Internet Reachability at Scale under the FedRAMP VDR/VER Rules

Matthew Venne
Chief Technology Officer, stackArmor

July 2026

Abstract

The FedRAMP Vulnerability Detection and Response (VDR) and Vulnerability Evaluation and Reporting (VER) rules require providers to determine, per finding, whether a vulnerability is internet-reachable. Direct exposure is often computable from information providers already possess: public routes, load-balancer listeners, firewall policy, running processes, affected software, and CVSS attack vector. Transitive reachability is different. Vulnerability sources and scanners rarely publish structured metadata describing whether a CVE can be triggered by internet-originating content after that content crosses a component boundary. Without that upstream fact, providers face two bad defaults: classify every vulnerability on every internet-touched component as reachable, destroying prioritization, or approximate reachability with direct accessibility and risk missing exceptional indirect triggers such as Log4Shell. Deployment-specific data-flow evidence can refine the second step—which affected components actually receive the content and what happens on the way—but vulnerability tracking systems do not generally integrate that evidence either.

This paper recommends an operational model between those extremes. It computes direct finding-level reachability and adds a reusable CVE-level `indirect_internet_trigger` promotion when vendor guidance, threat intelligence, exploit analysis, agentic enrichment, or analyst review establishes a transitive trigger. A coarse asset assertion identifies components that receive internet-derived content. Existing assessment and continuous-monitoring evidence supplies a standing, revocable prevention assertion for the input-control classes it actually tests. For Rev5 services, that evidence already includes recurring web-application scanning, remediation, and assessor validation; it does not require a new application-wide data-flow graph. This is the paper's present recommendation, not an interim step contingent on a future provider capability. It causes the operational IRV set to converge toward the directly accessible surface, plus known indirect triggers not covered by the standing assertion. A separate full-information equation documents how structured trigger profiles and optional deployment-edge evidence could refine the result if the vulnerability-intelligence ecosystem later supplies them. It is an interoperability reference, not a proposed FedRAMP evidence requirement. Agentic enrichment is treated as a controlled bridge over missing upstream metadata, not as an authoritative replacement for it.

1 Introduction

FedRAMP's 2026 rules require providers to evaluate detected vulnerabilities in the context of the cloud service offering and report whether each is an Internet-Reachable Vulnerability (IRV), whether it is a Likely Exploitable Vulnerability (LEV), and the Potential Agency Impact N-rating (PAIN) of exploitation. Those answers select the response timeframe under `VDR-TFR-PVR`; the IRV result is also included in the vulnerability detail report required by `VER-RPT-VDT`. The rules deliberately

use *internet-reachable*, not merely *internet-accessible*: a vulnerable resource may sit on a private network and still process a payload that originated on the public internet.[1, 2]

That distinction is technically correct and operationally difficult. A scanner can usually identify a CVE and affected component. It may supply a CVSS vector, a CWE, a short description, and references. It generally does not state, in machine-readable form, whether attacker-controlled content can cross a process boundary and still trigger the vulnerability, which content format or processing operation is required, or whether the attacker must remain the connection peer. Nor does the vulnerability record normally know the provider's application graph: which API writes which queue, which worker parses which upload, or which validation is applied before a database query.

This paper defines a method that can be implemented now and separately records optional extension points for upstream standardization. Sections 2 and 3 define the operational problem. Section 4 gives the recommended equation first and the full-information reference form second. Sections 5 through 7 describe the inputs. The remaining sections apply the method to triage, controls, worked examples, and validation.

IN PLAIN TERMS

FedRAMP asks a question scanners cannot fully answer from their normal output. Scanners are reasonably good at saying what software is present and which network service is exposed. They are not designed to say whether a string, file, message, or database record can carry a particular exploit through three application tiers. The solution is not to perform the same manual research for every asset. It is to enrich the vulnerability once, combine that result with reusable deployment and standing control evidence, and preserve uncertainty when either side is missing.

