Crowdsourcing Security

Vincent Tran

Tufts University

Abstract

Crowdsourcing promises a revolutionary solution to cybersecurity issues in the form of bug bounty programs. A company who utilizes a bug bounty program rewards users for successfully identifying security vulnerabilities in their products, be they information leaks or potential avenues of attacks. Apple, Microsoft, Google, and Facebook are few of the many top tech companies that crowdsource their security, illustrating the effectiveness of such bug bounty programs. By enlisting the help of the masses, companies can foolproof their services quickly, extensively, and, arguably more importantly, cheaply. This article will explore why crowdsourcing cybersecurity is effective, how to successfully (or poorly) implement a bug bounty program, and potential drawbacks.

Introduction

"Given enough eyeballs, all bugs are shallow"

-- Linus Torvalds

With the advent of the mobile computing age, the world has seen a dramatic increase in crowdsourcing applications. Uber, Yelp, Waze, and countless other popular apps have created profound societal changes in the past decade, allowing consumers to interact with their peers and the company in efficient symbiosis.

The rise of social media and personal devices also gives way to increased accessibility. This accessibility allows the ordinary person to contribute to larger projects and movements. The world has seen innovative applications of crowdsourcing, and in most cases, the success of these crowdsourcing campaigns demonstrates the lengths that people will go to help. Crowdfunding on Kickstarter gave rise to popular smartwatch company, Pebble. Ordinary citizens help the police daily through submitting info for manhunts and child abduction emergencies. Volunteers all over the world reviewed satellite images in the search for the missing Malaysia Airlines Flight 370.

While these successful crowdsourcing campaigns argue the goodness of humanity, the state of cybersecurity argues otherwise. Security is an ongoing, seemingly unsolvable issue. From defending against malicious attackers to educating engineers on secure coding practices to protecting the increasingly personal client data, tech companies nowadays face a daunting task in securing their products.

Traditionally, tech companies employ in-house or contracted penetration testers to ensure the security of their products. The happy medium between engineering and security is elusive; given scarcity of resources and the rampant competition in the tech industry, security oftentimes takes a backseat. As such, the brunt of security falls on the testers. Even with a team of security experts, however, their abilities are limited to their experiences, the size of their team, and the

resources allotted to them. Unsurprisingly, vulnerabilities are left uncaught, eventually found and abused by malicious hackers.

Crowdsourcing prevails where ad-hoc security measures fail; while penetration testers are limited by their numbers and experience, volunteer security researchers are plentiful, motivated, and may possibly be *more* insightful than employed testers. The application of crowdsourcing to security takes the form of bug bounty programs, where companies reward researchers for finding vulnerabilities in their product. As more and more companies adopt bug bounty programs, crowdsourcing security proves to be the most efficient yet economical manner of tackling the security dilemma.

To the Community

Computer and information security are hot topics in the modern world. With the pace of innovation in the technology realm, keeping security up to par is not an easy task. Although security is fundamental to most, if not all, pieces of technology, developers and companies need to employ new and improved tactics to ensure the privacy and safety of their users.

Crowdsourcing grows along with society's increasing interconnectivity. Crowdsourcing security is a promising new trend amongst the top tech companies, and it's likely to revolutionize the application security landscape. As consumers, potential bug bounty participants, and tech innovators, we need to follow the maturation of security, particularly in regards to fresh methods of ensuring security. Understanding the effectiveness of crowdsourcing security gives us valuable insight into the state of security. Can we be confident that the apps we use and make are secure? Through crowdsourcing, more eyes, more reports, and more transparency is available, helping to answer this question.

The History of Crowdsourced Security

Feedback. Developers want it, users have it. Receiving feedback from the consumer base is crucial to the success of a product. Still, incentivizing users to submit their feedback is similar to pulling teeth. For instance, the Facebook app, at one billion to five billion downloads, is one of the most installed apps in the Google Play Store. Even so, only 0.05% of users submitted a rating on the Play Store, showing the users' reluctance to provide feedback.

As incentive to provide feedback, the Netscape browser team hosted the first bug bounty program in 1995, where they gave monetary rewards to users who found and reported a bug they encountered [3]. While this was a successful campaign, with other companies occasionally hosting a similar program, bug bounties did not grow in popularity until 2010.

2010 saw Google's adoption of a massive-scale bug bounty program with prizes as much as $20,000 [2]. This set the stage for crowdsourcing security, as a reputable tech giant endorsed a then unproven method of enhancing security. At this point, unlike in 1995, mobile and personal computing, as well as the internet, had become commonplace, thus the medium and platform for volunteer researchers to participate in bounty programs was more accessible. In the years following, a flurry of tech companies started their own bug bounties, including Facebook and Microsoft, further increasing the acclaim of bug bounty programs. With the tech industry continuously and furiously growing, the quantity and quality of vulnerability reports grows as well, resulting in the widespread adoption of bug bounty programs as a reliable means to security.

