SC.L2-3.13.4

System & Communications Protection

Shared Resource Control

"Prevent unauthorized and unintended information transfer via shared system resources."

Objectives

[a]

Unauthorized and unintended information transfer via shared system resources is prevented.

Key Discussion Points

Object Reuse:
This is also called 'object reuse' — when the OS releases a resource (memory, disk block) from one user, it must be cleared before another user receives it.

Includes Encrypted Data:
The guide specifies this applies even to encrypted representations of information — encrypted residue in a shared resource still counts as a violation.

OS Controls This:
This requirement is primarily satisfied through OS-level controls — proper user isolation, patched firmware, and correct permission settings on shared directories.

Meltdown / Spectre:
The guide explicitly names processor-level exploits as in scope — unpatched Meltdown/Spectre vulnerabilities allow one process to read another process's memory.

Assessment Methods

EXAMINE

System and communications protection policy; procedures addressing application partitioning; system design documentation; system configuration settings; system security plan; system audit logs.

INTERVIEW

System or network administrators; personnel with information security responsibilities; system developer.

TEST

Separation of user functionality from system management functionality.

SC.L2-3.13.4

SYSTEM & COMMS PROTECTION — Shared Resource Control

Plain English

What this control is really saying:

When one user finishes working with a memory buffer or disk block, the OS reclaims it — but what was in that memory doesn't automatically get erased. If another user then gets access to the same resource, they might be able to read the previous user's data. This control requires that shared resources are cleared before being allocated to a new user or process.

How it is used:

- OS-level access controls prevent users from reading other users' file directories, temp files, and memory space — permissions are configured per hardening baseline.
- Firmware and OS patches addressing processor-level vulnerabilities (e.g., Meltdown, Spectre) are applied as part of the patch management program.
- Virtual machines in shared hosting environments are isolated from each other — hypervisor controls prevent inter-VM memory inspection.
- The system hardening baseline documents controls that prevent shared resource leakage — configuration settings are verified against the baseline on a defined schedule.

Real World Example

The Scenario

Acme Defense has a multi-user workstation in the CUI engineering bay used by three engineers who share the machine with a single shared login. All three users access each other's files without restriction.

What the assessor finds

An engineer working on a confidential proposal saves files to the temp directory. A co-worker on the same machine can read those files — they are accessible to all users. The machine has also not received the OS patches addressing the Spectre vulnerability.

SPRS Score Impact

3.13.4 carries a point value of 1. Shared resource leakage is a subtle but real risk — multi-user systems that don't isolate users allow CUI to leak between sessions through temp files, cached data, and memory residue.

What Good Looks Like

OS and firmware patched for known shared-resource vulnerabilities, user file permissions enforce isolation between accounts, temp directories not world-readable, VM hypervisor isolation configured, hardening baseline documents shared resource controls.

Common Gaps

What assessors actually find in the field:

✗ **OS not patched for processor CVEs**

Processor-level vulnerabilities (Meltdown, Spectre) allow reading other processes' memory — unpatched systems expose CUI from other users' sessions.

✗ **Temp file permissions too broad**

Temporary files created by CUI applications are world-readable — any user on the system can access CUI residue left in shared temp directories.

✗ **User home dirs not isolated**

Default OS installation allows users to traverse each other's home directories — CUI files in one user's profile are accessible to others.

✗ **Shared VM not hardened**

Virtual machines on shared infrastructure lack hypervisor-level isolation controls — guest VMs can potentially inspect host or neighboring VM memory.

✗ **No hardening baseline**

No hardening baseline defines the required shared resource isolation settings — configuration is ad hoc and controls may be missing.