Business logic flaws

What is it?

Business logic flaws often arise from different situations. They occur when users pass values to the target which are not expected. This can cause several unforseen things to occur. These things might not always be as impactful but sometimes they can be devistating. The analysts make assumptions about use behavior but these can be wrong. This will lead to flaws in the design and the implementation of the logic.

The reason i love Logic flaws so much is because they are really hard to look for. Normal use of the application will not always show these and we have to specifically look for them. This makes it nearly impossibly for automated tools to be created that will find logic vulnerabilities on a consistant basis.

Business process usually consists of:

```
Analysis > Development > Testing > Production
```

I know this process is very simplified but it's not important to know the details right now. Usually several "Stories" or "Features" get taken up into a release cycle. If the analysis from the start contains logic flaws, this is an entry point for our attack.

What also sometimes happens is that a piece of software might be developed and a couple of years later an expansion or adaptation might be requested. Usually documentation is a big problem in companies so when a change has to be made to a certain feature, the developers will have to dig into the code where they might remove certain important features that gaurd the sanity of the users actions.

What is the impact?

The impact is highly dependant on the specific target and logic flaw that you found. It is related to the impacted functionality as well.

- Client side calculations of prices in a clothing webshop - High/Crit
  - This is core business for the target so any issue related to the core business will automatically be more impactful
- When brute forcing usernames, you get a 200 OK status when the username you are trying to brute force exists and a 403 if it does not exist on the login page - Low
  - This is rather low unless those usernames really have to be secret, you'd have to brute force the login names and then you have to still guess the correct password. This is more usefull on a pentest job.
- Negative amounts of items on a webshop lead to negative prices - High/Crit
  - This is core business for the target so any issue related to the core business will automatically be more impactful above all, impact on money directly is very important
- If price = integer and amount = integer and total price = interger we can overflow total price when we price * amount - Critical
  - This might lead to the target returning us money which is certainly not desireable
- Registering with the same username as an existing user takes over the account - Critical
  - Account takeovers are always higher on the severity scale
- The user manual might tell you that you can't deactivate super admin users, but after trying it you can - Medium
  - You have to be a priviledged user to even be allowed into the user management system so this lowers the severity a bit
- Field in the response that's not in the original request but does get processed by the server when you add it - Nothing/Critical

    - This really depends on the fields that is being processed here. If you can change your accounttype from "User" to "Admin" This would ofcourse be a big problem.
- Importing products with the same name as existing ones overwrites them. Even if the products do not belong to you and you should not be able to overwrite them. - Medium
    - You are already in a priviledged position before you can import products, this lowers the severity