How to optimize your bug bounty program for success [Part 4]

BY ANNA HAMMOND · MAY 23, 2022 · LAST UPDATED ON JULY 14, 2025

If you've been following along with this series on the bug bounty process, you should have a good idea of setting up a bug bounty program by now and what to expect when launching.

In this last article in the series, we will explore how to *optimize* bug bounty programs for success.

People, Processes, and Systems

Given the ever-increasing levels of cybercrime, many organizations now consider cybersecurity integral to how they plan and execute their business. Maximize the effectiveness of your bug bounty program through the common business triad of people, processes, and systems, and you'll soon be squashing bugs like a champ.

People

With a focus on *cyber*security, it's easy to forget that people are critical to bug bounty programs. First, there is the crowdsourced ethical hacker community that tests your systems and software for vulnerabilities. Then, there are your internal teams who run the program, process reports, and get fixes scheduled and completed.

Let's look at these two groups in turn.

Maximize ethical hacker engagement

Keeping crowdsourced security experts motivated to work on your bug bounty program is one of the most important optimizations you can make.

A bug bounty platform makes it easy to communicate and build a relationship with the ethical hackers you work with. But it's also helpful to have some wider insights into the community. So, take a few minutes to understand what motivates the ethical hacker community.

The Ethical Hacker Insights Report 2021

Once you've got the big picture, adhere to the following guidelines, and you'll build a good relationship with your crowdsourced hackers:

1. Be responsive

Don't leave a community security expert dangling when a submitted report or communication comes through—these are busy people! Review and accept or reject reports quickly.

2. Pay on time

Don't pay on time if you want to become a pariah of the bug bounty program community. Otherwise, pay as soon as you've accepted a report submission. Platforms like Intigriti help by automating this part of the process.

3. Be clear

Bug bounty hunting is a complex business. If you leave gray areas in your program's definition and then reject reported findings in those areas, you'll get discouraging testimonials from the community. So be precise, and if you do slip up, own it and fix it!

4. Be respectful

They say successful companies treat their employees like they treat their customers. Acting the same way with your security experts can go a long way to building a great relationship and encouraging them to go the extra mile for you.

5. Be fair

Ethical hackers are skilled security researchers helping to locate vulnerabilities that could cost your organization a fortune. You won't get motivated, talented hackers working on your programs if you don't set a fair bounty.

6. Make it fun!

Suppose you set up your program parameters to read like a 1950s civil service exam. In that case, you're not going to attract the young, dynamic, motivated, and personable hackers that make up the crowdsourced community. Speak their language!

Get (and keep) your internal team onboard

In our article on preparing to launch a bug bounty program, we discussed how important it is to have your internal teams ready for what might be an overwhelming rush of vulnerability reports when you first launch your program. This stage will require a certain amount of admin work. It will also mean engineering or IT staff time. The advice is the same in terms of people and relationships: check in often.

That advice holds for Program Managers and Engineering too. Make sure both teams can keep up with the flow of work. Is the work stressing them out? If the answer is yes, go back to how you prioritize work and create a new agreement.

Finally, without resorting to fear, uncertainty and doubt, it can be helpful to remind both groups about the massive inconveniences and financial costs of dealing with ransomware and unauthorized access to resources. It's an old cliché that prevention is better than a cure, but it is very accurate in this case.

Processes

We see it every time: bug bounty programs get results fast! Often, the result is that your systems or software have vulnerabilities that leave them open to exploitation.

In most cases, the vulnerability is not an indicator of lax security unfortunately, it has a lot more to do with the inventiveness of malicious hackers. It's also why even the world's biggest and best-protected companies have turned to crowdsourced cybersecurity.

So, how can you optimize addressing vulnerabilities that arise?

Prioritize your vulnerabilities

When a vulnerability is reported, you need to decide:

1. If it is valid
2. The level of severity of the vulnerability
3. Who is going to fix it
4. When they are going to fix it.

If you're running on a bug bounty platform that provides triage services, the first two items in this list are already taken care of. For information on setting and changing severity levels, check out our knowledge base article on handling submissions.

Prioritize your responses

Next up is getting engineering or the IT department to fix the bugs promptly. Optimize here by building a process that everyone has signed off on. It should answer, *how do we handle each level of severity in*

terms of time and resources?

The key idea is that when the vulnerability report comes in, everyone knows what is required of them. You don't want to be making security decisions on-the-fly. Therefore, make sure these decisions are already made:

1. Are we clear and agreed on the security threat levels we have established?
2. How do we handle each level of reported security threat?
3. What is the established timeframe to fix each type of vulnerability?
4. Which resources will be assigned to each type of vulnerability?
5. Are we ready for high severity risks?
6. Who will spend the weekend working if a high-severity bug comes in on a Friday afternoon?
7. How do we respond if we are suddenly inundated with more vulnerability reports than we can handle?
8. What's the internal feedback process for fixed vulnerabilities?

Having clear answers to these questions will ensure your bug fixing goes smoothly when the reports start coming in.

Systems

For organizations that prefer to reinvent the wheel, the below list is a good place to start in creating a spec for your own systems:

- Low operational overhead
- Intuitive tools for scoping a bug bounty program
- Strong channel for communication with researchers and the platform owner
- Automated, fast payment
- Includes a submission log and history.

However, before diving into such a mammoth undertaking, we'd humbly suggest you explore just how affordable and effective Intigriti is. Intigriti provides all of these aspects, plus comes with expert support. It also comes with a success manager who will proactively check in with you.

Need a recap?

And that's it! We've come to the end of our four part blog series on the bug bounty process. We hope you've found this and this other articles in this series useful.

If you'd like to take another look at the other topics we covered, here are all the articles in order:

- The 3 key stages to setting up and managing a bug bounty program

- How to prepare for launching a bug bounty program
- What to consider when launching a bug bounty program

Learn more

Intrigued by what you have read? Want to know more about bug bounty programs? Get in touch to request a demo with a member of our team today.

REQUEST A DEMO	VISIT THE WEBSITE	GET IN TOUCH
intigriti.com/demo	intigriti.com	hello@intigriti.com