

PAIN Relief: A Verified-Control Method for Reducing FedRAMP VDR/VER PAIN and Exploitability

Matthew Venne
Chief Technology Officer, stackArmor

July 2026

Abstract

The companion memo *A Deterministic, CVSS-Environmental Method for FedRAMP VDR/VER Vulnerability Prioritization* derives a finding's Potential Agency Impact (PAIN) and its remediation deadline. That memo also establishes a mitigation *ladder*: PAIN comes down when cited, signed evidence lowers a Modified impact metric. This paper makes the ladder systematic. It defines **PAIN Relief**: a governed, versioned catalog that pairs a *machine-verified* security control with the specific weakness class (by CWE) it provably counters and the deterministic scoring reduction that pairing earns. Two levers move: verified controls that bound what a successful exploit can do lower a Modified impact metric and step the PAIN level down; verified controls that reduce the probability of successful exploitation lower a *local* estimate of that probability and can move a finding from likely-exploitable to not. Neither lever mutates a published input — not the CVSS base vector, not the EPSS score, not the CISA KEV clock. The method's guiding line is *enforced-and-machine-verifiable*, not *claimed*: a control earns relief only when enforcement infrastructure applies it and a collector can read it. The catalog is offered as an example; each provider governs its own.

Contents

1	Status and Relationship to the Companion Papers	2
2	The Problem: Good Hygiene Must Pay	2
2.1	The axis that matters: enforced, not claimed	2
3	Two Levers	2
3.1	Impact: lowering a Modified metric	3
3.2	Exploitability: lowering a local estimate	3
4	The Catalog	4
5	Governance	5
6	What PAIN Relief Never Does	5
7	Worked Examples	5
8	Conclusion	6

1 Status and Relationship to the Companion Papers

This document is informational. It does not define a FedRAMP requirement; it specifies a method for lowering the PAIN and exploitability of a finding within the existing VDR/VER rules, building directly on the PAIN memo’s mitigation ladder. It consumes the internet-reachability determination of *Internet Reachability at Scale under the FedRAMP VDR/VER Rules* and the disposition vocabulary of *A CycloneDX VEX Profile for FedRAMP VDR/VER*. The key words MUST, SHOULD, and MAY follow RFC 2119.

2 The Problem: Good Hygiene Must Pay

A provider that runs its workloads with least privilege, deny-by-default egress, read-only filesystems, and enforced runtime protection is measurably harder to harm than one that does not. Yet a raw scan scores an identical CVE identically on both. If secure configuration changes no number a provider must report, it is all cost and no credit, and the rational provider treats hardening as a burden. The PAIN method already refuses that outcome for one input — the asset’s security-impact profile, expressed through the CVSS Environmental requirements — by re-weighting a vulnerability’s impact by what is at stake on the specific asset. PAIN Relief extends the same principle from *what the asset requires* to *what protects it*.

IN PLAIN TERMS

A scanner scores a flaw as if every system running it were equally exposed. Two providers run the same flaw; one has locked the system down and one has not. PAIN Relief is how the locked-down provider’s work shows up in the score — lower impact, or a longer clock — so that doing security well is worth something on paper, not just in principle.

2.1 The axis that matters: enforced, not claimed

The tempting mistake is to reward controls a provider merely *asserts*. PAIN Relief admits a control only when it is **machine-verifiable** — read from enforced configuration or telemetry a collector can inspect (a NetworkPolicy, a **securityContext**, an RDS flag) — or carried by a signed, dated, reusable attestation for structure a platform cannot expose. A control that does not meet that automatic bar stays outside the catalog and is governed separately through reachability or VEX evidence. The line is not *a priori* versus *a posteriori*; it is *enforced-and-verifiable* versus *asserted*.

Input sanitization is handled separately by the reachability model’s P_0 predicate. Assessment, recurring DAST, remediation, and retest can support a scoped, revocable P_0 even when the control is not eligible for an automatic PAIN Relief catalog row. That evidence changes the reachability determination; it does not also earn a second likelihood or impact discount here.