In 2016, bug bounties are so commonplace that third-party organizations that optimize the bug bounty process have arisen. These organizations, such as HackerOne and BugCrowd, consolidate bug bounty listings and facilitate the communication between researcher and company, which is riddled with protocol a la responsible disclosure. The success of these

companies in the niche of organizing bug bounties demonstrates the overall demand for crowdsourced security, both by companies researchers.

The Effectiveness of Bug Bounties

Many tech giants and even the US Department of Defense have found success with crowdsourced security, but the question remains: why are bug bounties so effective? There are many aspects to the answer.

Firstly, effectiveness is relative. Theoretically, security should not be an issue that tech companies and users alike face. In practice, security issues are unchecked and arise from the traditional system of ad-hoc security testing. Bug bounties, though imperfect, show results where the traditional imperfect system does not. They are simply an extension of security testing, allowing more security experts to contribute on their own accord, resulting in a securer product. Given the understandable limitations of in-house security testing, it is no wonder that enlisting an unlimited amount of volunteers, offering rewards, and leaving bounties open indefinitely yields better results than a team of underappreciated security engineers who are on a deadline and have budget restrictions.

With policies such as the Computer Fraud and Abuse Act, many aspiring security experts have no outlet for their curiosity and learning. Capture-the-flag competitions and purposefully vulnerable virtual machines allow testers to practice their ethical hacking, but such exercises do not adequately capture the sentiment and challenge of real-world hacking. Bug bounty programs, however, provide a legal, challenging outlet for hacking desires, attracting the interest of both talented black-hat hackers and reputable white-hat experts.

Additionally, bug bounty programs offer incentives, usually in the form of cash rewards, for reporting legitimate vulnerabilities. Salaried security engineers earn a stable income, but cash

prizes as much as $20,000 motivates security researchers to go the extra mile to find the obscure or otherwise overlooked vulnerabilities. Other talented hackers who may not work for a company are also more to lend a hand and conduct their own security testing if there is a reward for their work. Rewards, even including mere recognition and praise, unlock the power of the commons, effectively expanding the size of a company's security efforts.

From the company perspective, bug bounties are just as favorable. By relying on crowdsourced security information, companies can count on a diverse set of expertise and backgrounds from their numerous volunteer testers. And, instead of the usual repeating pattern of development phase into security testing phase, the stream of testing is ongoing and incessant. On top of eliciting more researchers to check the software, bug bounty programs are a very economical strategy to improving security. Companies need only pay for valid vulnerability reports, thus saving significant money that would be otherwise invested in salaried security experts [7]. Unfortunately, the appeal of crowdsourcing leads some companies to adopt bug bounties without fully understanding the drawbacks, which will be discussed in a section below.

According to a report by BugCrowd, the cost of one professional penetration test, roughly $20,000 can be used as a bounty to incentivize 349 researchers to participate in a bug bounty program. It goes on to report that bug bounty researchers can find 38 vulnerabilities in eight hours, while a single penetration testing can find merely five vulnerabilities in eighty hours [1]. While the report is nondescript and published by BugCrowd, results like these are found across the board, evidencing the effectiveness of crowdsourcing.

Drawbacks to Crowdsourcing

Although economical and extensive, crowdsourcing is not a sufficient means of attaining security alone. As mentioned earlier, many companies with strained budgets opt to crowdsource security, but they often encounter many drawbacks. It is important for companies to understand

the disadvantages and potential repercussions bug bounty programs, particularly poorly organized programs.

The concept of bug bounty programs, and security testing in general, is flawed in that they are reactive measures. In theory, the optimal solution to security flaws in regards to time, effort, and money is to develop with security in mind and establish a sound threat model--to take *proactive* measures [4]. If emphasis is placed on secure code and design principles rather than ad-hoc security testing, many security issues could be avoided altogether. In practice, accounting for every possible vulnerability while developing software is not feasible, but more efforts can certainly be made to augment security education amongst developers. Bug bounties are effective catch-alls, allowing overlooked vulnerabilities to be reported and giving potential hackers a moral, mutually-beneficial option, dissuading malicious attacks, but they are not a reason to regard security as an afterthought. The best treatment is prevention.

