Organizational Learning Process on Bug-Bounty Platforms: The Role of Firm Experience and the Diversity of Hackers

Ali Ahmed[1] (ali_ahmed@student.uml.edu), Brian Lee[2] (lqh5190@psu.edu),
Amit Deokar[1] (amit_deokar@uml.edu)
[1] University of Massachusetts Lowell, [2] Penn State University

Abstract

Bug-bounty, a crowdsourcing way for vulnerability discovery, is an emerging practice for firms to detect security loopholes in their online systems. Unlike a typical crowdsourcing platform, in bug-bounty platforms, firms are required to collaborate with hackers on the platform continuously. Despite the growing interest in studying bug-bounty programs, it remains unclear how firms collaborate with hackers. In this paper, we examine how the firm's experience affects the efficiency in resolving security vulnerabilities on the platform. Using a dataset collected from the HackerOne bug bounty platform, we show that there exists an inverted U-shaped relationship between the organization's vulnerability resolution time and the number of vulnerabilities resolved in the past. Interestingly, the firm may perform worse (i.e., resolving in a long time) as they gain more experience initially. However, as the firm has resolved a sufficient number of vulnerabilities, the firm experience turns into a positive learning effect. Furthermore, our findings suggest that there are two advantages for firms continuously working with the same hacker. First, the positive learning effect kicks in earlier if the firm continuously works with the same hacker on the platform. Second, the repetitive working experience with the same hacker amplifies the positive learning effect. Finally, we found that working with the same hackers may lower the overall resolving time. The study provides theoretical contributions and some important implications in how organizations work with the online crowd through an open platform, especially under the context of vulnerability discovery, crowdsourcing, and organizational learning.

Keywords: Crowdsourcing, Bug-bounty, Organizational Learning, Vulnerability Disclosure

1. Introduction

Recently, crowdsourced vulnerability discovery, also known as *bug-bounty*, has received significant attention from firms for discovering security vulnerabilities in their online assets. A vulnerability is a security flaw that arises due to some system design, implementation, or maintenance issues (Krsul, 1998). By exploiting these vulnerabilities, malicious parties can gain unauthorized access to protected resources. Computer hackers often compromise information systems by exploiting software vulnerabilities on these systems (Cavusoglu et al., 2008). Recently, several bug-bounty platforms emerge to tackle the challenges of identifying vulnerability: firms can set up accounts on bug-bounty platforms announcing reward policies for ethical hackers to explore the possibilities of discovering vulnerabilities on the firm's systems. A key advantage of bug-bounties is that firms can leverage a large base of ethical hackers[1] to identify possible loopholes in their systems at a relatively low cost (M. Zhao, Grossklags, & Liu, 2015).

While bug-bounty programs are attractive for firms to identify security vulnerabilities, avoiding harmful eventualities such as data breaches, however, working with a crowd of hackers on bug-bounty platforms can be challenging. First, like many other online communities, the characteristics of online users on bug-bounty platforms can be dynamic and evolving (Faraj, Jarvenpaa, & Majchrzak, 2011). Inadequately fitting past experiences in working with the hackers into a new vulnerability resolution process may potentially harm the resolution efficiency for firms. Furthermore, unlike a typical offline working environment, online platforms lack a shared working identity and culture, which makes it hard for firms to estimate the commitment, communication dynamics, or even outcomes of the partnered online hackers

[1] From this point onwards, ethical hackers are referred to as hackers throughout the paper.

despite the availability of reputational systems (Lykourentzou, et al., 2016). As such, it is unclear whether firms can still learn from their past experience to improve their working efficiency on a bug-bounty platform.

Although research has examined the working relationship between teammates and how a professional's experience may alter their future working performance (Huckman, Staats, & Upton, 2009), studies related to how firms collaborate with online communities are scant. Further, information security literature mostly focuses on the incentives and characteristics of the hackers and the resolving efficiency at the organizational level (Arora et al., 2010; Arora, et al., 2006; Hata, et al., 2017; Kannan & Telang, 2005; Rescorla, 2005; Votipka, et al., 2018). In this paper, we explore and ask: *how does an organization work with the crowdsourcing hackers on an open platform?* Specifically, we analyze the firm performance from an organizational learning perspective. Using a dataset from HackerOne, a leading bug-bounty platform, we provide empirical evidence on the non-linear effect of a firm's experience on their vulnerability resolution-time and how this effect changes with the increase of experience of working with the same hacker.

We show an inverted-U shape relationship between the firm's experience and the vulnerability resolution time. The firm initially performs worse (i.e., resolving in a long time) as it gains more experience. However, after the firm has gained sufficient experience in working with hackers, experience may lead to a positive learning effect – a decrease in resolution time due to the additional experience in resolving vulnerability. Furthermore, we also find that there are two advantages for firms continuously working with the same hacker. First, the positive learning effect kicks in earlier if the firm works with the same hacker multiple times. Second, the repetitive working experience with the same hacker amplifies the positive learning effect. Our results also suggest that working with the same hackers may lower the overall resolving time. These findings provide some important implications for platform designers and organizations that seek to adopt a crowdsourcing approach for vulnerability discovery. To the best of our knowledge, this is one of the first studies to empirically investigate the learning process of an organization while resolving security vulnerability on an bug-bounty platform.

2. Current Literature and Research Gaps

This study draws literature from three distinct areas of research: economics of information security, organizational learning, and crowdsourcing. The field of economics of information security is recent and thriving; it aims to study the economic aspects of cybersecurity by understanding the incentive mechanism of organizations responsible for keeping the system secure, the users who use the system, and the hackers who want to exploit the system. Vulnerability discovery and the resolution of vulnerabilities are the two most important topics studied in this area (Jo, 2018). This study contributes empirically to this literature by studying the relationship between organizations' experience of vulnerability resolution and their performance of resolving such vulnerabilities.

The majority of the work on vulnerability resolution-time has been studied in the context of software patch management and the factors affecting the patch management process. Arora et al., (2010), showed that public disclosure of a vulnerability expedites the patch release process of an organization. A public disclosure pushes the vendor to release the patch much quickly as compared to no disclosure or delayed disclosure. Moreover, the critical vulnerabilities are patched faster as compared to low-risk vulnerabilities. Vulnerability patch management has also been studied within the context of organizational competition and strategy (Arora, Forman, Nandkumar, & Telang, 2006; Jo, 2018). Arora et al. (2006a), argued vendors face two separate threats while competing for patching with another vendor for the same vulnerability. The first effect is the disclosure threat, which comes from the possibility of a patch release by another vendor before them and implicitly disclosing the vulnerability. The second threat comes from the actual competition because users penalize laggards by comparing the responses of vendors that sell a similar product. Jo-Arrah, (2018) studied the competition effect on patch release for vulnerabilities in a web browser market. She focused on market concentration as a measure of competition intensity and found that higher market concentration positively impacts the vendors' responsiveness in patching vulnerabilities. However, a dominant position in the market (such as Google Chrome's) negatively impacts the promptness of releasing a patch. None of the previous studies have analyzed the patch management of an organization while accounting for their past patching experience. One of the possible reasons is the lack of availability of internal organizational data on vulnerability resolution. Generally, there is limited empirical work on issues related to vulnerability discovery and disclosure which is a concern for researchers in the area (Kannan & Telang, 2005; Ransbotham, Mitra, & Ramsey, 2012). In this study, we explore a unique dataset that brings insight into the organizational processes while resolving security vulnerabilities.

