{v_t}\right)^2, \text{ when } t < B^O \\ r^*\left(1 - \frac{c_h X^b}{r^*}\right)^2, \text{ when } t \ge B^O. \end{cases} \tag{9}$$

The equilibrium effort of the firm is $x_f = X^b[c_h(\sum_{j=1}^{B^O-1}\frac{n_j}{v_j} + \frac{N_{B^O}}{r^*})X^b + 1 - N_1] - d$, and its equilibrium expected payoff is

$$\pi_f^b = c_f\left(d - X^b\left[c_h\left(\sum_{j=1}^{B^O-1}\frac{n_j}{v_j} + \frac{N_{B^O}}{r^*}\right)X^b + 2 - N_1\right]\right). \tag{10}$$

Because the firm's payoff in (8) is discontinuous when r exceeds v_i (which changes the firm's expected cost from λ to r), the optimal r can be computed from a dynamic programming problem, namely, we must

Figure 4. (Color online) Hacker Distribution with BBP



solve a concave maximization problem across cases of the firm's profit function in (8).[13] We present an efficient algorithm to obtain the optimal r with time complexity $O(B^{nb})$ in the proof of Lemma 3.

The purpose of BBP contradicts the commonly held deterrence view of security protection in Section 3.1; instead of unequivocally deterring hacking and hoping to reduce the number of hackers, a BBP might incentivize more cooperative and coopetitive hackers to search for the vulnerability. A moderate degree of hacking by the right group of people might reduce a firm's security risk when a BBP is used.

4. Impact of the Bug Bounty Program

Comparing the firm's payoffs, π_f^{nb} from (7) and π_f^b from (10), the first proposition follows.

Proposition 1. *A BBP improves a firm's payoff if and only if*

i. *The efficiency of the firm in finding the vulnerability is sufficiently low relative to hackers (i.e., $\frac{c_f}{c_h} > \hat{c}$), where $\hat{c}$ is the unique solution to $\hat{c} - \frac{2(N_1 - N_{B^{nb}+1})\lambda \sum_{i=1}^{B^{nb}} \frac{n_i}{v_i}}{\hat{c} + \lambda \sum_{i=1}^{B^{nb}} \frac{n_i}{v_i}} + N_1 - N_{B^{nb}+1} - 2 + \lambda \sum_{i=1}^{B^{nb}} \frac{n_i}{v_i} = 0$.*

ii. *The market contains a sufficiently large portion of coopetitive hackers, that is, there exist hacker type(s) $t_0 \le B^{nb}$ such that $\frac{n_{t_0}}{N_{t_0}}\lambda - v_{t_0} + \frac{1}{N_{t_0}} E_{t_0} \ge 0$,[14] where $E_{t_0} \ge v_{t_0} N_{t_0} - n_{t_0}\lambda$ whenever $\frac{n_{t_0}}{N_{t_0}}\lambda - v_{t_0} \ge 0$. A sufficient condition is $\frac{n_{t_0}}{N_{t_0}}\lambda - v_{t_0} \ge 0$.*

The first condition in Proposition 1 is expected because it represents typical motivation for a firm to outsource a task (Quinn and Hilmer 1994). What is interesting in Proposition 1 is the second condition, which suggests that as long as a sufficient number of hackers is coopetitive, the BBP benefits the firm irrespective of the relative efficiency between the firm and hackers (note that the sufficient condition $\frac{n_{t_0}}{N_{t_0}}\lambda - v_{t_0} \ge 0$ does not depend on the cost parameters). Furthermore, in contrast to the common belief that a larger crowd enhances crowdsourcing effectiveness, here, more BBP participants might make a BBP less attractive. When the first condition in Proposition 1 is not met, more cooperative hackers, which increase N_{t_0} without affecting n_{t_0}, make the second condition less likely to hold.

To understand Proposition 1, we characterize two key benefits of BBPs. Rearranging the firm's payoff with and without BBP,

$$\pi_f^{nb} = -\sum_{i=1}^{B^O-1} \lambda \frac{n_i x_i}{d + x_f + \sum_{j=1}^{B^{nb}} n_j x_j} - \sum_{i=B^O}^{B^{nb}} \lambda \frac{n_i x_i}{d + x_f + \sum_{j=1}^{B^{nb}} n_j x_j} - c_f x_f, \tag{11}$$

$$\pi_f^{b} = -\sum_{i=1}^{B^O-1} \lambda \frac{n_i x_i}{d + x_f + \sum_{j=1}^{T} n_j x_j} - \underbrace{\sum_{i=B^O}^{B^{nb}} r \frac{n_i x_i}{d + x_f + \sum_{j=1}^{T} n_j x_j}}_{\text{attack diversion+protection delegation}} - \underbrace{\sum_{i=B^{nb}+1}^{T} r \frac{n_i x_i}{d + x_f + \sum_{j=1}^{T} n_j x_j}}_{\text{protection delegation}} - c_f x_f, \tag{12}$$

where B^O denotes the smallest index such that $v_{B^O} \le r$. The BBP changes the firm's payoff in two ways.

First, it incentivizes cooperative hackers indexed by $B^{nb}+1,\ldots,T$ to find the vulnerability. Their efforts add to the denominators in (12), namely, $d+x_f+\sum_{j=1}^{B^{nb}} n_j x_j$ becomes $d+x_f+\sum_{j=1}^{T} n_j x_j$. Because $r \geq v_t$ for coopetitive hackers, they exert more effort to compete with competitive hackers, who now face greater competition from other hackers. The firm's security risk is reduced; the first term in (12) decreases as the sum of efforts in the denominator increases. Meanwhile, a new cost in the third term in (12) appears because of an expected bug bounty payment to cooperative hackers.

Conceptually, this effect of a BBP is similar to delegating protection to cooperative and coopetitive hackers. We call this effect protection delegation. The less efficient a firm is at vulnerability identification than the hackers ($\frac{c_f}{c_h} > \hat{c}$), the more profitable such delegation becomes. This captures the conventional outsourcing wisdom: organizations with relatively low in-house efficiency can leverage the crowd to increase task efficiency (Terwiesch and Xu 2008). One difference here, however, is that the firm uses the crowd to combat competitive hackers instead of simply completing an outsourced task.

Another departure from conventional crowdsourcing is reflected in the second term of (11) and (12). Without the BBP, coopetitive hackers would be attackers inflicting damages, λ, on the firm if they had found the vulnerability first. The BBP reward, r, reduces such potential damage. Note that $v_t \leq r \leq \lambda$ for coopetitive hackers, that is, the bug bounty reward, r, is greater than their gain from misusing the vulnerability but smaller than the firm's loss, λ, when the vulnerability is misused. This suggests that the firm can take advantage of the gap between λ and v_t, where $t \in \{B^O,\ldots,B^{nb}\}$, and offer a bug bounty reward r to entice coopetitive hackers. This leads to a Pareto improvement for the firm and coopetitive hackers because the firm does not need to suffer a loss because of vulnerability misuse, and coopetitive hackers can receive a higher payoff. We call this benefit of the BBP attack diversion.

The second condition of Proposition 1 is more likely to hold when $\frac{n_{t_0}}{N_{t_0}}$ and the difference between λ and v_{t_0} are large. A large $\frac{n_{t_0}}{N_{t_0}}$ means that, prior to having the BBP, a large portion of hackers would already be looking for the vulnerability (recall $t_0 \leq B^{nb}$), but their valuation from misuse, v_{t_0}, is small relative to λ. In such a case, the firm might benefit from the BBP even when hackers are not more efficient at vulnerability discovery because the key benefit of the BBP lies in diverting the threat from this large group of coopetitive hackers.

It is important to recognize that, although the firm always has an incentive to offer a bug bounty reward $r \geq v_t$ provided that $v_t < \lambda$, it must balance the benefit from dissuading coopetitive hackers from attacking it and the influence on hackers with lower misuse benefit because the firm offers the same BBP reward to all hackers. The reward induces hackers with lower valuations, who would otherwise not search for the vulnerability, to exert more effort to pursue the bug bounty reward and increase the firm's expected bug bounty payment. Increased effort from these low-valuation (cooperative) hackers helps reduce the expected security loss, but the net benefit to the firm is ambiguous because of the extra cost incurred for the bug bounty payment. When cooperative hackers are inefficient but sizeable, their large size can undermine the return on investment from the BBP and renders it unprofitable.

Some large organizations, such as Oracle, criticize BBPs, whereas others, such as Alphabet and Meta, choose to adopt them. Proposition 1 can help reconcile this discrepancy. Oracle offers paid enterprise software that is accessible mostly to users with experience and skills. Hackers with such specialized knowledge might have high valuations from misusing the vulnerability (i.e., when v_{t_0} is large), and the size of such hackers might be small relative to the entire hacker distribution (i.e., $\frac{n_{t_0}}{N_{t_0}}$ is small). Furthermore, companies such as Oracle might have more resources to engage top-tier security experts with high efficiency to safeguard their systems. Accordingly, both conditions in Proposition 1 are less likely to hold, and this might explain why it does not favor BBPs.

By contrast, the products from Alphabet and Meta and similar online platforms, are often used in the public domain. They may attract significant numbers of casual hackers actively searching for vulnerabilities in their customer-facing products even without a BBP (i.e., $\frac{n_{t_0}}{N_{t_0}}$ is large). If many of these hackers are coopetitive and willing to trade the vulnerability finding for a bounty reward, the benefit of the BBP can be high because the cost of the bounty reward can be substantially lower than the cost of vulnerability misuse (Herley and Florêncio 2010). In such a case, the BBP is beneficial regardless of hacker efficiency.

We end the discussion of Proposition 1 by highlighting its relevance to some BBP phenomena that are not readily explained by existing crowdsourcing theories.

The first is underpaid hackers. If no hacker had a misuse benefit below v_{t_0}, then $n_{t_0} = N_{t_0}$, and hence, $\frac{n_{t_0}}{N_{t_0}}\lambda - v_{t_0} \geq 0$ would hold for any $v_{t_0} < \lambda$, suggesting that a firm is willing to offer a bug bounty reward to entice type t_0 hackers whenever $v_{t_0} < \lambda$. The presence of hackers with lower misuse benefits (i.e., the presence of the $\frac{n_{t_0}}{N_{t_0}}$ term), however, limits a firm's incentive to do so. This is a kind of pecuniary externality imposed by low-valuation hackers on other hackers; the mere presence of low-valuation hackers limits the firm's incentive to

offer larger bounty rewards that might otherwise have diverted more high-valuation (coopetitive) hackers.

Because of this pecuniary externality, a firm might underpay bug bounty rewards, leaving hackers with $v_t \in (r, \lambda)$ more willing to misuse the vulnerability than cooperating with the firm despite the fact that they are not too expensive to divert. Hackers often complain about low BBP payments (Krebs 2013). For example, "various BBPs have been criticized over the years for issues ranging from allegedly underpaying researchers … out of this frustration, researchers have publicly published apparent zero-days and sold their exploits to third parties such as zero-day brokers" (Culafi 2021, para. 2). Uber once paid $100,000 when extorted by a hacker for a vulnerability that could have led to a data breach, and yet the highest listed reward on Uber's public BBP is $10,000 (Perlroth and Isaac 2018).

Next is bug bounty reward discrimination. The pecuniary externality provides an interesting implication for BBP design. In general, the higher the v_{t_0}, the less likely $\frac{n_{t_0}}{N_{t_0}}\lambda - v_{t_0} \geq 0$ holds, suggesting that offering $r \geq v_{t_0}$ would more likely stimulate low-valuation hackers' excessive effort that might erode the firm's profit. In other words, when a firm offers a large bug bounty reward to gain the attack diversion benefit, the bug bounty payment itself might become a significant cost. The firm should manage the (unintended) payment to low-valuation hackers by disincentivizing them from exerting too much effort. One possibility is price discrimination; the firm might promise different award levels to different types of hackers by self-selection. For example, Apple's BBP, which offers rewards up to one million dollars, states that "reports that include a basic proof of concept instead of a working exploit are eligible to receive no more than 50% of the maximum payout amount" (Apple 2019). The discriminatory bug bounty reward policy might dissuade low-value hackers from exerting too much effort, reducing the pecuniary externality that they inflict on higher value (coopetitive) hackers.[15]

From this point onward, we assume that the condition described in Proposition 1 holds; that is, it is beneficial for the firm to offer a BBP.