3 Two Levers

PAIN Relief moves exactly two things, and keeps them strictly apart. A control belongs to one lever, chosen by its causal mechanism: a control that bounds *what a successful exploit yields* moves impact; a control that reduces *whether the exploit succeeds* moves exploitability. Because the assignment follows the mechanism, a control never scores on both axes, and nothing double-counts.

3.1 Impact: lowering a Modified metric

The PAIN memo scores a finding through the CVSS Modified Impact Sub-Score, built from the per-dimension impacts C, I, A and the asset’s requirements. A verified control that provably counters a weakness class lowers the affected Modified impact metric from High to Low before that score is computed:

$$\text{High} \longrightarrow \text{Low}, \quad \text{never None.} \quad (1)$$

The floor is deliberate. A claim of *zero* impact is an applicability claim, and it travels as a VEX disposition (`code_not_present`, `code_not_reachable`), not as relief. The reduced C, I , or A flows through the memo’s existing arithmetic unchanged; the PAIN level and deadline follow. There is one rung, and it does not stack: several controls that each argue a dimension down still lower it by a single step, with every contributing control named in the finding’s evidence.

3.2 Exploitability: lowering a local estimate

FedRAMP’s exploitability axis is binary: a finding is likely-exploitable (LEV) or not (NLEV), and the two select different remediation columns. The memo computes

$$\text{LEV} = \text{KEV} \vee (\text{EPSS} \geq \tau), \quad \tau = 0.70,$$

PAIN Relief never edits the published EPSS score — a global prediction is not a provider’s to rewrite — just as the impact lever never edits the CVSS base vector. Instead a verified control lowers a *local* estimate of exploitation probability, and the binary threshold does the rest:

$$\text{adjustedEPSS} = \max\left(\text{EPSS} \times \prod_{r \in R} \rho_r, \text{EPSS} \times \phi\right), \quad (2)$$

where R is the set of applicable verified likelihood controls, each carrying a *residual factor* $\rho_r \in (0, 1)$ — the share of exploitation risk it leaves behind — and ϕ is a governed floor on total reduction. LEV is then recomputed with `adjustedEPSS` in place of EPSS.

DEFINITION

ρ_r (*residual factor*): a control that halves exploitation likelihood carries $\rho = 0.5$; one that trims it modestly carries $\rho = 0.85$. Lower is stronger. ϕ (*stacking floor*): the most that stacking may reduce the estimate, e.g. $\phi = 0.5$ caps total reduction at one half. Both are governed, back-tested constants, documented and challengeable like the PAIN cut points.

Three properties of Equation (2) matter. First, **it stacks multiplicatively**: independent controls compound, so layered runtime defense is rewarded — this is the one place defense-in-depth earns more than any single control. Second, **the threshold self-calibrates**: a modest control drops a marginal finding ($0.72 \times 0.85 = 0.61$, now NLEV) but not a near-certain one ($0.99 \times 0.85 = 0.84$, still LEV), which is exactly right — one control should not defuse a vulnerability being exploited everywhere. Third, **KEV is frozen by default**, with one governed, off-by-default exception (below): a Known Exploited Vulnerability stays LEV regardless of ρ unless an adopter deliberately enables the exception, and its BOD 26-04 clock is always the deadline floor.

The KEV exception (governed, off by default). A KEV is *observed* exploitation, not a prediction, so by default no control moves it. An adopter MAY enable a narrow exception: a KEV is classified NLEV only if the effective residual is ≤ 0.7 *and* the resulting `adjustedEPSS` < 0.5 — a high bar that a near-certain KEV clears only with strong, stacked, verified controls. Even

then the exception relaxes *only* the PAIN-matrix column: the CISA KEV / BOD 26-04 due date remains in force (VDR-TFR-KEV: remediate by the CISA date “even if the vulnerability has been fully mitigated”). The finding’s deadline is therefore min(PAIN-matrix cell, CISA KEV date) — NLEV can loosen the clock *up to* the CISA date, never past it. It is off by default because a 3PAO may reasonably challenge an NLEV label on an actively-exploited CVE; an adopter who enables it does so knowingly and documents the controls.