The second stream of research relevant to our study is the organizational learning literature. The relationship between experience and performance has been a widely studied topic in organizational learning (Haleblian & Finkelstein, 1999; Kim, Kim, & Miner, 2009; Wright, 1936). We expect that organizations resolving security vulnerabilities on the bug bounty platform learn from the previously resolved vulnerabilities and the experience gained from such vulnerabilities affects the resolution-time of future vulnerabilities. However, the learning process may not be obvious because of the intrinsically complicated nature of vulnerabilities and the crowdsourcing effect of the bug-bounty platform. While there is limited research to guide our understanding of the learning process of a vulnerability resolution, the closest stream of research within this area is the organization's learning process in software development and maintenance literature. The work of Boh, Slaughter, & Espinosa (2007), on the effect of prior experience on software development and maintenance performance, is helpful but differs from a typical vulnerability resolution process. The software maintenance process studied in the Boh et al., (2007) is based on an incremental software development approach, in which software developers incrementally develop and update the software programs. In this process, there is apparently, no time and competition pressure. However, vulnerability resolution is a time-sensitive process (Ablon & Bogart, 2017; Frei, et al., 2006; Jo, 2018; Johnson, et al., 2015), organizations face disclosure, competition as well as an exploitation threat if they do not promptly resolve a vulnerability. Kim & Kim (2014) studied the learning process of an antivirus software company while resolving malware problems. They found that malware resolution-time decreases with the increase in experience and cross-family malware resolution experience has a greater impact than the within-family experience on the resolution-time. Kim & Kim's (2014) study is relevant to our study because of the organizational learning and security perspective, but the malware resolution process is very different from a vulnerability resolution process. A typical malware resolution process relies on human expertise, but a significant part of the process can be automated by computer algorithms (Kim & Kim, 2014; Szor, 2005). Whereas, a vulnerability resolution process is primarily based on human expertise and experience. Moreover, Kim & Kim's (2014) study doesn't consider any sort of crowdsourcing in malware discovery and identification.

This study also make a theoretical contribution to the literature of crowdsourcing. Our focus in this study is to investigate how firms may work with crowdsourced solutions. Most of the current literature in crowdsourcing is focused on how contributors behave and learn in crowdsourcing environment (such as Boudreau, et al., 2011; Riedl & Seidelc, 2019), the motivation and governance of contributors (Shah, 2006), contributors' submission quality (Bockstedt, et al., 2016), or why firms crowdsource in the first place (see review, Thuan, Antunes, & Johnstone, 2016). However, there is presently limited research on how firms work with crowdsourced solutions and how firms collaborate with the crowd in implementing such solutions. Hackers on a bug bounty platform collaborate with firms in two ways; first, they creatively identify a problem (vulnerability) in a system and, second, they help firms in resolving and testing the identified vulnerabilities (Finifter, et al., 2013). In return, firms incentivize hackers for their collaboration. Blohm et al., (2018) characterized crowdsourcing platforms into four categories based on the contribution types. The closest type of crowdsourcing to a bug-bounty platform is the crowdsourcing by heterogenous collaborators who perform a broadcasted search to identify and solve highly technical problems (Blohm, et al., 2018). Our study contributes to the theory of crowdsourcing by exploring the relationship between a firm's performance of working on a crowdsourced solution and how the hackers' (solver) constant collaboration affects the firm's performance.

Overall, this paper bridges gaps in the literature of information security economics, organizational learning, and crowdsourcing. Specifically, by studying the vulnerability resolution-time, we make a theoretical contribution to the vulnerability disclosure and patch management literature. Our findings also contribute to the organizational learning literature by analyzing the relationship between firm's experience and the firm's efficiency. Further, we contribute to the collaboration and team-familiarity aspects of the crowdsourcing literature. These theoretical contributions are novel for all three areas of literature; none of the previous studies have studied vulnerability resolution with an organizational learning and crowdsourcing perspective.

3. Theory & Hypothesis Development

Theoretical developments in organizational learning (Crossan, et al., 1999) and crowdsourcing literature (such as, Blohm, et al., 2018) guide our understanding of the process through which a firm's security team gains knowledge and experience over time as they interact with the hackers on a bug bounty platform. Organization learning is defined as the change in the organization's performance, such as problem-solving outcomes, production, financial outlook, or task completion times, as the organization acquires experience (Argote, et al., 2009; Dutton & Thomas, 1984). In the context of bug-bounty platforms, as organizations harness the wisdom of the crowd, i.e., hackers, organizational learning takes place while resolving security vulnerability reports, and it may reflect in a measurable metric such as the resolution-time of these vulnerabilities.

While it is known that organizational experience generally improves firm performance through utilizing the knowledge of involved stakeholders (e.g., Reagans, et al., (2005)), the dynamic nature of the online community leads to a higher uncertainty in the communication process and resolution outcomes. The working experience with one or a few hackers may not necessarily apply to the next work process with a new reporting hacker. That is, with limited learning, performance can suffer due to an "over-generalization" of experience. Researchers have studied over-generalization of experience by analyzing the relationship between limited experience and spurious successes or failures (Haleblian & Finkelstein, 1999; Musaji, Schulze,

& De Castro, 2020; B. Zhao & Olivera, 2006). On the one hand, a spurious success can reduce the motivation to learn from potential or near-failures. On the other hand, a spurious failure can replace or modify a potentially reliable problem-solving process with an unreliable process. Thus, both spurious successes, as well as spurious failures are determinantal to performance.

We posit a similar phenomenon in the vulnerability resolution process. Organizations with limited vulnerability resolution experience may encounter spurious successes or failures. In contrast, those with more experience may be able to identify and select reliable problem-solving methods and routines. Thus, we expect to see an increase in resolution time at low levels of experience due to the over-generalization before seeing a positive learning effect or an inverted-U shape relationship between the firm experience and the vulnerability resolution time. When there is a limited amount of working experience, the vulnerability resolution time increases as the experience increases, which leads to an upward trend in vulnerability resolution time. However, after the firm has gained a sufficient amount of working experience on the platform, the positive learning effect of experience kicks-in; this positive learning effect leads to a downward trend in vulnerability resolution time.

H1. Firm's vulnerability resolution time has an inverted U-shape relationship with the number of vulnerabilities resolved in the past.

