bugcrowd

Ultimate Guide to
Bug Bounty

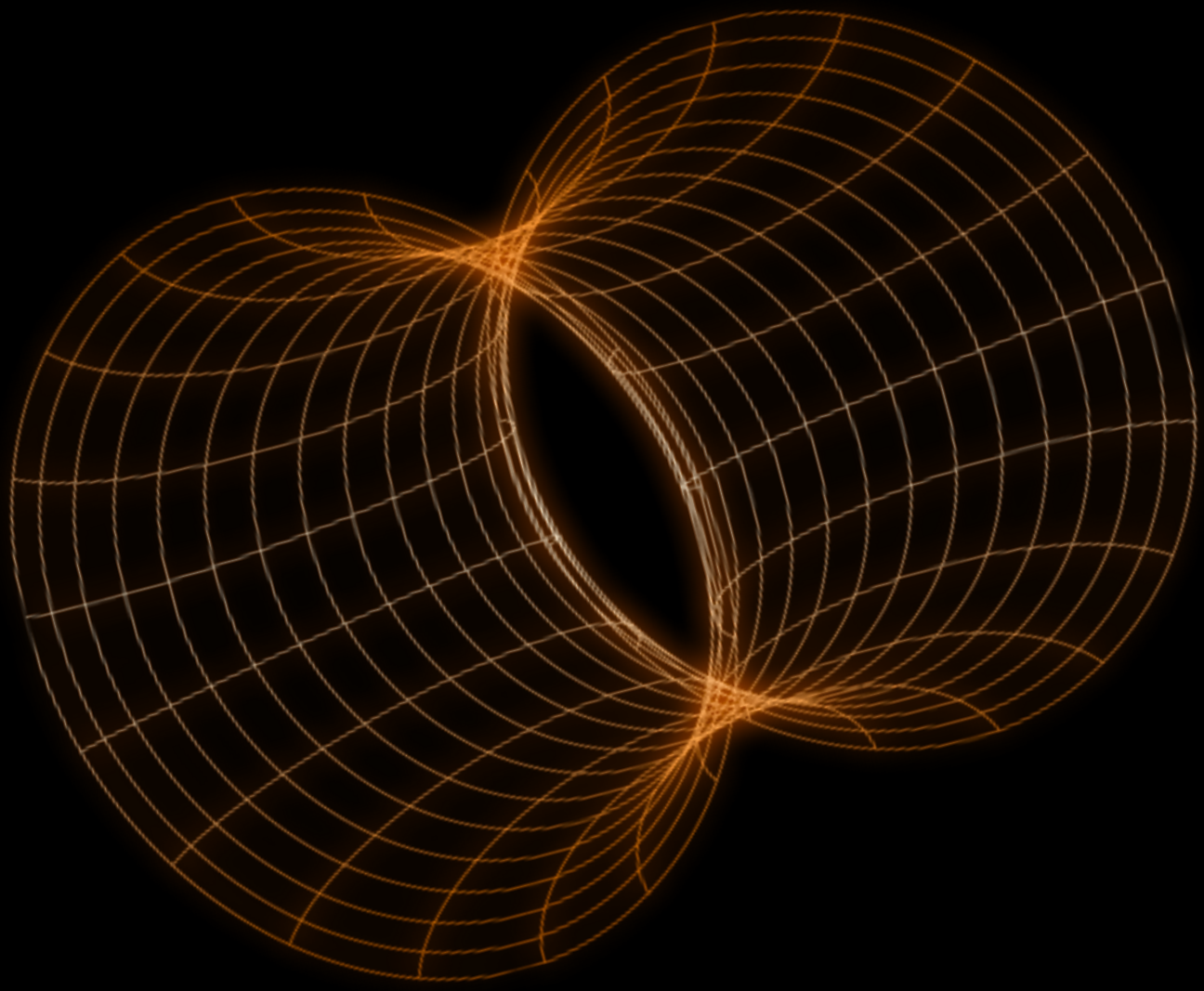


Table of Contents

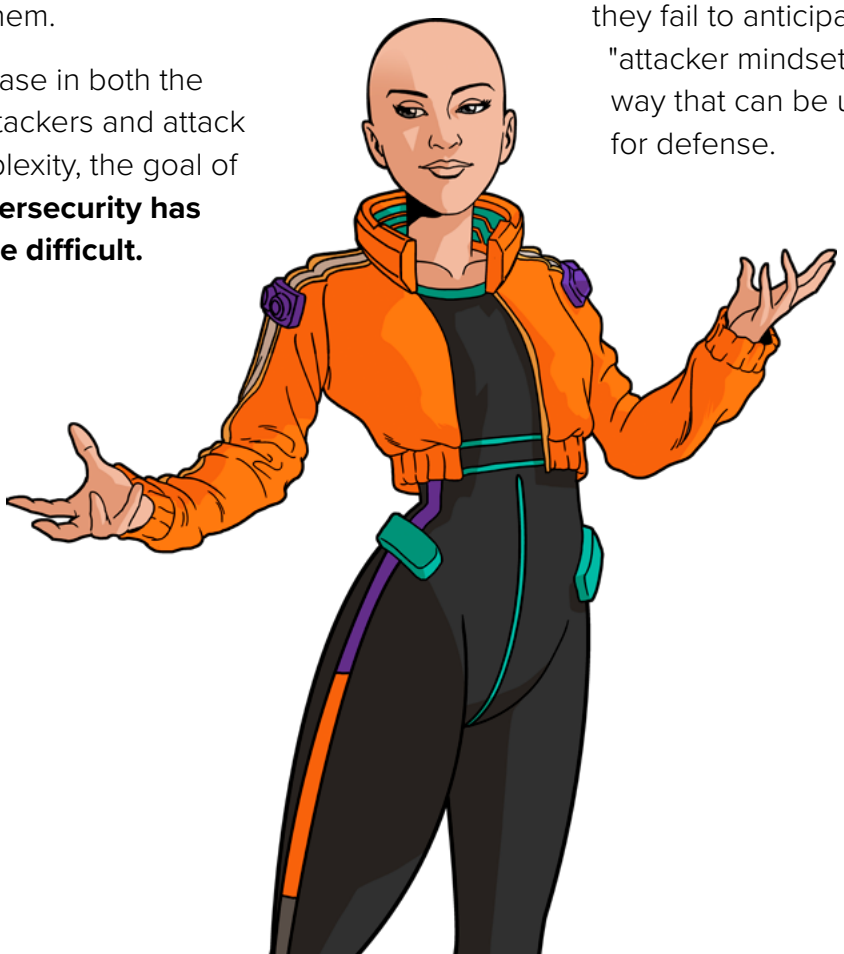
Everything you need to know about bug bounty programs