IN PLAIN TERMS

EPSS is a worldwide guess at how likely a flaw is to be attacked. We never change that number. We keep a second, local number that starts at EPSS and drops when we can verify a control that makes the attack less likely to work here. If the local number falls below the line, the finding earns the slower clock. Stacked controls push it lower together — but a flaw that is already being exploited in the wild (KEV) never moves, and the change is always shown beside the original so nothing is hidden.

4 The Catalog

A PAIN Relief entry is a row pairing a control, the CWE class it counters, and the move it earns. Rows are keyed to CWE because the finding carries a CWE and the control’s causal story is a statement about a weakness class — “deny-by-default egress breaks the second-stage fetch of injection and staged-execution flaws” is a claim about **CWE-917**, **CWE-94**, and their kin, not about a single CVE. Where a control bounds what *any* code execution yields regardless of how it began, the row keys to an outcome class (all execution-capable weaknesses, including memory corruption) rather than an enumerated list. A finding with no CWE, or an unrecognized one, earns nothing: relief fails open toward the higher score.

Verified control	Counters (CWE)	Lever	Move
Least-privilege DB grant	89	impact	MC/MI High→Low
No shell in image	78, 77	impact	MC/MI High→Low
SELinux domain confinement	ACE class	impact	MC/MI High→Low
Read-only root filesystem	22, 434, 73	impact	MI High→Low
Route rate limiting	400, 1333	impact	MA High→Low
Egress control (tiered)	917, 94, 98	likelihood	ρ 0.90–0.50 by firewall tier
Blocking-mode EDR/RASP	(any)	likelihood	$\rho = 0.85$

Table 1: Illustrative PAIN Relief rows. The maintained catalog is proprietary and provider-governed; this sample is an existence proof, not a standard — exactly the posture of the example archetype-to-profile catalog in the companion memo. Every applied credit ships its row identifier, rationale, and the catalog version in the finding’s evidence.

Verification is not limited to Kubernetes configuration. Host-level controls prove out from STIG/SCAP scan results just as well: SELinux domain confinement, for instance, is credited from the DISA RHEL 8 STIG (RHEL-08-010450 enforcing) plus a per-process confined-domain check, and bounds what an RCE can reach regardless of how the code executed. A public, illustrative subset of the catalog — six rows in this exact format, including the SELinux/STIG example — is published as the auditable format reference at [stackArmor/vdr-pain-relief-examples](#); the maintained catalog remains proprietary.

5 Governance

Adding a row grants a discount across the estate, so a row is a security setting and is reviewed like one. Each row MUST satisfy:

- **A causal story, not an adjective.** “Egress-deny breaks the staged fetch” qualifies; “EDR makes exploitation harder” does not. The row names the mechanism by which the control interrupts the weakness class.
- **Machine-verifiable or attested, never claimed.** The control is proven from enforced configuration, telemetry, or a signed reusable artifact. This keeps WAF signature coverage out of the automatic catalog and in the VEX lane. Input-control evidence is evaluated under the reachability model’s scoped, revocable P_0 predicate instead of being double-counted as PAIN Relief.
- **CWE-keyed and fail-open.** A row applies only to the classes it lists; missing or generic CWEs earn nothing.
- **One rung; never None.** Impact moves High to Low only; zero-impact is a VEX applicability claim. Exploitability moves the binary via Equation (2); nothing here touches reachability or KEV dates.
- **Versioned, pinned, and evidenced.** A score is reproducible only against a named catalog release; the row id and version travel in every evidence line. Refusals are recorded, so the catalog is curated, not accumulated.
- **Back-tested before adoption.** Residual factors, the stacking floor, and any new row are calibrated against real findings; a row that moves a large fraction of the estate is presumed miscalibrated until argued otherwise.