In the vulnerability resolution process, one of the sources of over-generalization comes from the diversity of crowdsourcing hackers. That is, since the working and communication patterns of hackers vary, the experience with one certain hacker may not necessarily apply to the other hackers. While generalizing the limited experience with the prior hackers to the future hackers could result in an over-generalization, partnering with the same hacker over a period of time could reduce the over-generalization. In the literature, researchers have shown that individual experience only benefits the team performance when they are familiar with others in the team (Huckman et al., 2009). In the case of a bug-bounty platform, the hacker and the firm's security team may work as a collaborative unit leading to an increase in performance, such as reduced vulnerability resolve-time. Under the virtual environment, teams collaborating in a concerted manner have shown higher performance (Riedl & Woolley, 2017). Thus, while the over-generalization effect leads to an inverted-U experience effect (H1) on a firm's performance, we anticipate that the positive learning effect may kick in earlier (at less experience) if the firm works with the same hacker repeatedly. In other words, we expect that the turning point of the inverted-U relationship between the firm's experience and vulnerability resolution time decreases as the firm is more familiar with the hacker of the reported vulnerability.

H2. The turning point of the inverted-U relationship between the firm's experience and vulnerability resolution time decreases as the firm gains more experience of working with the same hacker.

Generally, team familiarity yields superior performance, and firms with higher internal team familiarity have higher learning rates (Reagans et al., 2005). Team familiarity is based on increased coordination and willingness to engage in a relationship between team members. Uzzi (1996) has found that coordination also helps in developing trust. Trust between team members also promotes the exchange of private knowledge and information critical for learning (Uzzi & Lancaster, 2003). On bug-bounty platforms, with the increased experience with the same hacker, the firm's security team develops increased coordination. It develops a higher level of trust with the hacker to resolve security vulnerabilities. These factors subsequently impact the firms learning rates, which in turn yields a higher resolution efficiency. Formally, we operationalize the hypothesis as follows.

H3. The working experience with the same hacker increases the concavity of the inverted U-shape relationship between the firm's experience and vulnerability resolution time.

4. Study Context and Data

In this study, we use a dataset gathered from a bug-bounty platform HackerOne.com. This intermediary connects organizations with ethical hackers. HackerOne is currently one of the largest bug bounty platforms with the highest number of registered hackers, organizations, and reports submissions (Luna, Allodi, & Cremonini, 2019).

Once a hacker submits a vulnerability report to a firm, the firm evaluates its validity. If the report is valid, the firm deploys resources to find a solution to fix the vulnerability. Later, the report can be marked as closed by the firm once the vulnerability is fixed. To study the learning process of firms in resolving such security vulnerabilities, we collected 51,580 valid submitted and closed vulnerability reports for 314 firms since the inception of the platform in November 2013 till August 2019. For each vulnerability report, data about the firm, the creator (i.e., the hackers), whether or not the bounty is awarded, and the timestamp when the report was marked as resolved by the firm is available. In this data, 6,307 out of 51,580 vulnerability reports have been made publicly available after being resolved by the firm. These detailed reports provide information on the vulnerability reporting time, the severity (none, low, medium, high, critical) of the vulnerability, the bounty amount paid by the firm, the discussion comments between the hacker and the firm's security team, and the vulnerability resolution time.

4.1 Dependent Variable: Vulnerability Resolution Time

The first hypothesis examines the overall relationship between the time spent by a firm in resolving reported vulnerabilities and the firm's experience of resolving such vulnerabilities on the platform. The second and the third hypotheses study the moderating effect of working with the same hacker on the aforementioned relationship. The vulnerability resolution time is the main dependent variable used to examine all the hypotheses in our study. The vulnerability resolution time is defined as the number of days elapsed between the reporting of a vulnerability to a firm and its eventual resolution by the firm. We denote $ResolveTime_{i,t_r,h}$, as the number of days needed to resolve the r^{th} vulnerability report submitted to a firm i by hacker h at time t_r. Following Arora et al. (2010), we compute the $ResolveTime_{i,t_r,h}$ in days. From the dataset, we observe that $ResolveTime_{i,t_r,h}$ has a skewed distribution with a mean of 46.6 days, standard deviation of 100.3 days, skewness of 4.98 and kurtosis as 37.35 days. To address skewness and maintain consistency with prior learning literature (e.g., Kim & Kim, 2014), we perform log-transformation and use the dependent variable as $LogResolveTime_{i,t_r,h}$. Also, given that only certain reports on the platform are publicly disclosed, we conduct analysis using the data from the disclosed reports. While our analyses are based on these disclosed reports, we correct for the sample selection bias using a Heckman selection model in all our models.

4.2 Independent Variable: Firm's Vulnerability Resolution Experience

After validating the initial report filed by a hacker on the platform, the firm's security team works on a solution to fix the vulnerability and then marks it as resolved. Regardless of whether a report is publicly disclosed or not, each report's closing time is noted on the platform. Using these closing timestamps, we can track the number of reports closed on any particular day. Using information about the report creation date of the publicly disclosed reports, we compute the number of reports that have been closed before the arrival of a new report. Thus, our main independent variable, $FirmExperienceAll_{i,t_r,h}$, is the total number of reports resolved by the firm *i* before the creation time t_r of the report submitted the hacker h. Again, as with $ResolveTime_{i,t_r,h}$, we log-transform $FirmExperienceAll_{i,t_r,h}$ to $LogFirmExperienceAll_{i,t_r,h}$

to address skewness (mean = 202.73, standard deviation = 389.7, skewness = 3.76, kurtosis = 20.0).

4.3 Moderator: Hacker Experience Effect

We want to assess the impact of the experience of working with the same hacker as compared to a variety of hackers on the experiential learning effect of the firm. For this, we examine the moderating effect of the number of reports submitted by the same hacker on the relationship between the firm's overall experience and vulnerability resolve time. To capture the hacker experience effect on the resolving time of a focal report, we consider the prior reports submitted by the focal hacker of that report. First, we denote the prior experience of working with the focal hacker *h* by $FirmExperienceHacker_{i,t_r,h}$, i.e., the number of reports from the hacker *h* that have been resolved by firm *i* before the creation time t_r of the focal report *r* submitted by the same hacker *h*. Then, the ratio $RatioFirmExperienceHacker_{i,t_r,h}$ is the fraction of $FirmExperienceHacker_{i,t_r,h}$ reports among all reports closed by the firm prior to t_r.

$$RatioFirmExperienceHacker_{i,t_r,h} = \frac{FirmExperienceHacker_{i,t_r,h}}{FirmExperienceAll_{i,t_r,h}} \quad (1)$$

5. Empirical Models and Specification

We begin by empirically analyzing the relationship between the firm's vulnerability resolving time and the resolving experience using a linear regression model. In this model, we control for the time-invariant firm and hacker characteristics while using a firm fixed effect and hacker fixed effect. Furthermore, to capture the platform-wide policy changes over time, we incorporate a time fixed effect. Given the nature of the empirical study using archival dataset, we should note that the firm's vulnerability resolution time and its experience can still be endogenous. For example, firms may put more resources as they gain more experience on the platform, which in turn lower the resolution time as well. While finding a valid instrument is challenging, we use an instrument-free approach to address such a concern in the later section.

