bugcrowd

Managed Triage Guide

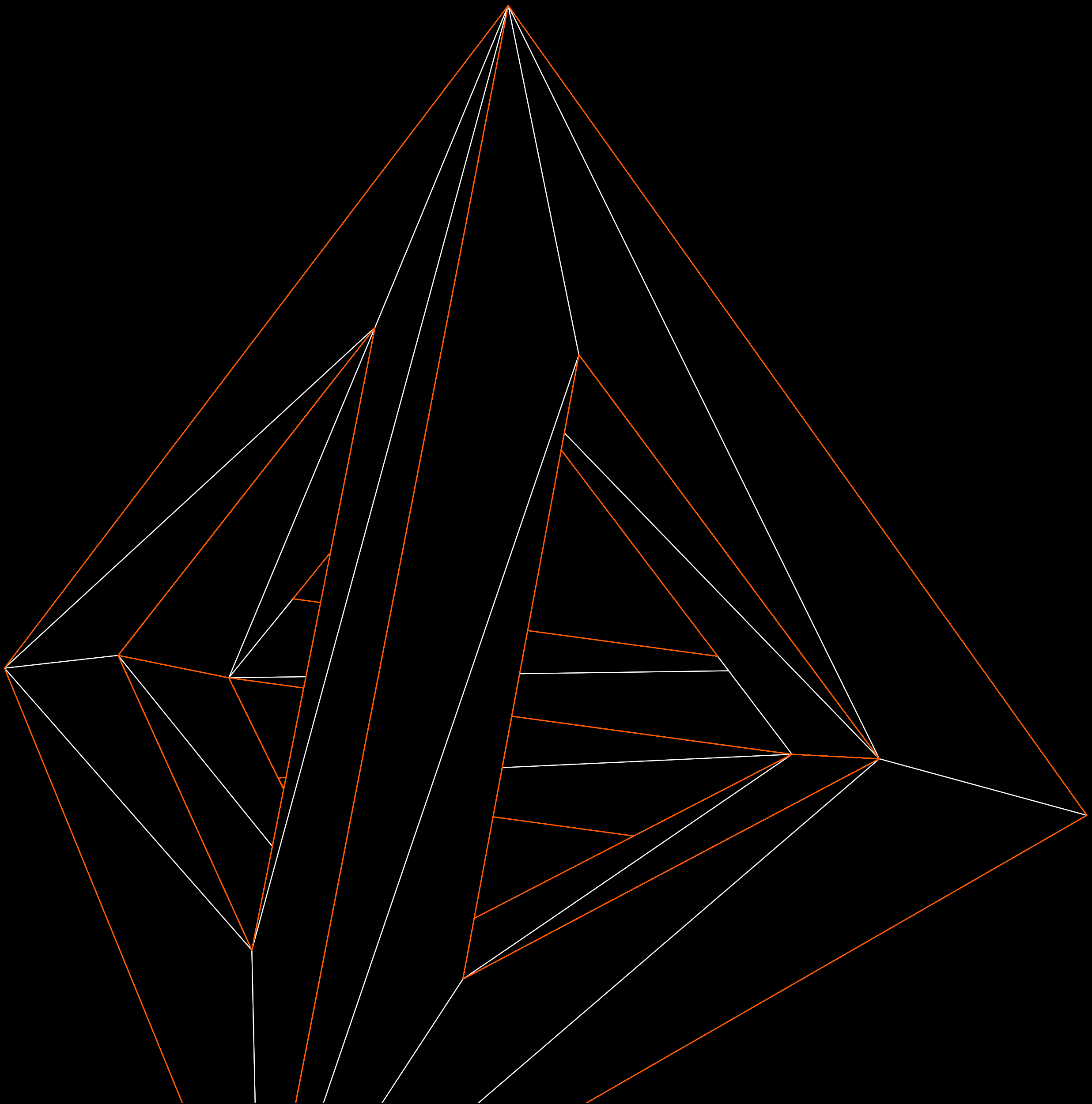


Table of Contents

Introduction

At Bugcrowd, one of our guiding principles is that adopters of our products are only as successful as the hackers and pentesters working on their security programs and the engagements that comprise them. There is shared interest across the Bugcrowd ecosystem in ensuring that processes, procedures, and communications are designed to **make hackers feel safe, valued, and motivated.** And we're proud to say that we've built the best triage experience in the industry to meet this goal.

For hackers specifically, **triage is the first and most critical point of contact with Bugcrowd.** It sets the tone for everything that follows, including rewards.

