Scudo: Google’s Hardened Allocator

Ben Brown, RITSEC VRIG 2026

What is a Scudo?

An Italian coin from the 17th and 18th centuries; translates to “Shield”

Also a dynamic user-mode allocator focused on security while still being mostly performant

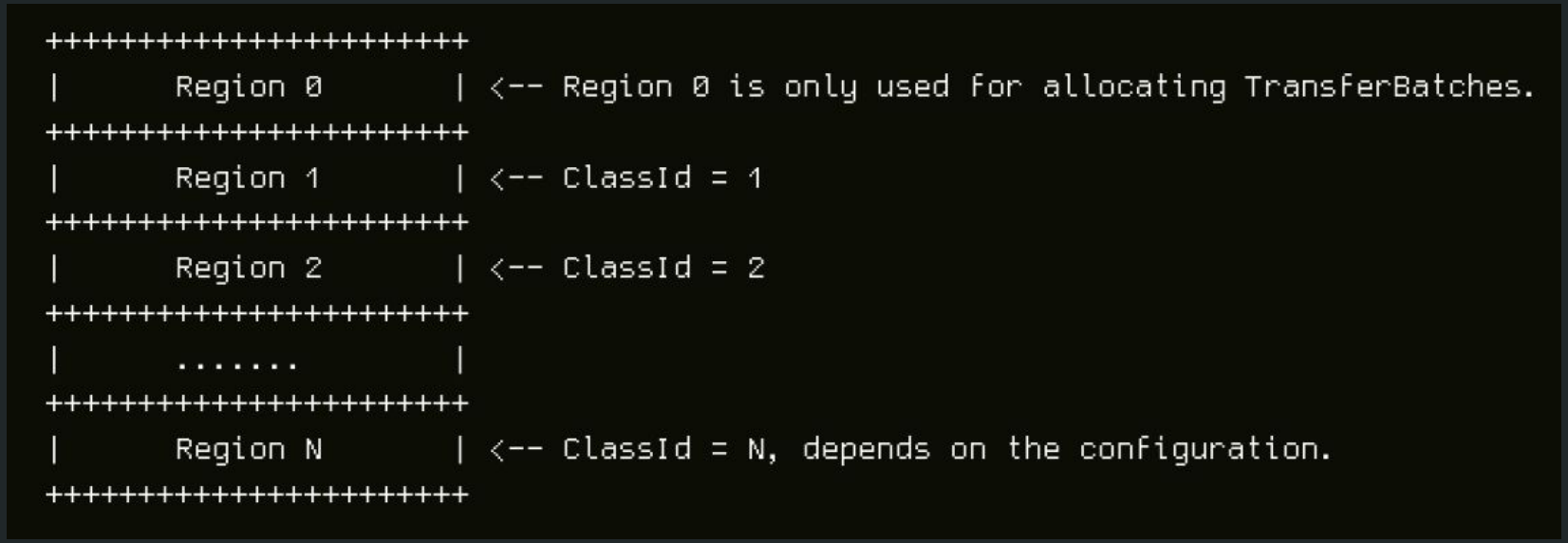
Used in Android 11+ and Fuschia; part of LLVM

Implemented in C++ with heavy feature use

High-Level Structure

I. Primary Allocator
   A. Fast and efficient
   B. Smaller allocation sizes
   C. Two implementations for 32 and 64 bit architectures
II. Secondary Allocator
   A. Larger allocations via mmap and friends
   B. Allocations surrounded by guard pages
III. Thread Specific Data Registry
   A. Defines local caches
   B. Two modes: Exclusive and Shared
IV. Quarantine
   A. Delayed freelist designed to prevent UAF
   B. Disabled by default; costly in terms of performance

Heap configuration



Primary Allocator

Entire heap for primary allocator is pre-mapped, then divided into N sizeClass regions

Optional configurations:

- Guard Pages
- Page-random mapping ([1, 16]*PageSize)

Total Heap Size: NumClasses * RegionSize

Block Lifecycle

I. Allocate
   A. Determine Allocator (Primary or Secondary)
   B. SizeClass region selected
   C. Check for free blocks (uses TransferBatch class in single linked list)
      1. Refill cache if blocks exist in TransferBatches
         a) Repopulate Freelist if no TransferBatch or Cache Blocks
         b) TransferBatches are shuffled during refills
   D. Header with metadata and checksum (CRC32) is prepended to block
   E. Return to program

II. Free
   A. Sanity checks for basic vulnerabilities (double free, etc)
   B. Checksum is validated
   C. Moved to quarantine if configured, sent back to cache or deallocated

Key Takeaways

I. Randomization is effective at adding security to an allocator
II. Some security practices lead to high overhead (Quarantine)
III. Caching and efficient memory transfer can balance security overheads

References and Further Reading

[1] “Advancing Cybersecurity: Introduction to the Scudo Allocator - Vectorize.” Accessed: Feb. 02, 2026. [Online]. Available: https://vectorize.re/blog/internals/introduction-to-scudo/

[2] “Behind the Shield: Unmasking Scudo’s Defenses,” Synacktiv. Accessed: Feb. 02, 2026. [Online]. Available: https://synacktiv.com/publications/behind-the-shield-unmasking-scudos-defenses

[3] “Scudo Hardened Allocator — LLVM 23.0.0git documentation.” Accessed: Feb. 02, 2026. [Online]. Available: https://llvm.org/docs/ScudoHardenedAllocator.html

[4] “Scudo, the Allocator (Part 1),” lolcads tech blog. Accessed: Feb. 02, 2026. [Online]. Available: https://lolcads.github.io/posts/2024/07/scudo_0/

[5] “un1fuzz ~.” Accessed: Feb. 02, 2026. [Online]. Available: https://un1fuzz.github.io/articles/scudo_internals.html