The model in Equation (2) analyzes hypothesis H1, i.e., the overall learning effect on firms when resolving vulnerabilities on the bug bounty platform.

$$LogResolveTime_{i,t_r,h} = \gamma_1 LogFirmExperienceAll_{i,t_r,h} + \gamma_2 LogFirmExperienceAll^2{}_{i,t_r,h} + Firm_i + Hacker_r + Day_t + VulnerabilityType_r + \gamma_3 InverseMillsRatio_r + Z_{i,r}\mathbf{\Gamma} + \varepsilon_{i,t_r} \quad (2)$$

In the model, the $LogFirmExperienceAll$ and $LogFirmExperienceAll^2$ capture the non-linear learning effect of the firms. If H1 is supported, we posit that there is an inverted U-shaped relationship between the resolving time and the experience of resolving reports. Accordingly, if H1 is supported, the coefficients $\gamma_1$will be positive and γ_2 will be negative.

$Firm_i$ and $Hacker_r$ are fixed effects controlling for the unobserved time-invariant characteristics of firms and hackers, respectively. $VulnerabilityType_r$ is the vulnerability fixed effect capturing the unobserved time-invariant characteristics of a specific vulnerability. There are 102 different vulnerability types (e.g., code injection, brute force attack) identified by hackers on the platform. Similarly, Day_t is the time fixed effect controlling for the platform-wide time-invariant effects. While these fixed effects will not directly be estimated, the inclusion of these fixed effects allows us to capture the potential heterogeneity across reports.

Since we can only observe resolution time for the disclosed reports, we correct for sample selection bias using a Heckman correction (Heckman, 1979) and estimate the inverse Mills Ratio ($InverseMillsRatio_r$) using a Probit model based on the temporal characteristics of firms and hackers, such as firm's resolution experience, hacker's overall experience, bounty amount if awarded, number of assets available for discovery, and the variety of assets available at the time of the disclosure of a report. The estimated Heckman model is presented in the appendix of the paper. The $InverseMillsRatio_r$ estimated from the Heckman model is incorporated in the main model in equation (2).

$Z_{i,r}$ is the matrix of the report characteristics such as the severity level, the number of comments received, the number of participants, the bounty amount, and so forth. We have also added time-varying hacker and firm characteristics in this matrix, such as hacker's experience on the platform, the number of firms' assets, and the number of different types of firm's assets in scope at a particular date. The full listing of control variables ($Z_{i,r}$) is given in Table 1. Finally, ε_{i,t_r} is the idiosyncratic error.

Table 1: Descriptive Statistics of All Variables

Variable	**Description**	**Mean**	**Std Dev.**
Total Number of Disclosed Reports = 6,307			
$LogResolveTime_{i,t_r,h}$	The log of the number of days elapsed between report creation date and the report resolving date for report r submitted to firm i by hacker h.	2.51	1.709
$LogFirm$ $ExperienceAll_{i,t_r,h}$	The log of the number of resolved reports by the firm *i* before the creation time t_r of report *r* submitted by hacker h.	3.816	2.045
$RatioFirm$ $ExperienceHacker_{i,t_r,h}$	The ratio of the number of reports closed with hacker h by the firm i to the overall reports resolved by the firm *i* before creation time t_r of report r submitted by hacker h.	0.014	0.070
$Numberof$ $Comments_{i,t_r,h}$	The number of comments between the hacker h and the firm *i* on the report *r* created at the time t_r.	11.57	7.51
$Severity_{i,t_r,h}$ $(Critical - None)$	Six different levels of severity indicators assigned by the firm *i* to the report *r* created by hacker h at the time t_r.	-	-
Log $BountyAmount_{i,t_r,h}$	The log of the amount of the bounty awarded by the firm *i* for report *r* created by hacker h at the time t_r.	2.981	3.134
$StatusIndicators_{i,t_r,h}$ $Resolved, Informative$ $Not - Applicable$	Three status indicators, Resolved, Information and Not Applicable based on the firms i final decision for report *r* created by hacker h at the time t_r.	-	-

$Numberof\ AssetsAvailable_{i,t_r,h}$	The number of assets on which the firm *i* has promised to offer a bounty at the time $t_r$when report *r* was created by the hacker *h*.	10.46	58.74
$Numberof\ TypesofAssets_{i,t_r,h}$	The number of types of assets on which the firm *i* is accepting reports at the time $t_r$when report r was created by hacker *h*.	0.499	1.17
$Hacker\ ExperienceTotal_{i,t_r,h}$	The number of resolved reports of the hacker *h* on the platform before the creation of report r for firm *i* at the time t_r from hacker *h*.	10.68	24.0
$InverseMillsRatio_{i,t_r}$	The value of the inverse mill ratio estimated using the Heckman first stage model at the time t_r when report r was created for firm *i*.	0.398	0.833

Equation (2) studies the hypothesis presented in H1; however, it does not consider the effect of the experience gathered while working with the same hacker. H2 and H3 capture the effect of learning from the reports while working with the same hacker over time. To test these hypotheses, the model in Equation (3) examines the moderating effect of the focal hacker experience using the moderator $RatioFirmExperienceHacker_{i,t_r,h}$.

$$\begin{aligned}LogResolveTime_{i,t_r,h} &\\ = \delta_0 &+ \delta_1 LogFirmExperienceAll_{i,t_r,h} + \delta_2 LogFirmExperienceAll^2{}_{i,t_r,h} \\ &+ \delta_3 RatioFirmExperienceHacker_{i,t_r,h} \\ &+ \delta_4\left(LogFirmExperienceAll_{i,t_r,h} \times RatioFirmExperienceHacker_{i,t_r,h}\right) \\ &+ \delta_5\left(LogFirmExperienceAll^2{}_{i,t_r,h} \times RatioFirmExperienceHacker_{i,t_r,h}\right) \\ &+ Firm_i + Hacker_r + Day_t + VulnerabilityType_r + InverseMillsRatio_r \\ &+ Z_{i,t_r}\mathbf{\Gamma} + \varepsilon_{i,t_r} \quad (3)\end{aligned}$$

where, ε_{i,t_r} is the idiosyncratic error, $Z_{i,t_r}$is the same matrix used in estimating the Equation (2).

6. Main Analysis

The main results are estimated using fixed-effect linear regression analysis. In Table 2, Column (1) only focuses on the linear experience effect. Column (2) analyzes the non-linear relationship between vulnerability resolution-time and experience and exhibits an inverted U-shape relationship between the firm experience and its resolving time. ($\gamma_1 = 0.3397$ and $\gamma_2 = -0.04441$).