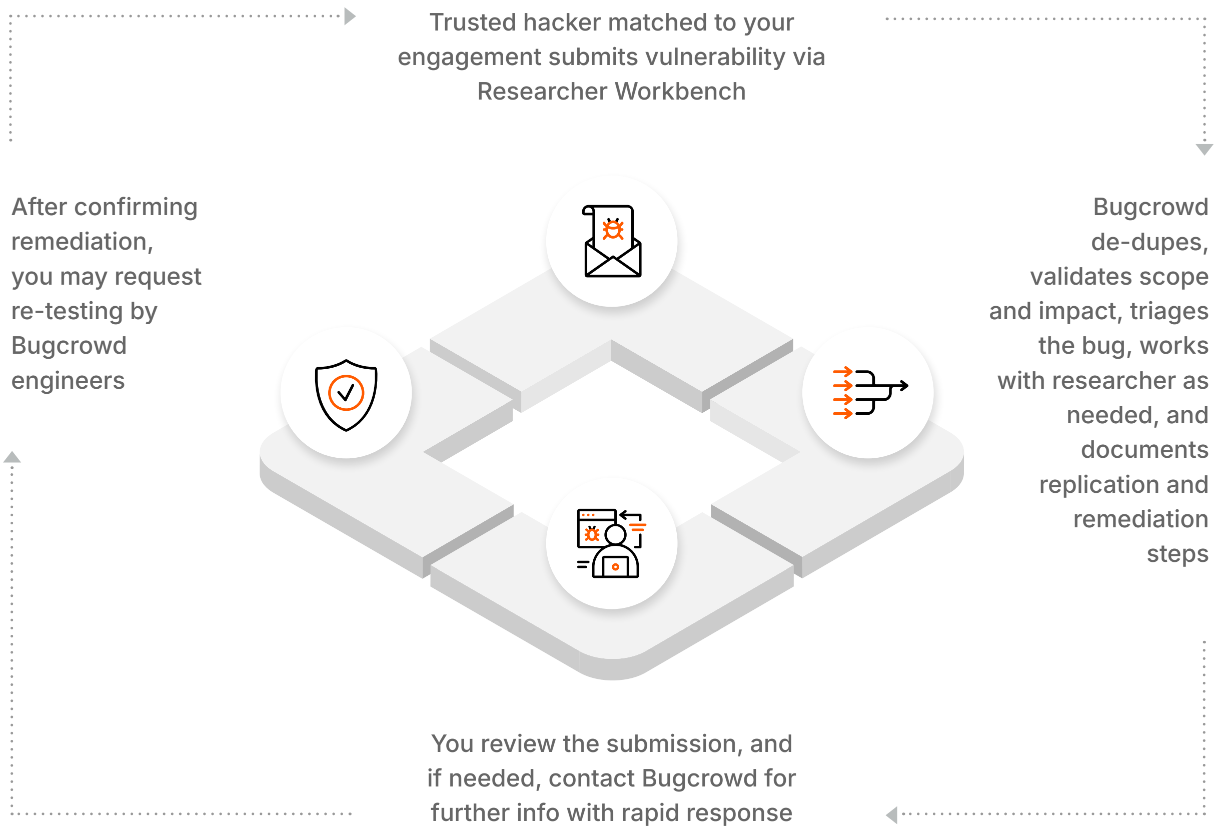
The Bugcrowd team works hard to get triage right (and has the track record to match), and in this guide, we will offer a preview of what to expect from our triage process and insights as to why it matters so much to Bugcrowd customers.

Triage process overview

First, let's outline the triage process itself. At its heart, triage is the **validation and prioritization** of findings between hackers and engagement owners.

At its heart, triage is the validation and prioritization of findings between hackers and program owners. We've invested years in making ***managed triage* a key built-in feature** of the Bugcrowd Platform. Not only have we worked hard to build a superb, globally distributed team of triage specialists, but we have also armed them with the best toolbox available to amplify their talents. These tools include AI models trained using unique data compiled from our 12+ years of experience managing thousands of crowdsourced security programs. That combination brings **speed, scale, and accuracy** to triage that the crowdsourced security industry has never seen before.

At a very high level, the workflow looks like the following (for an engagement owner):



Trusted hacker matched to your engagement submits vulnerability via Researcher Workbench

Bugcrowd de-dupes, validates scope and impact, triages the bug, works with researcher as needed, and documents replication and remediation steps

You review the submission, and if needed, contact Bugcrowd for further info with rapid response

After confirming remediation, you may request re-testing by Bugcrowd engineers

The triage process for **Bugcrowd’s triage team** looks like this:

1. Ensure the report is **in scope**
2. Ensure the report features a **valid vulnerability**

   E.g., it's not a functionality bug with no security impact
3. Check if the submission is a **duplicate**
4. **Replicate** the steps of the finding; if information is incomplete, work with the hacker to understand what’s missing to create simple, easy-to-follow replication steps
5. **Assess** the impact of the finding to:

   a. Guarantee that the reproducibility, impact, and significance of the issue are properly communicated to the customer via the submission

   b. Guarantee the appropriate priority of the finding is set in the platform
6. **Ensure the correct Vulnerability Rating Taxonomy (VRT) category** is selected for tracking and expectation setting
7. **Create an outline** for the program owner. Include the following:

   a. Reproducibility by Bugcrowd

   b. Any additional validation steps necessary
8. **Gather further information on impact** based on similar prior submissions
9. **Work with the hacker and the program owner** to align on outcomes

Now that you know the details of the triage process, we’ll cover Bugcrowd’s guiding values for the hacker/pentester experience, as well as how we build them into everything we do.