4.1. Security Impact

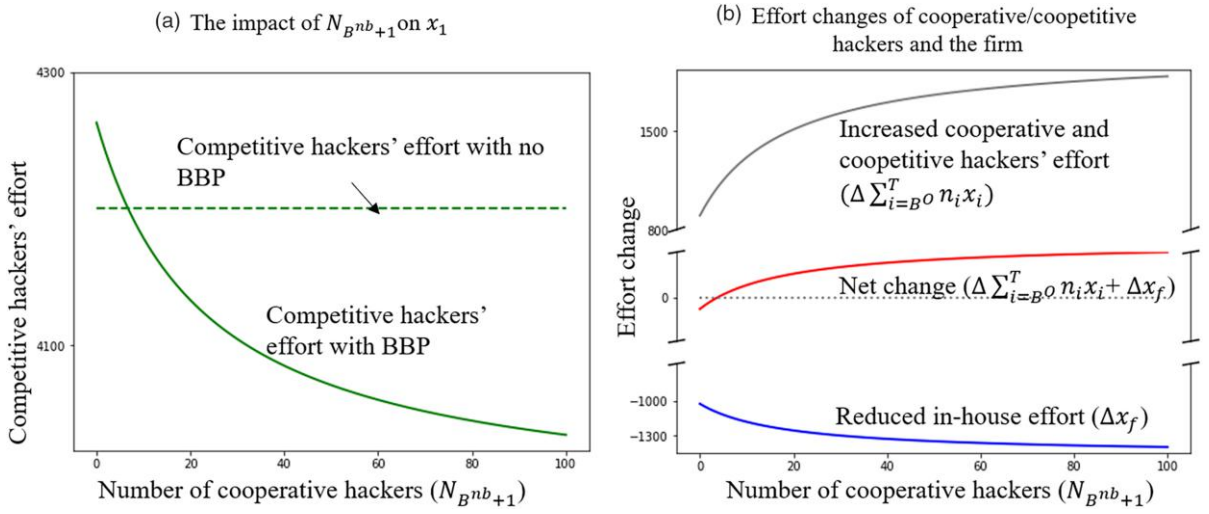
We now examine the security implications of a BBP. Let S denote the probability of vulnerability misuse; $1 - S$ then represents the security of the system. To examine the security with and without a BBP, we compare $S^{nb} = \sum_{i=1}^{B^{nb}} \frac{n_i x_i}{d + x_f + \sum_{j=1}^{B^{nb}} n_j x_j}$ and $S^b = \sum_{i=1}^{B^O - 1} \frac{n_i x_i}{d + x_f + \sum_{j=1}^{T} n_j x_j}$. From the analysis in Proposition 1, a BBP incentivizes coopetitive hackers not to attack the firm (i.e., eliminating $\sum_{i=B^O}^{B^{nb}} \frac{n_i x_i}{d + x_f + \sum_{j=1}^{T} n_j x_j}$ from the vulnerability misuse risk) and both cooperative and coopetitive hackers to compete against the competitive hackers (i.e., adding the effort terms $\sum_{j=B^{nb}+1}^{T} n_j x_j$ to the denominator of the vulnerability misuse risk). These two factors improve the security.

Nevertheless, changes in coopetitive and cooperative hackers' behaviors distort competitive hackers' and the firm's incentives. From the Tullock function in (1), the firm's effort is a strategic substitute of cooperative and coopetitive hackers' efforts under the BBP. Referring to the firm's payoff function in (8), its marginal benefit from in-house effort under a BBP, $\frac{\partial \pi_f^b}{\partial x_f} = \frac{1}{d + x_f + \sum_{j=1}^{T} n_j x_j} \left(\sum_{v_i > r} \lambda \frac{n_i x_i}{d + x_f + \sum_{j=1}^{T} n_j x_j} + \sum_{v_i \leq r} r \frac{n_i x_i}{d + x_f + \sum_{j=1}^{T} n_j x_j} \right) - c_f$, depends on two factors: (i) the difficulty of finding the vulnerability itself, $\frac{1}{d + x_f + \sum_{j=1}^{T} n_j x_j}$, and (ii) the expected cost when the vulnerability is found first by hackers in terms of either vulnerability misuse cost or bug bounty payment cost, $\sum_{v_i > r} \lambda \frac{n_i x_i}{d + x_f + \sum_{j=1}^{T} n_j x_j} + \sum_{v_i \leq r} r \frac{n_i x_i}{d + x_f + \sum_{j=1}^{T} n_j x_j}$. The bug bounty reward induces greater competition among hackers, and this increases the difficulty for the firm to find the vulnerability first. Because of the attack diversion and protection delegation benefits, it also decreases the expected cost of the firm when hackers find the vulnerability first. Both factors incentivize the firm to reduce its in-house effort.

Competitive hackers now face more competition from coopetitive and cooperative hackers but less competition from the firm. The net influence of a BBP on their effort depends on the balance of these two opposing forces. Figure 5 shows the equilibrium effort of competitive hackers in an illustrative market with three types of hackers, $T = 3$.[16] When the number of cooperative hackers is small and the bug bounty reward is not high, a BBP does not induce significant competition among the hackers. However, the firm would reduce its in-house effort because of reduced threats from coopetitive hackers. Accordingly, the competitive hackers might respond by exerting even more effort than when the BBP is absent because its chance of finding the vulnerability first can be higher now. This incentive of competitive hackers decreases as the number of cooperative hackers increases, and this increases the competitive intensity in finding the vulnerability because of the protection delegation effect.[17] In Online Appendix C.1, we present the exact conditions for competitive hackers to increase or decrease their efforts.

Proposition 1 suggests that, when too many hackers with low misuse benefits expend effort to find the vulnerability, the firm's payoff can decrease because of the bug bounty payment. These hackers' effort, however, always improves system security because they compete with competitive hackers and, thus, discourage competitive hackers from expending effort. The implication is that, when an organization cares more about security

Figure 5. (Color online) The Impact of $N_{B^{nb}+1}$ on Competitive Hackers' Effort (x_1)



Notes. Competitive hackers ($v_1 = 210{,}000$, $n_1 = 2$), coopetitive hackers ($v_2 = 150{,}000$, $n_2 = 20$), cooperative hackers ($v_3 = 0$, varying n_3), $c_h = c_f = 10$, $d = 3$, $\lambda = 210{,}000$, $B^{nb} = 2$, $r^* = 150{,}000$. The y-axis of panel (b) is broken to reflect variability across effects. Δ represents the change in values because of the BBP.

than economic benefits, the pecuniary externality described in Proposition 1 is moot. The BBP can be designed to entice as many cooperative hackers as possible. For example, the U.S. Department of Defense (2018) increased the reach of its BBP, "Hack the Pentagon," by inviting registered hackers on multiple bug bounty platforms such as HackerOne and Bugcrowd. Our analysis suggests that this might be security-driven rather than economically motivated.

Figure 6 shows how the various effects of BBP vary with the number of cooperative hackers. They collectively raise the system's security.

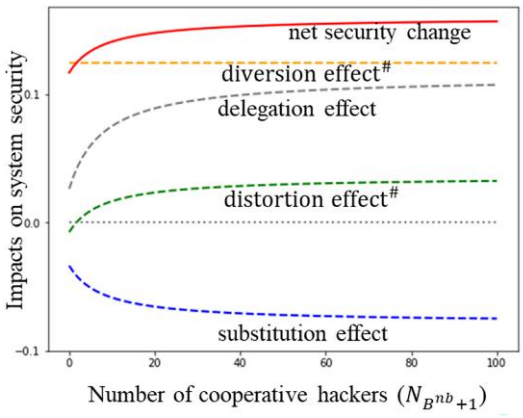
Proposition 2. *The system's security increases with a BBP.*

Despite the strategic substitution induced by the attack diversion and protection delegation benefits, Proposition 2 suggests that the firm retains sufficient in-house effort to ensure enhanced overall system security. In other words, unlike typical crowdsourcing, the firm should continue to compete with hackers, leveraging them for vulnerability identification using the BBP. The point of departure is that the bounty payment (i.e., the second term in (8)) decreases with the firm's effort, creating an incentive for the firm to use its own effort to economize the BBP payment.

4.2. Social Welfare

We now consider social welfare. In line with the cybersecurity literature (e.g., August and Tunca 2011,

Figure 6. (Color online) Different Effects on System Security



Effects	Representations
Diversion effect	$\sum_{i=B^o}^{B^{nb}} \frac{n_i x_i}{X}$ (removed coopetitive hackers' risk)
Delegation effect	$\Delta \sum_{i=B^o}^{T} \frac{n_i x_i}{X}$ (increased probability that the vulnerability is first found by cooperative and coopetitive hackers)
Substitution effect	$\Delta \frac{x_f}{X}$ (decreased probability that the vulnerability is first found by the firm)
Distortion effect	$\Delta \sum_{i=1}^{B^o-1} \frac{n_i x_i}{X}$ (increased/decreased probability that the vulnerability is first found by competitive hackers)
Net security change: $(1 - S^b) - (1 - S^{nb})$	

Notes. # indicates that we use positive (negative) values to represent positive (negative) effects on security. Parameters are the same as those used in Figure 5. Δ represents the change in values because of a bug bounty program. X represents X^{nb} or X^b in the corresponding scenarios.

Kannan et al. 2016, August et al. 2022), we assume society prefers minimizing security losses and all effort costs:[18]

$$W^b = -\underbrace{\sum_{v_i>r\lambda} \frac{n_i x_i}{d + x_f + \sum_{j=1}^{T} n_j x_j}}_{\text{security loss}} - \underbrace{c_f x_f}_{\text{in-house effort cost}} - \underbrace{\sum_{i=1}^{T} c_h n_i x_i}_{\text{hackers' effort cost}} \equiv \pi_f^b + \sum_{v_i \le r} n_i u_i^b - \sum_{v_i>r} c_h n_i x_i. \tag{13}$$

A bounty reward does not enter social welfare because it is a transfer payment from the firm to the winning hacker. However, it indirectly affects social welfare by influencing the security loss and effort costs. An excessive bug bounty reward reduces security loss but raises hackers' effort costs, and this may potentially lower social welfare, suggesting that the socially optimal bounty reward is finite.

Assume that a social planner can optimize the bounty reward and in-house effort to maximize social welfare. Comparing the firm's choice with that of the social planner, see the following proposition.

Proposition 3. *In a BBP, the firm offers (weakly) insufficient bug bounty rewards. There exists a threshold, $c_w > 0$, for which the firm's in-house effort is socially efficient only when $c_f = c_w$. Its in-house effort is excessive when $c_f > c_w$ and insufficient when $c_f < c_w$.*

A bounty reward provides greater benefits to society than the firm. Comparing W^b and π_f^b in (13), the firm does not internalize two positive externalities of the bounty reward. First, a reward diverts some hackers into nonharmful activities (additional u_i^b in $\sum_{v_i \le r} n_i u_i^b$). Second, a higher reward improves the payoffs of cooperative and coopetitive hackers (each u_i^b in $\sum_{v_i \le r} n_i u_i^b$ increases with r), and it reduces competitive hackers' effort too (reduced $\sum_{v_i>r} c_h n_i x_i$). The firm tends to undersupply bug bounty rewards because of these two externalities.