Organizations have widely adopted various tools and training to help find security vulnerabilities in digital assets and mitigate the introduction of vulnerabilities during coding.

Unfortunately, the hastened onslaught of cybersecurity threats suggests that these solutions are still behind the curve in finding and helping to quickly remediate the most critical issues. Both the prevalence and impact of cyberattacks are worsening. This suggests that status quo security solutions have largely failed to help security teams discover vulnerabilities before malicious attackers can find and exploit them.

With an increase in both the number of attackers and attack surface complexity, the goal of ensuring **cybersecurity has become more difficult.**

Many organizations invest in automation to address the increasing scale of attacks across massive assets, which is logical. However, scanners and other automated tools don't recognize emerging vulnerabilities—only those that are already well understood. They have little to no understanding of context, making it almost **impossible to separate signal from noise**. Furthermore, they fail to anticipate the "attacker mindset" in a way that can be utilized for defense.



For those reasons, throughout the past decade, the industry has discovered the **irreplaceable value** of incentivizing highly skilled hackers to uncover hidden critical flaws in today's massive attack surface—in the form of **bug bounty programs.**

How you design, measure, and implement a bug bounty program will have a major impact on its long-term success. To help you navigate this process, this guide explains:

- The evolution of crowdsourced security and the emergence of the Crowd.
- How a "bug bounty" is defined and its key benefits.
- The different components of a bug bounty program.
- How to get started, grow, and measure the impact of your bug bounty program over time.
- What to ask a prospective bug bounty provider to ensure a good fit with your resources.
- How to differentiate between a bug bounty program, a vulnerability disclosure program, and penetration testing.

The Basics of Bug Bounty Programs

What Is Crowdsourced Security?

Like other forms of crowdsourcing, **crowdsourced security** unites a disparate set of individuals to work toward a common goal. In this case, the goal is to discover and report hidden vulnerabilities in an attack surface.

Crowdsourced security is an approach to securing digital assets that draws from the **collective skills and experiences of the world's community of hackers.** These **highly capable** individuals are given the direction, scope, and incentives they need to effectively simulate the varied techniques employed by threat actors to identify and report vulnerabilities. To learn more about crowdsourced security, check out Crowdsourced Security 101.

What Is a Bug Bounty?

Previously, the term "bug bounty" was used synonymously with the term "crowdsourced security." With the arrival of additional ways to engage with a crowd, like penetration testing as a service (PTaaS) and attack surface management, the two terms have now been decoupled. Crowdsourced security is a resourcing model, while bug bounties involve an incentive ("pay for results") model that encourages the discovery of severe flaws based on the potential for monetary rewards.

For example, if a hacker involved in a bug bounty reports a cross-site scripting vulnerability but the same vulnerability was already noted by the customer's internal security team or if it was uncovered by another hacker first, the individual is not paid for that submission. In another example, two hackers may uncover different types of server security misconfigurations. If one is email spoofing and the other is using default credentials, both hackers would be paid, but the latter would command a higher rate due to greater potential business impact.

This model greatly reduces the average cost per vulnerability and ensures that customers are only paying for value received—which makes security return on investment (ROI) much easier to calculate.

History of Bug Bounties

In 1851, Charles Alfred Hobbs was paid 200 gold guineas by a lock manufacturer for rising to the challenge of picking one of its strongest locks. Flashing forward to the mid-90s and early 2000s, Netscape, IDefense, Mozilla, Google, and Facebook all had their own self-managed bug bounty programs, offering severity-based rewards to anyone who could identify vulnerabilities in their web applications.

Some organizations with large security teams and at an advanced stage of security maturity may still run their own bug bounty programs, but most that start in self-managed mode eventually migrate to "bug bounty as a service" solutions when they reach a certain scale. Generally, running bug bounty programs is outside the core competencies of most teams.

Who Participates in a Bug Bounty Program?

In crowdsourced security, **"the Crowd"** is the term used to refer to the massive, global community of hackers (also referred to as security researchers, ethical hackers, or white hats) who participate in bug bounty programs. These individuals are **independent actors** who work on crowdsourced security programs that they find to be fulfilling, lucrative, or both, either as their sole occupation or as a side hustle. The Crowd is the lifeblood of any crowdsourced security or bug bounty program and the main reason why the approach is so effective.

Hackers can and do hunt bugs on multiple platforms—no provider has an exclusive monopoly on them—so it's important for a platform to match **the right crowd** to the end user's needs at the right time.

Some crowdsourced security vendors boast a high number of hackers working on customers' programs, but quality is a more important metric to focus on when working with the Crowd. Organizations want to be sure they're working with a vendor that uses data to source and activate hackers with precisely the right skill sets and experience for their programs to boost engagement and critical findings—not just "throw bodies" at a problem.

Key Benefits of Bug Bounty Programs

Bug bounties are a pay-for-results approach to proactive security testing designed to maximize the discovery of high-impact vulnerabilities. Through managed bug bounty programs, organizations are given access to **thousands of highly skilled and thoroughly vetted hackers** ready to help organizations find vulnerabilities that other tools miss. The global nature of the Crowd means **24/7 talent availability,** with launch timelines that blow traditional utilization-based models out of the water. The ideal provider also offers 24/7 **vulnerability visibility and reporting,** fine-grained **crowd matching** to ensure access to the right talent, and **seamless business process integration** with a development team's favorite ticketing and vulnerability management solutions.