Bugcrowd values

To reiterate, we view **triage as the primary experience** for hackers and pentesters hunting on the Bugcrowd Platform, so it's critical that we get that experience right.

To help achieve this goal, everything Bugcrowd does during the triage process is based on these values:

Fair, impartial treatment for all

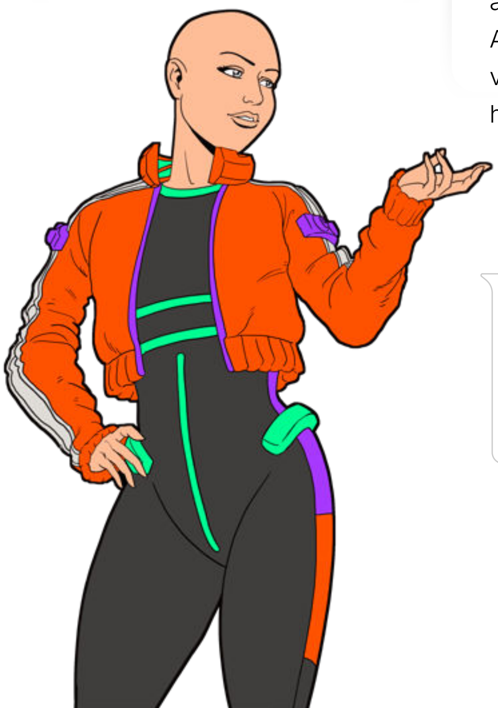
This includes customers/engagement owners, hackers, and Bugcrowd team members.

Focus on speed and accuracy

From a risk perspective, program owners and hackers alike want to see high-impact submissions triaged most quickly. Meeting this goal at scale requires a careful balancing of speed and accurate prioritization with AI augmentation adding value to workflows where it has the most impact.

Catching and learning from mistakes

Given the scale on which we operate, mistakes can happen. We strive to proactively identify, and ideally prevent, mistakes. When mistakes do occur, we acknowledge and correct them. Then, we do whatever is necessary to prevent them from happening again.



Let's break these principles down one by one, and look at how we implement them in the triage process.

Fair, impartial treatment for all

Three key aspects of the triage process help ensure equitable, respectful treatment for everyone in the Bugcrowd ecosystem: consistent hacker **experiences**, consistent bug and severity **classification**, and consistent **communications** and **outcomes**.

Consistent hacker experiences

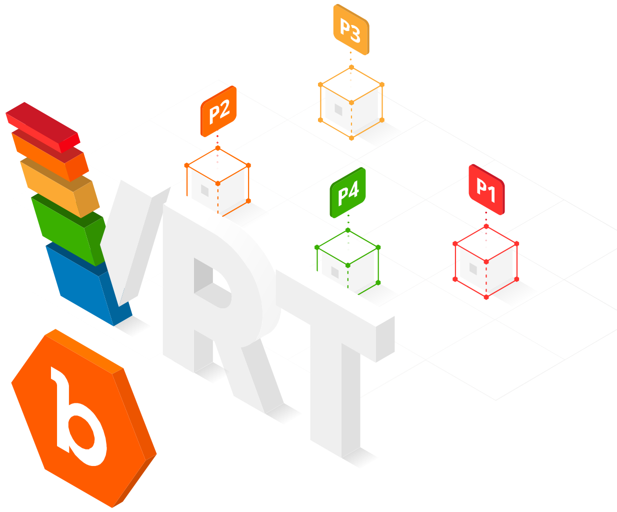
By hiring application security engineers—specifically former or current bug hunters—for in-house triage, we have intentionally built a team that fully, deeply understands the hacker experience. It uses that knowledge and awareness to advocate on behalf of hackers because our triage team has been where you are. Furthermore, we work hard to replace inconsistencies with repeatable outcomes on which hackers can rely.