Comparing the payoff of the social planner in (13) with that of the firm in (12), two disparities determine the in-house effort inefficiency: (i) A bounty payment directly reduces the firm's payoff but not that of the social planner. The incentive to reduce a bounty payment leads the firm to exert excessive in-house effort. (ii) The social planner accounts for all effort costs, but the firm only focuses on its own effort. When its efficiency is high relative to the hackers', the social planner prefers the protection effort from the firm than the hackers. The firm, however, ignores this social benefit, leading to insufficient in-house effort. Similarly, when the firm's efficiency is low relative to the hackers, its effort becomes excessive because more effort from hackers would save protection effort costs. Taken together, the firm exerts excessive effort when its efficiency is low and insufficient effort when its efficiency is high.

Market efficiency has always been central to the security outsourcing literature. Yet the inefficiency identified here is unique. In typical security-outsourcing scenarios (e.g., Lee et al. 2013, Hui et al. 2019), the firm undersupplies protection because it does not bear the full consequence of a security breach; instead, the managed security service provider absorbs part of the loss through compensation. In a BBP, this incentive to undersupply effort is absent because the firm suffers the full loss of a security breach. Yet it can either oversupply or undersupply security effort and reward because it does not internalize two novel social benefits of BBP: diverting some hacking into nonharmful activities and saving protection costs by reallocating security efforts between the firm and hackers.

We explore policies to increase social welfare. The BBP has two unique characteristics. First, unlike conventional contexts that typically feature insufficient protection (e.g., in security investment, e.g., Anderson and Moore 2006; outsourcing, e.g., Lee et al. 2013, Hui et al. 2019; and software patching, e.g., August and Tunca 2011), BBPs might lead a firm to oversupply effort. Addressing such inefficiency and its implications on existing cybersecurity policies is largely unexplored. Second, the inefficiencies in bounty rewards and the firm's in-house effort may call for different policies.

We focus on two broad categories of policies. The first directly increases the benefit for the firm to offer a large bounty reward. Examples include tax reductions and direct subsidies for bounty payments. For example, the European Union has sponsored BBPs for some open-source software since 2017 (European Commission 2019). This can be incorporated by representing the firm's bounty payment as $\sum_{v_i \le r}(r-s)\frac{n_i x_i}{d+x_f+\sum_{j=1}^{T} n_j x_j}$ in (8), in which s represents a reward subsidy or tax reduction. Another approach involves indirectly influencing the firm's incentive by penalizing or subsidizing data breaches. For example, many countries impose additional fines for data breach. This can be modeled by representing the firm's security loss as $\sum_{v_i>r}(\lambda+l)\frac{n_i x_i}{d+x_f+\sum_{j=1}^{T} n_j x_j}$ in (8), in which l represents a breach penalty or subsidy (when $l < 0$). We focus on these two policies because of their practicality, requiring minimal changes to existing practices. We can obtain other (more complex) approaches—such as collecting taxes to fund BBPs—by combining these two approaches. The key question is, should the government subsidize bounty rewards (or provide equivalent tax reductions), impose breach

penalties, or subsidize data breaches to induce more efficient BBP use?

Proposition 4. *A breach penalty is the preferred policy for social welfare when the firm's effort efficiency is high ($c_f < c_w$), whereas a bounty subsidy is the preferred policy for social welfare when the firm's effort efficiency is low ($c_f > c_w$).*

Table 2 summarizes the impacts of different policies on social welfare and inefficiencies. When the firm's effort efficiency is low, bounty subsidies align with the social planner's objectives. When the firm's effort efficiency is high, breach penalties help align the objectives. Breach subsidies always exacerbate reward inefficiency. The potential for bounty subsidies (breach penalties) to improve social welfare when breach penalties (bounty subsidies) reduce social welfare suggests their complementary relationship in addressing BBP inefficiencies.

A bounty subsidy affects the firm's incentive in two ways. First, it lowers the cost of paying a bounty reward, making the firm more willing to offer a higher reward. This aligns with the social planner's bounty reward preference. Second, the lower cost of paying cooperative and coopetitive hackers when they find the vulnerability first induces the firm to lower its in-house effort. This aligns with the social planner's protection effort preference when the firm is relatively inefficient. When the firm is efficient, reduced in-house effort exacerbates the protection effort inefficiency, which can reduce social welfare. Kannan and Telang (2005) advocate government-funded vulnerability reporting to improve social welfare without endogenizing firms' protection efforts. Our analysis suggests that such reporting could reduce social welfare when the firm has high protection efficiency.

Breach penalties are common in cybersecurity regulation, but their implications on social welfare with BBPs are not trivial. A breach penalty increases the firm's expected security loss and, in turn, the marginal benefit of a bounty reward, leading the firm to offer a higher reward. It can, thus, increase reward efficiency. Nevertheless, it raises the firm's costs when competitive hackers find the vulnerability first, and this encourages the firm to increase in-house effort. This increases protection effort efficiency when the firm is relatively efficient, but it can exacerbate protection effort inefficiency in other cases. Conversely, although a breach subsidy increases effort efficiency when the firm's efficiency is low, it lowers the firm's incentive to offer a high reward, hence exacerbating the firm's reward inefficiency.

Our findings highlight the need to review existing cybersecurity regulations to account for BBPs, particularly by considering a firm's capability. For example, current U.S. cybersecurity laws seldom consider companies' BBP adoption when levying breach fines. Our analysis suggests that it can be counterproductive if uniform penalties are applied to companies with BBPs but poor cybersecurity capabilities. In contrast, under the General Data Protection Regulation (GDPR), adoption of BBPs by companies can lead to reduced fines for data breach (HackenProof 2023). Although differential treatment is necessary, we advocate for a more tailored approach that considers firms' cybersecurity capabilities because leniency to competent companies might reduce social welfare. Furthermore, although most governments rely on reactive measures, such as imposing penalties after breaches occur, they should recognize the limitations of these measures and explore proactive approaches, such as selective BBP subsidies, as complementary policies.

5. The Optimal Bounty Reward

We perform a comparative static analysis to understand the properties of the bounty reward (i.e., r^* in Lemma 3). We use the main model to draw general conclusions. To avoid cluttered details and better highlight the key trade-offs and findings, we consider the following stylized scenario with three types of hackers.[19]

There are three types of hackers indexed by H, M, and L. Their respective sizes are n_H, n_M, and n_L and have valuations v_H, v_M, and v_L from misusing the vulnerability. Furthermore, $v_H > \lambda$, $v_M \in (\tau_s, \lambda)$, and $v_L \le \tau_s$, where $\tau_s = \frac{\lambda c_h(n_H+n_M)}{c_f+\lambda c_h(\frac{n_H}{v_H}+\frac{n_M}{v_M})}$ from Lemma 1.

This stylized model captures the essential characteristics of three hacker groups. As cooperative hackers, L-type hackers have a low misuse benefit and are incentivized by the bounty reward to help the firm find the vulnerability. M-type hackers have a moderate misuse benefit (which can be considered as the average black market price for the vulnerability), posing a security threat without BBP though they might participate in

Table 2. The Welfare Impact of Different Policies

Scenarios	Bounty subsidies	Breach penalties	Breach subsidies
$c_f < c_w$	Exacerbate effort inefficiency	Mitigate reward and effort inefficiency	Exacerbate both reward and effort inefficiency
$c_f > c_w$	Mitigate reward and effort inefficiency	Exacerbate effort inefficiency	Exacerbate reward inefficiency

Note. When $c_f < c_w$ ($c_f > c_w$), breach penalties (reward subsidies) increase social welfare, but alternative policies can reduce it.

the BBP if the reward is attractive. As competitive hackers, H-type hackers are unlikely to participate in the BBP because of their high misuse benefit.

Proposition 5. *The optimal bounty reward can increase or decrease with an increasing number of cooperative hackers. In the stylized scenario, the influence of increased cooperative hackers on the optimal bounty reward in various scenarios is reported in Table 3. The optimal bounty reward (weakly) increases with the breach cost, lower in-house efficiency relative to the hackers, and increased number of competitive hackers.*

Figure 7 shows how the optimal bounty reward changes with the number of L-type (cooperative) hackers. Factors such as general interest in BBPs and the firm's marketing and promotion of a BBP can influence the number of cooperative hackers.

Conventional crowdsourcing wisdom suggests that the optimal reward increases with participant size (e.g., Ales et al. 2017, Liu et al. 2018). However, Proposition 5 suggests that this principle applies only when a firm's efficiency and v_M are low as shown in Figure 7(a). Here, by offering a small reward, the firm can attract both L- and M-type hackers, deriving protection delegation and attack diversion benefits from the BBP. As the number of cooperative hackers increases, their overall capability grows, incentivizing the firm to invest more in the BBP by increasing the reward.

However, as Figure 7, (b) through (d), shows, in a broad range of scenarios, conventional crowdsourcing results do not apply because the firm's bounty reward might become insensitive, decrease, or change nonmonotonically as the number of cooperative hackers increases. First, in cases (b) through (d), the firm's reward is insensitive to increased L-type hackers when their numbers are small. With a small number of L-type hackers, the pecuniary externality described in Proposition 1 is not salient, so the firm is willing to offer a relatively large reward $r = v_M$ to obtain the attack diversion benefit. Yet, unlike case (a), the firm has little incentive to increase the reward beyond v_M because the high reward (cases (b) and (c)) or hackers' low efficiency (case (d)) make a large bounty reward influence the firm's payoff negatively ($\frac{\partial \pi_f^b}{\partial r} < 0$). Therefore, the optimal bounty reward is $r = v_M$, which is sufficient for an attack diversion benefit, minimizing the bounty cost.

Although the firm uses a similar reward strategy in cases (b) through (d) with a small number of L-type hackers, it should apply different strategies when the number of L-type hackers is large. Specifically, with a high v_M and efficient hackers, the firm would decrease reward initially and then increase it as the number of L-type hackers increases (Figure 7(b)). Why would increased L-type hackers influence the optimal reward in opposite ways in different ranges of n_L? A high v_M renders a large bounty reward, $r = v_M$, economically unsustainable because the bounty payment can become excessive with more L-type hackers. Once the number of L-type hackers reaches a threshold, the firm should first lower the bounty reward to save bounty costs and focus on the protection delegation benefit. However, if the number of L-type hackers increases further, the firm should increase the reward again as in case (a) (i.e., L-type hackers' increased overall capability). However, the reward will never return to the previous peak (i.e., v_M) because of the financial burden of rewarding the large pool of L-type hackers. In other words, the firm's bounty reward tends to be larger when the number of cooperative hackers is small.