2 The Metadata Gap Creates Two Bad Defaults

FedRAMP defines an IRV as a vulnerability that might be exploited or triggered by a payload originating on the public internet. Its notes expressly include resources with no direct internet route when they receive a payload or take an action triggered by internet activity. The evaluation guidance also says that payload interception, filtering, sanitization, rejection, or comparable prevention before vulnerable processing can remove IRV status.[1, 2]

Those rules establish the right semantic boundary: the unit is the *finding*, and the question is whether a qualifying trigger can reach the vulnerable processing operation. They do not, however, provide the metadata needed to automate that test. When a provider sees a local parser CVE on a queue worker, neither the scanner finding nor the CVSS vector necessarily says whether a public file upload can activate that parser. FIRST's CVSS guidance even notes that malicious network data passed from one system to a separate vulnerable system can be scored `AV:L`; base attack vector alone cannot resolve the transitive question.[5]

The gap creates two operational extremes.

Maximal propagation. Assume every vulnerability on every component that ever receives internet-derived data might be triggered by that data. This preserves recall, but a connected cloud service quickly approaches universal IRV classification: requests reach applications, records reach databases, messages reach workers, and strings reach logs. The IRV/NIRV distinction then contributes little prioritization while the provider still has to record and report the result.

Accessibility-only approximation. Evaluate only direct network exposure. This is scalable and reproducible, but it misses known transitive shapes such as a crafted string reaching a logging

library or a malicious file reaching an interior parser. It treats the absence of metadata as evidence that no indirect trigger exists, which is not the same claim.

The practical answer today is an explicitly metadata-constrained approximation: compute direct reachability from available evidence, promote known indirect-trigger CVEs through a reusable enrichment, and apply the standing prevention evidence already produced through assessment and continuous monitoring. In set terms,

$$R_{\text{operational}} = D \cup (K \setminus \mathcal{P}_0),$$

where D is the set of directly triggerable findings and K is the set of findings promoted by known indirect-trigger intelligence. $\mathcal{P}_0$ is the subset for which $P_0(v, a) = 1$ under a current assessment- and ConMon-backed prevention assertion. The true set may also contain triggerable members of an unresolved population U and depends on whether the asserted controls actually prevent the trigger:

$$R_{\text{true}} = D \cup (K \setminus \mathcal{P}_{\text{true}}) \cup U_{\text{triggerable}}.$$

Under current metadata constraints, $R_{\text{operational}}$ is therefore dominated by the directly accessible surface plus known, uncovered promotions. This is the paper's convergence claim: not that reachability and accessibility are semantically equivalent, but that recurring assessment, DAST, and remediation support a rebuttable operational presumption that tested downstream trigger classes are prevented. A failed, stale, out-of-scope, or contradicted assertion does not enter $\mathcal{P}_0$.

IN PLAIN TERMS

There are two easy answers, and both are bad. Calling every flaw touched by internet data reachable makes the label almost useless. Looking only at open internet services misses the rare but important flaws whose trigger rides inside data. Today's workable compromise starts with the direct surface and adds known indirect cases once per CVE. Existing assessment and recurring web testing then remove the cases for which the relevant input control is currently supported. This makes the answer look much like accessibility in steady state, with known uncovered triggers as explicit exceptions. The missing cases remain a disclosed limitation, and better upstream metadata is the durable fix.

3 Current Inputs and Optional Refinement

The recommended method joins two kinds of information at the resolution available today. Trigger status is chiefly a vulnerability-level fact; deployment applicability is a provider-level fact. Neither requires a provider to construct a complete application-wide taint or data-flow proof.

3.1 Vulnerability trigger metadata

The vulnerability side must describe how the flaw is activated, not merely where the affected package is installed. Useful fields include the carrier of the attacker input, its format and encoding, the operation performed on it, the degree of attacker influence required, and whether the attacker must remain the connection peer. This information should originate with vendors, CVE Numbering Authorities, coordinated-disclosure participants, or other authoritative vulnerability publishers, then be distributed through scanners and threat-intelligence feeds.

Today's scanners are a natural delivery point but not necessarily the authoritative source. Their ordinary findings often contain only a compact description whose purpose is detection and remediation, not a complete exploit contract. A CWE can route research toward a family of mechanisms; it cannot stand in for the exact trigger of a CVE.

