



The Use of Secure Design Patterns in Object-Oriented Programming

[Initial Plan]

I. Topic & Research Question

As mentioned in the Portfolio task, I feel that Design Patterns and Coding Design are the field that I interested in the most. So for the initial plan of this task, my chosen topic are: “The Use of Secure Design Patterns in OOP”.

Here are my 3 questions that I’ll try to answer at the end of this research project:

1. What types of security vulnerabilities can be mitigated through the use of common OOP design patterns (e.g. Singleton, Proxy, Factory, etc.)?
2. Which specific design patterns are most effective in addressing common software security concerns such as input validation, authentication, and access control?
3. How do secure design patterns compare to insecure or “anti-patterns” in terms of reducing security risks in object-oriented applications?

The aim of these questions is to understand how the design patterns are used in the cyber security field and all the aspects of it (strength and weakness)

II. The Research Method

Following the instruction in the Canvas file, I’ll propose 3 of my initial research methods (this is just my plans and I might need to change somethings in the final research):

1. **Experimental Comparision:** With vs Without Security Patterns
By implementing two versions: one using none of the security-focused design and the one use appropriate patterns like Proxy, Singleton, Factory, etc.
2. **Pattern Identification and Analysis in Open-Source projects:**
To examine how secure are these design patterns used in real-world programs and apps.
3. **Comparision of Secure Patterns and Anti-Patterns:**
Evaluate the security risks of anti-patterns to gain the whole picture of security-oriented design patterns.

In order to collect data to make conclusions about research methods above, I’ll initially use: Pattern evaluation rubric, existed research papers and manual comparision (using tools like IDE, Code Analysis)

III. How This Related To Unit Learning Outcomes

I think this short research project will explore how secure design-patterns can reduce risks in object-oriented applications, directly supporting the unit's learning outcomes. It promotes understanding of **abstraction**, **encapsulation**, **inheritance**, and **polymorphism** by comparing, reading and hand-on coding examples. This project involves using C# and class libraries to build working programs that reflect real security aspects. It also requires myself to spend time to read other resources (papers, peer-review research, videos, etc) to understand more about the chosen topic.

The process of designing, testing, and debugging both secure and insecure versions supports skill development in using appropriate coding analysis tools. The project also includes matrix rubrics and written comparisons to clearly communicate how structure and behavior change when patterns are applied. Finally, by reflecting on anti-patterns versus secure practices, the project encourages a deeper understanding of what makes a good object-oriented solution in real-world cases.