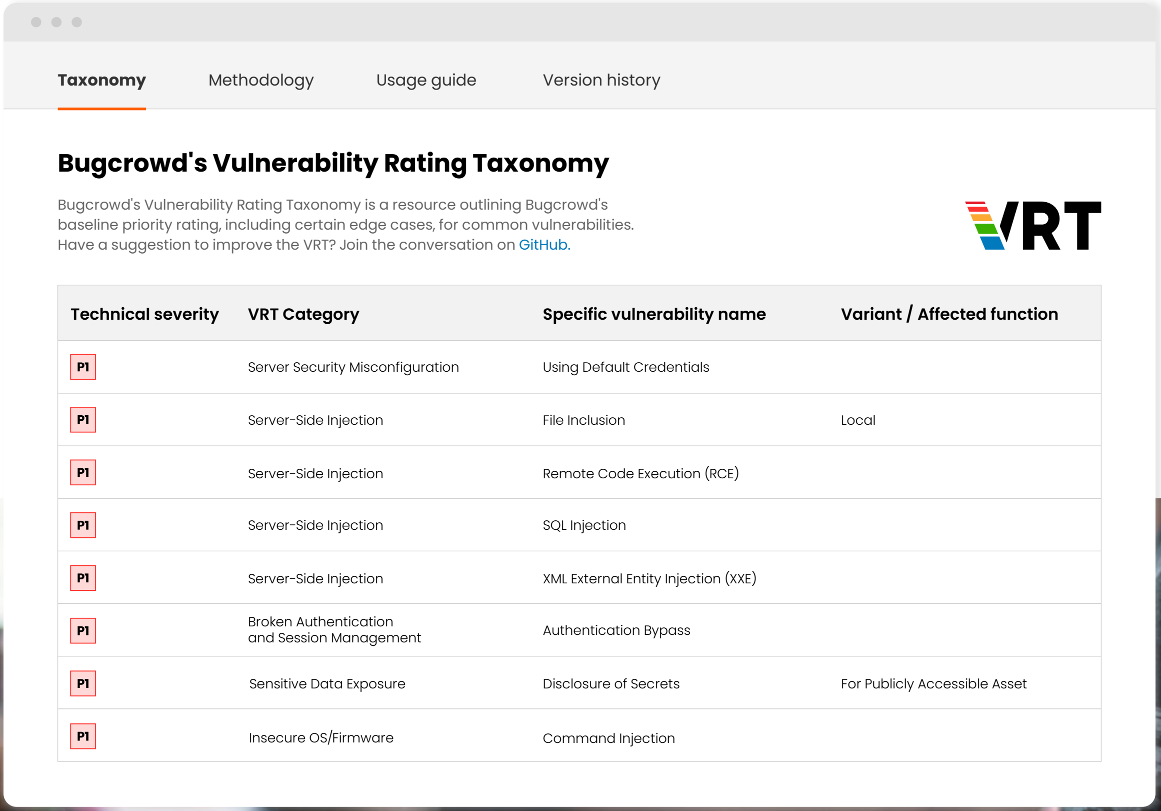
Another significant benefit of an in-house triage team is the **ability to amplify the impact** of members' work, **enabling speed, scale, and accuracy** beyond what any single organization can achieve on its own. That amplification is delivered by the Bugcrowd Platform itself, which arms our human specialists with guided workflows, standardized communications, and a rich security knowledge graph built on modern data infrastructure developed over more than a decade.

As a result, hackers (and program owners) can count on a consistent, predictable experience that includes rapid triage and **reward payments**—regardless of scale.

Consistent bug and severity classification

Another example of how standardization enables **respectful treatment** is our VRT. In the pre-VRT days, every program brief had a different definition of a “critical” issue—let alone the recognition of more nuanced issues. Despite the existence of tools like CVSS, frustration and confusion were common because briefs lacked context and specificity to a task. While disagreements are always possible, the **VRT provides a substantial (and open-source), shared severity taxonomy for everyone,** and with it, a foundation for consistent engagement between hackers and program owners.





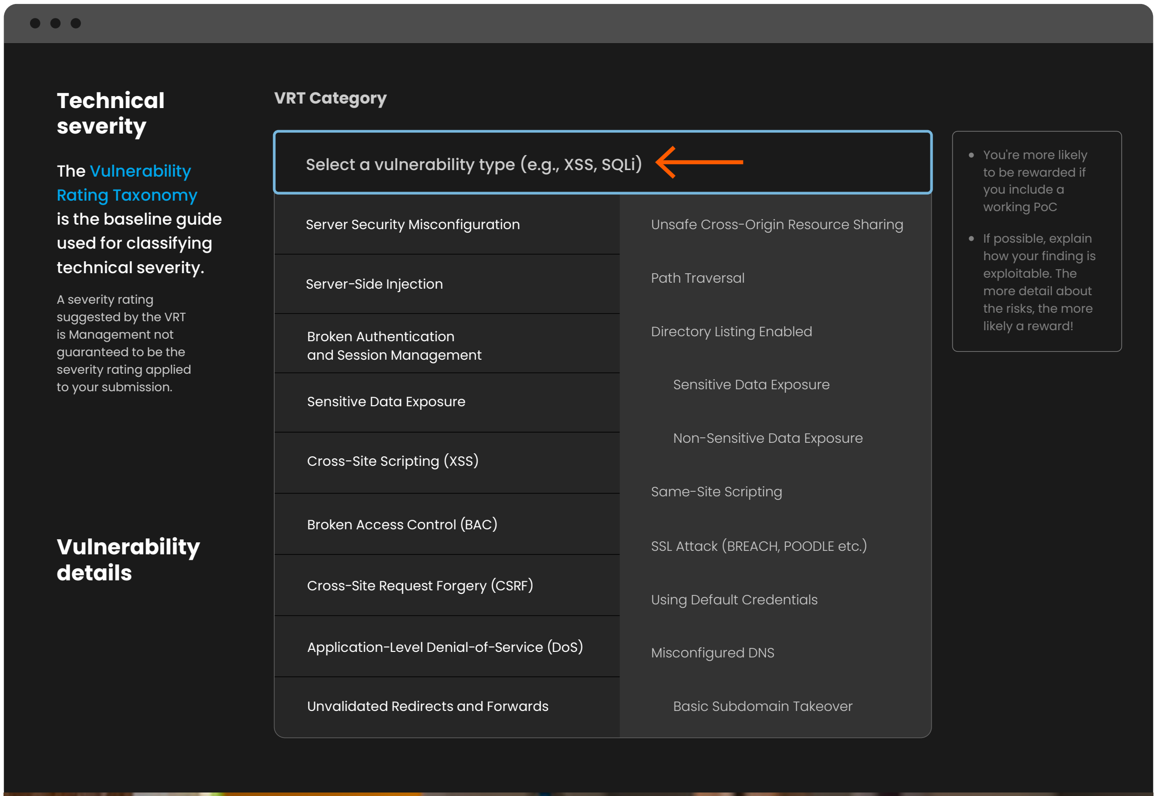
Bugcrowd's Vulnerability Rating Taxonomy