6 What PAIN Relief Never Does

No entry mutates the published EPSS score or the vendor CVSS base vector; both the original and the adjusted values appear in output. No entry alters the internet-reachability determination — that is the companion reachability model’s job, and PAIN Relief merely consumes its result. No entry extends or softens a CISA KEV / BOD 26-04 remediation date; a Known Exploited Vulnerability carries its CISA clock untouched no matter how thoroughly it is mitigated — the governed NLEV exception can only relax the PAIN-matrix column *beneath* that clock, never the clock itself. And no entry lowers a Modified impact dimension to None; that stronger claim is a signed VEX disposition, separately governed.

7 Worked Examples

Impact — a schema-scoped credential on SQL injection. A CWE-89 finding scores $C = I =$ High on a datastore, placing it at, say, N4. The application connects with a credential confined to a single schema — no superuser, no cross-database reach, no file or extension privileges — verified from a grants dump. A successful injection still executes with that credential’s authority, but that authority reaches only the one schema: *even on full exploitation the blast is bounded*. That is what makes this an impact control and not a likelihood one — there is no broader privilege for a bypass to restore. The row lowers Modified Confidentiality and Integrity High→Low; recomputed through the memo’s arithmetic the finding lands at N3. The reachability answer is unchanged; what changed is how much a successful exploit reaches.

Exploitability — egress control, and why Log4Shell is the exception. A CWE-917 Log4Shell-class lookup needs an outbound fetch to pull its second stage. The workload sits behind a verified egress firewall. Egress control is a *likelihood* lever, not impact: it does not shrink the RCE, it lowers the odds the chain completes — and an attacker who tunnels out via DNS, an allowed domain, or domain fronting still achieves full RCE. So its strength is the firewall’s, expressed as tiers: port-restricted egress (web ports open to any destination, non-web ports denied — IP-allow-listing tcp/80 and 443 is impractical, so 0.0.0.0/0 on the web ports is standard and domain filtering is the real control) earns $\rho \approx 0.90$, since it stops non-web second-stage ports but not an HTTP(S) fetch; domain/SNI filtering without DNS validation, 0.80; SNI with DNS validation, 0.70; and outbound break-and-inspect — which decrypts and checks the Host header, catching domain fronting — 0.50. *But Log4Shell is a KEV*, so by default it is frozen: the residual applies to nothing, the finding stays LEV, and the CISA clock governs. That is the honest outcome — egress hardening buys no time on an actively-exploited flaw. Take instead a non-KEV fetch-dependent RCE at EPSS 0.74 behind break-and-inspect ($\rho = 0.5$): $\text{adjustedEPSS} = 0.74 \times 0.5 = 0.37 < 0.5$, and it moves to NLEV, published EPSS untouched beside it.

IN PLAIN TERMS

Two practitioner notes on this one. First, if the egress firewall is *shared* — every workload routes through one gateway — discovering that path automatically grants the relief to all workloads behind it, not pod by pod: verify the gateway once, the estate inherits the tier. Second, the tier is the vendor’s and the config’s. A firewall that allow-lists a domain but never checks that the connection’s SNI matches the real destination is weaker than one that validates it against DNS, which is weaker than one that decrypts and inspects the Host header. The deeper the inspection, the more relief — because the more exfiltration and domain-fronting tricks it forecloses.

IN PLAIN TERMS

Same two flaws, two providers. On the hardened estate the first drops a PAIN level and the second drops onto a slower clock, purely because a scanner-adjacent tool could *see* the controls that make those flaws less dangerous here. On the unhardened estate, nothing moves. That difference is the entire point.

8 Conclusion

PAIN Relief closes the loop the PAIN memo opened. The memo re-weights a vulnerability by what is at stake on the asset; this paper re-weights it by the controls that verifiably protect the asset, using the same Environmental machinery and the same discipline of citing evidence. Impact steps down a rung when a verified control bounds the damage; exploitability crosses a threshold when verified controls make success less likely; and neither ever edits a published input, softens a KEV clock, or turns a claim into a discount. The result gives a provider what raw scanning never has: a deterministic, auditable reason that good security configuration lowers the number — and therefore a reason to build securely in the first place.