Some security experts also disapprove of the nature of bounty programs, claiming that bounties negatively reinforce behavior akin to blackmail. Tech journalist Kenneth van Wyk form ComputerWorld argues, “bug finders are in essence holding a metaphorical gun to the heads of software companies…” Van Wyk and other professionals often report security issues that they encounter for the benefit of the company and users, but without a proper communication protocol, companies can ignore or forget about reported vulnerabilities [10]. Although intentions behind disclosure may be good, white-hats and black-hats alike can grow impatient with non-communicative companies, leading to threats to disclose the vulnerability publicly. In a way, bug bounties support this behavior, but appease researchers with rewards so that exploits remain secret. Opposers of crowdsourced security assert that the inherent blackmail-like behavior remains even in bug bounties, evidenced by the behaviors of impatient bug hunters when interacting with slow, unresponsive companies. With established third-party bug bounty organizers and increased responsiveness from software companies, this issue is diminishing, but the burden remains on companies to hold a strong bounty program with comprehensive policies.

In dealing with bug bounty hunters, there is no assurance of professionalism, which is at least guaranteed when vetting and contracting a certified security team. When encouraging researchers to hack a product, as bug bounties do, companies expect the researchers to comply with responsible disclosure and other privacy protocols. Unfortunately, even if researchers agree to sign a non-disclosure agreement, the risk of intelligence leakage and improper disclosure remains.

While bug bounties seem to expand on the efforts of an internal security team, bug bounties cannot test internal systems as internal teams should. Bug bounties are public-facing by nature, thus participants can only conduct research on public-facing domains/products. Other internal protocols should be off-limits to bounty hunters as they may expose critical user or software information. As per the nature of bug bounties, researchers may hail from all over the world, including foreign countries. Releasing authentication tokens and giving researchers the appropriate privileges to sufficiently test internal systems or applications is risky, and giving internal privilege to a non-vetted, unidentified researcher should be strictly avoided. Even under a formal agreement, prosecuting a likely aliased hacker for violating an agreement is intangible [5].

In popular bug bounties where testing may not be restricted to web vulnerabilities, security teams may also find issue with the influx of network scans and other traffic. Large tech companies likely are not concerned with the increased traffic and potential for DoS by inexperienced researchers, but they will find their ability to spot truly malicious activity is heavily hindered. Blackhats may hide their activity amongst the activity of bounty hunters, which had been legitimized by the bug bounty program. Those with malicious intent can comfortably scan ports and probe areas that would otherwise set off alarms or leave a suspicious mark in the logs [5]. With this in mind, tech companies holding bug bounty programs must be mature and diligent in their policies and vigilance.

Running a Proper Bug Bounty Program (Applications)

After considering the aforementioned drawbacks, many major tech companies still find the potency of crowdsourced security to be enticing and worthwhile, and justifiably so. Negative experiences from bug bounty programs arise from poor policy and lack of preparation on the company's part. In order to maximize the effectiveness of a bug bounty program, a company needs to take several factors into account.

The essence of bug bounty programs is the bounty--the reward. Since one of the purposes of bug bounties is to deter cybercrime and encourage moral, responsible disclosure, companies need to carefully determine what prizes will appropriately award hackers but not overcompensate. Some bug bounties, particularly those held by smaller tech companies, reward participants with only hall-of-fame recognition or swag prizes, meaning they solely appeal to the altruism of researchers. Larger tech companies can afford to pay thousands of dollars per validated report. Companies that provide cash prizes weight certain vulnerabilities differently, rewarding finds that are more obscure or pose a severe risk with more money. For example, Google pays out $20,000 for exploits that may compromise user accounts, but as little as $100 for other exploits that they may deem as "lower priority" [2].

The first aspect of ensuring a comprehensible policy is to determine the scope of program. What is on or off limits should be made clear, as companies certainly do not want researchers to feel justified in attacking areas that they should not. Many companies, such as MIT and Google, provide explicit domains in which testing can be done. Sites outside of that domain, even within the same company, are strictly off-limits and will not be rewarded. Once established, researchers will not be able to blackmail, as mentioned earlier, for payment with out-of-scope reports.

As with life, in bug bounty programs, communication is key. Companies preparing to host a bug bounty program often misallocate their time and resources, believing that fixing the many bugs reported should be their priority. In actuality, many more resources should be spent on communication with researchers, including parsing through the many submissions, determining appropriate prizes, and updating the researchers in a timely fashion . When researchers submit a vulnerability report, they are either concerned for the security of the application and/or eagerly await their prizes. If a company is unresponsive after a researcher submits a report or refuses to pay due to policy misinterpretations, the past shows that researchers will find a way to get the company's attention, usually in a brash and aggressive way.