Bugcrowd's Vulnerability Rating Taxonomy is a resource outlining Bugcrowd's baseline priority rating, including certain edge cases, for common vulnerabilities. Have a suggestion to improve the VRT? Join the conversation on GitHub.

Technical severity	VRT Category	Specific vulnerability name	Variant / Affected function
P1	Server Security Misconfiguration	Using Default Credentials	
P1	Server-Side Injection	File Inclusion	Local
P1	Server-Side Injection	Remote Code Execution (RCE)	
P1	Server-Side Injection	SQL Injection	
P1	Server-Side Injection	XML External Entity Injection (XXE)	
P1	Broken Authentication and Session Management	Authentication Bypass	
P1	Sensitive Data Exposure	Disclosure of Secrets	For Publicly Accessible Asset
P1	Insecure OS/Firmware	Command Injection	

Consistent communications and outcomes

At the scale at which Bugcrowd operates, it's important to have guided workflows across the triage process. We've invested in ensuring prompt, clear, and detailed triage communication (read about our groundbreaking Request A Response feature here), and we arm hackers with tools that help them create high-quality submissions that reduce noise, such as Submission Templates. Additionally, we provide guidelines on how to safely demonstrate the impact behind findings. These efforts to help hackers create high-impact submissions make validation, triage, and payment happen more quickly.



Technical severity

The Vulnerability Rating Taxonomy is the baseline guide used for classifying technical severity.

A severity rating suggested by the VRT is Management not guaranteed to be the severity rating applied to your submission.

Vulnerability details

VRT Category

Select a vulnerability type (e.g., XSS, SQLi)

Server Security Misconfiguration
Server-Side Injection
Broken Authentication and Session Management
Sensitive Data Exposure
Cross-Site Scripting (XSS)
Broken Access Control (BAC)
Cross-Site Request Forgery (CSRF)
Application-Level Denial-of-Service (DoS)
Unvalidated Redirects and Forwards

