Web Science Challenges in Researching Bug Bounties

HUW FRYER, University of Southampton
ELENA SIMPERL, University of Southampton

The act of searching for security flaws (vulnerabilities) in a piece of software was previously considered to be the preserve of malicious actors, or at least actors who wished to cause chaos. Increasingly, however, companies are recognising the value of running a bug bounty program, where they will pay 'white hat' hackers to locate and disclose security flaws in their applications in order that they can fix it. This is known as a 'bug bounty' or a 'vulnerability reward program', and at present has seen comparatively little research. This paper introduces two existing research on bug bounties in two areas: as a means of regulating the sale of vulnerabilities; and as a form of crowdsourcing. We argue that the nature of bug bounties makes Web science particularly suitable to drive forward research. We identify gaps in the current literature, and propose areas which we consider to be particularly promising for future research.

CCS Concepts: • **Information systems → Crowdsourcing**; • **Security and privacy → Social aspects of security and privacy**;

Additional Key Words and Phrases: Crowdsourcing, Vulnerability research, Bug bounties

1 INTRODUCTION

A 'bug bounty' or 'vulnerability reward program' (VRP) is the process for rewarding the discovery of a flaw or vulnerability in a piece of software. The concept has been around for a long time, notably Donald Knuth offering rewards for omissions in his *The Art of Programming* books, or flaws in his LaTeXsoftware, and in the 1990s Netscape offered a reward for flaws in its browser. Despite this history, examples of its application have been sparse up until the last few years where its popularity has increased, as this decade high profile programs from companies such as Mozilla and Google[26], and even the US Department of Defense in 2016 have started. There now exist services which act as middlemen in connecting companies with people who are prepared to search their systems for weaknesses.

This paper will provide a brief review of some of the key research into bug bounties. For the most part, this has been tangential, merely acknowledging their existence, as a part of overall Web or application security. Nevertheless, there has been more recent research which has considered bug bounties in their own right, analysing the behaviour of the participants, or the means in which a company operating a bug bounty might seek to optimise the quality of the results of their bug bounty. We will identify the gaps in the literature, and identify areas where crowdsourcing and Web Science research can assist in driving research forward in this area.

In Section 2, we introduce some terminology, and background research, before describing our literature review in Section 3. We then present opportunities for future research in Section 4 based on crowdsourcing literature, and discuss the suitability of this research area for Web Science. We conclude with a summary of our paper, and our recommendations.

2 TERMINOLOGY AND BACKGROUND

Within software engineering, a ‘bug’ can have the meaning of any flaw in the application. However, the term ‘bug bounty’ as it is generally used is the sort of program where members of a crowd will locate security flaws in a Web application or piece of software. These flaws will be reported responsibly to the vendor of the software or the administrator of the Web application so that they may be fixed. A security flaw is commonly known as a ‘vulnerability’, a weakness which could be the subject of an ‘exploit’ – something a malicious actor could use to gain some form of unauthorised access to a system. Usually a seller of a vulnerability will be required to show a potential buyer that an exploit exists [27], otherwise a vulnerability is merely theoretical and of no practical danger to the application in question. Vulnerabilities are distinguished by their level of severity and ease of exploitation, with the most severe allowing total control of a system running the software.

The reason security flaws should arguably be discovered in the first place, is that a black market for these vulnerabilities already exists. A hacker who discovers a flaw may be less inclined to inform the victims, but instead choose to sell it to a criminal who would use it to make a profit[14]. The criminal market has reportedly relied upon specialisation of tasks, including one task being to locate security flaws in popular software or websites and selling them to others who may be better positioned to monetise the information or access gleaned in some other way. There is reportedly a complex, sophisticated black market where different actors specialise in different areas, so the person who makes money out of illicit access to a system will likely be an entirely different person.

Additionally, state actors may wish to acquire vulnerabilities. The ability to exploit a particular system is regarded as valuable to nation states for a variety of reasons - whether to spy on each other; their citizens; or to have a means of retaliation following an attack from another state. A nation state has resources way in advance of almost all other actors and are therefore in a position to offer a considerably higher sum should an individual wish to sell them. A discussion of the government role in vulnerability disclosure is beyond the scope of this paper, but see [34] for more details about this area.