Other Key Benefits Include:

SHARED ACCESS TO TOP TALENT

The crowdsourced model allows all participants to share the value of something impossible to replicate alone. Additionally, bug bounties provide an **elastic workforce** as needed.

RAPID LAUNCH AND TIME TO VALUE

A community of skilled, trusted hackers looking for bugs inside a pay-for-results framework drastically **reduces the time to launch.** A competitive, first-to-find incentives model also compresses the time needed to discover truly impactful bugs.

CONTINUOUS ASSURANCE

Not paying per head or per hour means organizations can afford to have a testing practice that matches today's agile and continuous development cycles—because **attackers don't take days off.**

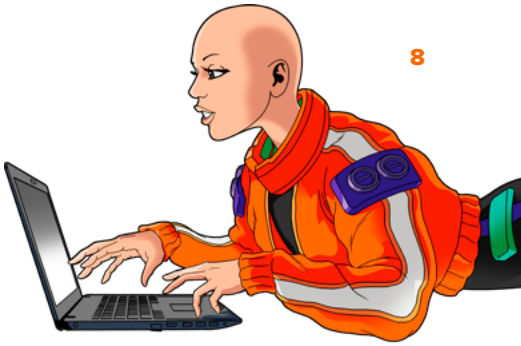
UNIQUE SKILLS ON DEMAND

Many organizations have a great in-house team, but even the most resourced teams can't cover all the skills needed to find all potential vulnerabilities within systems. The Crowd offers the largest rolodex of vetted, ranked, and highly active hackers with **infinite combinations of skill and experience.**

RAPID RISK REDUCTION

Competitive, incentive-based testing motivates hackers to **think creatively** and find high-impact vulnerabilities that present the greatest risk to businesses.

What Do Bug Bounty Providers Offer?



Unless their security maturity level is extreme, most organizations will choose to work with a managed bug bounty provider. Providers differ in robustness, comprehensiveness, and depth, so when comparing them, it's important to understand each of their approaches.

Hacker engagement

Some bug bounty providers rely on leaderboards or other coarse-grained methods for building a bug bounty team. However, custom-curating a crowd based on skill, interest, ability, performance, and other dimensions (as Bugcrowd does) can make a huge difference in a program's success. So when evaluating a provider, it's important to dig into its crowd matching methodology to ensure it aligns with goals and expectations.

Validation and triage

For bug bounty to deliver its full value, submissions must be validated and prioritized according to severity to help the program owner filter out noise. Be aware, however, that some providers offer triage as an add-on or afterthought (if at all), not a core competency. When choosing what is right for you, be aware that even "invite-only" bug bounties result in significantly more vulnerabilities than you may be used to.

SDLC Integration

Most serious providers will offer integrations into popular developer workflow tools, such as JIRA, GitHub, and ServiceNow, as well as an API. So it's important to ask whether the integration is robust enough to meet your team's most common use cases. In addition to these common developer tools, integrations into Slack and Trello can also improve communication and alerting workflows, while integrations with vulnerability management tools like Qualys can help contextualize and prioritize vulnerabilities from all discovery solutions.

Reward payouts

Valid, non-duplicate, and in-scope vulnerabilities are rewarded from a set-aside sum of money known as the "bounty pool," which can be topped off when it runs low. The actual reward is set in advance based on the incentive model determined by the program owner. By allowing an intermediary/provider to handle compliant payments, organizations avoid the headaches of individual tax procedures that differ by state and country.

Reporting and analytics

As the saying goes, "You can't improve what you can't measure." To ensure the long-term success of bug bounty programs, it's important to have the ability to monitor program health, uncover insights into trends and opportunities, and benchmark against internal or industry KPIs. This requires access to historical data that provide context about what success looks like, but not all providers (particularly newer ones) will have such data.

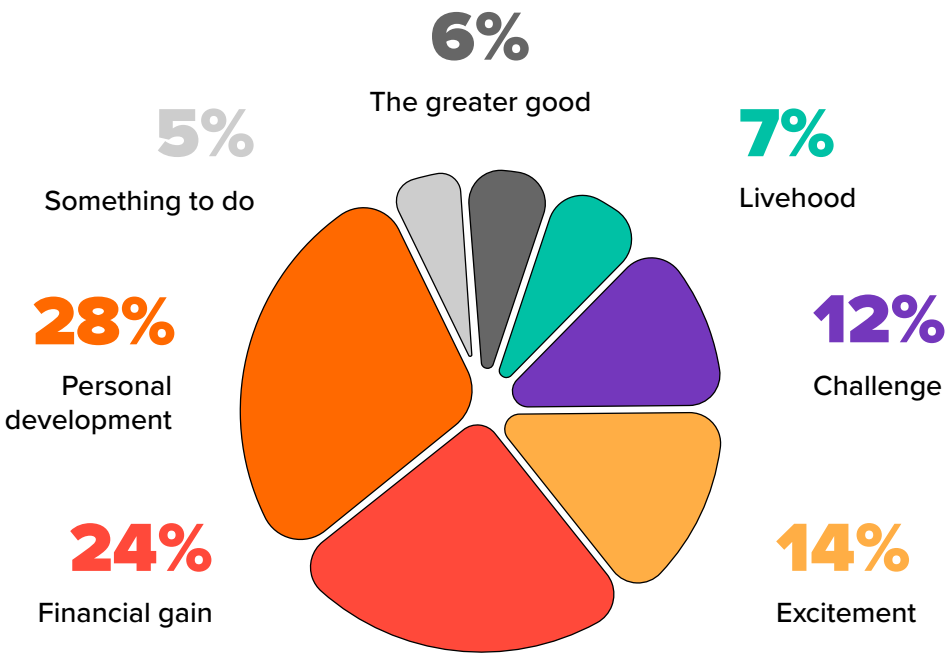
What Motivates Hackers?

