1

The Evolution of Bug Bounty Programs

1.1 Making History

Understanding the evolution of bug bounty programs first requires familiarity with the hacking landscape, or as many in the information security field know it, penetration testing. Security researchers haven't always been respected or given the opportunity to shine. Throughout history, hacking has been a word that scares the public and creates waves of fear inside a company when rumors of a "hack" spread. The first bounty paid for breaking into something (in recorded history) was in 1851. Charles Alfred Hobbs was paid roughly the equivalent of $20,000 to pick a physical lock. (https://www.itspmagazine.com/itsp-chronicles/history-and-interesting-facts-about-bug-bounties-an-appsec-usa-2017-panel-recap).

The first actual bounty program was run by Netscape and it began in 1995. The primary scope was application testing for Netscape Navigator 2.0., their primary product. Slowly, other enterprises started to adapt their own bug bounty programs and offer awards. Bug bounty crowdsourcing platforms introduced the new wave, compiling enterprise programs into a neat catalogue in which security researchers could hop into various programs and begin to participate. Bugcrowd was known as the first crowdsourcing platform in bug bounty history and has been a key player in enterprise bug bounty program management. The pioneers – Casey Ellis, Chris Raethke, and Sergei Belokamen – believed in connecting latent potential to unmet demand with the overall goal of making security easier for everyone. In addition, Ellis firmly believed in assisting security researchers in keeping their records clean. Casey Ellis has also expressed a desire to help educate the youth toward the idea of ethical hacking, rather than a life of crime, and part of the inspiration for creating such a company has to do with the ideal of destigmatizing security research.

In all actuality, reviewing the state and history of bug bounty programs gives the reader a valuable positive perspective, but enterprises are slow to adapt. Even since 1995, there are still fewer than 400 bug bounty programs and 1600 vulnerability disclosure programs that exist in the world. The surprisingly small number of programs that exist in the world represent the resistance and conservatism of the field of legal hacking, otherwise known as security research.

Corporate Cybersecurity: Identifying Risks and the Bug Bounty Program, First Edition. John Jackson.

1.2 Conservative Blockers

When information security specialists learn about bug bounty programs, many of them are excited to get involved. Application security is a growing field, and modern day web, mobile, and hardware assets need to be protected. With such an essential requirement to protect applications, enterprises still resist the absolute necessity of making vulnerability reporting management a prioritized incentive. As with everything, there's not a "one-size-fits-all" answer for why an enterprise would ignore application security; however, many factors play a role in the resistance that is widespread, even today. For example, here are some of the reasons a company may decide to ignore the idea of a bug bounty program:

- Increased threat actor activity.
- Security researchers scamming.
- Applications being a small consideration.
- Enormous budgetary requirements.
- Other security tooling as a priority.

There are obviously several other reasons an enterprise may believe a bug bounty program will cause unnecessary risk or negative effects. Debunking the above five defined points will give people a better understanding of why being afraid is natural, but it can be detrimental to the overall health of a good application security program.

1.3 Increased Threat Actor Activity

An enterprise may be fearful that establishing a bug bounty program will cause an increase of malicious threat actors attempting to hack into or successfully exploiting applications. The logic can be portrayed as such, "If an enterprise bug bounty program is established, then security researchers will be allowed to hack, and it will be impossible to tell who is malicious." The problem with this statement's assumption that threat actors are hiding among security researchers is one of a common philosophical logical fallacy: the Slippery Slope.

The Slippery Slope logical fallacy is best defined as, "A course of action that seems to lead inevitably from one action or result in another with unintended consequences." In layman's terms, the translation of the Slippery Slope in the security research scenario is, "If the enterprise allows security researchers to conduct research, we will be maliciously exploited." It's best to imagine the scenario of increased threat actor activity with the other perspective in mind. Without a bug bounty program, flaws may never be identified – vulnerabilities that could compromise an organization's sensitive information or intellectual property.

Enterprises considering operating bug bounty programs should learn effective logging and prevention through logging mechanisms and web application firewalls, which are discussed later in this book.

1.4 Security Researcher Scams

Any type of business that relies on services rendered by another party should always be weary of scamming. Understanding the vulnerability types, criticality, and assessing payment amounts will always be the best course of action for a company running a bug bounty program. Still, the idea of scamming isn't a new one. Potential program managers have to learn best practices and understand the basics of vulnerability management. Nonetheless, protections for programs are in place. Managed services offered through bug bounty crowdsourcing platforms such as Bugcrowd and HackerOne will become useful tools. The triage team will assist in validating the legitimacy of a vulnerability which can assist in preventing scamming. Program managers shouldn't solely rely on the validation, but scamming happens far more infrequently than enterprises that are on the fence imagine.