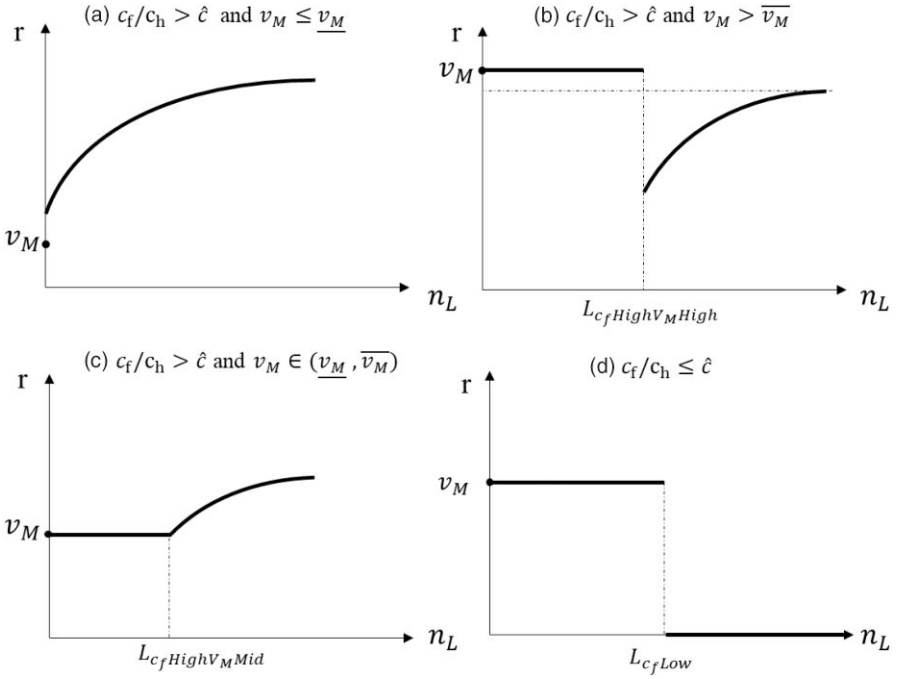
In contrast, if v_M is moderate and hackers are relatively efficient, the firm's bounty reward is larger when the number of cooperative hackers is large as shown in Figure 7(c). Although a small number of L-type hackers does not affect the optimal reward, their increased size increases the firm's incentive to offer a larger reward, which can eventually exceed v_M if v_M is below the threshold $\bar{v_M}$. Beyond this point, more L-type hackers would further

Table 3. The Influence of L-Type Hackers on the Bounty Reward

Comparison with conventional crowdsourcing	Scenarios	Reward change with increased number of L-type hackers
Similar	$\frac{c_f}{c_h} \geq \hat{c}$ and $v_M \leq \underline{v_M}$	Increases
Different	$\frac{c_f}{c_h} \geq \hat{c}$ and $v_M > \overline{v_M}$	With a few L-type hackers ($L < L_{c_f High V_M High}$), the reward stays stable; with more L-type hackers ($L \geq L_{c_f High V_M High}$), the reward first decreases then increases. The reward level with many L-type hackers is never higher than with few.
	$\frac{c_f}{c_h} \geq \hat{c}$ and $v_M \in (\underline{v_M}, \overline{v_M})$	The reward stays stable with a few L-type hackers ($L < L_{c_f High V_M Mid}$) and then increases with more L-type hackers ($L \geq L_{c_f High V_M Mid}$).
	$\frac{c_f}{c_h} < \hat{c}$	With a few L-type hackers ($L < L_{c_f Low}$), the reward stays stable; it drops to zero when $L \geq L_{c_f Low}$.

Notes. $\hat{c}$ is the unique solution of $\hat{c} - \frac{2(n_H+n_M)\lambda\left(\frac{n_H}{v_H}+\frac{n_M}{v_M}\right)}{\hat{c}+\lambda\left(\frac{n_H}{v_H}+\frac{n_M}{v_M}\right)} + n_H + n_M - 2 + \lambda\left(\frac{n_H}{v_H}+\frac{n_M}{v_M}\right) = 0$. The definitions of $\underline{v_M}$ and $\overline{v_M}$ are reported in the proof.

Figure 7. The Influence of L-Type Hackers on Bounty Rewards



Note. Definitions for $L_{c_f High V_M High}$, $L_{c_f High V_M Mid}$, and $L_{c_f Low}$ are reported in the Online Appendix.

drive the optimal reward above v_M. Comparing Figure 7, (b) and (c) (or (a)), and if we interpret v_M as the black market price for the vulnerability, one surprising implication is that, for firms that are otherwise similar, those with high black market vulnerability prices—potentially because of their connections to more valuable targets—tend to reduce their bounty rewards as the market attracts more cooperative hackers, whereas those with low vulnerability prices tend to increase their rewards.

Figure 7(d) shows that, sometimes, the firm's bounty reward drops abruptly from a high value to zero. As discussed in Proposition 1, inefficient cooperative hackers offer little protection delegation benefits and would, in fact, erode attack diversion benefits through pecuniary externality, gradually negating the BBP benefits as their number increases. We also find that breach costs, hackers' efficiency relative to the firm, and the number of competitive hackers have expected positive influences on a bounty reward. These factors heighten a firm's security risks, incentivizing it to raise the bounty reward to address such risks.

Proposition 5 highlights a nuanced relationship between the optimal reward and participant size in crowdsourcing with coopetition. Although extant studies examine how participant size affects rewards, none has considered the influence of coopetition on the optimal reward choice, which we show can lead the optimal reward to exhibit remarkably different properties. Our findings underscore the importance of accounting for coopetitive dynamics when optimizing rewards for BBPs and other similar crowdsourcing contests.

6. Extensions

Our model can serve as the foundation to analyze various BBP policies and regulation of settings involving coopetitive relationships. This section examines two prominent BBP policy designs: safe harbor terms and policies for managing duplicate report submissions. Online Appendix D discusses other extensions, including general cost functions, incorporating altruistic hackers, and a sequential decision setting.

6.1. Safe Harbor Terms

Although BBPs allow firms to benefit from hackers' efforts, traditional cybersecurity regulations often suppress hacking, including many security-testing activities that are necessary for vulnerability identification. For example, the Computer Fraud and Abuse Act (CFAA) in the United States makes it illegal to "intentionally access a computer without authorization or exceed authorized access, and thereby obtain … information from" almost any computer. The Computer Misuse Act of 1990 in the United Kingdom and China's criminal law contain similar provisions (e.g., Qi et al. 2009, Swinhoe 2020). These laws have long been criticized for their ambiguity regarding what constitutes authorized access, creating a challenging environment for hackers to participate in BBPs because of the potential risk of legal repercussions for their security research. As a canonical incident, the drone manufacturer DJI once threatened legal action under the CFAA against a hacker regarding a nondisclosure

agreement amid the hacker's efforts to report a vulnerability through DJI's BBP (Gallagher 2017).

One solution recommended by legal scholars and industry experts is to include an SHT in BBP policies to explicitly authorize ethical security testing (e.g., Thompson 2008, Elazari 2018). A typical SHT reads, "Safe Harbor: we will not threaten or bring any legal action against anyone who makes a good faith effort to comply with this BBP, or for any accidental or good faith violation of this policy. This includes any claim under the DMCA for circumventing technological measures to protect the services and applications eligible under this policy. As long as you comply with this policy: We consider your security research to be 'authorized' under the CFAA" (from Mozilla's BBP).[20] Many big companies, such as Amazon, GitHub, and Tesla, include SHTs in their BBPs though numerous other firms do not amid criticisms from researchers (Elazari 2018).

Is the SHT beneficial to a firm using BBPs? Our analysis so far assumes no legal risk in security testing, which is a reasonable simplification because people conducting vulnerability research have rarely been convicted criminally (e.g., Etcovitch and van der Merwe 2018).[21] We now extend the model to analyze an SHT's influence by assuming that testing a firm's system brings nontrivial legal risks, represented by a disutility, ξ.[22] A hacker's payoff function without an SHT becomes

$$u^{b}_{t,k}=\begin{cases}\dfrac{rx_{t,k}}{d+x_f+\sum_{j=1}^{T}\sum_{g=1}^{n_j}x_{j,g}}-\xi x_{t,k}-c_h x_{t,k} \text{ if } v_t\le r,\\ \dfrac{v_t x_{t,k}}{d+x_f+\sum_{j=1}^{T}\sum_{g=1}^{n_j}x_{j,g}}-\xi x_{t,k}-c_h x_{t,k} \text{ if } v_t> r,\end{cases} \tag{14}$$

which incorporates the key consideration that a hacker who tests a system ($x_{t,k}>0$) incurs a disutility $\xi x_{t,k}$ because of the legal risks. Ceteris paribus, this disutility term reduces the incentive for cooperative or coopetitive hackers to help a firm find vulnerabilities, a key reason many people recommend an SHT in a BBP.

An SHT introduces two changes. First, the firm exempts hackers from legal liabilities when they participate in the BBP ($\xi x_{t,k}$ disappears in the first scenario of (15)). Ideally, competitive hackers should still be held accountable for testing the firm's system, but as noted by some legal scholars (e.g., Thompson 2008), it is practically challenging for a firm to differentiate ethical testing from malicious ones because the firm observes only its systems being accessed by outsiders. We consider the payoff function

$$u^{bs}_{t,k}=\begin{cases}\dfrac{rx_{t,k}}{d+x_f+\sum_{j=1}^{T}\sum_{g=1}^{n_j}x_{j,g}}-c_h x_{t,k} \text{ if participate in}\\ \qquad\qquad \text{the BBP } (\rho(v_t)\le r),\\ \dfrac{v_t x_{t,k}}{d+x_f+\sum_{j=1}^{T}\sum_{g=1}^{n_j}x_{j,g}}-\alpha\xi x_{t,k}-c_h x_{t,k} \text{ if not}\\ \qquad\qquad \text{participate in the BBP}\\ \qquad\qquad (\rho(v_t)> r).\end{cases} \tag{15}$$

We introduce the parameter $\alpha\in[0,1]$ to model the change in legal risks when competitive hackers test the system. The extreme case $\alpha=1$ represents an ideal situation in which competitive hackers cannot benefit from a SHT, and so they face the same legal risks as before. The other extreme, $\alpha=0$ represents the case when the SHT broadly authorizes security research. For example, Disclose.io, a website that promotes SHTs, recommends a template: "When conducting vulnerability research, according to this policy, we consider this research conducted under this policy authorized concerning any applicable antihacking laws … "[23] The previous example from Mozilla mentions that the term applies when hackers "comply with this BBP." Limited provisions such as this increase α above zero because, if competitive hackers' misuse of a vulnerability is investigated and convicted, they might still face legal penalties associated with their research activities. Therefore, different values of α reflect variations in SHT design. We are interested in how α affects the benefit of BBPs.

Second, an SHT changes hackers' incentive compatibility constraints, reflected by $\rho(v_t)\le r$ and $\rho(v_t)>r$ in (15), in which $\rho(\cdot)$ is an endogenous discount function (the technical details are available in the proof of Proposition 6). When deciding whether to report or misuse a vulnerability, hackers need to consider the additional legal risk of $\alpha\xi x_t$ associated with misuse. We assume that hackers use the function $\rho(v_t)$ to discount their misuse benefit to decide whether to report a finding. Clearly, $\rho(v_t)<v_t$ whenever $\alpha>0$.

Proposition 6. *The firm's payoff from including an SHT in the BBP decreases as α decreases. There exists a threshold $\hat{\alpha}\in(0,1)$ such that an SHT increases the firm's payoff when $\alpha>\hat{\alpha}$. When $\alpha\le\hat{\alpha}$, an SHT increases the firm's payoff only when the firm's efficiency in finding the vulnerability is sufficiently low relative to hackers (i.e., $\frac{c_f}{c_h}$ is sufficiently large). It reduces the firm's payoff otherwise.*

Proposition 6 suggests that an SHT is beneficial only when α is close to one, and it might bring net harm otherwise. This might explain the paradox of why many companies are reluctant to adopt SHTs despite benefiting from ethical hacking and being criticized. First, firms face uncertainty regarding whether an SHT brings net benefits. Second, even if they realize that the benefits of SHTs depend on α (i.e., the SHT's influence on competitive hackers), they might anticipate a low α if they adopt a recommended SHT such as the one from Disclose.io.

An SHT brings direct benefits and indirect costs. The first benefit is intuitive: it incentivizes ethical hacking by mitigating the associated legal risks. Second, and more subtly, it can help a firm save bounty costs. The minimum bounty the firm needs to offer to attract type t hackers is $r=v_t$ without an SHT, but with an SHT, this minimum decreases to $r'=\rho(v_t)\le v_t$ as long as $\alpha>0$. In other words, an SHT provides the firm with a lever to conserve the bounty reward. However, the cost is

that an SHT might be abused by competitive hackers. When α is small, an SHT incentivizes competitive hackers just as it incentivizes cooperative and coopetitive hackers, negating its positive influence on ethical hacking. Moreover, if $\alpha \to 0$, then $\rho(v_t) \to v_t$, meaning that the firm would lose the lever to conserve the bug bounty reward when α is small. An SHT, thus, increases the firm's payoff mostly when α is large, and it can reduce the firm's payoff otherwise except for a small parameter range (i.e., when $\frac{c_f}{c_h}$ is sufficiently large).[24]

The key implication is that, to make SHTs beneficial to a firm, simply including authorizing language in BBP policies might not suffice; a firm should take proactive measures to ensure that α is sufficiently large. In other words, the firm should prevent hackers from interpreting SHTs as a general reduction of legal risks when probing vulnerabilities. Two strategies might increase α. One is informational. According to perceptual-deterrence theory (e.g., Apel 2013, Chalfin and McCrary 2017), when potential offenders (competitive hackers here) are made more explicitly aware of the legal consequences of their criminal actions (security testing associated with vulnerability misuse), the deterrence effect of the law strengthens. We conjecture that α would be higher if an SHT delivers a clear message about the consequences of security research associated with vulnerability misuse. Meta's BBP includes explicit terms that certain research activities disqualify one from SHT protection: "You make a good faith effort to avoid privacy violations and disruptions to others … access another person's data or company data may demonstrate a lack of good faith and disqualify you from any benefit of the Safe Harbor … " (Facebook 2022).

