SC.L2-3.13.2

System & Communications Protection

Security Engineering

"Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems."

Objectives

[a]
Architectural designs that promote effective information security are identified.

[b]
Software development techniques that promote effective information security are identified.

[c]
Systems engineering principles that promote effective information security are identified.

[d]
Identified architectural designs that promote effective information security are employed.

[e]
Identified software development techniques that promote effective information security are employed.

[f]
Identified systems engineering principles that promote effective information security are employed.

Key Discussion Points

Identify Then Use:
Both steps are required — [a,b,c] require identifying the designs and principles; [d,e,f] require actually using them. Documentation without application fails.

Layered Protections:
Defense in depth, least privilege, and separation of duties are engineering principles — applied through architecture, not just policy.

Covers the Full Lifecycle:
Security engineering applies from initial design through upgrades and modifications — not just at deployment. Legacy systems must be reviewed when modified.

Threat Modeling:
Identifying attack vectors during design is a core security engineering technique — it produces security requirements that drive implementation decisions.

Assessment Methods

EXAMINE

Security planning policy; enterprise architecture documentation; system security plan; system and communications protection policy; procedures addressing security engineering principles; security architecture documentation; security requirements and specifications; system design documentation; system configuration settings.

INTERVIEW

Personnel with responsibility for determining system security requirements; personnel with system design, development, and modification responsibilities; personnel with security planning responsibilities; personnel with information security responsibilities.

TEST

Processes for applying security engineering principles in system specification, design, development, implementation, and modification; automated mechanisms supporting the application of security engineering principles.

SC.L2-3.13.2

SYSTEM & COMMS PROTECTION — Security Engineering

Plain English

What this control is really saying:

Security is easier to build in than to bolt on. This control requires that security be considered at the design and architecture level — not just through tools and policies applied after the fact. Identify the designs, coding practices, and engineering principles you will use, and then actually use them consistently across your systems and their lifecycle.

How it is used:

- The organization maintains a documented security architecture that defines the layered defense approach — network segmentation, least privilege, defense in depth — and references it during system upgrades.
- Developers follow a secure coding standard (e.g., OWASP Top 10 mitigations) and complete secure development training before working on systems that process CUI.
- New systems and major upgrades include a threat modeling exercise during design — identified threats drive security requirements that are tracked through implementation.
- Legacy system upgrades include a review against current security engineering principles — components that cannot meet requirements are flagged for lifecycle replacement.

Real World Example

The Scenario

Acme Defense acquired a CAD system for CUI design work five years ago. The system runs on an unsupported OS. No threat modeling was done when it was deployed. No security engineering principles were applied — it was purchased, plugged in, and connected to the CUI network.

What the assessor finds

The CAD system runs an OS with dozens of known unpatched CVEs. The vendor no longer supports it. No architectural review was ever conducted. The system has direct access to all CUI files — it was never subjected to any security engineering review and no lifecycle plan exists for it.

SPRS Score Impact

3.13.2 carries a point value of 1. Organizations that build security in from the start spend far less on remediation than those that bolt it on after deployment — and produce systems that are fundamentally more resistant to attack.

What Good Looks Like

Security architecture documented, architectural principles identified and applied in system design, secure coding standards in use, developers trained, threat modeling conducted during design and upgrades, legacy systems reviewed against current principles with lifecycle plans.

Common Gaps

What assessors actually find in the field:

✗ No security architecture

Systems are built based on immediate functional need — no documented architectural principles guide design decisions or security tradeoffs.

✗ Security bolted on after build

Systems are deployed and then hardened reactively — security is not incorporated during design, only after vulnerabilities are discovered.

✗ No secure coding standard

Developers write code without a defined secure coding standard — common vulnerabilities like injection and insecure deserialization are not systematically addressed.

✗ No threat modeling

New systems and major upgrades skip threat modeling — attack surfaces are not identified and mitigated during design.

✗ Legacy systems never reviewed

Aging systems that cannot meet modern security principles continue in service indefinitely — no lifecycle process triggers review or replacement.