Table 2: Main Analysis

	(1)	**(2)**	**(3)**
Dependent Variable	$\boldsymbol{Log\ ResolveTime_{i,t_r,h}}$	$\boldsymbol{Log\ ResolveTime_{i,t_r,h}}$	$\boldsymbol{Log\ ResolveTime_{i,t_r,h}}$
$LogFirmExperienceAll_{i,t_r,h}$	0.1185***	0.3397***	0.3184***
	(0.04066)	(0.08275)	(0.08447)
$LogFirmExperienceAll^2{}_{i,t_r,h}$		-0.04441***	-0.03904***
		(0.01485)	(0.01500)
$RatioFirmExperienceHacker_{i,t_r,h}$			-2.2038*
			(1.1585)
$LogFirmExperienceAll_{i,t_r,h} \times RatioFirmExperienceHacker_{i,t_r}$			2.6364**
			(1.1477)
$LogFirmExperienceAll^2{}_{i,t_r,h} \times RatioFirmExperienceHacker_{i,t_r}$			-0.5509**
			(0.2296)
$SeverityCritical_{i,t_r,h}$	-0.7762***	-0.7576***	-0.7455***
	(0.1867)	(0.1869)	(0.1869)
$SeverityHigh_{i,t_r,h}$	-0.5164***	-0.5036***	-0.5101***
	(0.1244)	(0.1238)	(0.1236)
$SeverityMedium_{i,t_r,h}$	-0.2673**	-0.2639**	-0.2683**
	(0.1048)	(0.1047)	(0.1046)
$SeverityUnknown_{i,t_r,h}$	-0.5614***	-0.5432***	-0.5239***
	(0.1331)	(0.1330)	(0.1333)
$SeverityNone_{i,t_r,h}$	-0.8047***	-0.7869***	-0.7862***
	(0.1770)	(0.1761)	(0.1763)
$NumberofComments_{i,t_r,h}$	0.06272***	0.06287***	0.06280***
	(0.005488)	(0.005486)	(0.005514)
$LogBountyAmount_{i,t_r,h}$	-0.03237*	-0.02836*	-0.02577
	(0.01722)	(0.01721)	(0.01728)
$NotApplicableStatus_{i,t_r,h}$	-0.2177	-0.2036	-0.1912
	(0.2902)	(0.2839)	(0.2826)
$ResolvedStatus_{i,t_r,h}$	1.1048***	1.0988***	1.0981***
	(0.1347)	(0.1347)	(0.1349)
$NumberofAssetsAvailable_{i,t_r,h}$	0.002805***	0.002508**	0.002438**
	(0.001084)	(0.001078)	(0.001095)
$NumberofTypesofAssets_{i,t_r h}$	-0.3166***	-0.2842***	-0.2948***

	(0.09696)	(0.09748)	(0.09794)
$HackerExperienceOverall_{i,t_r,h}$	0.1572***	0.1667***	0.1865***
	(0.05396)	(0.05373)	(0.05482)
$InverseMillsRatio_r$	0.04685	0.03799	0.03603
	(0.04904)	(0.04897)	(0.04914)
Firm & Asset Type Fixed Effects	Yes	Yes	Yes
Day Fixed Effects	Yes	Yes	Yes
Hacker Fixed Effects	Yes	Yes	Yes
Vulnerability Type	Yes	Yes	Yes
Observations	4354	4354	4354
R-squared	0.785	0.786	0.787

* p<0.1, ** p<0.05, *** p<0.01

To further test the U-shape relationship postulated by H1, we use a three-step procedure suggested by Lind & Mehlum (2010). First, we ensure that γ_2 is significant and of the opposite sign of the γ_1, i.e. $\gamma_1 > 0, \gamma_2 < 0, both\ are\ significant$. Second, we note that the slope is sufficiently steep at both ends of the data range. Third, the turning point is observed to be well within the data range. Taking the first derivative of Equation (2) and setting it zero, yields the turning points at 3.82*** (std. error 0.651), 95% CI [2.548 – 5.101]. Our turning point is well within the data range (i.e. 0 – 8.03). The firm has to resolve at least 3.825% of reports (or $e^{3.825} \approx 47.087$ reports on average) before the resolution time starts decreasing. Figure 1 supplements these tests and shows the non-linear relationship between the firm's experience and its vulnerability resolution time.

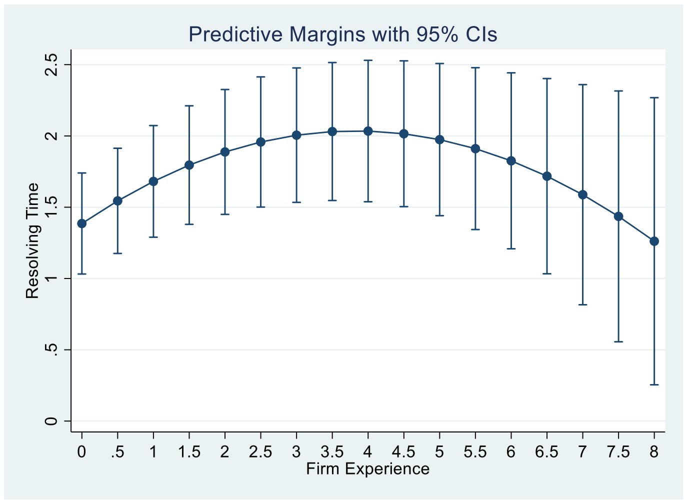

Figure 1. Estimated Relationship between the Firm's Experience and its Resolving Time

Column (3) analyzes the relationship between experience and resolution time while moderating the experience of working with the same hacker. In the U-shaped relationship, the moderation by a variable can work in two distinct ways: it can shift the turning point of curve left or right, and it can flatten or steepen the curve (Haans, Pieters, & He, 2016). If the moderator $RatioFirmExperienceHacker$ shifts the curve to the left and also steepens it at the same time, we can say that hypotheses H2 and H3 are supported. Following the guidelines provided by Haans et al. (2016), we test how the moderation affects the turning point in this U-shaped relationship. To find the turning point in Equation (3) in regard to the firm's experience, we find the first-order condition with respect to $LogFirmExperienceAll$ which gives us Equation (4).

$$LogFirmExperienceAll^{*} = \frac{-\delta_1 - \delta_4(RatioFirmExperienceHacker\,)}{2\delta_2 + 2\delta_5(RatioFirmExperienceHacker)} \quad (4)$$

From Equation (4), we observe that the turning point $LogFirmExperiencAll^{*}$ depends on the moderator, $RatioFirmExperienceHacker$. To show how the turning point changes with $RatioFirmExperienceHacker$, we take the derivative of Equation (4) with respect to $RatioFirmExperienceHacker$. This gives us:

$$\frac{\partial LogFirmExperienceAll}{\partial RatioFirmExperienceHacker} = \frac{-\delta_1\delta_5 - \delta_2\delta_4}{2(\delta_2 + \delta_5(RatioFirmExperienceHacker)} \quad (5)$$

To test formally whether a shift in the turning point occurs, we assess whether Equation (5) as a whole is significantly different from zero at specific and meaningful values of the moderator (Haans et al. 2016). A positive and significant Equation (5) suggests that the turning point is shifting to the right as the moderator increases, while a negative and significant Equation (5) suggests that the turning point is shifting to the left as the moderation increases. Since a very small number of firms have the experience of working with the same hacker, the mean and standard deviation of the R$atioFirmExperienceHacker$ is at 0.0577 and 0.03216 respectively. Upon formally testing the Equation (5) at the mean value as well as the 1-2 standard deviations above the mean, we find that Equation (5) is significant and negative for all these values of the $RatioFirmExperienceHacker$, suggesting a significant shift to the left. Thus, H2 is supported. Table 3 shows the shift in the turning point as moderator increases using Equation (4).

