All this text wasn’t written by me (Tiago a.k.a TigaxMT).

All of it was transcribed by me from HackerOne video lessons.

All credits and thanks go to them.

Writing Good Reports

WHY?

If you're a bug bounty hunter and you're finding great bugs, why does it matter whether or not you write a good report?
At the end of the day, is there anything more important than the technical quality of your work?

Your goal with a report should be to give the most useful information possible to the product team. This means they can triage and confirm your bug faster; this gets the money into your pocket even sooner.
It also leads to fewer questions from the team, making everyone's lives easier.

WHAT IS A GOOD REPORT?

- Clear description of the bug
- Real-world impact
- Concise reproduction steps
- Working Examples
  - Proof of concept links/payloads
  - Screenshots
  - Source code snippets

DESCRIPTION – BAD

"When submitting feedback, the title tag value isn't escaped, allowing XSS attacks."

Problems:
- Where does the title come from?
- What privileges are required to execute this attack?
- Which page(s) are actually affected?

DESCRIPTION – GOOD

"Within the administration panel of the site, administrators are able to set the default value for the `<title>` tag, to which page names are prepended. On the "Submit Feedback" page (https://example.com/portal/feedback), this value is not escaped properly. This means that an attacker with administrative privileges can inject arbitrary HTML into a page, thus effecting a cross-site scripting attack.

1. Log into application as an administrator
2. Go to https://example.com/admin/settings
3. Set the page title field to `</title><script>alert(document.cookie);</script>`

4. Go to https://example.com/portal/feedback
5. Note the alert dialog that occurs

IMPACT – BAD

“Attackers can add any HTML to a page, which is cross-site scripting.”

Problems:

- What can an attacker accomplish with this?
- Does “a page” mean a specific page? If so, which?
- What does the attack flow look like in practice?

IMPACT – GOOD

“An attacker is able to execute JavaScript in the context of the “Submit Feedback” page. This code is able to perform any action that the victim could ordinarily perform, e.g. making posts and sending messages. As this page is accessible to all users and is clicked commonly, this may allow an attacker to compromise a large number of users without any new interaction being forced.”

SCREENSHOTS – BAD

This doesn’t tell you anything about:

- The affected page
- Parameters/fields affected

SCREENSHOTS – GOOD

This shows the URL, affected parameter, and the cookies readable from the browser.

DETERMINING THE IMPACT

When it comes to determining the impact of a vulnerability, the key thing is thinking like an attacker.
What things are important to the business behind the application? If you can figure out what's important to the business, you can figure out what an attacker would be targeting and thus how your vulnerability can be used for that.

When you're talking about the impact of a bug, talking about the technical details misses the point. Businesses don't care about SQL injection, they care about user information being accessed or destroyed. They don't care about cross-site scripting, they care about fraudulent orders.

SOFT SKILLS

Once you've started finding bugs, the single biggest thing you can do to increase bounty payouts is to work on your soft skills. Do you have issues with spelling or grammar? There are a plenty of resources online to work on those. Have your reports been closed N/A when you think they shouldn't be? Maybe you're not presenting it to the team in a useful and actionable way.

READ!

On HackerOne you can find thousands of reports from real hackers. These hackers range from absolute beginners to the best in the business, but you can learn from every one of them. If you read a lot of reports, you'll notice the things that make them particularly good or bad. Use those to make your own reports better.