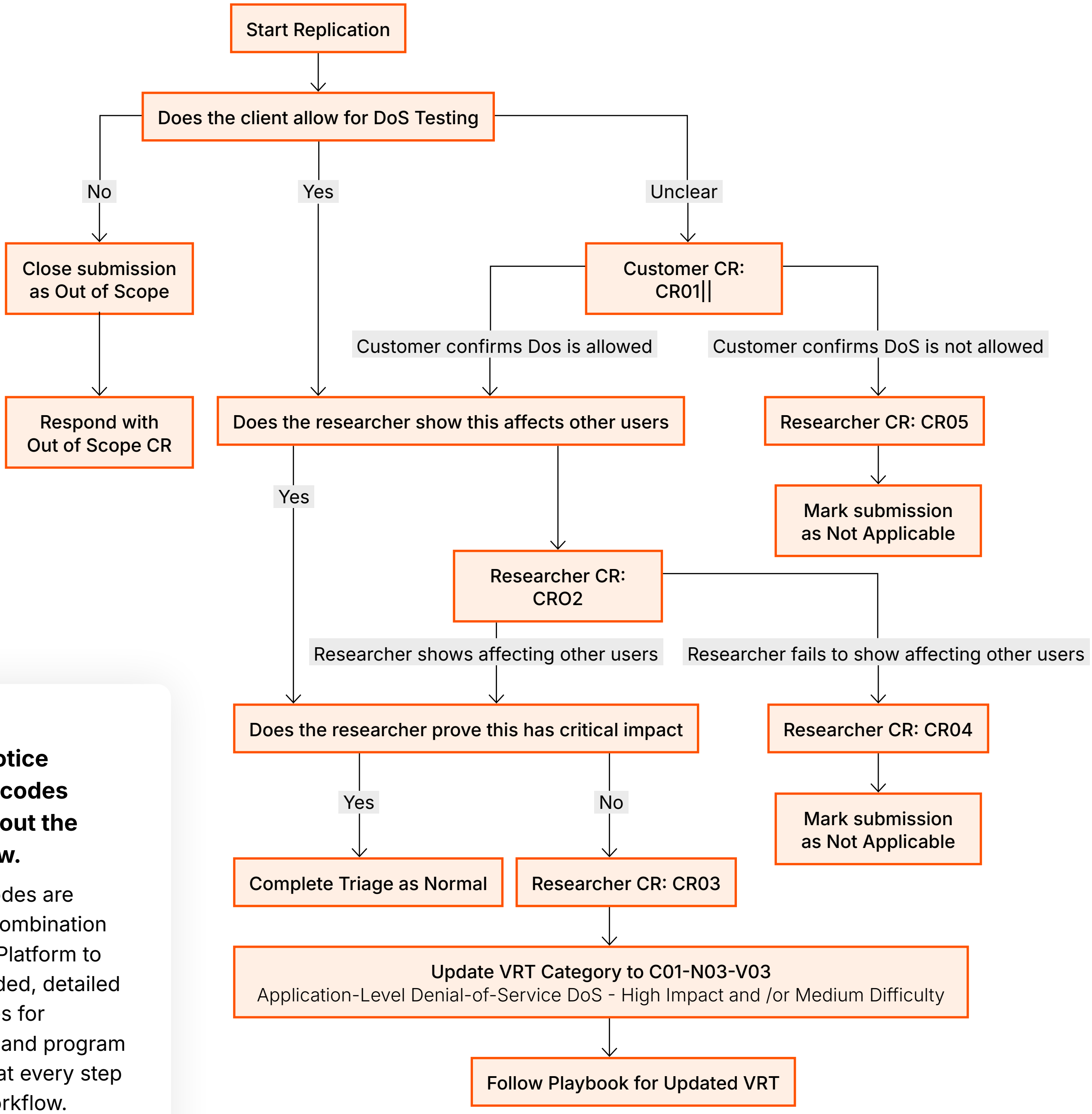
Unsafe Cross-Origin Resource Sharing
Path Traversal
Directory Listing Enabled
Sensitive Data Exposure
Non-Sensitive Data Exposure
Same-Site Scripting
SSL Attack (BREACH, POODLE etc.)
Using Default Credentials
Misconfigured DNS
Basic Subdomain Takeover

- You're more likely to be rewarded if you include a working PoC
- If possible, explain how your finding is exploitable. The more detail about the risks, the more likely a reward!

The Bugcrowd Platform gives our triage team process flows to follow for each item it works on.

This helps us build new templates that provide **consistent guides** for hacker and engagement owner outcomes. It also extends beyond the VRT, with **playbooks** for internal validation processes like de-duplication and customer-specific workflows such as validating against certain environments.

Here's an example:


Start Replication
Does the client allow for DoS Testing
No
Yes
Unclear
Close submission as Out of Scope
Customer CR: CR01||
Customer confirms Dos is allowed
Customer confirms DoS is not allowed
Respond with Out of Scope CR
Does the researcher show this affects other users
Researcher CR: CR05
Yes
Mark submission as Not Applicable
Researcher CR: CRO2
Researcher shows affecting other users
Researcher fails to show affecting other users
Does the researcher prove this has critical impact
Researcher CR: CR04
Yes
No
Mark submission as Not Applicable
Complete Triage as Normal
Researcher CR: CR03
Update VRT Category to C01-N03-V03
Application-Level Denial-of-Service DoS - High Impact and /or Medium Difficulty
Follow Playbook for Updated VRT

You'll notice various codes throughout the workflow.

Those codes are used in combination with our Platform to build guided, detailed responses for hackers (and program owners) at every step of the workflow.

Focus on speed and accuracy

It’s no secret that the Bugcrowd Platform offers the fastest triage services in the industry–with time-to-first touch usually within a single business day.

The global Log4j incident is an example of how investing in engineered triage as a core competency helps us deliver good customer and hacker outcomes (rapid remediation and rapid payouts, respectively) as quickly as possible, **even during massive incidents.**

→

Next, we’ll cover your expectations regarding timelines.

Time to touch

Few things are more frustrating for hackers than reporting a bug, only to feel like it was lost to the void. Hackers deserve to know when their submissions will be actioned upon, so predictable and consistent timelines are a key goal for Bugcrowd.

Because our triage team takes the first pass at every submission that arrives through the Platform, we're able to commit to specific expectations. Thus, we hit our service level objectives (SLOs), shown below, **99+% of the time:**

P1	FIRST TOUCH	1 day	Defined as from the moment a P1 finding is submitted to the Platform, to when the report is first evaluated by an Application Security Engineer**
P1	TRIAGE	2 days	Assuming no blockers or need for additional information
P2 ▸ P5	FIRST TOUCH	3 days	With the exception of a handful of programs that do self-triage for logistical or legal reasons. We monitor such programs closely and hold them to the same SLOs to which we hold ourselves
P2 ▸ P5	TRIAGE	7 days	It may take multiple interactions or days to get to the root of the issue. This does not include time spent on blockers from hackers or program owners

Business day*

Most tickets are triaged at first touch, but if there's a need for additional information, our timeline steps in. **This way, you know exactly what to expect from us within a given time frame.**

Premium service levels/SLAs are also available for customers who need them.

