4 bug bounty mistakes and how to avoid them

BY TRAVISINTIGRITI · APRIL 17, 2024 · LAST UPDATED ON MARCH 6, 2025

Getting into bug bounties is no easy task, we know. There's so much to consider and your path to becoming a bug bounty hunter can vary in so many ways. Bug bounty hunting can be fraught with challenges, and even the most skilled individuals can fall victim to common mistakes.

1. Striking the wrong balance: learning vs doing

Okay, we're starting with a hard hitter here because this is a tricky one to balance. **How much is too much**? **How much is too little**? We think new bug hunters should ask themselves these questions often.

A lot of people spend their days **constantly** learning new techniques, completing labs or CTFs, making notes, and other awesome cybersecurity activities. To be clear, these are amazing, you should 100% not stop doing these things, but if your goal is to become a bug bounty hunter – ***get hunting before you burn out!***

Lab environments and CTFs do their best to mimic the real world, but nothing will be more real than… real! So maybe try to interlace some real hunting alongside some learning and make sure you don't lose your touch for the real world when you're popping shells for flag.txts!

On the other side of the coin, we do see a lot of people focusing on just a few common areas of bug hunting and instantly start trying to find bugs with limited knowledge. For example, people are often well-versed in XSS, SSRF, XXE, path traversal, etc as they're quite common and easy to learn. But learning about cache-related bugs, Android bugs or electron vulnerabilities, will give you an edge over other people who are putting all their energy into XSS on a target's main web app. ***Widening your net means more fish for you***. But of course, if you're a champion shark hunter, go deep too.

2. Reliance on common automation

Automation can be great and certainly has its place in bug bounties. Some of the biggest and most successful bug hunters in the world made their success using automation to find vulnerabilities. Arguably, in some instances (especially for recon) automation is pretty much a requirement.

The key word here is "reliance". Automation should be just another tool in your toolbox. Bug hunters of the distant past often tell stories about the successes that automation brought them back in the early days of bug bounties, but since then things have changed. Here's why:

- The industry has matured, meaning less low-hanging fruit is available for the taking by common public automation.
- The industry has grown massively~~,~~ to a point where automated traffic can become troublesome to companies, so rate-limiting rules often apply.

- Seasoned veterans have become experts at streamlining automation to a level that most common automation tools can't keep up with.
- WAFs are becoming commonplace, limiting the impact of automation and potentially becoming a barrier between you and a bounty.

On top of this, automated findings are by far the most likely to get duplicates when making a report. Learning how to manually hunt for bugs instead of running a suite of common tools such as nuclei or SQLmap over your targets is a great way to level up as a bug hunter.

3. Failing to consider impact

Impact is everything when it comes to bug bounties. A common mistake new bug hunters make is assuming that the bug type is what changes the impact. For example, clickjacking = low right? How much harm can one click do? And what about SSRF? High! That's what other reports show, so yours should be too right?

Not always. We need to consider the business impact of the bug found. Just because you can prove a scenario to be vulnerable doesn't mean it's automatically going to get triaged at the impact you expected.

For the examples above, clickjacking is often reported and given a $0 bounty, but in some cases such as with metamask UGWST was able to claim $120,000 for a single clickjacking vulnerability!

And as for SSRF, there are circumstances where a found SSRF just isn't of any actual use. If you're inside a tightly contained system, the SSRF may not provide any actual impact to the business at all. In some cases, it could even be considered a product feature.

So next time you want to make a bug report, put yourself in the shoes of the company and ask yourself "Does this have a business impact?".

4. Not understanding your target

Learning your target should be step 1 when starting on a target in most cases. Getting to know your target inside and out will help to understand what is important to the company, what key areas they're focussing on and just understand how the application works.

Understanding what features exist, how they are created, and spotting common patterns or technologies can all be used to help you in your bug-hunting journey. This information will allow you to get in the minds of the people working for that company and this will in turn increase your ability to find bugs.

In addition, understanding your target will help you understand what is or is not a vulnerability. For example, many web apps might have some sort of points system and viewing those points for another user may not be important, or even intended. However, an app like a loyalty app could give insights into certain spending habits of a victim or other components that may cover how those points are allocated, to begin with. Without knowing exactly what the points system is, how points are allocated and why the company deems it important, you might not consider leaking a user's points to be an actual security-impacting issue.

Final thoughts: tread your own path to success

That wraps up the most common bug bounty mistakes that we see! Before we part ways, it's important to mention that there are exceptions to every point mentioned above. This only serves as a guide based on what we see in the community and what helps a wide number of people. But if you feel you're gaining success the way you are going, don't worry about switching it up to join the masses. However, if you're struggling, give some of the tips above a thought and see how you can change your workflow and hopefully push you further into success!

Want to keep learning? Then head on over to our recent article where we unveil the most common ways of finding security vulnerabilities in static websites.

REQUEST A DEMO

intigriti.com/demo

VISIT THE WEBSITE

intigriti.com

GET IN TOUCH

hello@intigriti.com