Given that malicious actors are selling vulnerabilities, how should the defenders respond? There is conflicting opinion in the literature about whether searching for vulnerabilities constitutes a social good. On the one hand, Rescorla argues that the probability of a defender finding the same vulnerability before a malicious actor is incredibly small[31], whereas Ozment pointed to a depletion in the amount of reported vulnerabilities in FreeBSD as being an indication that searching for bugs made it more difficult for adversaries to do the same[29]. Miller reported a personal experience in 2007 of an attempt to sell an exploit in Microsoft Powerpoint (before bug bounties were generally used) where the flaw was discovered and patched before an agreement was reached as to the price[27].

Amongst Miller’s other points, was the difficulty that a security researcher had in selling a flaw once they had located it [27]. Even were they able to find a buyer, they would have no way of knowing the value which a purchaser would place on it, and being able to prove the effectiveness of the vulnerability without giving away the value of their discovery was also a non-trivial solution. Bug bounties alleviate these difficulties: the price is specified up front, the researcher is aware what kind of vulnerabilities would be in scope, and services which manage bug bounties can act as a trusted third party to ensure they get paid.

This is not to say that bug bounties are the only solution as a means of regulating this. From an economic point of view, the concept of vulnerability markets in general has received some scrutiny in the literature [5, 27, 28, 30]. Böhme identified a typology of different possible means of regulating vulnerability markets, concluding that bug bounties were not the best possible option, having weakness in relation to their efficiency[5]. As an example of an alternative format, there are also competitions related to security flaws, notably ‘pwn2own’ which occurs at the CanSecWest security conference[1].

3 BUG BOUNTIES

3.1 Methodology

Despite the increasing popularity of bug bounties [6], and their seeming relationship with crowdsourcing, we were unaware of any work which considered bug bounties within the context of crowdsourcing. The one exception to this was Su & Pan, who proposed a system to introduce microtasking to the process, where additional actors would test and verify the vulnerability submitted by another researcher [36].

As a result, we conducted a literature review, based on the methodology of Mao et al’s review of the related area of crowdsourced software engineering [24]. The search was for the phrases “bug bount(y|ies)", “vulnerability reward program", “vulnerability disclosure", in any available field in seven online search engines: ACM Digital Library, IEEE Digital Library, Springerlink Online Library, Wiley Online Library, Elsevier ScienceDirect, ProQuest, and Google Scholar. As a fallback, we additionally used snowballing of references where further titles were identified. To identify relevant literature, the title, abstract and introduction sections of each paper were read, which was usually enough to identify it as being outside the criteria for inclusion. Where this was not the case, the whole paper was read.

As an exploratory study, our research question was: what are the gaps in the existing literature related to bug bounties, which can be addressed by crowdsourcing? As a result, the inclusion criteria for the literature review was that the paper in question was about bug bounties specifically, or contained analysis of a bug bounty program, platform, or behaviour of the workers in a program. Literature was excluded where it merely mentioned the existence of bug bounties, or it focused on vulnerability management more generally. In future work, it is intended that this inclusion criteria be widened, because these are all relevant with regards to policy implications, as well as assessing cost-effectiveness for starting a bug bounty.

In conducting this search, a total of 11 papers were discovered which were primarily about bug bounties. This includes a paper which is unpublished[40], and also non-academic work by Bugcrowd[6].

Having established the literature, we make use of established crowdsourcing literature as a means of solving the overall research question. We do not attempt to conduct a survey of crowdsourcing literature, since it is beyond the scope of this research and existing works have conducted surveys of crowdsourcing generally[9], and within software engineering in particular[24]. We contend that according to Estellés-Arolas criteria[12], considering a bug bounty program as crowdsourcing is valid. There is a defined crowd (security researchers) with a clear goal (locate vulnerabilities) and a defined benefit for both the worker and (clearly defined) requester. It is an online process to solve a problem, which uses the Internet, and has some degree of open call. Many calls will have restrictions imposed by the platform based on the reputation of the researcher, particularly in regards to signal to noise ratio yet many will have a completely open call subject only to self-selection by those considering themselves to have enough skill.