3.2 Deployment applicability and prevention evidence

The current profile needs governed assertions about which component archetypes receive internet-derived content and which tested trigger classes are prevented before vulnerable processing. FedRAMP assessment and ConMon already produce a recurring evidence stream for the latter: web-application scan scope and configuration, results, remediation and retest history, control implementation evidence, and assessor validation. A provider may refine those assertions with evidence about how content moves through the service and what each boundary does to it. A higher-resolution representation can include API-to-service calls, queue publication and consumption, database and object storage flows, logging paths, file-processing pipelines, protocol translation, and validation or sanitization behavior. Each claim should be bound to the source revision, built artifact, deployed configuration, and evidence that supports it.

Individual ingredients exist today: infrastructure as code, service catalogs, flow telemetry, API specifications, SAST results, unit tests, integration tests, and CI/CD provenance. But the industry lacks a broadly adopted, machine-readable schema and assurance method for combining those artifacts into an independently verifiable, vulnerability-specific reachability record that a vulnerability-management platform can ingest, map to affected findings, and re-evaluate when the supporting code, configuration, topology, or evidence changes. This paper treats such a record as an optional interoperability refinement, not as a prerequisite for using the recommended method.

Vulnerability Trigger Profile + Deployment and Standing Control Evidence = Finding-Level Reachability

IN PLAIN TERMS

A provider can usually identify broad groups of components that receive internet-derived data—APIs, queues, workers, databases, and logging systems. FedRAMP assessment and recurring web scans also provide a standing assertion that the input-control classes in their validated scope are working and that findings are remediated. That is enough for a scoped triage presumption; it is not proof against an unknown trigger or an untested path. Proving the exact path, what happens at each handoff, and which deployed code implements a validation control is harder. The pieces exist across infrastructure code, telemetry, tests, and CI/CD records, but vulnerability systems cannot yet combine them into a standard, continuously valid reachability record. The recommended method therefore uses governed component and control archetypes today. A higher-resolution claim still needs a verified prevention point on every relevant path before the vulnerable operation; not every edge must sanitize the data.

4 Two-Surface Finding-Level Model

The original network descriptor remains useful, but network conversations and relayed content are different objects. The model therefore uses two linked surfaces rather than forcing both into one tuple.

4.1 The network surface

Let $N(a)$ be the set of network descriptors directly deliverable to component a, and let $X_N(v)$ be the network descriptors under which vulnerability v can be directly triggered. A network descriptor is

$$n = \langle \kappa, \tau, \pi, \nu, \delta \rangle,$$

where κ is the CVSS attack-vector class, τ is transport and port, π is application protocol, ν is protocol version or ALPN, and δ is direction. Direct reachability is

$$D(v, a) = \mathbb{1}[N(a) \cap X_N(v) \neq \varnothing] \,. \tag{1}$$

CVSS `AV:N` is evidence that $X_N(v)$ may contain a network trigger. `AV:L`, `AV:A`, and `AV:P` exclude a vulnerability from this *direct* branch; they do not establish that v has no indirect content-trigger path.

4.2 The recommended operational profile

Structured per-CVE content-trigger profiles and per-edge transformation graphs are not generally available in current scanners and vulnerability tracking systems. The method recommended by this paper uses three available, governed predicates:

- $J(v)$ is the CVE-level `indirect_internet_trigger` status: confirmed, ruled out, or unknown.
- $C_0(a)$ asserts that component a receives internet-derived content, derived from an asset archetype, an imported or observed flow, or a governed operator assertion.
- $P_0(v, a)$ asserts that current assessment and ConMon evidence covers the relevant trigger or control class and supports prevention before vulnerable processing. It is resolved by application or deployment archetype rather than independently for every asset.

The recommended equation is

$$\mathrm{IRV}_0(v, a) = D(v, a) \;\vee\; (\mathbb{1}[J(v) = \text{confirmed}] \cdot C_0(a) \cdot (1 - P_0(v, a))) \,. \tag{2}$$

The status `unknown` is retained as data even though the baseline classifier treats it as not promoted. This is a deliberate scalability presumption, not proof of non-reachability. Vendor guidance, a CNA update, threat intelligence, exploit analysis, agentic enrichment, or analyst review can promote $J(v)$ once and apply the result across every affected finding.