A second promising approach is to require BBP participants to disclose identifier information during security testing and make SHT contingent on such disclosure. The idea is to induce hackers to self-select, which can make competitive hackers more distinguishable. For example, Amazon's BBP requires participants to create accounts and include account information in their network traffics when testing Amazon's systems.[25] This leaves competitive hackers with two options: test without identifier information, which Amazon can detect and block easily, or create fake accounts by incurring additional costs, which might still leave traces for investigation once they misuse a vulnerability. In both cases, because the research activities of competitive hackers become more identifiable, SHTs need not reduce the legal risks for competitive hackers.

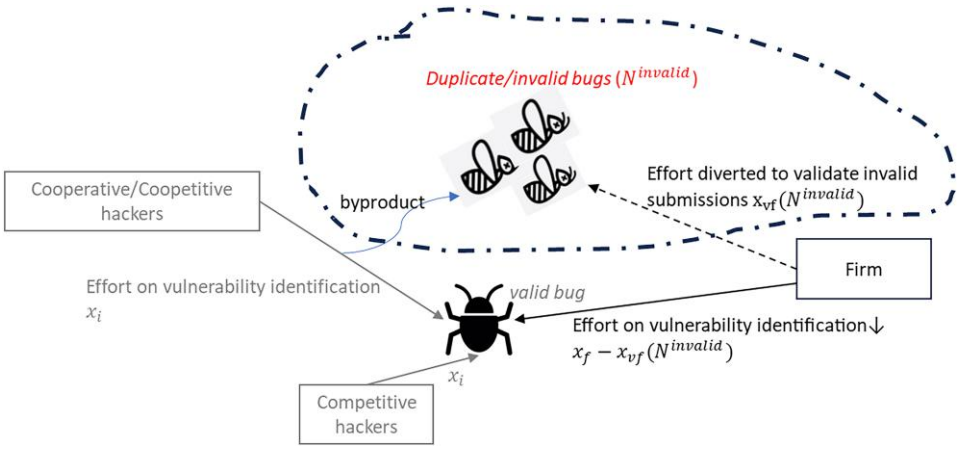
Although a few studies have analyzed SHTs (e.g., Zhang et al. 2024), none considers SHTs' influence on competitive hackers. Our findings demonstrate that this omission leads to overestimation of SHTs' benefits, and it may result in suboptimal designs of SHTs.[26] Our study, thus, underscores the importance of considering the strategic interactions between competitive hackers and the firm in evaluating SHTs.

6.2. Duplicate/Invalid Reports

We extend the model to analyze the duplicate/invalid report problem.[27] Hackers may obtain duplicate or invalid findings when trying to identify vulnerabilities. They might submit such findings without validating them (see Endnote 27 for an example). Such invalid findings do not provide additional security information or value to the firm, and yet processing and validating them consumes the firm's effort.

For brevity, we highlight the key model changes and present detailed discussions of model choices in Online Appendix D.1. Referring to the dashed area in Figure 8, the new model feature is that cooperative and coopetitive hackers might submit invalid reports with the expected number positively related to their effort, and so the total number of invalid reports can be represented by $N^{invalid} = \sum_{v_i \le r} b n_i x_i$, where $b > 0$ is a positive

Figure 8. (Color online) Invalid Reports in Bug Bounty Programs



scale parameter. The firm has to validate invalid submissions to avoid paying wrong bounty rewards. The need to handle invalid submissions reduces the resources that the firm would otherwise spend on identifying valid vulnerabilities, so the firm's remaining effort dedicated to vulnerability identification becomes $x_f - x_{vf}(N^{invalid})$, where $x_{vf}(N^{invalid}) = \theta_f N^{invalid}$ represents effort diverted to validating invalid reports and θ_f represents the firm's efficiency at handling invalid findings with a smaller θ_f representing greater efficiency. Therefore, the firm's payoff after considering invalid submissions becomes

$$\pi_f^b = -\sum_{v_i>r} \lambda \frac{n_i x_i}{X^{nv}} - \sum_{v_i \le r} r \frac{n_i x_i}{X^{nv}} - c_f[x_f - x_{vf}(N^{invalid})] - c_{fv} x_{vf}(N^{invalid}),$$

where $X^{nv} = d + [x_f - x_{vf}(\sum_{v_j \le r} b n_j x_j)] + \sum_{j=1}^{T} n_j x_j$ represents the aggregate effort that the firm and hackers invest in vulnerability discovery and c_f and c_{fv} represent the effort parameters for vulnerability identification and validation. We assume $c_f = c_{fv}$ and show that our results are robust in Online Appendix D.1. With this setting, we confirm that invalid reports reduce a firm's BBP payoff and the optimal bounty reward, and hackers have little incentive to validate their findings (see Online Appendix D.1).

We are interested in policies to address the invalid report problem. The industry and academic literature primarily discuss two approaches to handle invalid reports:

- Taxing duplicate reports: The idea is to raise the cost of submitting invalid reports, similar to a tax on spam emails (Rao and Reiley 2012). For example, HackerOne assigns a reputation score to bug reporters, and duplicate report submissions lower a reporter's reputation, and this may influence their eligibility for future BBPs (HackerOne 2025). BugCrowd follows a similar approach. A firm can incentivize hackers to validate submissions using penalties such as low reputation scores, hence reducing invalid or duplicate report submissions.
- Algorithms to handle duplicate reports: A stream of literature (see Zhang et al. 2023 for a review) proposes to enhance a firm's efficiency in identifying duplicate reports through machine learning models. Recent studies (Xiao et al. 2020) show that deep neural network–based algorithms can identify duplicate reports with accuracy as high as 98% based on textual descriptions and other metadata in the reports. This efficiency corresponds to a lower θ_f.

Proposition 7. *Taxing duplicate reports reduces the firm's payoff when $\frac{c_f}{c_h}$ is sufficiently large and $\theta_f b$ is sufficiently small (i.e., $\frac{c_f}{c_h} > \overline{c}$ and $\theta_f b < \overline{\mu}$, where $\overline{c}$ and $\overline{\mu}$ are positive thresholds), and it increases the firm's payoff otherwise. Efficient handling of duplicate reports increases the firm's payoff whenever the efficiency-enhancing technology is not too costly to implement.*

Proposition 7 reveals a paradox. Whereas taxing duplicate reports substantially reduces invalid report submissions, it may end up reducing a firm's payoff. Conversely, improving efficiency in handling reports does not reduce invalid submissions, but it often improves the firm's payoff. This insight has important implications for the duplicate report literature, which focuses on identifying policies that most effectively reduce duplicate submissions (e.g., Schechter 2002, Zhao et al. 2017). With that focus, their implication might be that firms should prioritize policies that minimize duplicate submissions. Our findings suggest that this approach may be unfounded as fewer duplicate submissions can be associated with a lower payoff.

Taxing duplicate reports encourages hackers to validate reports, saving the firm efforts but burdening hackers. Increased time spent by hackers on validating findings reduces their time available to identify vulnerabilities, which can impede their vulnerability identification efficiency. When the firm's vulnerability identification efficiency is low ($\frac{c_f}{c_h}$ is high), it relies on cooperative and coopetitive hackers' greater efficiency for vulnerability identification (protection delegation). In such cases, reduced hacker efficiency can diminish the firm's BBP return, particularly if duplicate submissions are less critical than identifying vulnerabilities, for example, when the chance of duplication is low because of insufficient submissions or diverse system scopes (i.e., small b) or when the firm has an efficient system to handle duplicate submissions (i.e., small θ_f). Therefore, taxing duplicate reports is beneficial primarily when the firm is less concerned about hackers' vulnerability identification efficiency (e.g., when the firm is primarily seeking attack diversion benefits) and might backfire otherwise.

By contrast, enhanced efficiency in handling invalid reports reduces the influence of such submissions on in-house effort. The firm is then able to devote more effort to vulnerability identification, and this reduces both security and bounty payment costs. The firm's payoff, thus, improves as long as implementing the efficiency-enhancing algorithm is not too costly.

One implication that contradicts the extant literature is that organizations may strategically choose to embrace duplicate report submissions even with the option of penalizing such uninformative submissions. In fact, some BBPs reward reporters for duplicate submissions by providing acknowledgment (e.g., Hao et al. 2022).

7. Concluding Remarks

"Do I not destroy my enemies when I make them my friends?" said Abraham Lincoln at the conclusion of the

American Civil War. BBPs capture this spirit in cyberspace and inspire how firms and governments may approach cybersecurity. Although the idea of enlisting help from hackers is not new (Felicetti 2016, Leswing 2016, McGee and Breslauer 2016), BBPs stand out in comparison with traditional ways of hiring white-hat hackers directly because they allow organizations to convert a large number of hackers from the open market into white hats without specifying individual wages or ex ante differentiating between white- and black-hat hackers. This study contributes an analytical framework to scrutinize the implications of BBPs, including their economic and security benefits, social efficiency, and policy design. Our analysis generates multiple insights and implications as Table 4 summarizes.

Our findings offer organizations two key benefits. First, some organizations and security experts question BBPs because of concerns about economic returns and potential negative influences on in-house protection (e.g., Oracle.com 2015, Dayan 2022). Such concerns are often based on intuition instead of formal analysis, which can lead to undue hesitation or suboptimal BBP decisions. We provide a theoretical framework that clarifies these trade-offs. For example, we identify two distinct benefits of BBPs, illustrating BBPs' value to organizations with strong internal security capabilities, and show that BBPs' positive security benefits outweigh their negative influence on in-house protection (points 1 and 3 of Table 4). These insights are especially relevant as AI tools (e.g., Mythos) increasingly enhance firms' ability to detect vulnerabilities, prompting questions about whether BBPs remain necessary (e.g., Okunytė 2026). Our findings suggest that BBPs remain valuable even when firms possess highly capable AI-based security tools, particularly when their primary benefit arises from attack diversion. Moreover, we offer guidance on the strategic design of SHTs and duplicate report policies to improve the effectiveness of BBPs (points 10 and 11). We believe that these insights can enable organizations to make more informed decisions and enhance the acceptance of BBPs.

Second, we demonstrate that BBPs require strategies that contradict conventional crowdsourcing or security protection practices. For example, conventional crowdsourcing emphasizes the benefits of a larger crowd and increasing rewards with the crowd's size, but we show that more participants can diminish BBP returns and lead optimal rewards to decrease or change nonmonotonically (points 2, 8, and 9). Similarly, conventional protection measures focus on disincentivizing all external hackers, but BBPs convert some hackers into allies though they potentially encourage some malicious hackers to exert more efforts; organizations also need tailored strategies for different types of hackers (points 3 and 10). By highlighting these unique characteristics, our findings can help organizations more effectively adapt to the distinctive challenges and complexities posed by BBPs.