1.5 Applications Are a Small Consideration

Enterprises that avoid bug bounty programs because of the idea of applications being a small attack surface are asking for trouble. When employees tasked with the security of a company evaluate vulnerability potential, the obvious go-to is to secure the network and related assets. However, web and mobile applications in particular have become exceedingly complex. With multiple development languages and servers, the attack surface is far greater than one might imagine. Consider the following example:

Server → Hosts one part of the web application → One assigned IP address
Web application → Connected to multiple servers → Multiple IP addresses

The deployment of an enterprise's assets will always be the determinant factor in the attack service; however, modern applications are becoming more interconnected than they ever were in the past. It's easy to think about a "server" as an asset with a wide attack surface, and in many cases, that is true, and the attack vectors will always vary. Regardless, enterprises should not consider the value of a bug bounty program as something minute and ineffective. In addition, flawed application logic may result in the exploitation of the network and enterprises may not consider that. For example, SQL (Structured Query Language) injection can result in a full server-database dump or remote code execution on the network. Server side request forgery can result in the exposure of sensitive information leading to unauthorized server access or pivoting to other parts of the network. Application security is a large undertaking and neglecting it can result in the full compromise of an enterprise.

1.6 Enormous Budgetary Requirements

Bug bounty programs scale. The size and operation of the bug bounty program is up to the enterprise to decide. In addition, if the company isn't giant, it's unrealistic to assume that the enterprise would have to pay a large sum of money to get a program up and running.

With bug bounty crowdsourcing becoming the norm, companies like Bugcrowd and HackerOne are willing to have scoping calls with leadership to identify a fair pricing model for program management. The price of program management is well worth the cost of identifying vulnerabilities that can result in the loss of hundreds of thousands, if not millions, of dollars in assets or compliance violations such as GDPR (General Data Protection Regulation) or the California Privacy Act. Application security, like any other subbranch of security, is an investment – and security doesn't typically see hefty returns on investment. Information security doesn't make a company money: it protects the company from losing money, allowing the acquisition of money.

1.7 Other Security Tooling as a Priority

Out of all of the other potential worries for setting up a program, security tooling is a legitimate concern. Balancing a budget requires coordination with all levels of leadership and an overall evaluation of security posture. For example, establishing a bug bounty program isn't likely a good idea if the enterprise does not have a web application firewall, or a decent endpoint protection and response solution. Coordination with the security team will have to occur, but if all other bases are covered, there's no reason a basic bug bounty program cannot be established.

1.8 Vulnerability Disclosure Programs vs. Bug Bounty Programs

Even for the most technical of individuals, understanding the difference between a vulnerability disclosure program (VDP) and a bug bounty program (BBP) can be mind boggling. Even still, engineers who run bug bounty programs may make the mistake over calling a bug bounty program a vulnerability disclosure program, or vice versa. Understanding the difference between the two is essential to communicating expectations clearly and educating the general public on the day-to-day processes involved.

1.8.1 Vulnerability Disclosure Programs

Vulnerability disclosure programs are the method used when an enterprise wants to facilitate the disclosure of vulnerabilities but not offer any sort of paid incentive. Vulnerability disclosure programs can be considered a goodwill type of vulnerability management process. The two types of vulnerability disclosure programs are managed and unmanaged. An unmanaged program would be a vulnerability disclosure program that is offered in-house, with an associated good faith based effort. In contrast, a managed vulnerability disclosure program could be one where program managers are assisted by a triage team from a bug bounty crowdsourcing platform such as Bugcrowd or HackerOne. As an incentive to researchers, they are offered points in return for reports, which is an essential part of leveling-up and getting invited to private programs, which typically have less competition for security researchers and a better chance of vulnerability finding.

Private vulnerability disclosure programs are also allowed through crowdsourcing platforms, reducing the costs associated with paying bounties as points will be rewarded.

1.8.2 Bug Bounty Programs

Bug bounty programs are typically more mature vulnerability disclosure programs, offering rewards in place of points. When program managers want to convert their vulnerability disclosure programs to bug bounty programs, the process is typically as simple as initiating a financial incentive for security research. Bug bounty programs carry more weight and attract more professional hackers. For example, some of the best security researchers may never participate in vulnerability disclosure programs because the time they spend evaluating bug bounty programs could easily be time converted to a cash flow. An enterprise's end state should always be aspiring to reach paid-program participation. Security research consumes a lot of time and an enterprise should want to pay its researchers for the time spent. If confused, think of it like this: how many people are willing to do a full-time job for free versus paid? Hobbyists will always exist, but the participation of some of the greatest security researchers can only be obtained with monetary incentives.