Hackers go by a variety of names, but all **share one critical trait**—a desire to not only improve their families' and their own lives but also to improve customers' lives.

Per Bugcrowd's Inside The Mind of a Hacker report, **75% of hackers** identify non-financial factors, such as personal development, the greater good, and enjoying a challenge, as their main motivators to hack. Only **29% of elite hackers** hack full time, while the majority split hours between part-time hacking and full-time employment as analysts, engineers, and even CISOs. Furthermore, **77% of hackers** work in IT or cybersecurity. The bug bounty community is a global group of well-intentioned individuals from all walks of life, with diverse backgrounds, technical skills, and expertise.

The bug bounty community is a global group of well-intentioned individuals from all walks of life, with diverse backgrounds, technical skills, and expertise.

This diversity is what makes bug bounties so impactful—the crowd offers the opportunity to connect uniquely skilled individuals with organizations that need fresh perspectives.



Can I Trust Hackers?

Ten years ago, a handshake and a background check were all that stood between aspiring pentesters and an organization's data. Trust was binary and assumed by employment. **But times have changed.**

Bug bounty programs provide us with the opportunity to think critically about how trust is defined, measured, allocated, and revoked. Does a background check make someone more trustworthy than 3,000+ vulnerability submissions? For some, yes. Others would say that the promise of a reward serves as a greater incentive to follow the rules.

Modern bug bounty programs now have the technology required to dynamically assess talent by using both traditional measures of trust and those that consider **performance and behavior.**

Code of Conduct

While each provider decides how it allocates and manages trust, most start with a pre-program agreement. For example, hackers on the Bugcrowd Platform must agree to our Standard Disclosure Terms, which include a charter to do no harm in testing or in subsequent communications. This is similar to a non-disclosure agreement, but oftentimes, customers choose to work with hackers to safely disclose resolved findings for the good of all involved. In addition to this understanding, bug hunters on our platform also agree to abide by our Code of Conduct, which outlines the behavioral expectations for all Bugcrowd community members, both on and off platform.

Measures of Trust

There are several ways to assess trust:

Background checks

Background checks look for felony and major misdemeanor criminal convictions at the country, state, and federal level, as well as in international watch lists. Hackers must provide their full name, email, and countries in which they have lived in the last 7 years. While some platforms conduct background checks on all their hackers, Bugcrowd takes the reverse approach—only hackers who have proven themselves to be both skilled and professional are invited to complete a confidential criminal background check, if they so choose. This approach ensures that participants in programs requiring background checks comprise the most elite hackers.

ID-verification

ID verification is sometimes required to ensure hackers are who they say they are and operating from the locations that we expect. Bugcrowd hackers can choose to be verified through a service known as NetVerify. Hackers initiate this process by uploading a picture of their face and photo ID. NetVerify then uses this information to validate identity and confirm location. This is useful for programs that require only nationals to be invited, but it also helps hackers who are not operating from any areas on the OFAC global banned list.

Behavior and communication

Behavior and communication are also indicators of trustworthiness. Like security testing, trustworthiness cannot be determined based on a point-in-time assessment. Trust is built on a holistic and continuous view of a person's behaviors and interactions. Bugcrowd's Hacker Success team vigilantly monitors every active hacker's interactions both on and off the platform to address any gaps or de-escalate misunderstandings. Just as a full-time employee can be removed from employment for egregious behaviors on social platforms, so too can a member of the Crowd be banned from individual programs or the platform as a whole.

Factors to Consider When Getting Started with a Bug Bounty Program

Bug bounty programs can take on many different forms depending on an organization's goals, budget, testing timelines, and interest in specific skills.

Before engaging with a bug bounty provider, ask the following questions:

Q Self-Managed Or Managed?

While a few large enterprises do have the team required 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 a few reasons:

Payment processing

This can be a nightmare for even the largest organizations to manage because it involves draining the resources of departments outside of security, such as finance. Managing who gets paid when and ensuring that tax forms are provisioned for each is no easy task.

Relationship management

Like all communities, every member of the Crowd has their own unique way of interacting, hunting, and communicating that is often more an exercise in psychology than it is in operations. Most providers typically employ a dedicated team of experts who either come from the community or are well trained in the nuanced issues that hackers face. This can drastically reduce the chance of miscommunication or misunderstanding between the two parties.

Triage and prioritization

Bugcrowd offers engineered, best-in-class triage to drastically reduce the burden of vulnerability validation and prioritization and puts an enormous amount of time into caring for the hacker community and ensuring that their needs are met. Without an objective intermediary, relationships can sour, resulting in a sudden drop in engagement that can spread quickly throughout the community.

Q Narrow Scope or Open Scope?

A scope is the defined set of targets listed by an organization as assets that are to be tested as part of a particular engagement. Things that are listed as "in scope" are eligible for testing, and things that are "out of scope" are not to be tested. Within the context of a bug bounty, what's in scope is what hackers are incentivized to report (and are rewarded for), and what's out of scope is off limits, and no compensation is given for findings related to those targets. Generally, it's best to reach a **maturity stage** that implements an open scope as quickly as is feasible because attackers have no limits where targets are concerned.

DEFINING SCOPE

Limited scope **Wide scope** **Open scope**

Limited scope

A limited scope on a bug bounty program is one that only includes single or specific targets. For instance, listing "example.com" as the only in-scope domain is considered a limited scope. Even if "accounts.example.com" and "api.example.com" were added to create a larger scope, this is still considered limited. Any time the scope is made up of precisely specific targets, it's generally considered a limited scope.