Despite this, as a format it is quite different to most crowdsourced programs. The nature of bug bounties is that it accepts all valid unique submissions, provided they are within the scope of the call. Consensus is generally not required, making it differ from many HIT tasks such as image tagging. Competition or innovation crowdsourcing calls will generally accept only one, or a handful of submissions which best solve the problem[8].

[1] See http://blog.trendmicro.com/pwn2own-returns-for-2017-to-celebrate-10-years-of-exploits/

3.2 Literature

Whatever reservations there may be about ethics, efficiency or cost effectiveness, and with initial scepticism from major players[13], bug bounties have been embraced by many of the major technology companies as well as gaining support in other industries[6]. Some research about the participants in the programs has started, as well as means of getting around some of the difficulties with running a bug bounty program.

As indicated by [27], prior to the introduction of a formal mechanism for buying and selling bugs through bug bounties, obtaining a seller was challenging. Two reports by Ring illustrated this in further detail, discussing the competing opinion of whether companies should offer bounties to vulnerability research - and additionally of some companies prosecuting those discovering vulnerabilities[32, 33]. Kuehn & Mueller[16, 17] consider the changing dynamics in information security towards bug bounties being considered a norm. After case studies on Microsoft & Facebook's bug bounty they conclude that bug bounty programs exist as a way of reducing uncertainty when exchanging an information good as a reason for their development.

There are currently two major operators who have had mention made in the literature who facilitate bug bounties: Bugcrowd[2], and Hackerone[3], although other websites offer a list of other Web applications offering a bounty. Bugcrowd now publish an annual report on current trends in the bug bounty area, the most recent being in 2016[6]. Previously, Wooyun offered a forum for researchers to disclose bugs, and had a more coercive model - the Web applications in question were given a certain period of time to fix the flaws, before the flaw was made public. However, the website has been out of action since July 2016 when the founder Fang Xiaodun was reportedly arrested[22]. As of March 2017, the website still displays a message indicating that it is not operational[4].

Two of the older bounty programs, those of Mozilla and Google for their Web browsers Firefox and Chrome respectively were studied in 2013 [26]. Both were found to be better value for the company than hiring a security researcher on a permanent basis when considering the severe security flaws they discovered relative to the cost. They found Google's bug bounty program gleaned more vulnerabilities for a comparable amount of money, which they suggested was due to the tiered reward system they operated compared to Mozilla's flat fee.

In two separate papers, Zhao analysed the behaviour of white hats on the Wooyun[38] and Hackerone[39] platforms. In both platforms they observed the behaviour of white hats in the different systems. In [38], it was observed that the distribution of effort followed a power law, similar to that observed Lotka about academic publication frequency and supporting observations by[26], with a maximum of 291 submissions and an overall average of 4.8. Analysis of both revealed that when divided into categories of productivity each group reported a comparable amount of vulnerabilities, in addition to the severity of the vulnerability and the ranking of the website.

Maillart et al. focus more on the misaligned incentives involved between the companies running a bug bounty program and the researchers themselves[23]. The interest of the company is to exhaust the amount of flaws to a residual level, whereas the interest of the researcher is the cumulative payoff they will gain from discovering bugs. This is best served for the researchers by diversifying their efforts across different programs, since there will be bugs to discover which are easier to locate, and there should be less competition. Analysing 35 programs on Hackerone, they follow [38, 39] and observe a *windfall* effect within a few weeks of the start of the program, after which the amount of reports reduce significantly in quantity. They attribute this to timing effects, where researchers switch to a new program, or possibly stockpile vulnerabilities in advance of the program opening.

[2] https://bugcrowd.com/

[3] https://www.hackerone.com/

[4] http://wooyun.org/ Last accessed 15 March 2017

These papers regard the interest of a company running a bug bounty that they should seek to encourage as many researchers as possible, in the hope of finding flaws. The diversification meaning that, the higher proportion of white hats, the higher the probability that any flaw discovered is by a white hat rather than an attacker. This is supported by is supported by [10], however they also discovered a positive correlation ($r = 0.3591$, $p = 0.0307$) between the amount of accurate reports, and the amount of false positives. This indicates some tension here in regards to the costs of obtaining too many results of low quality.