CWE-seeded mandatory review. A CWE is a routing signal, not proof of $J(v)$, but selected CWE mappings should always place a CVE into a one-time, CVE- or product/version-level review for indirect triggerability. A non-exhaustive high-priority seed set includes `CWE-78` (OS command injection), `CWE-79` (especially stored XSS), `CWE-89` (SQL injection), `CWE-94` and `CWE-95` (code or eval injection), `CWE-134` (externally controlled format strings), `CWE-409` (highly compressed data), `CWE-502` (deserialization of untrusted data), `CWE-611` (external entity references), `CWE-917` (expression-language injection), `CWE-1333` (inefficient regular-expression complexity), and `CWE-1336` (template-engine injection). The review asks whether the trigger can remain effective when carried in a string, file, message, record, log entry, or other content through an intermediary, without requiring the attacker to remain the vulnerable operation's connection peer. A CWE match alone must set neither `confirmed` nor `ruled out`. When authoritative or reproducible evidence confirms the behavior, the enrichment records $J(v) =$ confirmed — equivalently, `indirect_internet_trigger: true` in a Boolean representation — with provenance. Every affected finding then enters the indirect branch of Equation 2; it becomes IRV under that branch only where $C_0(a) = 1$ and $P_0(v, a) = 0$. The seed set is a review accelerator, not a definition or exhaustive allowlist of indirect-trigger vulnerabilities.

$P_0(v, a) = 1$ is a standing assurance claim, not a mathematical proof. It requires current evidence of the relevant scan or test scope, assessor-validated configuration, remediation and successful retest of relevant findings, and no known bypass or material change that invalidates the evidence.

A confirmed trigger outside the tested class, a new relevant finding, a failed retest, missing or stale coverage, or an alternate path sets $P_0(v, a) = 0$ until the record is re-established. This makes the presumption explicit and auditable instead of silently assuming that all downstream content is safe.

Equation 2 is not a placeholder for a provider-side system that FedRAMP expects but industry has not yet built. It is the operational classifier this paper recommends under the current rules and metadata environment. Its CVE-level promotion makes the exceptional transitive cases explicit, while P_0 reuses evidence providers and assessors already handle. Neither requires every provider to rediscover an unpublished trigger or prove every application data-flow edge.

4.3 Full-information reference extension (non-normative)

For interoperability and upstream metadata design, it is useful to show how more structured inputs fit without changing the recommended classifier's logic. Let $C(a)$ be the set of internet-originating content capabilities that reach component a, and let $X_C(v)$ be the content trigger profiles capable of triggering v. A content descriptor is

$$c = \langle \gamma, \phi, o, \iota, b \rangle,$$

where γ is the carrier (request field, uploaded file, queue message, database record, or log entry), ϕ is format and encoding, o is the processing operation (parse, deserialize, query, log, render, execute, or decompress), ι is the degree of attacker influence (data-only, structural/control, or arbitrary bytes), and b records whether exploitation requires the attacker to remain the connection peer.

Each documented data-flow edge e may apply a transformation $T_e : 2^{\mathcal{C}} \to 2^{\mathcal{C}}$. For a path p from a public entry point to a,

$$C(a) = \bigcup_{p \in \text{paths}(\text{internet}, a)} (T_{e_n} \circ \cdots \circ T_{e_1})\,(C_{\text{internet}}). \tag{3}$$

The full-information reference form is

$$\text{IRV}_F(v, a) = \mathbb{1}[N(a) \cap X_N(v) \neq \varnothing \;\vee\; C(a) \cap X_C(v) \neq \varnothing]\,. \tag{4}$$

Equation 4 is a reference semantics for publishers and tools that elect to supply richer inputs. It is not a claim that current providers possess those inputs, a recommendation that they build them now, or a new evidence obligation inferred from FedRAMP. Better upstream data may refine results over time without making Equation 2 unreasonable or incomplete for its declared operational purpose.

IN PLAIN TERMS

The first tuple is simply a detailed label for attacker-controlled content. It records five facts: what carries the input, its format, what the receiving component does with it, how much control the attacker retains, and whether the attacker must remain on the same connection. For example, it could describe an uploaded image containing arbitrary attacker-chosen bytes that a private worker later parses.

Equation 3 says to start with everything the internet can send and follow it along every path to the component. At each handoff, update the description to reflect what actually happens. A queue changes the carrier; protocol translation changes the protocol; schema validation may restrict the allowed format; and a parameterized query changes SQL influence from instructions to data. The transformations are applied in path order. The large union symbol means to combine the result from every possible path: if any path still delivers a capability, it remains in $C(a)$.