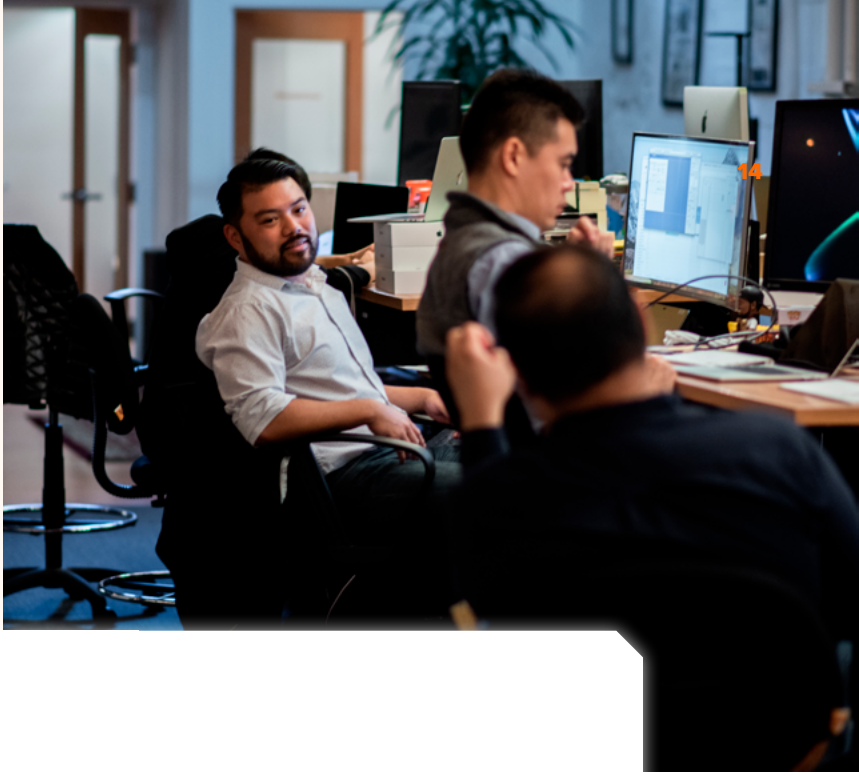
Wide scope

A wide scope bounty program is one that includes a wildcard in the in-scope targets, such as "*.example.com." This signifies that any subdomain of example.com is in scope. For instance, part of that wildcard could include previously unmentioned or unexplored attack surfaces, such as "staging-2019.example.com" or "admin.example.com," both of which could offer rich opportunities for identifying vulnerabilities. Hackers are particularly adept at finding and exploiting assets that have been forgotten or hidden by the sands of time. By including a wildcard, an organization increases the probability of identifying security risks across a much broader swath of an attack surface.

Open scope

An open scope bounty program is one that has no limitations on what hackers can or cannot test, so long as the target/asset belongs to a specific organization. Open scopes generally look something like "any externally facing asset belonging to Example Org," where nothing is excluded. Hackers are highly effective at identifying assets here—some may find and exploit an old marketing page for an event from a decade ago. Additionally, they may find keys or sensitive information stored on GitHub or Pastebin. There may be remnants from mergers, acquisitions, and any other artifacts that live in a litany of different places on the web. Running an open scope leverages the power of the whole crowd in finding and identifying any exposures an organization may have online, and most of the time, there's a lot more out there than an organization realizes.

Organizations generally choose between a limited scope, a wide scope, or an open scope for their programs. Once an organization establishes what's in scope, it can begin writing the **"bounty brief"** that will help communicate to hackers its targets, priorities, exclusions, and incentive scheme.



Q Staging Or Production?

After determining what's in scope, it's time to consider where in the **development lifecycle** focused testing is most appropriate. Where possible, we suggest utilizing preproduction/staging environments, as opposed to production. There are many reasons to consider this option, including **reduced impact on customers** and ease of credential provisioning for hackers, and much more:

- Hackers can help identify issues in new app versions before each release.
- There's no chance of staging environments made unstable by the volume and type of hacker testing affecting users.
- It's typically far less expensive to fix vulnerabilities identified in preproduction before a service is made widely available.
- It's typically easier to mass-create staging credentials for hackers.
- If there is a purchase point in an application, it's usually much easier to provide fake credit cards, SSNs, etc., in non-production environments.
- It's much easier to restrict access to only certain hackers (only allowing access from a specific IP address), thereby providing better visibility into hacker testing/coverage.

Q Public or Private?

The power of crowdsourced security stems from its numbers. While this can refer to the total number of people involved in a program, it also refers to the broader network of available talent. More thoroughly vetted and continuously ranked hackers means that organizations will always have the team that best fits their testing environments.

Because more people on a program also means more vulnerabilities, **Bugcrowd recommends starting small,** with invite-only access, until vulnerabilities reach a manageable level and organizations feel comfortable graduating to public access (if appropriate).

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

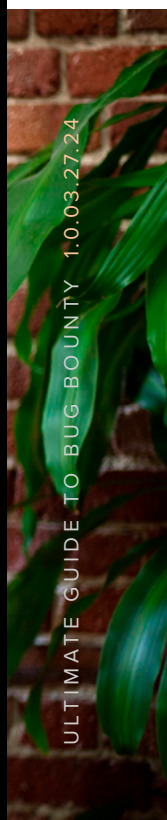
With public programs, any registered hacker can see, access, and work on their stated scope of assets. Public programs typically have a much broader scope, which allows for a wider range of potential vulnerabilities to be identified by a larger set of unique skills and experiences. Check out some of Bugcrowd's public programs on our website.

Q 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 an organization to only ~2-week testing sprints.

On-Demand
A time-boxed, point-in-time program may run in isolation, or periodically throughout the year.

Continuous
An ongoing program is a good fit for high-value targets or agile development environments, where the asset may face frequent change.



Q Which Integrations Matter Most?

A strong vulnerability discovery solution is weak without a way to facilitate rapid remediation. While security teams aren't responsible for providing the fix, they are better served if they can make the remediation process as easy as possible for the development team.

