Jason Luttrell

CSD 370

Module 4.2: **Secure Design Principles**

A comparison of Three Sets of Secure Design Principles

I compared the Secure Design Principles (SDPs) in the textbook *CSSLP Certified Secure Software Lifecycle Professional All-in-One Exam Guide* with a similar set published by the University of California, Berkeley, and a third set from OWASP.

The textbook’s and Berkeley’s lists are shown in the table below. As can be seen, they take different approaches and do not align directly. The textbook presents higher-level concepts, design heuristics, and strategies. In contrast, Berkeley’s list is more specific and focused on implementation topics. There is some overlap—for example, *Error Handling and Logging* from Berkeley is mostly synonymous with the textbook’s *Fail Safe*. However, overall, the two sets categorize practices quite differently.

For learning about SDPs, the textbook list is effective. However, Berkeley’s list seems better suited as a rubric for implementation.

|  |  |
| --- | --- |
| Textbook | Berkeley Secure Coding Guidelines |
| Good Enough Security | Input Validation |
| Least Privilege | Output Encoding |
| Separation of Duties | Authentication and Password Management |
| Defense in Depth | Session Management |
| Fail Safe | Access Control |
| Economy of Mechanism | Cryptographic Practices |
| Complete Mediation | Error Handling and Logging |
| Open Design | Data Protection |
| Least Common Mechanism | Communication Security |
| Weakest Link | System Configuration |
| Leverage Existing Components | Database Security |
| Single Point of Failure | File Management |

# 

The third list I reviewed is the OWASP *Secure Coding Practices Checklist*. This resource is exhaustive, with 14 sections and 243 items—it leaves no stone unturned. As I read through it, I recognized many items I’ve encountered during Fortify scans at work. In a typical development environment, the OWASP checklist might seem like overkill. However, since I work on classified systems, this checklist—though daunting—might be exactly what I need. It would simplify security scans before and after merges if our team implemented these practices during development, rather than retroactively. It would also help us meet our sprint commitments, since security findings are rarely accounted for in sprint planning.

**My pick is the OWASP list**, because the systems I work on require the highest security standards available.

References

Conklin, W. A., & Shoemaker, D. P. (2022). *CSSLP Certified Secure Software Lifecycle Professional All-in-One Exam Guide, Third Edition.* New York: McGraw Hill.

OWASP. (2024). *Secure Coding Practices Checklist*. Retrieved from owasp.org: https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/stable-en/02-checklist/05-checklist

UC Berkely. (2025). *Secure Coding Practice Guidelines*. Retrieved from UC Berkely Information Security Office: https://security.berkeley.edu/secure-coding-practice-guidelines