Equation 4 then asks two independent questions:

- Does the component's directly exposed network surface overlap with the network conditions the vulnerability needs?
- Does any internet-derived content capability that survives the path overlap with a content trigger the vulnerability needs?

The intersection symbol $\cap$ means "overlap," $\neq \varnothing$ means "at least one match exists," and $\vee$ means "or." If either the direct-network comparison or the carried-content comparison has a match, the finding is IRV.

Current tools generally cannot populate these detailed content descriptions and transformations. That is why the recommended operational profile uses the coarser CVE promotion $J(v)$, content-receipt tag $C_0(a)$, and standing prevention assertion $P_0(v, a)$. The full-information form shows how richer metadata could improve precision later; it does not make a complete data-flow graph a present requirement.

5 Computing the Direct Network Branch

The direct branch is the mature part of the model. It is an OR across every public ingress path and an AND of conditions within each path:

$$N(a) = \bigcup_{P \in \text{paths}_{\text{direct}}(a)} N_P(a).$$

A path contributes descriptors only when a route reaches a live listener and the relevant protocol/version survives the controls on that path. The following gates are implementation procedures, not independent definitions of IRV.

5.1 Gate 1: Addressing and routing

A direct path requires a public address or public ingress, an active route, and forwarding to the component. A public host path normally requires a public IP and an internet-gateway route. A load-balanced backend requires neither on the backend itself; the public load balancer and its target binding establish the path. Egress-only NAT does not create an inbound path.

5.2 Gate 2: Source restrictions

Firewall and load-balancer source restrictions determine who can attempt the path and should be recorded. Attribution and prefix breadth are useful triage signals, but they do not automatically prove NIRV: traffic from an attributed partner range can still originate on the public internet, and a compromised authorized source can still send a trigger. A source restriction removes the descriptor only when the provider's documented interpretation and evidence establish that the qualifying public source population cannot use the path. Otherwise it informs LEV and residual likelihood.

An aggregate source range broader than a `/20` may remain a useful review threshold, but it is a heuristic for examining allowlist quality, not a reachability boundary.

5.3 Gate 3: Edge authorization boundaries

A separate, bypass-free remote-access boundary can remove a backend from the direct internet surface. The method removes a path from $N(a)$ when an IAP, VPN, or comparable edge component authorizes the connection before traffic reaches the backend; fronts organizational personnel and tooling; is the backend's sole ingress path; and prevents direct invocation or spoofing around the boundary. The edge component remains directly internet-accessible, but the protected backend is

NIRV for that direct path. This is a path decision, not a claim that the boundary sanitizes payload content; a separately confirmed indirect content path is still evaluated under $J(v)$ and $C_0(a)$.

Application-level authentication is not an IRV/NIRV gate in this method. By the time an application's login, session, or authorization logic runs, traffic has reached the component. Application authentication may inform LEV or an attested mitigation. It may support a lower PAIN rating only when evidence shows that the enforced role, tenant scope, or privilege boundary reduces the customer effect of successful exploitation; the existence of a login alone does not lower PAIN.

5.4 Gate 4: Kubernetes and container routing

Container exposure is determined from the union of public Gateway API routes, Ingress resources, public `LoadBalancer` services, reachable `NodePort` services, and `hostNetwork`/`hostPort` bindings. Cluster state must be joined with cloud routing and firewall state: Kubernetes alone cannot say whether a node port is reachable from the internet. Missing either evidence source leaves the path unresolved.

5.5 Gate 5: Process, listener, protocol, and version

An exposed host does not make every installed package directly reachable. The component must map to a running process and a reachable listener that speaks a protocol and version in $X_N(v)$. Package inventory joined with process and socket ownership supplies the strongest direct mapping. Protocol termination also matters: an HTTP/2 flaw on a backend is not directly reachable through a load balancer that terminates HTTP/2 and speaks only HTTP/1.1 to that backend, provided no alternate HTTP/2 path exists.

No-listener evidence closes only the direct branch. A socketless worker can remain indirectly reachable if a confirmed trigger rides inside a file, message, record, or log entry it processes.