Table 3: Turning Point Shifting to the Left with Increase in Moderator

(1)	(2)	(3)
RatioFirmExperienceHacker	***LogFirmExperienceAll**** **(Std. Err.)**	**95% Conf. Interval**
0.0	4.070*** (0.830)	[2.449 – 5.704]
0.1	3.091***(0.302)	[2.497 – 3.684]
0.2	2.837***(0.270)	[2.307 – 3.366]
0.3	2.715***(0.270)	[2.186 – 3.245]
0.4	2.645***(0.274)	[2.107 – 3.183]
0.5	2.600***(0.279)	[2.052 – 3.147]
0.6	2.568***(0.283)	[2.012 – 3.123]
0.7	2.544***(0.278)	[1.981 – 3.107]
0.8	2.526***(0.290)	[1.957 – 3.094]
0.9	2.511***(0.292)	[1.938 – 3.085]
1	2.499***(0.294)	[1.921 – 3.077]

To test hypothesis H3, i.e. whether the moderation is steepening the curve, we test the significance of the coefficient on the interaction between the moderator and the quadratic term in the specification model (Haans et al., 2016). The results show that the coefficient on δ_5 (-0.5723**) is negative and significant, and thus the curve steepens for the firms who have experience working with the same hacker. This steepening also suggests that working with the same hacker increases the rate of learning. Thus, hypotheses H3 is also supported.

Another important observation from the Column (3) is the decrease in the intercept for the moderated curve. The coefficient on the $RatioFirmExperienceHacker$ is negative and significant at 10% significance level. It suggests that working with the same hacker shifts the overall curve downward for the firm, implying a lower resolution time for working with the same hacker, at the same level of experience, as compared to working with a variety of hackers.

7. Endogeneity Correction Using Copula Approach

Although we control for time-invariant fixed effects for the firms, hackers, platform, and the characteristics of the vulnerability types, there is a possibility of other confounding factors that we have not observed, which can affect the firm's experience as well as the resolution performance at the same time. Typically, to alleviate the endogeneity problem, researchers can model the endogenous variable using exogenous variables in an instrument variable approach. The modeled endogenous variable would be considered as uncorrelated with the error term. However, finding a valid IV is typically challenging, and the exclusion restriction assumption is usually untestable. Another way to tackle the possible endogeneity is to explicitly model the correlation between the endogenous variable and the error term (see Ebbes, et al., 2009) such as the copula approach (Park & Gupta, 2012). Park & Gupta (2012) address the endogeneity problem by estimating the joint distribution of the endogenous regressor and the error term using a Gaussian copula function. Copulas are functions that join or "couple" multivariate distributions to their one-dimensional marginal distribution functions (Balakrishna & Lai, 2009). Park & Gupta (2012) proposed to construct a multivariate distribution that effectively captures the correlations between the regression and the structural error with the assumption that the endogenous variable does not have a normal distribution. Once the correlation is explicitly modeled, the biased estimates due to the endogeneity problem can be alleviated.

Consider the model presented in Equation 2, if $LogFirmExperienceAll$ is endogenous, the correlation between $LogFirmExperienceAl$ and error term ill not be zero i.e. $E(LogFirmExperienceAll\ \epsilon) \neq 0$. To model this correlation, we can assume that the endogenous regressor, $LogFirmExperienceAll$ is comprised of two variables, x_1 and X_2, which represents an endogenous part and an exogenous regressor respectively. We assume that the CDF of the error term (ε_{i,t_r}) in the model follows normal distribution with mean 0 and variance ${\sigma^2}_{\varepsilon}$. The joint CDF of the Gaussian copula between error term and x_1 can be represented as:

$$G(x_1, \varepsilon) = N(x_1^*, \varepsilon^*\ ) \tag{6}$$

Where $x_1^* = \Phi^{-1}(F_x(x_1))$, $\varepsilon^* = \Phi^{-1}(F_\epsilon(\epsilon))$, F_x signifies the CDF of x_1, Φ denoted the standard normal CDF and N is the bivariate standard normal distribution with correlations coefficient ρ. By differentiating the joint probability density functions in Equation 6:

$$g(x_1, \epsilon) = \frac{\delta\ \delta}{\delta x_1 \delta \varepsilon} f_x f_\epsilon,$$

Where f_x and f_ϵ are the marginal densities of $x_1$and ϵ respectively. Following the Park & Gupta (2012) method, we can rewrite Equation 2 by splitting up $LogFirmExperienceAll$ in an exogenous and endogenous part by including the normal inverse copula as follows:

$$
\begin{aligned}
&LogResolveTime_{i,t_r,h}\\
&\quad = \gamma_1 LogFirmExperienceAll_{i,t_r,h} + \gamma_2 LogFirmExperienceAll^2{}_{i,t_r,h}\\
&\quad + \gamma_3 InverseCopulaCorrection_{it_r}{}^{*} + Firm_i + Hacker_r + Day_t\\
&\quad + VulnerabilityType_r + \gamma_4 InverseMillsRatio_r + Z_{i,r}\mathbf{\Gamma} + \varepsilon_{i,t_r} \qquad (6)
\end{aligned}
$$

Following Park & Gupta method, we estimate the normal inverse copula correction created from an estimated density function for x_1 recovered from the Epancechnikov kernel function. Table 4 shows the estimation results with the copula correction plugged into Equation (2) and (3).

Table 4: Endogeneity Correction Using Copulas

	(1)	**(2)**	**(3)**
Dependent Variable	$\boldsymbol{Log\ ResolveTime_{i,t_r,h}}$	$\boldsymbol{Log\ ResolveTime_{i,t_r,h}}$	$\boldsymbol{Log\ ResolveTime_{i,t_r,h}}$
$LogFirmExperienceAll_{i,t_r,h}$	0.4855***	0.4055***	0.3864**
	(0.1483)	(0.1555)	(0.1610)
$LogFirmExperienceAll^2{}_{i,t_r,h}$		-0.03712*	-0.03179
		(0.02159)	(0.02191)
$InverseCopulaCorrection_{it_r}$	-0.9375***	-0.2602	-0.2626
	(0.3631)	(0.5294)	(0.5421)
$RatioFirmExperienceHacker_{i,t_r,h}$			-2.0517*
			(1.1666)
$LogFirmExperienceAll_{i,t_r,h} \times RatioFirmExperienceHacker_{i,t_r}$			2.5368**
			(1.1484)
$LogFirmExperienceAll^2{}_{i,t_r,h} \times RatioFirmExperienceHacker_{i,t_r}$			-0.5404**
			(0.2292)
Baseline Control[†]	Yes	Yes	Yes
Observations	4355	4355	4355
R-squared	0.786	0.786	0.787

* p<0.1, ** p<0.05, *** p<0.01

[†]All the control variables and fixed effects were not reported in the Table 4 due to the space limitation.