Laszka et al use economic modeling to analyse various models employed by Hackerone to mitigate this and hopefully ensure a higher quality of submission[18]. They found that policies such as restricting access to those of high reputation, or rate-limiting submissions could be effective, although care needed to be taken in implementing them otherwise the overall utility would go down. Zhao et al expanded on this, although this work does not yet appear to have been published[40].

4 BUG BOUNTY RESEARCH CHALLENGES

From the existing literature on bug bounties and related research themes in crowdsourcing, we identified four key areas which we argue should drive forward the research agenda in relation to bug bounties. This section takes issues identified in the bug bounty literature, and considers it in the context of the wider crowdsourcing literature.

4.1 Incentives and Budget

Bugcrowd have a set of recommendations for new programs given the 'maturity' of their security processes, and the severity of the flaws to be identified. Based on three years of their data, for a new company they recommend paying between $100 and $1,500, with an average of $300[7]. However, this is still a young industry, and Bugcrowd do not publish their methodology for deciding on their prices. The more general research on incentives can play a role here. In particular, to identify the extent to which bug bounty researchers are motivated by purely financial means. Previous research on open source projects identified a multitude of different motivations amongst participants, and more recent research into crowdsourcing microtasks revealed diverse motivations, such as the owner of the call, and the end result of the research.

Whilst existing work has identified possible models for setting a price, or explaining behaviour researchers, these are theoretical models, and make assumptions such as the fact that the researcher is motivated by ensuring the maximum cumulative payoff [18, 23]. There is evidence to suggest that this may be true for a lot of cases. Algarni conducted attempted to contact top discoverers to identify their motivations finding them to be largely financial (although with a small sample)[2]. Additionally, there has been discussion in the security community prior to the popularity of bug bounties about researchers not reporting vulnerabilities for free [32, 33]. However [39] observed that of the 33 public programs without a monetary reward there were 1201 valid reports from the community. Wooyun also offered no reward for disclosure of vulnerabilities[38], suggesting other motivations at play.

That said, a company with the means to offer higher rewards is at a competitive advantage over others, and more likely to have their tasks completed. With bug bounties in particular, the nature of the task and expertise required, means the highest price could be significant. Increasing the amount of money offered to a worker can lead to an increase in participation[11], although this can also lead to an increase in spammers. More generally, there is some danger that the higher incentives do not necessarily improve the quality of submissions as workers can regard their work as being worth more so do not take correspondingly more care[25].

Whatever value a company sets the incentives at, there is an additional overhead in any crowdsourcing program. Simperl observed that having a large crowd can lead to overhead in terms of processing the submissions, as illustrated by the cases of both Google and Netflix[35]. In the case of bug bounties, a company additionally needs to allocate resources to validate and prioritise the vulnerabilities, as well as implement fixes for valid submissions which an attacker might not even find in any event[31]. The question then arises, how best to optimise a budget for a bug bounty program, or if it is even financially worthwhile at all. The results presented in [26] suggest that bug bounties might be cost effective, at least for two major technology companies, yet the additional overheads and potentially high cost of labour suggest that smaller companies might have different experiences.

4.2 Task Decomposition

The selection of tasks is important in many crowdsourcing applications, in order to ensure that the overall goal is achieved with the minimum of overhead. This is frequently done in a way comparable to the MapReduce programming paradigm, with its advantages for parallelism, and a reduction in the requirements for prior skill of the workers. It is possible to do these in a 'microtask' scenario, but with a 'macrotask' it is not possible to break down the tasks into anything smaller[8, 35], requiring a different set of criteria for consideration. Areas such as validation of reports represent an obvious area where a bug bounty could include additional tasks[36], which are more akin to microtask crowdsourcing where a consensus would be desirable. However, there are obvious security risks with this, so great care would be required before any practical implementation.

Task decomposition is an open problem in software engineering crowdsourcing [24], though there has been some effort in the area. LaToza et al. considered the idea of a software development process where it is possible to introduce microtasking [20, 21], and built a microtasking IDE (integrated development environment) based on that. Though they had some success, they noticed an additional overhead compared to traditional software engineering[19]. Adriano also have a draft proof of concept paper on whether bugs can be discovered in small sections of code, and are planning future work on more complex code[1].

