MOAD Cheat Sheet

MOTHER OF ALL DEFECTS — QUICK REFERENCE · ALL ACTIVE RESEARCH PROGRAMS · Russell John Ballestrini · Aaron Wes Schacht

ID	CODENAME	CWE	> DEFECT / < FIX	STATUS
MOAD-0001	**A Sedimentary Defect**	CWE-407	> `list.contains()` inside a loop, O(N²) membership test / < Replace list with hash set. O(1) lookup, same behavior.	ACTIVE
MOAD-0002	**An Intertangled Defect**	TBD	> Subsystems coupled through shared mutable global state, no phase boundary / < Phase snapshot, clean interface, separate context per subsystem	ACTIVE
MOAD-0003	**A Leaked Context**	TBD	> `ThreadLocal` holds request identity, bleeds into next request on thread reuse / < `ScopedValue` (Java 21+) · `context.Context` (Go) · `ContextVar` (Python)	ACTIVE
MOAD-0004	**A Logged Secret**	CWE-312	> HTTP headers (`Authorization`, `Cookie`) logged verbatim, tokens on disk / < Credential denylist at log serialization, strip before write	ACTIVE
MOAD-0005	**A Thundering Herd**	CWE-362	> Cache miss + compute + put with no lock, N threads all compute simultaneously / < `computeIfAbsent` · singleflight · lock around compute path	ACTIVE
MOAD-0006	**A Glass Safe**	CWE-257	> Credentials stored reversibly: plaintext, XOR, base64, or recoverable encryption / < One-way hash: `bcrypt` · `argon2`, no recovery path	CANDIDATE
MOAD-0007	**A Flatland Defect**	CWE-407	> Spatial objects in flat list, every raycast and collision query scans O(N) / < Spatial index (BVH, octree, k-d tree), O(log N) per query	CONFIRMED
MOAD-0008	reserved		> Reserved / <	RESERVED
MOAD-0009	**A Metered Heart**	TBD	> Scheduled job fires on a clock, not events. State-repair or blind recompute on a timer. / < Event-driven design: fire when something actually changes	CANDIDATE
MOAD-0010	reserved		> Reserved / <	RESERVED
MOAD-0011	**A Catastrophic Inheritance**	CWE-1333	> User-supplied pattern compiled by backtracking NFA (PCRE, Oniguruma). O(2^N): at N=25, 6,805ms vs 1.65ms linear (4,120,453x). / < RE2 or Thompson NFA: google-re2 (Python), re2 gem (Ruby). Timeout wrapper as fail-safe fallback.	CANDIDATE

Coupled risk — MOAD-0001 + MOAD-0005: Fix O(N²) at a high-throughput node without staging downstream capacity and freed throughput floods every queue simultaneously. Assign caretakers in `vendors.json` before any patch lands. Surge estimate: `speedup × in_degree`.