1.9 Program Managers

Throughout the book, the phrase "program manager" will come up frequently. A program manager isn't to be thought of as a traditional manager who coordinates employee activity. Rather, program managers are any employee who deals with the configuration or management of an enterprise bug bounty program. For example, the title of the employee doesn't matter: an application security engineer or a chief information security officer could be a program manager. The only consideration is that the employee must have oversight of the program and the ability to make changes. After all, even an employee who is remediating bugs is managing the day-to-day workflow of the program.

1.10 The Law

Historically, the law hasn't always been kind to security researchers. Even today, hacking is still considered dangerous or controversial to nontechnical people. A substantial part of society does not view hacking as an art, but as a criminal behavior in all circumstances. When most people view hacking as an overwhelmingly criminal activity, it is unsurprising that legitimate researchers often find themselves working in a hostile environment, and one that threatens to punish them. Many documented instances of security researchers being threatened with legal action exist. A quick search on the Internet of the phrase "security researcher threatened" will bring up quite a bit of news.

Redefining the expectations of security research starts with educating the community – and bug bounty programs play a gigantic role in helping society understand that hacking can be ethical. Vulnerability disclosure programs are a great start, but the end state is a

transition to a bug bounty program that allows hackers to receive fair compensation for their efforts. Nonetheless, security research without utilizing a bug bounty program can be highly dangerous and can risk the livelihood of the individual conducting the research. A bug bounty program and the safe harbor clauses it contains can help to guarantee researcher safety. Vulnerability research has changed the landscape of what category hackers fall into, and has allowed quite a bit of flexibility and protection from punishment from the law.

1.11 Redefining Security Research

During the course of this book, the reader will see what skills are necessary to create, manage, and refine bug bounty programs. The one important aspect to remember when reading this book is that establishing or managing a bug bounty program is only one small part of a much bigger picture. History is being made, in real time, and the expansion of ethical hacking into the enterprise space is a necessary component of ensuring the safety of company assets and user data. Understanding how important programs can be is a way of information security that should be shared in a positive light. The best way to bring attention to the ethical nature of thousands of security researchers while they hack and make a difference is to operate with an open mind and attempt to give honest disclosure, while awarding processes a fair evaluation on every occasion.

Security research, or in other words the art of hacking, needs the assistance of enterprises that operate bug bounty programs – to adequately reshape the landscape of hacking. As a community, we cannot let the fear of hacking prevail as the action of shaming individuals that care about the security of an organization ends up causing more harm than good. Reshaping the world will take the cooperation and understanding of all individuals involved in the process. In addition, enterprises should maintain a neutral state of mind. Security researchers hack for various reasons: money, credibility, press, portfolio building, or fun. The reason vulnerability research is conducted should hardly matter: the only responsibility of the enterprise is to provide a safe environment and to patch the vulnerabilities. Fear of the press, while a legitimate concern, can be redirected into positive energy that rewards and values the researchers. If the organization engages openly with the researcher, it could well result in a positive outcome, in terms of media spin or as a learning outcome.

1.12 Taking Action

It shouldn't come as a surprise that word of mouth is a powerful tool. The enterprise space is ever-expansive and companies will constantly compete to be better than their competition. As a society, the establishment of honest programs and disclosure processes can influence the entire enterprise space. Here are some ways program managers or potential program managers can help assist researchers and the security research space.

1.12.1 Get to Know Security Researchers

Be involved in the community aspect of research. Whether reading publications on CVEs (common vulnerabilities and exposures) or bug writeups, participating in Twitter conversations, or connecting with hackers on LinkedIn, it's important to understand all aspects of the landscape.

1.12.2 Fair and Just Resolution

Running a program isn't the final solution. Managing an enterprise program requires collaboration and fair resolution processes. Ensuring that the program stays ethical and cares about the security researchers is a key part of spreading the positive aspects of bug bounty programs.

1.12.3 Managing Disclosure

While not recommended until a program is more established, eventually enterprises should strive to help researchers disclose their findings to the public when patched, if they wish to do so. Research disclosure helps inspire new generations of hackers and also receives enterprise, and potentially media, attention. Nonetheless, within a program security researchers should maintain the ability to disclose in any circumstance if the information is redacted enough or if a CVE exists on an enterprise product/there's user or customer PII exposure that needs to go public.

1.12.4 Corrections

Program managers should strive to speak highly of researchers and the great work that is provided as a service. "Hackers" aren't malicious by default, and program managers receive first-hand experience of ethical behavior. When hackers are called malicious, program managers should strive to set the record straight and describe the differences between an ethical hacker (security researcher) and a malicious hacker (threat actor).

1.12.5 Specific Community Involvement

Joining the movement for better disclosure is the first step to a greater collaboration between researchers and programs. Casey Ellis built a one-of-a-kind community named Disclose as a way for companies to participate in the conversation. (https://disclose.io).