6 Proposed Vulnerability Trigger Profiles

The Vulnerability Trigger Profile is a proposed missing ecosystem artifact, not a record that current scanners, CNAs, or vulnerability-management platforms consistently publish or consume. A provider can implement the coarse $J(v)$ status today as a custom field or enrichment, but the structured profile below has no broadly adopted interchange or assurance standard. Its purpose is to show what upstream publishers and scanners would need to supply so the indirect branch can be automated consistently. A profile may contain several trigger alternatives because a CVE can have different mechanisms under different configurations.

```
cve: CVE-202x-xxxxx
product: example-parser
indirect_trigger:
  status: confirmed
  carrier: uploaded_file
  format: image/png
  operation: parse_metadata
  attacker_influence: arbitrary_bytes
  crosses_process_boundary: true
  source: vendor
  evidence: https://vendor.example/advisory
```

```
reviewed_at: 2026-07-09T00:00:00Z
```

At minimum the record needs affected product/version scope, one or more trigger profiles, provenance, publication time, assertion status, review status, and supersession information. Scanner vendors should carry the record next to the finding, but the authoritative assertion should normally originate upstream with the vendor, CNA, or publisher that knows the vulnerable code.

6.1 Agentic enrichment is a bridge, not the source of truth

Agentic analysis can read advisories and propose a profile once per CVE or product/version family. It cannot reconstruct facts that were never published. The operational tradeoff is unavoidable:

- A conservative agent abstains frequently, leaving the scalability gap.
- An aggressive agent increases coverage but creates unsafe false-negative risk when it concludes that no indirect path exists.
- Human review improves reliability but recreates the manual burden if it is applied to every finding rather than selected CVE groups.

Agent output should therefore cite the exact source for every asserted field, retain unknowns, record model and prompt versions, and require stronger review for negative conclusions than for promotions. A proposed indirect trigger can raise a CVE into the candidate set. An agent's unsupported conclusion that no indirect trigger exists must not be treated as authoritative evidence.

The enrichment queue should be risk-ranked: KEVs, active exploitation, high PAIN, high prevalence, and unresolved CVEs on components that commonly parse untrusted content go first. Results are cached and invalidated when vendor or threat intelligence changes.

7 Optional Deployment-Evidence Refinement

The current $C_0(a)$ predicate may come from a service archetype or governed assertion; that is sufficient for Equation 2. A provider or tool developer may elect to refine $C_0(a)$ with a graph whose directed edges describe actual content handoffs. Where used, declared policy and observed telemetry have different roles:

- Infrastructure policy, NetworkPolicy, security groups, and service configuration describe which paths are permitted.
- Flow, mesh, queue, trace, and host telemetry prove which paths have been observed and help discover undeclared edges.
- Absence of observed traffic does not prove a future path is impossible. Only enforcement or a governed completeness assertion can rule it out.

An optional higher-resolution edge assertion can bind behavior and evidence to the deployed artifact:

```
source: public-api
destination: order-database
channel: postgres
```

```
carries_internet_derived_data: true
transformations:
  - operation: parameterized_query
    effect: sql_structural_influence_to_data_only
evidence:
  source_commit: abc123
  artifact_digest: sha256:...
  sast_report: urn:...
  unit_test_report: urn:...
  integration_test_report: urn:...
  attested_by: ci.example.com
```

The binding is the important part:

edge → transformation assertion → test evidence → source revision → deployed artifact.

Vulnerability-management platforms generally cannot consume this chain as a portable finding-level reachability assertion today. The following are ecosystem extension points, not provider maturity levels or requirements imposed by this method:

Profile	**Capability**
Recommended now	Scanner findings, CVSS, direct exposure, CVE-level indirect-trigger promotion, coarse content-receipt tags, and standing assessment/ConMon prevention assertions.
Upstream reuse	Reusable CVE trigger profiles from authoritative or reviewed enrichment.
Optional graph	Imported or dynamically derived application data-flow graphs.
Optional attestation	CI/CD-generated edge assertions and evidence bound to deployed artifacts.
Optional automation	Continuous recomputation when vulnerabilities, code, topology, controls, or threat intelligence change.