GitHub user XiphosResearch is a prime example of what ensues when companies are uncooperative. XiphosResearch published to GitHub an exploit that he found on Zoho.com, claiming that the company refuses to patch their code, even after six months. They also refused to acknowledge him in their hall-of-fame, claiming that his exploit is not cross-site scripting based, although it can be used to trigger cross-site scripting. He went on to post on Reddit, claiming the bug bounty program by Zoho is merely a "market exercise" and that the company is plagued with "corporate dysfunction". Both the public disclosure and defamation of companies is not unheard of in situations like this, which is all the more reason for companies to maintain open channels of communication with their researchers and have clearly-worded and specific policies [10].

Conclusion

After reaching a stable state of security in an application, the logical next step for a company is to open a bug bounty program, making use of the power of the commons. Provided that a company is comfortable with the potential drawbacks and have strong policies regarding bounties, crowdsourcing is a powerful tool to perfect the application's security. Large tech companies, startups, and government agencies alike find success with crowdsourcing, showing

that the trend of bug bounties will continue to grow and may even become a staple in the process of attaining application security.

As we enter the age of the Internet of Things, security threats grow and evolve in unprecedented ways. Old vulnerabilities remain yet still, new vectors of attack emerge. Attackers can now abuse the proliferation of smart devices to perform DDoS attacks as large as one terabyte per second, or hijack a Tesla car while it is in motion, driving on the street. Tesla recently announced their own bug bounty program, allowing researchers to report such vulnerabilities in their cars, which is a critical step in the right direction to solve such pressing issues. Security expert, Rajesh Krishnan of HackerOne, also fully expects bug bounties to "...scale to the Internet of Things." He goes on to predict the future of crowdsourcing, where exploits become harder to find and prerequisite knowledge of hardware becomes necessary. He argues that companies may soon need to offer bounties for *attempting* to find a vulnerability [6]. How Internet of Things companies proceed after this prediction is to be seen, but crowdsourcing is a growing phenomenon that can only benefit them, so we can expect them to come out with their own crowdsourcing solution soon.

Crowdsourced security marks a shift in perception of security and hackers. While security before was plagued with criminal and illegal connotations, bug bounty programs are opening the eyes of both people and companies. The widespread acceptance of bounty programs shows a newfound reverence for knowledgeable security experts, and also supports overall security education in users and developers. Companies like Microsoft and Oracle, who sought to persecute hackers, even when they disclosed vulnerabilities responsibly, are now embracing the crowdsourcing model, with Microsoft implementing a bounty program of their own [9].

Crowdsourcing relies on the efforts of the common people. Anyone who is interested in studying application or information security is highly encouraged to participate in bug bounties. Ultimately, with more knowledgeable and determined researchers submitting reports to bug bounty programs, attacks will grow sparser. These efforts lead to increased confidence amongst

users in the security of applications and also provides an inexpensive, persistent way to improve security for the company. Given the mutually beneficial nature of crowdsourcing, I expect all tech companies to eventually adopt bug bounty programs, or at the very least remain open to accepting charitable help from volunteer hackers.

References

1. *5 Tips for a Successful Bug Bounty Program*. Perf. Casey Ellis and Jonathan Cran. *Bugcrowd*. Bugcrowd, 30 June 2015. Web. 14 Dec. 2016.
2. *Google Application Security*. N.p., n.d. Web. 14 Dec. 2016.
3. "History of Bug Bounties." *Bugcrowd.com*. BugCrowd, n.d. Web. 14 Dec. 2016.
4. Keane, Jonathan. "Who's Keeping Your Data Safe? With Bug Bounties, It's Would-be Hackers." *Digital Trends*. N.p., 10 Apr. 2016. Web. 14 Dec. 2016.
5. Kolochenko, Ilia. "Can Bug Bounties Replace Traditional Web Security?" *SC Magazine UK*. SC Magazine, 16 Jan. 2016. Web. 14 Dec. 2016.
6. Krishnan, Rajesh. "Will Bug Bounties Scale To The Internet of Things?" *The Security of Things*. N.p., 09 July 2016. Web. 14 Dec. 2016.
7. Rubens, Paul. "How Bug Bounty Programs Bring Big Savings and Better Security." *CIO*. CIO, 23 July 2013. Web. 14 Dec. 2016.
8. Schuman, Evan. "When Bug Bounties Are Counter-Productive." *Veracode*. Veracode, 26 Aug. 2016. Web. 14 Dec. 2016.
9. Weinberger, Matt. "Oracle's Security Chief Made a Big Gaffe in a Now-deleted Blog Post." *Business Insider*. Business Insider, 11 Aug. 2015. Web. 14 Dec. 2016.
10. Wyk, Kenneth Van. "Bug Bounties: Bad Dog! Have a Treat!" *Computerworld*. Computerworld, 23 July 2013. Web. 14 Dec. 2016.
11. XiphosResearch. "XiphosResearch/exploits." *GitHub*. N.p., 28 Oct. 2016. Web. 14 Dec. 2016.