See the data sheet for details

AI-augmented triage

We complement our world-class triage team service with powerful AI tools that enhance outcomes behind the scenes. These tools help us rapidly validate and prioritize submissions at scale. For instance, we've deployed a model that proactively identifies and filters spam before it reaches your inbox. Additionally, we are testing new models for identifying out-of-scope issues, detecting duplicates, assigning VRT categories and severity level, and providing initial remediation advice.

* Bugcrowd's business hours are 9am-5pm PST, Monday through Friday, holidays excluded. Bugcrowd triage teams are also located in EU and APAC time zones, facilitating 24/7 tracking and rapid local response. **One-day triage isn't guaranteed if additional information is necessary.

Treatment of aging submissions

No discussion on time to first touch would be complete without a **conversation** on the program owner’s role post-triage.

After a submission has been validated and confirmed as in-scope, unique, and replicable, it is transferred from Bugcrowd to the owner for acceptance, payment approval, and remediation. Because this part of the process can take some additional time, we’re highly focused on streamlining it.

The fastest path to a program's success is for its owner to accept and reward applicable triaged reports within two weeks of them being validated by Bugcrowd, and 90% of programs operate within this timeline. We consider any submissions that sit for more than two weeks to be “aging” and address them promptly via a dedicated ticketing instance that assigns responsibility to the program’s Technical Customer Success Manager (TCSM). It’s the TCSM’s job to ensure all aging submissions are escalated to the program owner and actioned upon. In extenuating circumstances, we will pause the program and notify the affected parties.

There are several reasons a program owner might be slow to respond, but our process works hard to speed things up as much as possible and get hackers paid. We make a targeted push to go after aging subs and are continuously working to reduce the number of findings that fall into that category.

If a submission is tagged as “aging,” rest assured that, as soon as it crosses the two-week threshold, a ticket has already been created and we’re already working on getting it across the line. However, if you have any questions or are looking for updates, reach out to our Support team.

Making specializations scale

Understandably, not all programs are alike. With these differences in mind, to **keep timelines short and accuracy high,** we have specialized resources for addressing specific requirements, such as exotic asset types (e.g., IoT devices and Web3 assets), and automated processes to ensure they're assigned to the right things quickly.

Over time, as we learn more about a program, we give it a categorization. In some cases, this may mean a dedicated application security engineer who maintains virtual machines and other supporting infrastructure. In other cases, it may mean assigning a dedicated group of engineers who specialize in the technology associated with that engagement.

This is all supported by a workflow engine in the Bugcrowd Platform that allows us to assign triage specialists to groups and then assign those groups to relevant engagements. Assignments are based on the composition of groups, the priority of submissions within those groups, and the SLOs or SLAs associated with them. As a result, the right engineers are assigned to the right issues automatically, not through some manual, subjective process. This leads to shorter triage time and improved accuracy.

Outside of program-specific specializations, we also offer program owners priority-specific SLAs. For example, on weekends, a dedicated team focuses on tracking critical issues (P1s) to shorten timelines as much as possible. This group also helps us to respond to significant industry-wide security incidents, such as the Log4j vulnerability, and to develop new guided processes and playbooks.

Catching and learning from mistakes

Consistency, speed, and accuracy are all fine and good, but catching, acknowledging, and learning from mistakes are values on which customers and hackers won't compromise.

The Bugcrowd engineered triage process is designed to reflect that concern through rigorous quality assurance, organizational learning, and support for appeals.

Quality assurance

Quality assurance is a major ingredient in the triage process. Each week, we pressure test a representative sample of submissions with a series of quality assurance checks that account for technical accuracy and communications with hackers and program owners, among other things. These scores are aggregated, shared across the team, and analyzed over time to identify opportunities for improvement.

In cases where a submission receives a low assurance score, we'll sample more widely to see what kinds of changes need to be made to prevent the same mistake. These changes can include the following:

- **Staff training**
- **Establishing or improving a program-specific playbook**
- **Updating notes specific to a program**
- **Overhauling a specific triage process or submission category playbook**

In summary, we emphasize not only catching mistakes but also preventing them in the future.