8 Standing Prevention Evidence and Optional Transformations

The operational $P_0(v, a)$ predicate does not depend on the optional full-information graph. FedRAMP's Rev5 ConMon Playbook requires monthly scans of web applications and all web interfaces or an approved sample, individual tracking of findings, and assessor-validated scanner configuration. Providers maintain the evidence and remediate findings; assessors verify the integrity and results of provider-run scans. The 2026 VDR rules more generally require persistent detection, validation, mitigation, and remediation. [3, 4]

Together, those existing activities create a standing behavioral assertion for the application and input-control classes actually covered by the validated test program. P_0 makes that assertion explicit and revocable. It does not claim that a clean DAST result discovers every trigger, proves every internal path, or covers a vulnerability mechanism that the test plan does not exercise. OWASP likewise cautions that automated scanners have coverage and false-negative limits and should be used as part of a balanced testing program. [7]

Where a provider elects to support a higher-resolution transformation-based NIRV assertion, the content algebra makes the same reasoning more precise. A transformation does not remove a CWE label; it changes a capability carried by the content.

- **Queue or storage handoff:** changes the carrier while normally preserving attacker influence and format.
- **Parameterized SQL:** changes SQL influence from structural/control to data-only. The attacker string still reaches the database, but not as SQL grammar.
- **Contextual output encoding:** changes HTML structural influence to text for the specific rendering context.
- **Protocol termination:** changes protocol or version, such as `h2` to `http/1.1`.
- **Schema validation:** restricts content to the language actually enforced by the schema; it does not generically remove deserialization or parser risk.
- **Content disarm and reconstruction:** replaces unsafe document structures with a reconstructed format under an explicit policy.
- **Unknown transformation:** preserves a confirmed applicable trigger capability until evidence establishes otherwise.

A content-trigger finding may earn NIRV operationally when the applicable P_0 assertion is current. A higher-resolution claim may show that every path carrying internet-derived input to the vulnerable operation contains a transformation that prevents the specific trigger. The evidence hierarchy is:

Weight	Examples
Operational	Assessor-validated scanner configuration and scope; recurring DAST of the relevant interface and control class; relevant findings remediated and retested; and no known bypass or material invalidating change.
Higher-resolution	Enforced path absence; protocol termination; affected code absent; complete code/data-flow proof; typed or parameterized APIs; and targeted regression or exploit tests bound to the deployed artifact.
Supporting	Scoped SAST, runtime observations, flow telemetry, unit and integration tests, and general control assessment.
Insufficient alone	A clean scan with unknown scope or configuration, quiet telemetry, an undocumented assertion, or absence of observed execution during a finite window.

FedRAMP's guidance expressly recognizes prevention before vulnerable processing as relevant to IRV.[2] The provider produces and maintains the operational evidence and remediates findings. During assessment, the 3PAO collects or reviews that evidence, validates its scope and method, and verifies provider-run scan results. The assessor is not being asked to invent a complete data-flow graph or perform routine CVE trigger enrichment on the provider's behalf.

IN PLAIN TERMS

Regular DAST and remediation already say, in operational terms, "the tested input controls are working." The model uses that as a current, scoped presumption, not as proof that every possible payload is safe. If a new CVE has a trigger the testing did not cover, the presumption does not apply. The optional graph can support the stronger statement that a named control blocks a named trigger on every path, but the current method does not require providers to build that graph.

9 Triage, Grouping, and Reporting

The model retains more internal state than FedRAMP's binary reported field:

Confirmed IRV A direct match exists, or a promoted indirect trigger applies and is not covered by current prevention evidence.

Provisional IRV An applicable trigger is known, but deployment or prevention evidence is incomplete.

Evidenced NIRV The direct branch is empty and each promoted indirect trigger is ruled out or covered by a current P_0 assertion.

Indirect unknown The upstream record does not establish whether an indirect trigger exists; the baseline has not promoted it.

The last state exposes the principal operational presumption. Defaulting every unreviewed `AV:L/A/P` vulnerability to IRV would recover maximal propagation and defeat scalability. The present profile therefore treats an unknown indirect trigger as not promoted while preserving that status for risk-ranked enrichment. This can temporarily miss a rare undocumented indirect-trigger case. Threat-intelligence monitoring, KEV handling, vendor updates, agentic proposals, analyst promotion, and re-evaluation are the controls for that residual risk. The paper does not characterize the presumption as proof.