4.3 Quality of Submissions

Obtaining the highest possible amount of submissions, whilst minimising noise and associated costs with running a vulnerability program is also an issue. In particular, for smaller companies this represents a problem. They may not have the resources to pay the high costs for bounties or to manage the high volume of invalid responses, and as such may lose out on the benefits of utilising the crowd. This has already received some attention in researching bug bounties, with Laszka et al providing an economic model [18].

In crowdsourcing research, ensuring quality of submissions has received considerable scrutiny in the literature with many different systems proposed. Bernstein et al. introduced the *Find - Fix - Verify* model, where the task is divided into subtasks in which the location, fixing and verifying are split between different workers[4]. Alternatively, the aggregation of different answers can be used to determine the accuracy based on a different criterion such as a simple majority vote. More refined models include basing the decision on the expertise of the worker, and then iteratively adjusting it based on the continued accuracy of submissions, e.g., Whitehill[37].

The work described earlier by Su & Pan about adding falls into this category specifically relating to bug bounties of adding a microtask for verification[36]. This could additionally be considered as part of task decomposition, but it is better to consider it here, since it is an additional task rather than decomposing the vulnerability discovery.

However, the poor quality of submission for any task can potentially be made up for by the increased diversity. Edmunson et al ran an interesting study where they presented participants with a codebase with known vulnerabilities in, and asked to provide a security review[10]. They discovered that, whatever the experience of the individual researcher, none found all the vulnerabilities, but any randomly selected group of 15 participants had a 95% chance of discovering all seven known vulnerabilities, demonstrating the value of diversification of expertise.

4.4 Relevance to Web Science

As a more general research challenge, we call upon the Web Science community to invest in research in this area since Web Science is particularly suited to it. As an exploratory study, we only scratched the surface of bug bounties as a research area in relation to crowdsourcing, but we additionally consider wider challenges identified in this area. Halford et al. argue that Web Science has never been just about researching wires and protocols, but additionally considers the Web as being a constantly adapting organism which is a product of the people who use it[15]. Web security is a significant cost to society from both preventative and reactive measures[3], and as such is a key element of researching on the Web.

As vulnerability management, disclosure, and mitigation have matured, the effect on the practices of people developing software for the Web, as well as general Web users is something which is worth researching[16]. Much of the work on vulnerability markets has been theoretical, now there is the opportunity to analyse the effect it is having. Similarly, in addition to work carried out by [29, 31] whether these vulnerability markets have caused a depletion of vulnerabilities in Web applications. Beyond the effects of markets, the distortions and ethics of selling to governments is also an important area to consider, in particular where that relates to national security.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we presented an exploratory study into bug bounties as a concept and argued it should be of interest to those working in Web Science in general, and crowdsourcing in particular. It represents an area which has seen comparatively little research, and yet would benefit significantly from a Web science approach. We conducted a literature review, and identified a small corpus of papers, using them to identify promising research areas for the future mapped with crowdsourcing literature. As an extension to this work, we intend to expand the scope of our review into bug bounties, and how they relate to vulnerability markets; ethics; and regulation. In addition, we intend to concentrate more fully on the different dynamics between microtask based crowdsourcing and bug bounty research.

REFERENCES

[1] Christian Medeiros Adriano and Andre van der Hoek. 2016. Exploring Microtask Crowdsourcing as a Means of Fault Localization. *arXiv preprint arXiv:1612.03015* (2016). https://arxiv.org/abs/1612.03015

[2] Abdullah M. Algarni and Yashwant K. Malaiya. 2013. Most Successful Vulnerability Discoverers: Motivation and Methods. In *Proceedings of the International Conference on Security and Management (SAM)*. 1.

[3] Ross Anderson, Chris Barton, Rainer Böhme, Richard Clayton, Michel JG Van Eeten, Michael Levi, Tyler Moore, and Stefan Savage. 2013. Measuring the cost of cybercrime. In *The economics of information security and privacy*. Springer, 265–300.