As BBPs gain traction and their ecosystem evolves, firms may face new design problems. For example, should BBP platforms promote information sharing among hackers? The trade-offs revealed in our analysis

Table 4. Practical Insights

Economic payoff	1. BBPs bring protection delegation and attack diversion benefits. BBPs might increase a firm's payoff even when hackers are inefficient in vulnerability identification.
	2. Contrary to the crowdsourcing wisdom that a larger crowd enhances crowdsourcing effectiveness, more cooperative hacker participants might make BBPs less attractive.
Security	3. BBPs might encourage competitive hackers to exert more efforts. Organizations should continue to compete with hackers even with a BBP.
	4. Organizations that care more about security than economic payoff may entice as many cooperative hackers as possible.
Social welfare	5. Governments need to correct BBP inefficiencies because firms' provision of rewards and in-house effort might be inefficient. The inefficiency differs from those in information security outsourcing.
	6. Governments might need to revise breach penalty regulations in response to a wider BBP adoption; current approaches may reduce social welfare with BBPs.
	7. There is complementarity between breach penalties and bounty reward subsidies in addressing BBP inefficiencies.
Bounty reward	8. The conventional crowdsourcing wisdom—that rewards increase with participant size—applies to BBPs only when a firm's efficiency and the misuse benefit of potentially coopetitive hackers (v_M) are low.
	9. With a high misuse benefit (of potentially coopetitive hackers) but low firm efficiency, the optimal reward stays stable initially, then decreases, and eventually increases with more cooperative hackers; with a moderate misuse benefit and low firm efficiency, the optimal reward stays stable initially and then increases with more cooperative hackers; with high firm efficiency, the optimal reward stays stable initially but eventually reduces to zero with more cooperative hackers.
Policy design	10. Adopting an SHT might reduce a firm's payoff even when the firm benefits from ethical hacking. Organizations should consider supplementary policies alongside SHTs.
	11. Policies that more effectively reduce duplicate submissions can lead to lower payoffs. Firms should choose duplicate report policies strategically.

call for different design considerations when compared with traditional crowdsourcing platforms. In conventional crowdsourcing, knowledge sharing among the crowd—such as sharing technologies or tools—is often encouraged to boost task efficiency (Jin et al. 2021). In BBPs, however, such shared knowledge could empower competitive hackers, and this may reduce the firm's payoffs. Promoting such knowledge sharing is beneficial only when the firm obtains protection delegation benefits from the higher efficiency of hackers (see Proposition 1) or when the shared information primarily benefits cooperative/coopetitive hackers without aiding competitive hackers (e.g., the firm may design the bug bounty platform so that only verified hackers have access to the shared information). Similarly, we can design BBPs to influence hackers' perception of competition. In conventional crowdsourcing, firms often downplay competition to elicit more crowd efforts (Terwiesch and Xu 2008). Here, firms need not prefer more effort from hackers because hackers who find the vulnerability first may exploit it. Instead, firms may dissuade hacker efforts by using bug bounty forums to intensify perceived competition or congestion—such as by inserting duplicate hackers as external participants or publicizing a large number of active hackers—especially when firms are primarily obtaining attack diversion benefits (i.e., when firms have high in-house capabilities; see Online Appendix D.5 for a brief analysis).

We emphasize two insights for the government. First, it should recognize the value of BBPs in hacking regulation, particularly through the attack diversion benefit of BBPs. Unlike traditional regulatory approaches that rely on punitive measures or hiring selected hackers, BBPs allow society to exploit hackers' expertise and efforts productively and on a greater scale. If coopetitive hackers can find more legitimate opportunities in BBPs, they might gradually divert effort away from harmful hacking. This unique social benefit of BBPs warrants government attention. Second, the third row (points 5–7) of Table 4 highlights the need for the government to address BBP inefficiency. A pertinent challenge is the difficulty of a single policy to address effort inefficiency, which can manifest as either insufficient or excessive protection efforts in different scenarios and reward inefficiency. Therefore, we recommend multifaceted policies to address BBP inefficiencies. More broadly, our findings suggest that the government must recognize and deliberate the intertwined inefficiencies when evaluating existing policies and developing new policies to accommodate BBPs and other emerging protection methods.

This study advances the crowdsourcing and security investment literature. For crowdsourcing, we incorporate coopetitive interactions between an organizer and participants. Although this study focuses on cybersecurity, its findings might apply to other contexts that involve coopetition. Consider an industry facing potential disruptive innovations. A company might use crowdsourcing preemptively to internalize innovative ideas and prevent them from being proposed to competitors. Such an incentive for adopting crowdsourcing—preempting the threat that competitors have access to innovative ideas first—is not documented in extant crowdsourcing research, but it is similar to the attack diversion mechanism discussed in this study. Future research should explore how such coopetition influences strategy and social outcomes, including innovation, competition, and social welfare, in other contexts.

For security investment, this study represents the first step toward developing the theoretical underpinning of how firms can work with potential enemies, namely, hackers, to enhance information security. Conventional security strategies often attempt to prevent hackers from finding a vulnerability (Gordon and Loeb 2002). However, those solutions cannot minimize the potential damage when hackers already found a vulnerability before a firm.[28] BBPs allow firms to address such risks. Future research should cultivate methods to recruit instead of desert hackers. Our research offers a novel framework in that direction.

Our analysis is subject to several limitations. To maintain tractability and focus on the main insights, we made several simplification assumptions. For example, we assume complete information on the relative cost parameters. One future research direction is to examine how introducing uncertainty affects our findings. We model only hacker heterogeneity in terms of valuations toward the vulnerability. Hackers can be heterogeneous in other ways, such as their preferences for addressing different bugs. Future research might consider other hacker heterogeneity. We focus on a single reward, but a natural extension is to explore how firms can set multiple reward levels, including designing reward discrimination. We also focus on security risks because of vulnerability misuse, but organizations experience other security risks, such as system interdependency risks (e.g., Kunreuther and Heal 2003, Hui et al. 2012, August et al. 2022) or risks that do not involve vulnerabilities (e.g., social engineering or distributed denial of service). Future research can examine whether BBPs have externalities on other security risks and how they collectively influence system security. We focus on how a firm's choice of BBPs affects the probability of various parties finding a vulnerability first and how a BBP broadly affects hackers' choices of misuse or protection. Future research can assess other effects because of BBPs. For example, a BBP might offer additional benefits such as greater specialization, learning from hackers, reputational gain, and costs such as the operational cost of trialing reports or communicating with hackers, GDPR data breach notification liability when participants obtain customer data accidentally

(Porup 2020), and wage payments as BBP participants might be considered employees under the U.S. federal labor law and California's laws (Porup 2020). These considerations can be analyzed by fine-tuning our model without major changes to its basic structure. Our contribution, thus, lies in both the study's results and insights and the foundation it builds for future BBP research.

To conclude, previous security measures based on deterrence focus on sticks: how to punish hackers for attacking others. Contrary to this tradition, a BBP acts as a carrot through which a firm converts an enemy (i.e., coopetitive hackers) into an ally. As shown in our analysis, combining carrots and sticks is often superior to using sticks alone. Future cybersecurity research and practice should systematically investigate ways to use underground hacking resources more effectively.

Acknowledgments

The authors thank the Department Editor, the Associate Editor, and three anonymous reviewers for their constructive comments and guidance throughout the review process. The authors also thank participants at the 2019 Workshop on the Economics of Information Security, 30th Workshop on Information Systems and Economics, as well as seminar participants at the Hong Kong University of Science and Technology, American University, Shanghai Jiao Tong University, and Peking University HSBC Business School for their helpful comments. Finally, the authors thank an audience at the SAS ERM-ESSEC CREAR Cyber Risk Conference 2018 in Singapore for inspiring this research.

Endnotes

[1] See Forbes top 100 digital companies, https://www.forbes.com/top-digital-companies/list/ (accessed August 17, 2024).

[2] Major data breach (≥ 30,000 breached records) information from HAVEIBEENPWNED (Hunt 2022), a breach database developed by Troy Hunt and CloudFare and the breach data provider for Firefox and 1Password (Li et al. 2019). We supplement this database with breach information recorded on Wikipedia (Wikipedia 2022). Bug bounty program information retrieved from the internet.

[3] For example, in uTest (https://www.utest.com/), a crowd-based usability test platform, a user's test performance was affected by the platform's internal effort to improve the service. We discuss another example in the concluding remarks.

[4] It is customary to analyze a single vulnerability (e.g., Kannan and Telang 2005, August and Tunca 2006) in a single organization because IT security vulnerabilities are often unique to the design and configuration of individual systems, and hence, competition of IT security investment is not a key concern (Gordon and Loeb 2002, Kannan and Telang 2005).

[5] Firms usually respond promptly to vulnerability reports in BBPs. For example, Github's average response time for bug reports was 12 hours in 2021 (Github 2022). Zoom claimed that its initial response time was less than four hours (Davis 2022).

[6] The size n_t can also be interpreted as the proportion of type t hackers in the market. Note that, for ease of exposition, we use k as a running index to denote any single hacker of type t.

[7] We can add a group of benign hackers who would never misuse a vulnerability, but doing so does not offer new analytical insights. Conceptually, we can interpret a firm's protection as summarily including contributions from benign hackers. We discuss altruistic hackers in one extension.

[8] We generalize the setting in Kannan and Telang (2005), which considers two types of representative hackers. Our model can be applied to different BBPs and vulnerabilities by modifying the hacker distribution.

[9] Realistically, the firm's security risk depends on its own nature and the effort that it expends relative to the hackers in searching for the vulnerability. The firm may gauge its potential security risk by referring to industry reports, security vulnerability intelligence, or its own track record and experience. For example, HackerOne and the National Institute of Standards and Technology National Vulnerability Database regularly report the distribution and scoring of vulnerabilities across industries and organizations. Such intelligence helps the firm assess how likely its system may be targeted. The firm may also refer to the Dark Web or online hacker discussion forums (Yue et al. 2019) to estimate its potential risk because of hacking knowledge development and possible hacker actions. The assumption that the firm forms rational expectations of its own vulnerability and hacker actions is common in the literature (see, e.g., Kannan and Telang 2005, Png and Wang 2009, Kannan et al. 2016). It allows us to focus on how the BBP shifts the strategic interaction between the firm and hackers without the distraction of cost uncertainty.

[10] The Tullock contest function assumes that the firm's and hackers' efforts exert an externality on each other by reducing the chance of others finding the vulnerability first. We assume that the firm and hackers form rational expectations of such externalities, including integrating information about the competitive intensity of vulnerability identification from third-party sources such as online bug bounty forums (e.g., bugbountyforum.com) or other information channels.

[11] The corresponding explicit function is $x_{t,k} = \sqrt{\frac{v_t}{c_h}\left(d + x_f + \sum\sum_{(j,g)\neq(t,k)} x_{j,g}\right)} - (d + x_f + \sum\sum_{(j,g)\neq(t,k)} x_{j,g})$. We present the implicit function because it is common in the Tullock contest literature (e.g., Cornes and Hartley 2005), and it is more concise to represent hackers' equilibrium efforts in relation to overall competition (see Lemmas 2 and 3).

[12] $\arg\max_{r\in Z}\pi_f^b(r)$ represents the set of arguments in set Z that maximizes the target function $\pi_f^b(r)$. That is, $\arg\max_{r\in Z}\pi_f^b(r) := \{r | r \in Z \wedge \forall x \in Z : \pi_f^b(r) \geq \pi_f^b(x)\}$. We provide the technical details in the proofs.

[13] We do not use a continuous distribution for v_i because, with a continuous distribution, the probability of each hacker finding the vulnerability is zero, making modeling the hackers' behavior difficult. As argued in Moorthy (1984), a continuous distribution is equivalent to assuming that the types in the discrete distribution are dense.