Therefore, it is important to ensure that a provider can provide **vulnerability-specific remediation advice** and decide which integrations matter the most to a development team for the presentation of that information. The Bugcrowd Platform offers pre-built integrations with JIRA, GitHub, ServiceNow, Trello, and Slack, in addition to webhooks and APIs, making it easier for security professionals to enqueue prioritized vulnerabilities and for developers to see what should be addressed first, how to go about this, and whether anything else stands in the way. **Context is key.**

As JIRA is the most common ticketing and management system for most users, it's important to accommodate the following top three use cases:

Centralized JIRA security project

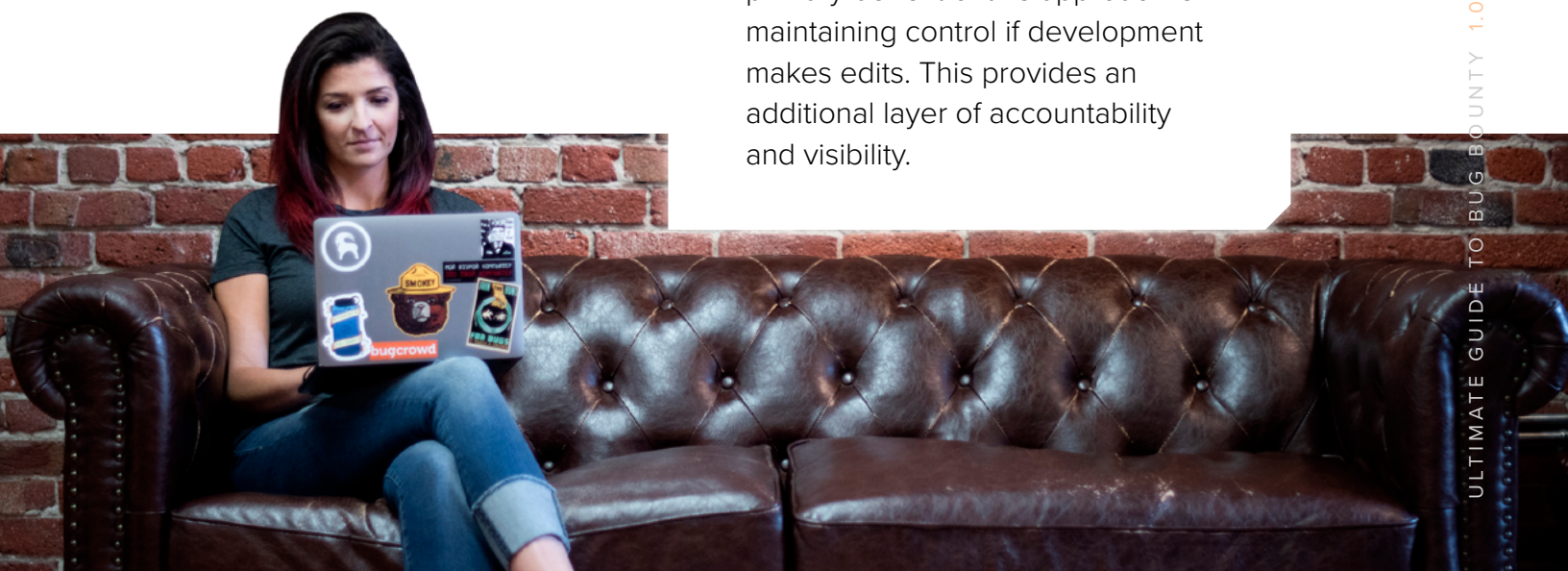
The AppSec team has one "security" JIRA project to manage its security work. Having one security JIRA project between security and development is a great way to centralize work; it is simple to maintain, as there is no logic needed to understand where tickets are created.

In developer JIRA projects

The AppSec team pushes security tickets into a developer's JIRA projects while respecting the developer's ownership. Enterprise organizations typically have more than one development team or business application, which requires more than one JIRA project.

Hybrid

This hybrid of both features one "security" project and a linked issue in a developer's JIRA projects. The primary benefit of this approach is maintaining control if development makes edits. This provides an additional layer of accountability and visibility.



Achieving Long-Term Success with a Bug Bounty Program

Setting Expectations

Bug bounties can greatly reduce the risk of vulnerabilities to an organization. Leveraging a solution like the Bugcrowd Platform can relieve a lot of the burden. But this doesn't remove the importance of program owners being **active participants** in their programs.

Bug bounty programs take time to maintain and grow over time, and **broad organizational commitment is required** to make them successful.

It's important for security professionals to have open dialogue with their executive teams about the implications of such a program. This could include potential impacts on budget structures (to accommodate a variable bounty pool), as well as impacts on engineering should a sudden influx of vulnerabilities disrupt current processes.

Additionally, bug bounty program owners must commit to **timely platform responses,** including accepting validated vulnerabilities or addressing program issues raised by a provider.

Crawl, Walk, Run

Approaching bug bounty programs with a **"crawl, walk, run**" mindset is a recipe for success for any organization of any size. Big public launches drive press coverage and broader awareness, but these aren't always appropriate if a security team is not quite ready for that volume. Processing and payment is one matter, but once a program owner knows about an issue, they should also be prepared and equipped to promptly resolve them. An example of the "crawl, walk, run" approach includes:

Launch private bug bounty with limited scope

Transition to public program

Increase rewards, add targets, boost engagement

Tips to Growing a Bug Bounty Program

Finally, iteration is an important part of any successful bug bounty program. **What worked to fuel hacker engagement yesterday might not work today.** Bugcrowd has a decade of experience in identifying any risks to growth, and as a result, will rely on three key "levers" to encourage long-term success.

Evolve a program's scope

Re-evaluate the attack surface: Many programs start with publicly available web targets. As time goes on, it's important to explore an organization's full attack surface. Hackers are most committed to programs with a varied and evolving scope.

Consider the product pipeline: New and recent product features or code changes are often overlooked for bug bounty inclusion. Working a bug program into any security/product interface can reduce the chances that something is missed.