[4] Michael S. Bernstein, Greg Little, Robert C. Miller, BjÃűrn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. 2015. Soylent: A Word Processor with a Crowd Inside. *Commun. ACM* 58, 8 (July 2015), 85–94.

[5] Rainer Böhme. 2006. A comparison of market approaches to software vulnerability disclosure. In *Emerging trends in information and communication security*. Springer, 298–311. http://link.springer.com/chapter/10.1007/11766155_21

[6] Bugcrowd. 2016. The State of Bug Bounty. (June 2016).

[7] Bugcrowd. 2017. Defensive Vulnerability Pricing Model. (2017). https://pages.bugcrowd.com/whats-a-bug-worth

[8] Thierry Burger-Helmchen and Julien Pénin. 2010. The limits of crowdsourcing inventive activities: What do transaction cost theory and the evolutionary theories of the firm teach us. In *Workshop on Open Source Innovation, Strasbourg, France.* 1–26.

[9] A. I. Chittilappilly, L. Chen, and S. Amer-Yahia. 2016. A Survey of General-Purpose Crowdsourcing Techniques. *IEEE Transactions on Knowledge and Data Engineering* 28, 9 (Sept. 2016), 2246–2266.

[10] Anne Edmundson, Brian Holtkamp, Emanuel Rivera, Matthew Finifter, Adrian Mettler, and David Wagner. 2013. An Empirical Study on the Effectiveness of Security Code Review. In *Engineering Secure Software and Systems.* Springer, Berlin, Heidelberg, 197–212.

[11] Carsten Eickhoff and Arjen de Vries. 2011. How crowdsourcable is your task. In *Proceedings of the workshop on crowdsourcing for search and data mining (CSDM).* 11–14.

[12] Enrique Estellés-Arolas and Fernando González-Ladrón-De-Guevara. 2012. Towards an integrated crowdsourcing definition. *Journal of Information science* 38, 2 (2012), 189–200.

[13] Dennis Fisher. 2010. Microsoft Says No to Paying Bug Bounties. (July 2010). https://threatpost.com/microsoft-says-no-paying-bug-bounties-072210/74249/

[14] Chris Grier, Lucas Ballard, Juan Caballero, Neha Chachra, Christian J Dietrich, Kirill Levchenko, Panayiotis Mavrommatis, Damon McCoy, Antonio Nappa, Andreas Pitsillidis, and others. 2012. Manufacturing compromise: the emergence of exploit-as-a-service. In *Proceedings of the 2012 ACM conference on Computer and communications security.* ACM, 821–832.

[15] Susan Halford, Catherine Pope, and Leslie Carr. 2010. A manifesto for Web Science. *Journal of Web Science* (2010).

[16] Andreas Kuehn and Milton Mueller. 2014. Analyzing bug bounty programs: An institutional perspective on the economics of software vulnerabilities. In *TPRC Research Conference on Communication, Information and Internet Policy.*

[17] Andreas Kuehn and Milton Mueller. 2014. Shifts in the Cybersecurity Paradigm: Zero-Day Exploits, Discourse, and Emerging Institutions. In *Proceedings of the 2014 New Security Paradigms Workshop (NSPW '14).* ACM, New York, NY, USA, 63–68. DOI:http://dx.doi.org/10.1145/2683467.2683473

[18] Aron Laszka, Mingyi Zhao, and Jens Grossklags. 2016. Banishing misaligned incentives for validating reports in bug-bounty platforms. In *European Symposium on Research in Computer Security.* Springer, 161–178. http://link.springer.com/chapter/10.1007/978-3-319-45741-3_9

[19] Thomas D. LaToza, W. Ben Towne, Christian M. Adriano, and AndrÃĺ van der Hoek. 2014. Microtask Programming: Building Software with a Crowd. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14).* ACM, New York, NY, USA, 43–54. DOI:http://dx.doi.org/10.1145/2642918.2647349

[20] Thomas D LaToza, W Ben Towne, André Van Der Hoek, and James D Herbsleb. 2013. Crowd development. In *Cooperative and Human Aspects of Software Engineering (CHASE), 2013 6th International Workshop on.* IEEE, 85–88.

