[Skip to main content](https://lms.alnafi.com/xblock/block-v1:alnafi+DCCS102+2025_DCCS+type@vertical+block@1e424eed0cd94c90bf646218c370ea31?exam_access=&recheck_access=1&show_bookmark=0&show_title=0&view=student_view#main)

**Testing for Improper Error Handling**

Errors will be generated by all types of applications (web apps, web servers, databases, and so on) for a variety of reasons. Developers frequently ignore handling these errors or dismiss the possibility that a user will intentionally cause an error (e.g. sending a string where an integer is expected). When a developer-only considers the happy path, he or she overlooks all other possible user input that the code can receive but cannot handle. Errors can occur when:

1. Stack traces,
2. network timeouts,
3. input mismatches,
4. and memory dumps are all examples of errors.

Incorrect error handling can enable attackers to:

1. Learn about the APIs that are used internally.
2. Map the various services that are integrating by learning about the internal systems and frameworks that are being used, which opens the door to attack chaining.
3. Compile a list of the application versions and types that are in use.
4. DoS the system by causing a deadlock or an unhandled exception that sends a panic signal to the engine that is running it.
5. Controls bypass where a specific exception is not restricted by the happy path logic.

Please click on this URL to perform this lab <http://wstg.alnafi.com/>

Here’s an example of improper error handling.