[14] $E_{t_0} = X^{t_0}[c_h\lambda\sum_{i=1}^{t_0-1}\frac{n_i}{v_i} - c_f c_h(\sum_{i=1}^{t_0-1}\frac{n_i}{v_i} + \frac{N_{t_0}}{v_{t_0}})X^{t_0} + c_h N_{t_0} - c_f + c_f N_1] - X^{nb}[c_h\lambda\sum_{i=1}^{B^{nb}}\frac{n_i}{v_i} - c_f c_h\sum_{i=1}^{B^{nb}}\frac{n_i}{v_i}X^{nb} - c_f(1 - N_1 + N_{B^{nb}+1})] + \lambda(N_{t_0+1} - N_{B^{nb}+1})$, and $X^{t_0} = \frac{(N_1 - N_{t_0})\lambda + N_{t_0}v_{t_0}}{c_h\lambda\sum_{i=1}^{t_0-1}\frac{n_i}{v_i} + N_{t_0}c_h + c_f}$.

[15] In particular, with a stylized hacker distribution with hackers' bargaining power depending on their misuse benefits, we can show that the firm has greater incentives to practice reward discrimination when the coopetitive hackers' misuse benefit is high. Such an analysis is available upon request.

[16] We plot Figure 5 by varying $N_{B^{nb}+1}$ (i.e., the number of cooperative hackers). With three hacker types, $N_{B^{nb}+1} = n_3$. For any hacker distribution, the graph can be obtained by first computing B^{nb} to obtain $N_{B^{nb}+1}$ (using Lemma 1). Then, we can vary $N_{B^{nb}+1}$ to obtain hackers' effort x_t according to Lemma 3. Changing $N_{B^{nb}+1}$ does not affect B^{nb} according to Lemma 1; x_t and x_f are affected only through changing $N_{B^{nb}+1}$.

[17] The only exception is when there exists a dominant competitive hacker who has more than a 50% chance of finding the vulnerability first. Then, with more cooperative hackers, that dominant hacker increases its effort, whereas all other competitive hackers reduce theirs. In all other cases, all competitive hackers reduce their efforts with more cooperative hackers in the market. To focus on general and more realistic cases, we assume that no individual hacker has more than a 50% chance of finding the vulnerability first (i.e., $\frac{x_1}{X^b} < 1/2 \Longleftrightarrow \frac{v_1}{2c_h} < \frac{(N_1 - N_{BO})\lambda + N_{BO} r^*}{c_f + N_{BO} c_h + \lambda c_h \sum_{j=1}^{BO-1} \frac{n_j}{v_j}}$). See the additional discussion in Online Appendix C.1.

[18] As suggested in (13), our social welfare measure includes the utilities of the firm and cooperative and coopetitive hackers minus competitive hackers' effort costs. Government subsidies or fines are considered transfer payments (Becker 1968). Our results are robust to excluding competitive hackers' effort costs. This measure implies that society prefers increasing the gain from reporting but not misusing the vulnerability (Levin and Trumbull 1990). See Online Appendix C.2 for a more detailed discussion of different social welfare measures.

[19] For example, with four types of hackers, there are $\binom{5}{2} = 10$ possible combinations of cooperative, coopetitive, and competitive hackers along with more than 10 potential optimal reward ranges based on the distribution of v_t. The general insights, however, are similar.

[20] Extracted from Mozilla's BBP; see https://www.mozilla.org/en-US/security/bug-bounty/ (accessed July 22, 2022).

[21] However, misuse of vulnerabilities carries substantial legal liability. Our model assumes that the hacker valuation, v_t, captures such legal risks.

[22] In the economics of crime literature (e.g., Becker 1968, Chalfin and McCrary 2017), this disutility is the product of the probability of apprehension and penalty. We do not separate apprehension and penalty in this study.

[23] See https://github.com/disclose/dioterms/blob/master/core-terms-bbp.md (accessed July 20, 2022).

[24] The intuition behind this parameter range is that cooperative and coopetitive hackers' increased efficiency (facilitated by the SHT) is more beneficial to the firm when the firm has low in-house efficiency in vulnerability identification (i.e., $\frac{c_f}{c_h}$ is large) as Proposition 1 suggests.

[25] The details can be found at https://hackerone.com/amazonvrp?type=team (accessed July 20, 2022). Similar policies can be found in the A. S. Watson Group (2020) and Virginia Tech IT Security Office (2021) BBPs.

[26] For example, our analysis suggests that a better designed SHT can improve social welfare by improving payoffs to both a firm and cooperative/coopetitive hackers.

[27] Our model focuses on deriving general insights about duplicate/invalid reports, so we use the terms "invalid report" and "duplicate report" interchangeably. An example of a duplicate/invalid report is a leaked password to a firm's internal server when the firm has already changed the password because of an earlier vulnerability discovery. Hackers may still report the leaked password because they do not spend time locating the server and verifying the password. For simplicity, we exclude the possibility of hackers intentionally sending invalid reports to spam a firm as it is a spamming instead of hacking problem. Alternatively, we could assume that the firm has an effective spam filter.

[28] Insurance is an exception. Similar to BBPs, insurance reduces security losses when hackers find a vulnerability before a firm. However, BBPs and insurance reduce security loss in fundamentally different ways. Insurance allows a firm to receive compensation after security incidents, but BBPs prevent security incidents by diverting hackers away from misusing a vulnerability.

References

Akgul O, Eghtesad T, Elazari A, Gnawali O, Grossklags J, Mazurek ML, Votipka D, Laszka A (2023) Bug {hunters'} perspectives on the challenges and benefits of the bug bounty ecosystem. *Proc. 32nd USENIX Security Sympos*. (USENIX Association, Anaheim, CA), 2275–2291.

Ales L, Cho SH, Körpeoğlu E (2017) Optimal award scheme in innovation tournaments. *Oper. Res.* 65(3):693–702.

Anderson R, Moore T (2006) The economics of information security. *Science* 314(5799):610–613.

Anglin B (2020) Six years of the GitHub security bug bounty programs. Accessed March 29, 2020, https://github.blog/2020-03-25-six-years-of-the-github-security-bug-bounty-program/.

Apel R (2013) Sanctions, perceptions, and crime: Implications for criminal deterrence. *J. Quant. Criminology* 29(1):67–101.

Apple (2019) Apple security bounty. Accessed June 29, 2022, https://developer.apple.com/security-bounty/.

Arora A, Nandkumar A, Telang R (2006) Does information security attack frequency increase with vulnerability disclosure? An empirical analysis. *Inform. Systems Frontiers* 8(5):350–362.

A. S. Watson Group (2020) A.S. Watson group bug bounty program. Accessed May 7, 2022, https://hackerone.com/watson_group?type=team.

August T, Tunca TI (2006) Network software security and user incentives. *Management Sci.* 52(11):1703–1720.

August T, Tunca TI (2011) Who should be responsible for software security? A comparative analysis of liability policies in network environments. *Management Sci.* 57(5):934–959.

August T, Dao D, Niculescu MF (2022) Economics of ransomware: Risk interdependence and large-scale attacks. *Management Sci.* 68(12):8979–9002.

Bandyopadhyay T, Liu D, Mookerjee VS, Wilhite AW (2014) Dynamic competition in IT security: A differential games approach. *Inform. Systems Frontiers* 16(4):643–661.

Baye MR, Hoppe HC (2003) The strategic equivalence of rent-seeking, innovation, and patent-race games. *Games Econom. Behav.* 44(2):217–226.

Becker GS (1968) Crime and punishment: An economic approach. *J. Polit. Econom.* 76(2):169–217.

Boudreau KJ, Lacetera N, Lakhani KR (2011) Incentives and problem uncertainty in innovation contests: An empirical analysis. *Management Sci.* 57(5):843–863.

Bouncken RB, Gast J, Kraus S, Bogers M (2015) Coopetition: A systematic review, synthesis, and future research directions. *Rev. Managerial Sci.* 9(3):577–601.

Brandenburger AM, Nalebuff BJ (2011) *Co-Opetition* (Crown Currency, New York).

Cavusoglu H, Raghunathan S, Cavusoglu H (2009) Configuration of and interaction between information security technologies: The case of firewalls and intrusion detection systems. *Inform. Systems Res.* 20(2):198–217.

Cezar A, Cavusoglu H, Raghunathan S (2014) Outsourcing information security: Contracting issues and security implications. *Management Sci.* 60(3):638–657.

Chalfin A, McCrary J (2017) Criminal deterrence: A review of the literature. *J. Econom. Literature* 55(1):5–48.

Chawla S, Hartline JD, Sivan B (2019) Optimal crowdsourcing contests. *Games Econom. Behav.* 113:80–96.

Cornes RC, Hartley RT (2005) Asymmetric contests with general technologies. *Econom. Theory* 26:923–946.

Cremonini M, Nizovtsev D (2009) Risks and benefits of signaling information system characteristics to strategic attackers. *J. Management Inform. Systems* 26(3):241–274.

Culafi A (2021) Bug bounty programs in 2021: High payouts, higher stakes. Accessed May 7, 2022, https://www.techtarget.com/searchsecurity/news/252509175/Bug-bounty-programs-in-2021-High-payouts-higher-stakes.

Dasgupta P, Stiglitz J (1980) Uncertainty, industrial structure, and the speed of R&D. *Bell J. Econom.* 11(1):1–28.

Davis R (2022) Zoom's bug bounty program: 2021 in review. Accessed May 29, 2022, https://blog.zoom.us/zoom-bug-bounty-program-2021/.

Dayan LB (2022) What happens when bug bounties don't work? Accessed June 7, 2022, https://vulcan.io/blog/what-happens-when-bug-bounties-dont-work/.

Dey D, Fan M, Zhang C (2010) Design and analysis of contracts for software outsourcing. *Inform. Systems Res.* 21(1):93–114.

Dixit A (1987) Strategic behavior in contests. *Amer. Econom. Rev.* 77(5):891–898.

Elazari A (2018) The law and economics of bug bounties. Accessed March 1, 2022, https://www.usenix.org/conference/usenixsecurity18/presentation/elazari-bar.

Etcovitch D, Van Der Merwe T (2018) *Coming in from the Cold: A Safe Harbor from the CFAA and the DMCA 1201 FOR Security Researchers* (Berkman Klein Center Research Publication, Cambridge, MA), 1–43.

European Commission (2019) EU bug bounty programme for open source software gives awards of up to eur 25,000. Accessed May 29, 2020, https://digital-strategy.ec.europa.eu/en/news/eu-bug-bounty-programme-open-source-software-gives-awards-eur-25000.

Facebook (2022) Meta bug bounty programme info. Accessed May 7, 2022, https://business.facebook.com/whitehat.

Felicetti K (2016) You'll never guess which industry is now hiring hackers. Accessed May 29, 2020, https://www.monster.com/career-advice/article/banks-courting-hacker-cybersecurity-talent.

Finifter M, Akhawe D, Wagner D (2013) An empirical study of vulnerability rewards programs. *Proc. 22nd USENIX Security Sympos.* (USENIX Association, Washington, DC), 273–288.

Fryer H, Simperl E (2017) Web science challenges in researching bug bounties. *Proc. 2017 ACM Web Sci. Conf.* (ACM, New York), 273–277.

Gallagher S (2017) Man gets threats—Not bug bounty—After finding DJI customer data in public view. Accessed May 29, 2024, https://arstechnica.com/information-technology/2017/11/dji-left-private-keys-for-ssl-cloud-storage-in-public-view-and-exposed-customers/.

Gal-Or E, Ghose A (2005) The economic incentives for sharing security information. *Inform. Systems Res.* 16(2):186–208.

Gao X, Zhong W, Mei S (2015) Security investment and information sharing under an alternative security breach probability function. *Inform. Systems Frontiers* 17(2):423–438.

Garfinkel MR, Skaperdas S (2007) Economics of conflict: An overview. Sandler T, Hartley K, eds. *Handbook of Defense Economics*, vol. 2 (Elsevier, Amsterdam), 649–709.

GitHub (2022) Eight years of the GitHub security bug bounty program. Accessed May 29, 2022, https://github.blog/2022-05-23-eight-years-of-the-github-security-bug-bounty-program/.

Google (2024) Google and Alphabet vulnerability reward program (VRP) rules. Accessed June 29, 2024, https://bughunters.google.com/about/rules/google-friends/6625378258649088/google-and-alphabet-vulnerability-reward-program-vrp-rules.

