bugcrowd

6 Questions

TO ASK BEFORE STARTING A BUG BOUNTY PROGRAM

They say there is no such thing as a dumb question, and with how many different forms a bug bounty program can take, you want to make sure you're asking the right questions. This way, you can find a solution that aligns with your organization's goals, budget, testing timelines, and interest in specific skills.

First, let's make sure we're on the same page about the definition of a bug bounty. Previously, the term "bug bounty" was used synonymously with the term "crowdsourced security." Now, crowdsourced security is seen as a resourcing model, while bug bounty represents a particular way of incentivizing and engaging those resources. Bug bounties leverage a competitive model that encourages testing through potential for reward. If hackers are the first to find a vulnerability within the scope defined, they are rewarded with monetary payments dependent on validity and impact. This model greatly reduces the average "cost per vulnerability" versus other security solutions and ensures that customers are truly only paying for value received.

Now that we understand the basics of how a bug bounty program works, let's go over six crucial questions to ask yourself before engaging with a bug bounty provider.

GET THE CHECKLIST

1 SELF-MANAGED OR **MANAGED?**

While a few large enterprises do have the team in place to manage their own bug bounty programs, these are usually highly visible, well-known, and reputable brands that can attract the attention of the broader security community. Organizations of all sizes typically opt for managed programs for easier payment processing, better relationship management, and the option for triage and prioritization.

If you decide to go the self-managed route, it's important you build a team with technical depth, VRT or CVSS expertise, strong communication skills, patience, and commitment to security above all else. You also need someone with marketing skills to ensure your program is visible to all of the right talent.

In either case, be sure to have defined processes in place for handling vulnerabilities.

2 IN SCOPE OR **OUT OF SCOPE?**

The first step in choosing which assets should be part of your bug bounty program is understanding your entire attack surface. This can be daunting. A 2020 ESG report found that two thirds of organizations say attack surface management is more difficult today than it was two years ago. Luckily, there are several solutions that help make this easier, such as Bugcrowd Attack Surface Management.

By understanding your attack surface, you can more readily convey to researchers what is (and isn't) in scope, ensuring that you get the results you want. Do you have multiple web apps? APIs? All of these can go in "scope" for a single program, or you can divide them if you have teams to support each. Once you establish what's in scope, you're ready to begin writing the "bounty brief" that will help communicate to researchers your targets, priorities, exclusions, and incentive scheme.

3 STAGING OR **PRODUCTION?**

Now that you've determined what's in scope, it's time to consider where in your development lifecycle it's more appropriate for focused testing. Where possible, we suggest utilizing pre-production/staging environments, as opposed to production. There are many reasons to consider this option, including reduced impact to customers, ease of credential provisioning for researchers, and much more.

4 PUBLIC OR **PRIVATE?**

The power of crowdsourced security is power in numbers. While that can refer to the total number of people on your program, it also refers to the broader network of available talent. More thoroughly vetted, and continuously ranked security researchers means you'll always have the team that's best fit for your testing environment.

Private programs are invite-only programs which target a select group of researchers based on technical and business requirements. No one else in the community or beyond will be able to see details on or access private programs.

With a public program, anyone registered through your platform can see, access, and work on your program. These typically have a much broader scope which allows for a wider range of potential vulnerabilities to be identified by a larger set of unique skills and experiences.

Because more people on your program also means more vulnerabilities, many organizations start small with invite-only access, until vulnerabilities reach a manageable level and you feel comfortable graduating to public access, if appropriate for your business.

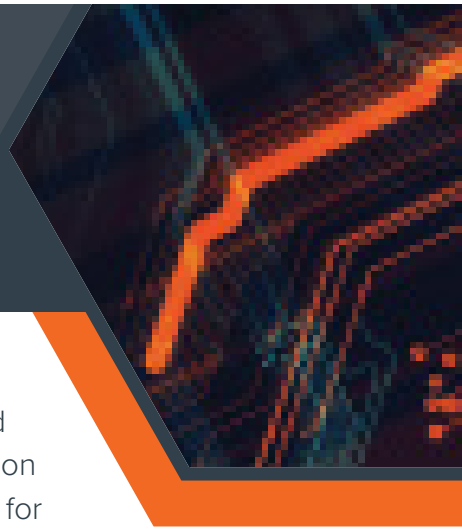
5 ON-DEMAND OR ONGOING?

While the structure of crowdsourced security programs enables continuous testing where it was previously not possible, it may be the case that testing or budget cycles limit you to only two-week testing sprints.

On-demand testing is a time-boxed point in time testing engagement that may be run in isolation, or periodically throughout the year.

Continuous testing is an ongoing testing engagement that is the best fit for high-value targets or agile development environments where the asset may face frequent change.

6 WHICH INTEGRATIONS MATTER MOST?



A strong vulnerability discovery solution is weak without a way to facilitate rapid remediation. Ensuring your platform can provide vulnerability-specific remediation advice and deciding which integrations matter most to your development team for presentation of that information is crucial.

Most modern bug bounty platforms offer developer tool integrations like JIRA, GitHub, ServiceNow, Trello, and Slack, making it easier for security to enqueue prioritized vulnerabilities, and for developers to see what should be addressed first, how to do it, and whether anything else stands in the way. It's important to make sure your platform can accommodate a centralized JIRA security project, in developer JIRA projects, and hybrid JIRA projects.

Find out more about the basics of bug bounty programs, best practices to launch a program, and how to achieve long-term success in

→ **The Ultimate Guide to Bug Bounty**