Appeals

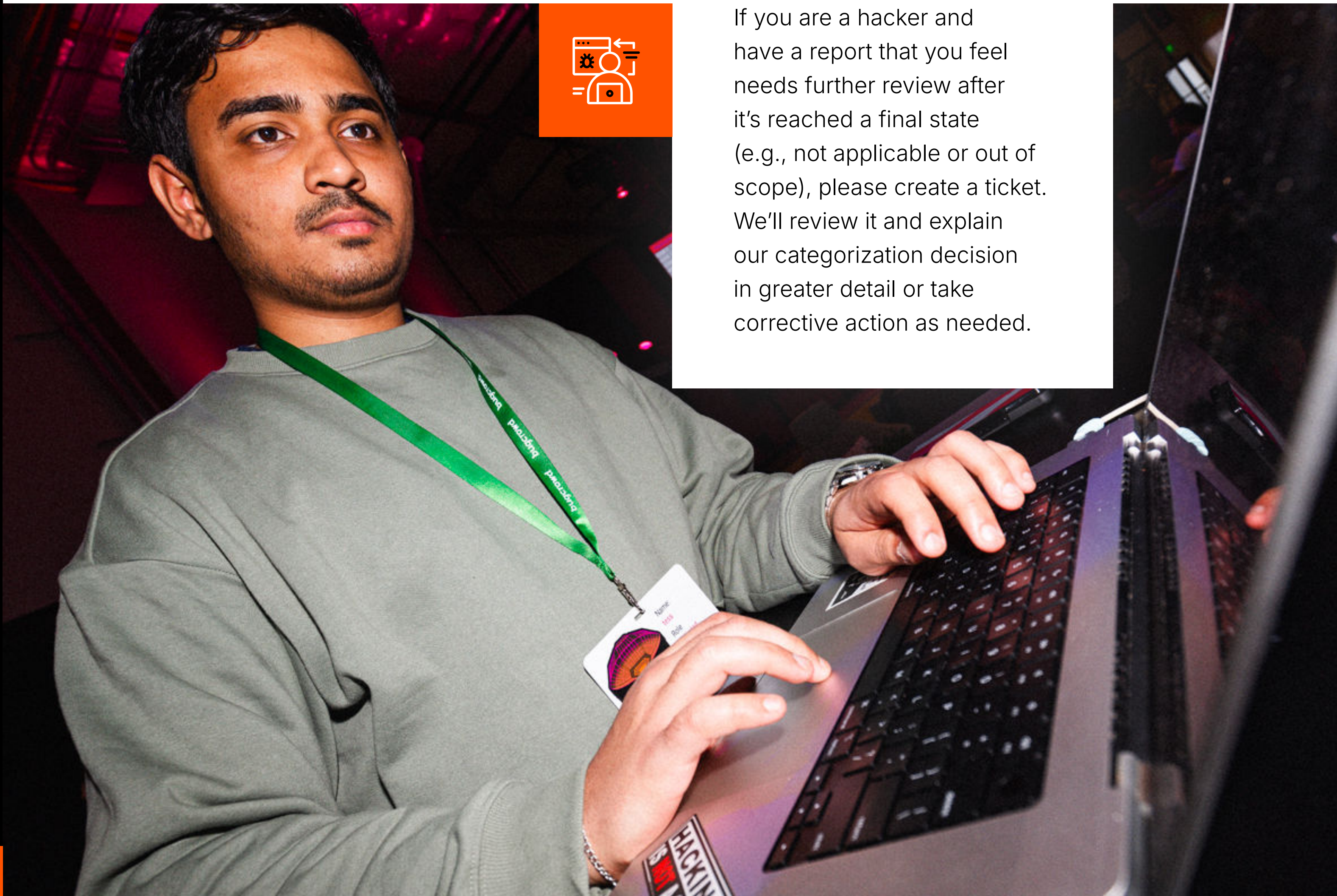
As you've gathered by now, there is nothing that Bugcrowd values more highly than a **good hacker experience.**

And part of that experience is the expectation that Bugcrowd will advocate for hackers with program owners when necessary, as well as acknowledge and correct our own mistakes.

For example, when we do make a mistake–such as erroneously marking a submission as NA when a subsequent report of the same issue results in a payout–we'll make it right by paying the hacker the expected reward out of our "Make It Right" Fund.

However, correcting past mistakes is just table stakes. We believe that the industry as a whole has approached hacker appeals the wrong way–as backward-looking damage control. At Bugcrowd, we strongly believe that many appeals can be prevented in the first place by listening more carefully and communicating more clearly.

If you are a hacker and have a report that you feel needs further review after it's reached a final state (e.g., not applicable or out of scope), please create a ticket. We'll review it and explain our categorization decision in greater detail or take corrective action as needed.



Why triage matters for Bugcrowd customers

Elevated triage doesn't just provide value for hackers; it is **immensely valuable** for customers. Our **global team** of in-house triage engineers consists of hackers with specializations in web apps, mobile apps, APIs, hardware/IoT, AI, and crypto.

This team quickly triages and validates all findings to ensure you get critical insights as fast as possible. Ultimately, vulnerabilities that aren't dealt with are massive risks for organizations. Our triage team is dedicated to making sure you can find and fix these vulnerabilities faster than threat actors can spot them.

Getting help

bugcrowd

We hope that you've learned what you wanted to know about Bugcrowd's managed triage process. **Should you ever run into any issues with triage on the Bugcrowd Platform, the Bugcrowd Support team is here to help!**

Be sure to contact us if you have any questions or issues with any of the following:

- ✓ General inquiries regarding the Bugcrowd Platform
- ✓ Dispute issues
- ✓ Platform access
- ✓ Program scope clarifications
- ✓ Credentials or payments
- ✓ Requests for submission comment response or appeals
- ✓ Technical support inquiries