In Table 4, we find that our results are consistent with the main model qualitatively. Due to the limitation of space, we only reported the main variables and the copula regressor denoted by $InverseCopulaCorrection_{it_r}$. The results in Table 4 are all consistent with the main results: there is an inverted-U shape relationship between firm experience and vulnerability resolution time. In Column (3), $LogExperience^2$ is insignificant with a negative sign. One possible reason for this is the high multicollinearity between the linear term, the square term, and the inverse copula term. However, the moderations with the linear term, the quadratic term, and the intercept of the moderation is significant and with the expected signs as in Table 2. Therefore, we argue that Park & Gupta's instrument-free endogeneity correction method has produced consistent estimates and alleviated the unknown sources of endogeneity from our model.

8. Conclusion and Discussion

Theoretically, our study contributes to the growing literature of organizational learning, the economics of information systems, and crowdsourcing. The previous literature has not fully explored how firms work with online communities or in an open platform. Moreover, the information security literature has not studied the firm's performance of vulnerability resolution with changing experience.

On the practical side, our findings address one of the major pre-adoption fears of a bug-bounty program, which is related to the high cost of processing the reported vulnerabilities (Al-banna, et al., 2018). Firms also fear over-burdening their security teams and reducing their overall efficiency. Our initial results suggest that firms can leverage bug bounty programs while working with a small number of hackers. Working with a few hackers not only reduces the period of negative resolution performance but also increases the rate of learning. Once the firm's security teams have gained adequate experience, they may leverage their programs with a larger crowd. Therefore, a viable strategy in online communities such as bug bounty platforms is to start with a limited pool of crowdsourced workers while keeping high resolution efficiency.

References

Ablon, L., & Bogart, A. (2017). *Zero Days, Thousands of Nights: The Life and Times of Zero-Day Vulnerabilities and Their Exploits*. *Zero Days, Thousands of Nights: The Life and Times of Zero-Day Vulnerabilities and Their Exploits*. RAND Corporation. https://doi.org/10.7249/rr1751

Al-banna, M., Schlagwein, D., Bertino, E., Barukh, M. C., & Schlagwein, D. (2018). Friendly Hackers to the Rescue : How Organizations Perceive Crowdsourced Vulnerability Discovery Friendly Hackers to the Rescue : How Organizations Perceive Crowdsourced Vulnerability Discovery. *PACIS 2018 Preceedings*.

Argote, L., Beckman, S. L., & Epple, D. (2009). The persistence and transfer of learning in industrial settings. In *The Strategic Management of Intellectual Capital*. https://doi.org/10.1287/mnsc.36.2.140

Arora, A., Forman, C. M., Nandkumar, A., & Telang, R. (2006). Competitive and strategic effects in the timing of patch release. *October*, (October 2005). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.79.6891&amp;rep=rep1&amp;type=pdf

Arora, A., Krishnan, R., Telang, R., & Yang, Y. (2010). An empirical analysis of software vendors' patch release behavior: Impact of vulnerability disclosure. *Information Systems Research*, *21*(1), 115–132. https://doi.org/10.1287/isre.1080.0226

Arora, A., Nandkumar, A., & Telang, R. (2006). Does information security attack frequency increase with vulnerability disclosure? An empirical analysis. *Information Systems Frontiers*, *8*(5), 350–362. https://doi.org/10.1007/s10796-006-9012-5

Balakrishna, N., & Lai, C. D. (2009). *Continuous bivariate distributions*. *Continuous Bivariate Distributions*. https://doi.org/10.1007/b101765

Blohm, I., Zogaj, S., Bretschneider, U., & Leimeister, J. M. (2018). How to manage crowdsourcing platforms effectively? *California Management Review*, *60*(2), 122–149. https://doi.org/10.1177/0008125617738255

Bockstedt, J., Druehl, C., & Mishra, A. (2016). Heterogeneous Submission Behavior and its Implications for Success in Innovation Contests with Public Submissions. *Production and Operations Management*, *25*(7), 1157–1176. https://doi.org/10.1111/poms.12552

Boh, W. F., Slaughter, S. A., & Espinosa, J. A. (2007). Learning from experience in software development: A multilevel analysis. *Management Science*, *53*(8), 1315–1331. https://doi.org/10.1287/mnsc.1060.0687

Boudreau, K. J., Lacetera, N., & Lakhani, K. R. (2011). Incentives and problem uncertainty in innovation contests: An empirical analysis. *Management Science*, *57*(5), 843–863. https://doi.org/10.1287/mnsc.1110.1322

Cavusoglu, H. H., Cavusoglu, H. H., & Jun, Z. (2008). Security patch management: Share the burden or share the damage? *Management Science*, *54*(4), 657–670. https://doi.org/10.1287/mnsc.1070.0794

Crossan, M. M., Lane, H. W., & White, R. E. (1999). An organizational learning framework: From intuition to institution. *Academy of Management Review*, *24*(3), 522–537.

https://doi.org/10.5465/AMR.1999.2202135

Dutton, J. M., & Thomas, A. (1984). Treating Progress Functions as a Managerial Opportunity. *Academy of Management Review*, *9*(2), 235–247. https://doi.org/10.5465/amr.1984.4277639

Ebbes, P., Wedel, M., & Böckenholt, U. (2009). Frugal IV alternatives to identify the parameter for an endogenous regressor. *Journal of Applied Econometrics*. https://doi.org/10.1002/jae.1058

Faraj, S., Jarvenpaa, S. L., & Majchrzak, A. (2011). Knowledge collaboration in online communities. *Organization Science*, *22*(5), 1224–1239.

Finifter, M., Akhawe, D., & Wagner, D. (2013). An empirical study of vulnerability rewards programs. *Proceedings of the 22nd USENIX Security Symposium*, 273–288.

Frei, S., May, M., Fiedler, U., & Plattner, B. (2006). Large-scale vulnerability analysis. In *Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense* (pp. 131–138).

Haans, R. F. J., Pieters, C., & He, Z. (2016). Thinking about U: Theorizing and testing U-and inverted U-shaped relationships in strategy research. *Strategic Management Journal*, *37*(7), 1177–1195.