Gordon LA, Loeb MP (2002) The economics of information security investment. *ACM Trans. Inform. System Security* 5(4):438–457.

Gottlieb B (1999) Hack, counterhack. Accessed October 3, 2023, https://archive.nytimes.com/www.nytimes.com/library/magazine/home/19991003mag-hackers.html.

Gupta A, Zhdanov D (2012) Growth and sustainability of managed security services networks: An economic perspective. *MIS Quart.* 36(4):1109–1130.

HackenProof (2023) HackenProof: Bug bounty programs and GDPR compliance. Accessed January 1, 2024, https://hackenproof.com/blog/industry-news/hackenproof-bug-bounty-programs-and-gdpr-compliance.

HackerOne (2025) Reputation. Accessed March 1, 2026, https://docs.hackerone.com/en/articles/8369865-reputation.

Hao R, Li Y, Feng Y, Chen Z (2022) Are duplicates really harmful? An empirical study on bug report summarization techniques. *J. Software Evolution Process* 35(11):e2424.

Harper A, Harris S, Ness J, Eagle C, Lenkey G, Williams T (2011) *Gray Hat Hacking: The Ethical Hacker's Handbook* (McGraw-Hill, New York).

Harrington T (2022) Can bug bounty programs replace dedicated security testing? Accessed June 7, 2022, https://resources.infosecinstitute.com/topic/can-bug-bounty-programs-replace-dedicated-security-testing/.

Hausken K (2006) Income, interdependence, and substitution effects affecting incentives for security investment. *J. Accounting Public Policy* 25(6):629–665.

Herley C, Florêncio D (2010) Nobody sells gold for the price of silver: Dishonesty, uncertainty and the underground economy. Tyler M, David P, Christos I, eds. *Economics of Information Security and Privacy* (Springer, Berlin, Heidelberg), 33–53.

Horton JJ, Chilton LB (2010) The labor economics of paid crowdsourcing. *Proc. 11th ACM Conf. Electronic Commerce* (ACM, Cambridge, MA), 209–218.

Hui KL, Hui W, Yue WT (2012) Information security outsourcing with system interdependency and mandatory security requirement. *J. Management Inform. Systems* 29(3):117–156.

Hui KL, Kim SH, Wang QH (2017) Cybercrime deterrence and international legislation: Evidence from distributed denial of service attacks. *MIS Quart.* 41(2):497–523.

Hui KL, Ke PF, Yao Y, Yue WT (2019) Bilateral liability-based contracts in information security outsourcing. *Inform. Systems Res.* 30(2):411–429.

Hunt T (2022) Have I been pwned? Accessed June 21, 2022, https://haveibeenpwned.com/PwnedWebsites.

Jensen MC, Meckling WH (1976) Theory of the firm: Managerial behavior, agency costs, and ownership structure. *J. Financial Econom.* 3(4):305–360.

Jeppesen LB, Lakhani KR (2010) Marginality and problem-solving effectiveness in broadcast search. *Organ. Sci.* 21(5):1016–1033.

Jin Y, Lee HCB, Ba S, Stallaert J (2021) Winning by learning? Effect of knowledge sharing in crowdsourcing contests. *Inform. Systems Res.* 32(3):836–859.

Kannan K, Telang R (2005) Market for software vulnerabilities? Think again. *Management Sci.* 51(5):726–740.

Kannan K, Rahman MS, Tawarmalani M (2016) Economic and policy implications of restricted patch distribution. *Management Sci.* 62(11):3161–3182.

Katyal NK (2001) Criminal law in cyberspace. *Univ. Pennsylvania Law Rev.* 149(4):1003–1114.

Kigerl AC (2016) Deterring spammers: Impact assessment of the can spam act on email spam rates. *Criminal Justice Policy Rev.* 27(8):791–811.

Krebs B (2013) The case for a compulsory bug bounty. Accessed January 29, 2020, https://krebsonsecurity.com/2013/12/the-case-for-a-compulsory-bug-bounty/.

Kunreuther H, Heal G (2003) Interdependent security. *J. Risk Uncertainty* 26(2):231–249.

Lacity MC, Khan SA, Willcocks LP (2009) A review of the IT outsourcing literature: Insights for practice. *J. Strategic Inform. Systems* 18(3):130–146.

Lee CH, Geng X, Raghunathan S (2013) Contracting information security in the presence of double moral hazard. *Inform. Systems Res.* 24(2):295–311.

Leeson PT, Coyne CJ (2005) The economics of computer hacking. *J. Law Econom. Policy* 1:511.

Leswing K (2016) Apple hired the hackers who created the first Mac firmware virus. Accessed May 29, 2020, https://www.businessinsider.com/apple-hired-the-hackers-who-created-the-first-mac-firmware-virus-2016-2.

Levin J, Trumbull W (1990) The social value of crime. *Internat. Rev. Law Econom.* 10:271–284.

Li L, Pal B, Ali J, Sullivan N, Chatterjee R, Ristenpart T (2019) Protocols for checking compromised credentials. *Proc. 2019 ACM SIGSAC Conf. Comput. Comm. Security* (ACM, New York), 1387–1403.

Liu B, Lu J, Wang R, Zhang J (2018) Optimal prize allocation in contests: The role of negative prizes. *J. Econom. Theory* 175:291–317.

Loury GC (1979) Market structure and innovation. *Quart. J. Econom.* 93(3):395–410.

Maillart T, Zhao M, Grossklags J, Chuang J (2017) Given enough eyeballs, all bugs are shallow? Revisiting Eric Raymond with bug bounty programs. *J. Cybersecurity* 3(2):81–90.

McGee C, Breslauer B (2016) Meet the hacker mom big companies hire for cybersecurity. Accessed May 29, 2020, https://www.nbcnews.com/tech/internet/meet-hacker-mom-big-companies-hire-cyber-security-n964291.

Mendez T, Göhmann D (2024) Google VRPs in review—2025. Accessed March 29, 2026, https://bughunters.google.com/blog/google-vrps-in-review-2025.

Mitra S, Ransbotham S (2015) Information disclosure and the diffusion of information security attacks. *Inform. Systems Res.* 26(3):565–584.

Moorthy KS (1984) Market segmentation, self-selection, and product line design. *Marketing Sci.* 3(4):288–307.

Munaiah N, Meneely A (2016) Vulnerability severity scoring and bounties: Why the disconnect? *Proc. Second Internat. Workshop Software Anal.* (ACM, New York), 8–14.

Okunytė P (2026) Curl creator who called mythos a "PR stunt" says AI will not take human jobs, but might kill bug bounties. Accessed June 5, 2026, https://cybernews.com/security/curl-bug-bounty-ai-security-reports-daniel-stenberg/.

Oracle.com (2015) No you really cannot. Accessed January 29, 2022, https://gist.github.com/michaeldyrynda/9b5eac6c02e6089052a6.

Perlroth N, Isaac M (2018) Inside Uber's $100,000 payment to a hacker, and the fallout. Accessed January 12, 2022, https://www.nytimes.com/2018/01/12/technology/uber-hacker-payment-100000.html.

Png IP, Wang QH (2009) Information security: Facilitating user precautions vis-à-vis enforcement against attackers. *J. Management Inform. Systems* 26(2):97–121.

Png IP, Wang CY, Wang QH (2008) The deterrent and displacement effects of information security enforcement: International evidence. *J. Management Inform. Systems* 25(2):125–144.

Porup J (2020) Bug bounty platforms buy researcher silence, violate labor laws, critics say. Accessed June 7 2022, https://www.csoonline.com/article/3535888/bug-bounty-platforms-buy-researcher-silence-violate-labor-laws-critics-say.html.

Qi M, Wang Y, Xu R (2009) Fighting cybercrime: Legislation in China. *Internat. J. Electronic Security Digital Forensics* 2(2):219–227.

Quinn JB, Hilmer FG (1994) Strategic outsourcing. *Sloan Management Rev.* 35(4):43–55.

Ransbotham S, Mitra S (2009) Choice and chance: A conceptual model of paths to information security compromise. *Inform. Systems Res.* 20(1):121–139.

Ransbotham S, Mitra S, Ramsey J (2012) Are markets for vulnerabilities effective? *MIS Quart.* 36(1):43–64.

Rao JM, Reiley DH (2012) The economics of spam. *J. Econom. Perspect.* 26(3):87–110.

Romanosky S, Telang R, Acquisti A (2011) Do data breach disclosure laws reduce identity theft? *J. Policy Anal. Management* 30(2):256–286.

Schechter S (2002) How to buy better testing using competition to get the most security and robustness for your dollar. *Internat. Conf. Infrastructure Security* (Springer, Berlin, Heidelberg), 73–87.

Spence M (1973) Job market signaling. *Quart. J. Econom.* 87(3): 355–374.

Swinhoe D (2020) Groups promote computer misuse act update to enable security research. Accessed May 7, 2021, https://www.csoonline.com/article/3566140/groups-promote-computer-misuse-act-update-to-enable-security-research.html.

Szidarovszky F, Okuguchi K (1997) On the existence and uniqueness of pure Nash equilibrium in rent-seeking games. *Games Econom. Behav.* 18(1):135–140.

Terwiesch C, Xu Y (2008) Innovation contests, open innovation, and multiagent problem solving. *Management Sci.* 54(9):1529–1543.

Thompson TA (2008) Terrorizing the technological neighborhood watch: The alienation and deterrence of the white hats under the CFAA. *Florida State Univ. Law Rev.* 36:537–585.

Tullock G (1980) Efficient rent seeking. Buchanan J, Tollison R, Tullock G, eds. *Toward a Theory of the Rent-Seeking Society* (Texas A&M University Press, College Station, TX), 97–112.

U.S. Department of Defense (2018) Department of defense expands 'hack the pentagon' crowdsourced digital defense program. Accessed May 7, 2022, https://www.defense.gov/News/Releases/Release/Article/1671231/department-of-defense-expands-hack-the-pentagon-crowdsourced-digital-defense-pr/.

Virginia Tech IT Security Office (2021) The Virginia Tech bug bounty program. Accessed May 7, 2022, https://bugbounty.aws.cloud.iso.vt.edu/.

Wikipedia (2022) List of data breaches. Accessed June 21, 2022, https://en.wikipedia.org/wiki/List_of_data_breaches.

Winder D (2020) U.S. Army hacked by 52 hackers in five weeks. Accessed March 29, 2020, https://www.forbes.com/sites/daveywinder/2020/01/16/us-army-hacked-by-52-hackers-in-five-weeksheres-why/#6fd351e81669.

Wired (2000) Apache site defaced. Accessed January 29, 2020, https://www.wired.com/2000/05/apache-site-defaced/.

Xiao G, Du X, Sui Y, Yue T (2020) HINDBR: Heterogeneous information network based duplicate bug report prediction. *2020 IEEE 31st Internat. Sympos. Software Reliability Engrg.* (IEEE, Coimbra, Portugal), 195–206.

Yue WT, Wang QH, Hui KL (2019) See no evil, hear no evil? Dissecting the impact of online hacker forums. *MIS Quart.* 43(1):73–95.

Zhang L, Demirezen EM, Kumar S (2024) How to make my bug bounty cost-effective? A game-theoretical model. *Inform. Systems Res.* 36(2):1031–1053.

Zhang T, Han D, Vinayakarao V, Irsan IC, Xu B, Thung F, Lo D, Jiang L (2023) Duplicate bug report detection: How far are we? *ACM Trans. Software Engrg. Methodology* 32(4):1–32.

Zhao M, Grossklags J, Liu P (2015) An empirical study of web vulnerability discovery ecosystems. *Proc. 22nd ACM SIGSAC Conf. Comput. Comm. Security* (ACM, New York), 1105–1117.

Zhao M, Laszka A, Grossklags J (2017) Devising effective policies for bug-bounty platforms and security vulnerability discovery. *J. Inform. Policy* 7:372–418.