Review the bounty brief: A bug bounty vendor will be well practiced in restructuring a bounty brief to ensure an organization's interests are being clearly communicated. Perhaps you were hoping for more testing in a certain area. If you're not explicit about it in your brief, it's possible that hackers missed the information.

Keep up with reward rates

Grow with the program: The longer programs run, the higher the rewards should be to reflect the increased difficulty of finding new vulnerabilities. Bugcrowd can also provide insight into the market rate for vulnerabilities, as they change over time. Don't forget that hackers can choose from many different programs; the right reward range can help programs stay competitive.

Demonstrate code confidence: Increasing rewards typically signifies confidence in the products launched. Higher rewards attract more skilled hackers, who are happy to accept the challenge. "Hardened" targets with a narrower scope should increase rewards to ensure proper attention from skilled hackers.

Highlight areas of interest: If a scope includes multiple targets but an organization recently updated the code to one particular asset and wants to ensure it's thoroughly tested, temporarily increasing rewards for that asset is likely to boost engagement.

Focus on relationships

Reduce response time: Response time is measured as the time between when a customer receives a triaged bug and the time the submission is reviewed by the customer. Typically, customers should at least respond to (accept or reject) submissions within one week.

Invest in communication: Managed bug bounty programs thrive off a very symbiotic relationship; hackers and customers must work together and understand one another's needs.

Act with empathy: It's important to remember that hackers are human and have families and lifestyles to support. Having a reputation for accepting vulnerabilities in a timely fashion helps hackers identify which programs they can rely on for timely payouts.which programs they can rely on for timely payouts.

Bug Bounty Programs vs. Pen Testing vs. Vulnerability Disclosure Programs (VDPs)

Bug bounty programs, pen testing, and VDPs are standard offerings of an elite crowdsourced security platform. However, the difference between these three offerings can be a little confusing, especially for organizations looking to combine products as part of a **layered security approach.**

VDPs

A VDP is a secure, publicly available channel for anyone to submit security vulnerabilities to organizations, helping them **mitigate risk** by enabling the disclosure and remediation of vulnerabilities before they are exploited by bad actors.

In contrast to bug bounties, VDP submissions are not incentivized by cash rewards. Publishing a vulnerability report after it has been fixed is another common attribute of VDPs and gives hackers the opportunity to share knowledge and enhance their own reputation in the process.

A VDP is also open scope, meaning that anybody can participate and attempt to find vulnerabilities on any target/asset belonging to an organization.

Whether an organization also has a bug bounty program, we highly recommend that every organization leverage a VDP. A VDP should be a baseline security standard for everyone. A VDP establishes a “see something, say something” mindset within an organization that carves out a global channel for vulnerability reports and publicly demonstrates that a company is doing everything possible to protect its customers, partners, and suppliers.

87%

of organizations report receiving at least one P1 vulnerability through their VDP.

Pen Testing

According to the National Institute of Standards and Technology (NIST), pen testing is defined as “security testing in which assessors mimic real-world attacks to identify methods for circumventing the security features of an application, system, or network.”

In other words, pen testing is a **simulated cyberattack** carried out by an authorized third party (known as pentesters), who tests and evaluates the security vulnerabilities of a target organization’s computer systems, networks, and application infrastructure.

Pen tests have **three defining characteristics:** they are performed by external testers, are typically time bound, and usually follow a testing methodology. Many organizations also expect a final report to demonstrate regulatory compliance to an auditor.

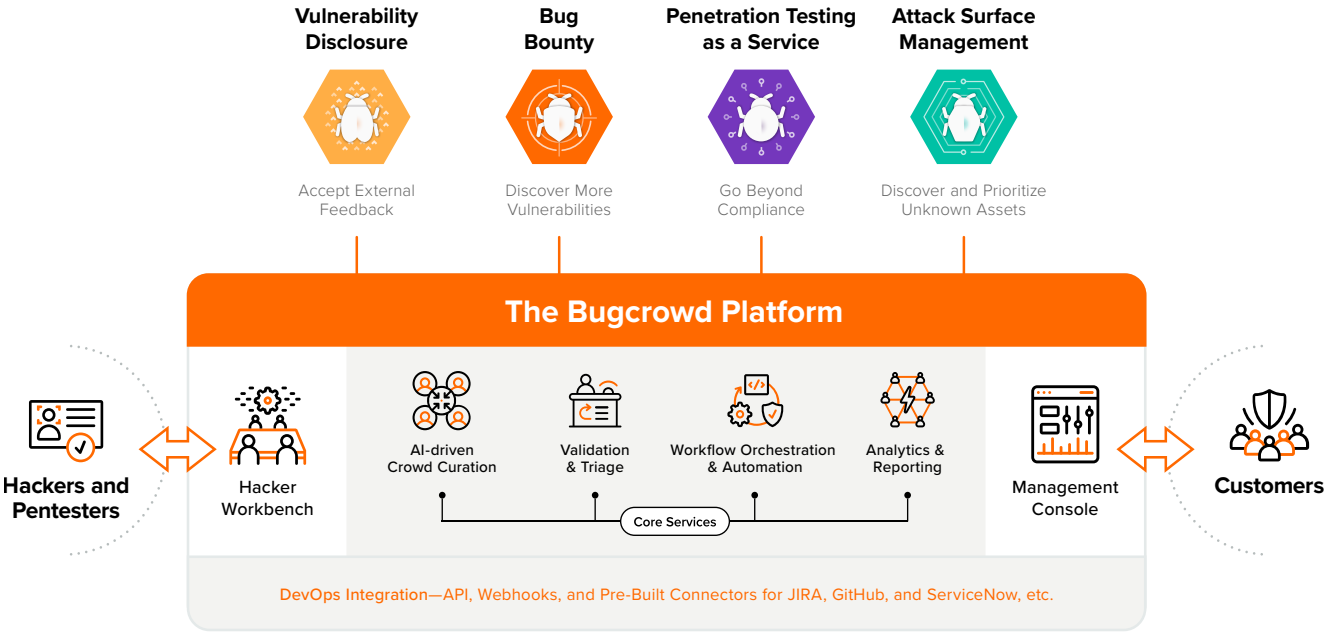
It’s common to conflate bug bounty programs and pen testing because they rely on attacker tools, techniques, and mindsets for vulnerability discovery under a predefined scope. Pen testing and bug bounty programs have very similar goals but differ with respect to the intensity of the assessment. Pen tests are methodology driven and are best for coverage, whereas bug bounties are better for risk reduction.