Haleblian, J., & Finkelstein, S. (1999). The influence of organizational acquisition experience on acquisition performance: A behavioral learning perspective. *Administrative Science Quarterly*. https://doi.org/10.2307/2667030

Hata, H., Guo, M., & Babar, M. A. (2017). Understanding the Heterogeneity of Contributors in Bug Bounty Programs. *International Symposium on Empirical Software Engineering and Measurement*, *2017-Novem*, 223–228. https://doi.org/10.1109/ESEM.2017.34

Heckman, J. J. (1979). Sample Selection Bias as a Specification Error. *Econometrica*, *47*(1), 153–161. https://doi.org/10.2307/1912352

Huckman, R. S., Staats, B. R., & Upton, D. M. (2009). Team familiarity, role experience, and performance: Evidence from Indian software services. *Management Science*, *55*(1), 85–100. https://doi.org/10.1287/mnsc.1080.0921

Jo, A. (2018). The effect of competition intensity on software security - An empirical analysis of security patch release on the web browser market ∗, (August), 1–31.

Johnson, B., Laszka, A., & Grossklags, J. (2015). Games of Timing for Security in Dynamic Environments. In M. H. R. Khouzani, E. Panaousis, & G. Theodorakopoulos (Eds.), *Decision and Game Theory for Security* (pp. 57–73). Cham: Springer International Publishing.

Kannan, K., & Telang, R. (2005). Market for software vulnerabilities? Think again. *Management Science*, *51*(5), 726–740. https://doi.org/10.1287/mnsc.1040.0357

Kim, J. Y., Kim, J. Y. J., & Miner, A. S. (2009). Organizational learning from extreme performance experience: The impact of success and recovery experience. *Organization Science*, *20*(6), 958–978. https://doi.org/10.1287/orsc.1090.0439

Kim, S. H., & Kim, B. C. (2014). Differential effects of prior experience on the malware resolution process. *MIS Quarterly: Management Information Systems*, *38*(3), 655–678.

https://doi.org/10.25300/MISQ/2014/38.3.02

Krsul, I. V. (1998). *Software vulnerability analysis*. Purdue University West Lafayette, IN.

Lind, J. T., & Mehlum, H. (2010). With or without U? The appropriate test for a U-shaped relationship. *Oxford Bulletin of Economics and Statistics*, *72*(1), 109–118.

Luna, D., Allodi, L., & Cremonini, M. (2019). Productivity and patterns of activity in bug bounty programs: Analysis of hackerone and Google vulnerability research. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3339252.3341495

Lykourentzou, I., Wang, S., Kraut, R. E., & Dow, S. P. (2016). Team dating: A self-organized team formation strategy for collaborative crowdsourcing. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 1243–1249).

Musaji, S., Schulze, W. S., & De Castro, J. O. (2020). How Long Does It Take to Get to the Learning Curve? *Academy of Management Journal*, *63*(1), 205–223. https://doi.org/10.5465/amj.2017.1145

Park, S., & Gupta, S. (2012). Handling endogenous regressors by joint estimation using copulas. *Marketing Science*, *31*(4), 567–586. https://doi.org/10.1287/mksc.1120.0718

Ransbotham, S., Mitra, S., & Ramsey, J. (2012). Are markets for vulnerabilities effective? *Mis Quarterly*, 43–64.

Reagans, R., Argote, L., & Brooks, D. (2005). Individual experience and experience working together: Predicting learning rates from knowing who knows what and knowing how to work together. *Management Science*, *51*(6), 869–881.

Rescorla, E. (2005). Is finding security holes a good idea? *IEEE Security and Privacy*, *3*(1), 14–19. https://doi.org/10.1109/MSP.2005.17

Riedl, C., & Seidelc, V. P. (2019). Learning from mixed signals in online innovation communities. *Organization Science*, *29*(6), 1010–1032. https://doi.org/10.1287/orsc.2018.1219

Riedl, C., & Woolley, A. W. (2017). Teams vs. Crowds: A Field Test of the Relative Contribution of Incentives, Member Ability, and Emergent Collaboration to Crowd-Based Problem Solving Performance. *Academy of Management Discoveries*. https://doi.org/10.5465/amd.2015.0097

Shah, S. K. (2006). Motivation, governance, and the viability of hybrid forms in open source software development. *Management Science*, *52*(7), 1000–1014. https://doi.org/10.1287/mnsc.1060.0553

Szor, P. (2005). *The Art of Computer Virus Research and Defense: ART COMP VIRUS RES DEFENSE _p1*. Pearson Education.

Thuan, N. H., Antunes, P., & Johnstone, D. (2016). Factors influencing the decision to crowdsource: A systematic literature review. *Information Systems Frontiers*, *18*(1), 47–68. https://doi.org/10.1007/s10796-015-9578-x

Uzzi, B. (1996). The sources and consequences of embeddedness for the economic performance of organizations: The network effect. *American Sociological Review*. https://doi.org/10.2307/2096399

Uzzi, B., & Lancaster, R. (2003). Relational The Embeddedness of and Bank Loan Learning: Managers Their Clients. *Management Science*.

Votipka, D., Stevens, R., Redmiles, E., Hu, J., & Mazurek, M. (2018). Hackers vs. Testers: A Comparison of Software Vulnerability Discovery Processes. In *Proceedings of the 2018 IEEE Symposium on Security and Privacy* (pp. 374–391). IEEE. https://doi.org/10.1109/SP.2018.00003

Wright, T. P. (1936). Factors affecting the cost of airplanes. *Journal of the Aeronautical Sciences*, *3*(4), 122–128.

Zhao, B., & Olivera, F. (2006). Error reporting in organizations. *Academy of Management Review*. https://doi.org/10.5465/AMR.2006.22528167

Zhao, M., Grossklags, J., & Liu, P. (2015). An empirical study of web vulnerability discovery ecosystems. *Proceedings of the ACM Conference on Computer and Communications Security*, *2015-Octob*, 1105–1117. https://doi.org/10.1145/2810103.2813704

Appendix

Heckman selection model to remove selection bias of disclosure of a report.

	(1)
Dependent Variable	$\boldsymbol{Disclosed_{i,t_r,h}}$
$LogFirmExperienceAll_{i,t_r,h}$	0.1617***
	(0.01756)
${LogFirmExperienceAll^2}_{i,t_r,h}$	-0.03005***
	(0.001930)
$BountyAwarded_{i,t_r,h}$	-0.8944***
	(0.03433)
$LogBountyAmount_{i,t_r,h}$	0.2077***
	(0.005578)
$HackerExperienceOverall_{i,t_r,h}$	-0.1117***
	(0.004784)
$ScopesAvailable_{i,t_r,h}$	0.5085***
	(0.03207)
$NumberofAssetsAvailable_{i,t_r,h}$	0.0001185
	(0.0001455)
$NumberofTypesofAssets_{i,t_rh}$	-0.03632***
	(0.01033)
$Year2012_i$	-
	-
$Year2013_i$	0.1710
	(0.2186)
$Year2014_i$	0.6601***
	(0.03687)
$Year2015_i$	0.6753***
	(0.03643)
$Year2016_i$	0.6633***
	(0.03111)
$Year2017_i$	0.7034***
	(0.02791)
$Year2018_i$	0.2194***
	(0.02642)
$Constant$	-1.4938***
	(0.04703)
Observations	51,570
R-squared	0.142

* p<0.1, ** p<0.05, *** p<0.01