[21] Thomas D LaToza and André Van Der Hoek. 2015. A vision of crowd development. In *Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE International Conference on*, Vol. 2. IEEE, 563–566.

[22] Gene Lin. 2016. Founder of China's largest 'ethical hacking' community arrested. (July 2016). https://www.hongkongfp.com/2016/07/30/founder-chinas-largest-ethical-hacking-community-arrested/

[23] T Maillart, M Zhao, J Grossklags, and J Chuang. 2016. Given enough eyeballs, all bugs are shallow? Revisiting Eric Raymond with bug bounty markets. (2016).

[24] Ke Mao, Licia Capra, Mark Harman, and Yue Jia. 2016. A survey of the use of crowdsourcing in software engineering. *Journal of Systems and Software* (Sept. 2016).

[25] Winter Mason and Duncan J Watts. 2010. Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter* 11, 2 (2010), 100–108.

[26] Matthew Finifter, Devdatta Akhawe, and David Wagner. 2013. An Empirical Study of Vulnerability Rewards Programs. In *Proceedings of the 22nd USENIX Security Symposium.* USENIX Association, Washington DC, 273–288.

[27] Charlie Miller. 2007. The legitimate vulnerability market: Inside the secretive world of 0-day exploit sales. In *In Sixth Workshop on the Economics of Information Security.*

[28] Andy Ozment. 2004. Bug auctions: Vulnerability markets reconsidered. In *Third Workshop on the Economics of Information Security.* 19–26.

[29] Andy Ozment. 2005. The Likelihood of Vulnerability Rediscovery and the Social Utility of Vulnerability Hunting.. In *WEIS.* Citeseer. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.479.7888&rep=rep1&type=pdf

[30] Sam Ransbotham, Sabyasachi Mitra, and Jon Ramsey. 2008. Are markets for vulnerabilities effective? *ICIS 2008 Proceedings* (2008), 24.

[31] E. Rescorla. 2005. Is finding security holes a good idea? *IEEE Security Privacy* 3, 1 (Jan. 2005), 14–19. DOI:http://dx.doi.org/10.1109/MSP.2005.17

[32] Tim Ring. 2014. Why bug hunters are coming in from the wild. *Computer Fraud & Security* 2014, 2 (Feb. 2014), 16–20.

[33] Tim Ring. 2015. White hats versus vendors: the fight goes on. *Computer Fraud & Security* 2015, 10 (Oct. 2015), 12–17.

[34] Ari Schwartz and Rob Knake. 2016. *Government's Role in Vulnerability Disclosure: Creating a Permanent and Accountable Vulnerability Equities Process.* Technical Report. Discussion Paper 2016-04, Cyber Security Project, Belfer Center for Science and International Affairs, Harvard Kennedy School.

[35] Elena Simperl. 2015. How to Use Crowdsourcing Effectively: Guidelines and Examples. *LIBER Quarterly* 25, 1 (Aug. 2015).

[36] H. J. Su and J. Y. Pan. 2016. Crowdsourcing platform for collaboration management in vulnerability verification. In *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS).* 1–4.

[37] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R. Movellan, and Paul L. Ruvolo. 2009. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta (Eds.). Curran Associates, Inc., 2035–2043.

[38] Mingyi Zhao, Jens Grossklags, and Kai Chen. 2014. An Exploratory Study of White Hat Behaviors in a Web Vulnerability Disclosure Program. In *Proceedings of the 2014 ACM Workshop on Security Information Workers (SIW '14)*. ACM, New York, NY, USA, 51–58.

[39] Mingyi Zhao, Jens Grossklags, and Peng Liu. 2015. An Empirical Study of Web Vulnerability Discovery Ecosystems. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*. ACM, New York, NY, USA, 1105–1117. DOI:http://dx.doi.org/10.1145/2810103.2813704

[40] Mingyi Zhao, Aron Laszka, Thomas Maillart, and Jens Grossklags. 2016. Crowdsourced Security Vulnerability Discovery: Modeling and Organizing Bug-Bounty Programs. (2016). http://aronlaszka.com/papers/zhao2016crowdsourced.pdf