With this in mind, one can easily envision a layered strategy for both compliance and risk reduction that combines:

- Ongoing vulnerability discovery and assessment—when the exploitability of vulnerabilities is confirmed, this is what some might consider a “basic” pen test.
- Periodic, human-driven pen testing to find common flaws that Option 1 may have missed (what some might consider a “standard” pen test).
- A continuous bug bounty running “over the top” to pick up emerging vulnerabilities not yet reflected in the methodologies used in Options 1 and 2.

The Bugcrowd Platform

Bug bounty programs aren't the only way to leverage the power of the Crowd. The multi-solution Bugcrowd Platform brings the right crowd into all your workflows at the right time, allowing you to run bug bounties, penetration tests, VDPs, and more at scale and in an integrated, coordinated way.



✓ Best Security ROI from The Crowd

We match you with the right trusted hackers for your needs and environment across hundreds of dimensions using AI.

✓ Instant Focus on Critical Issues

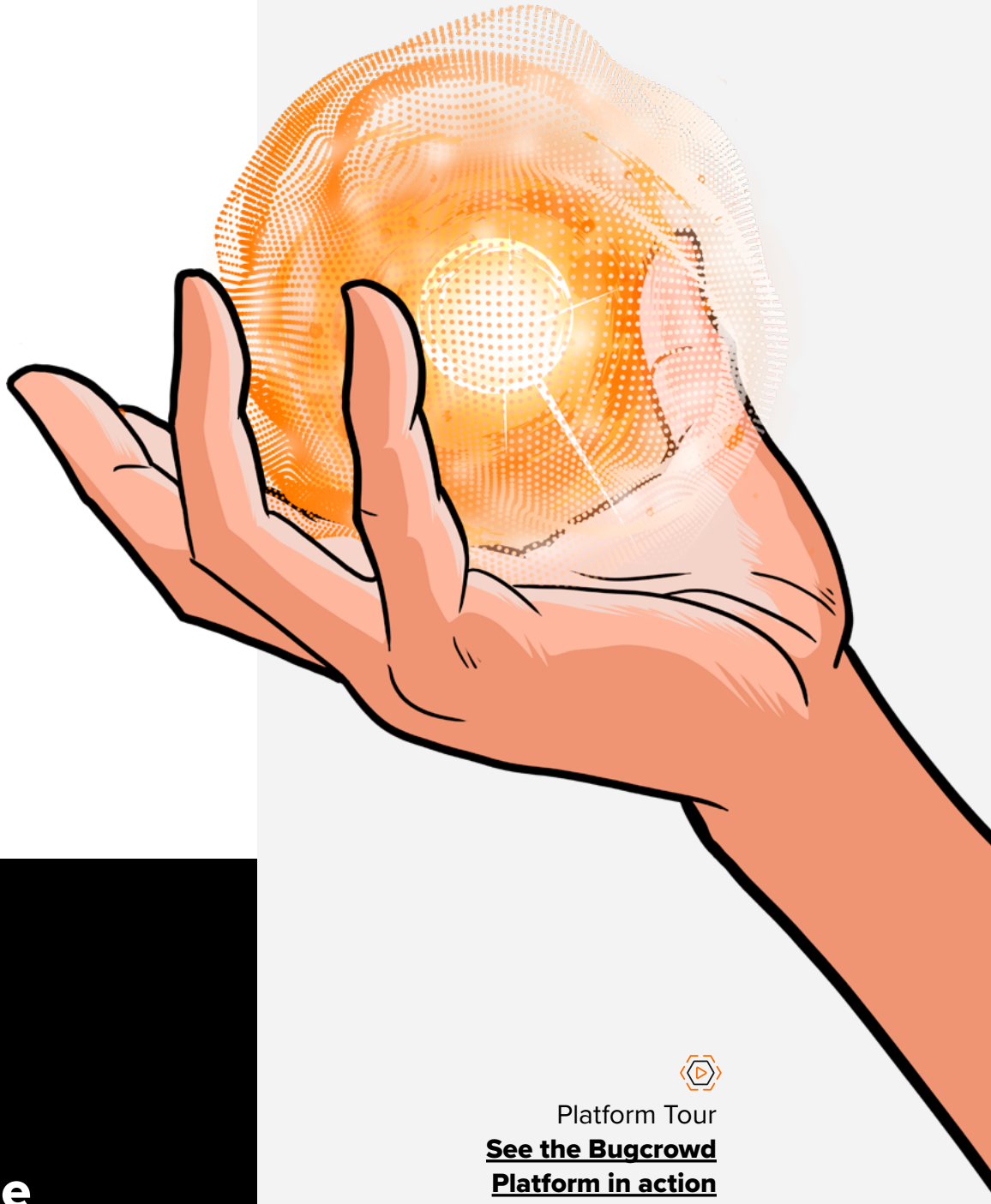
Working as an extension of the platform, our global security engineer team rapidly validates and triages submissions, with P1s (critical vulnerabilities) often handled within hours.

✓ Continuous, Resilient Security for DevOps

The platform integrates workflows with existing tools and processes to ensure that applications and APIs are continuously tested before they ship.

✓ Contextual Intelligence for Best Results

We apply accumulated knowledge from over a decade of experience crafting thousands of customer solutions to your goals for better outcomes.



Unleash the ingenuity of the global hacking community now

Try Bugcrowd

Platform Tour
See the Bugcrowd Platform in action

Data Sheet
Managed Bug Bounty

bugcrowd