The second operational presumption is that a current P_0 remains valid for its documented interface, application archetype, and tested control class. Recurring scans and remediation refresh it. A relevant finding, material change, scope gap, new trigger technique, or contrary threat intelligence revokes or narrows it. This is what makes convergence conditional and defensible rather than a blanket claim that downstream content is sanitized.

Evaluation should be amortized across

$$(\text{CVE},\ \text{product/version family},\ \text{deployment archetype},\ \text{input/control archetype})$$

rather than repeated for every instance. One reviewed trigger profile can serve thousands of scanner findings; one deployment archetype can supply the same path and standing-control conclusion to a fleet of identically managed assets. Instance-level exceptions remain possible and must break out of the group.

The resulting IRV value plugs into the existing PAIN/LEV matrix without changing PAIN:

$$\text{column} = \begin{cases} \text{LEV+IRV} & \text{LEV} \wedge \text{IRV}(v, a), \\ \text{LEV+NIRV} & \text{LEV} \wedge \neg\text{IRV}(v, a), \\ \text{NLEV} & \neg\text{LEV}. \end{cases}$$

The provider may retain richer internal statuses and evidence while reporting the binary value required by `VER-RPT-VDT`. Qualifying IAP/VPN edge authorization is already reflected in $N(a)$. Source allowlisting, application authentication, exploit maturity, and attacker prerequisites inform LEV or attested mitigation rather than being forced into the reachability predicate. Application authentication affects PAIN only when its enforced scope demonstrably reduces the customer effect of successful exploitation.

10 Worked Examples

1. **Local vulnerability on a public web host.** The CVE is `AV:L`, so it has no direct network descriptor and $D(v, a) = 0$. Its indirect status is retained separately. If $J(v)$ is unknown, the current profile does not promote it; if later analysis shows that an uploaded document invokes the vulnerable local parser, $J(v)$ becomes confirmed and the indirect branch is evaluated across all affected assets. A generic web scan does not set $P_0 = 1$ for that parser unless its validated scope actually exercises the relevant upload and parser trigger class.

2. **SSH CVE on a public web host.** The host exposes HTTPS publicly, but SSH is reachable only through a path that the provider has evaluated outside the qualifying public population. The finding's required surface contains SSH; $N(a)$ for the qualifying internet path does not. Their intersection is empty even though the host is public. Source attribution alone is not the proof; the recorded path interpretation and enforcement are.

3. **HTTP/2 flaw behind protocol termination.** A public load balancer accepts HTTP/2 but opens HTTP/1.1 connections to the backend. The backend finding requires `h2`. With the load balancer as the only path, the transformation removes `h2` from N(backend) and the finding is NIRV. The same flaw on a self-managed load balancer remains IRV.

4. **Administrative service behind a qualifying IAP.** The IAP is the service's only ingress path and authorizes organizational personnel before forwarding any request. The IAP remains internet-accessible; the backend is removed from $N(a)$ and is NIRV for the direct branch. A login implemented by the backend application would not produce the same result, because the request reaches that component before the login logic runs.

5. **Known indirect logging trigger.** Threat intelligence establishes that a crafted string can trigger a logging library after crossing process boundaries. The CVE-level $J(v)$ is promoted once. Every asset archetype tagged $C_0(a) = 1$ and running the affected library enters the IRV candidate group. Ordinary SQL-injection or cross-site-scripting DAST coverage does not establish P_0 for a logging-expression trigger, so the finding remains IRV unless a targeted control and its current evidence cover the trigger. This is the current-profile treatment of the Log4Shell shape.

6. **SQL construction behind an application tier.** Internet-derived values reach the database, but every application path uses parameterized queries. The values therefore reach SQL only as data; they cannot become SQL instructions. If the validated assessment and recurring DAST program covers SQL injection on the relevant interface, relevant findings have been remediated and retested, and no bypass or material change is known, then $P_0(v, a) = 1$ and the database finding is NIRV. The operational method does not require a new proof of every query edge. A new SQL-injection finding, an uncovered interface, or an unparameterized alternate path resets P_0 to zero.

7. **Installed but unobserved package.** Runtime telemetry shows that a package did not execute during the observation window. That is supporting evidence, not proof that its code is absent or can never execute. CycloneDX `code_not_present` means the vulnerable code has actually been removed or omitted; `code_not_reachable` requires a durable execution-path conclusion, not merely a quiet window.[6]

11 Perimeter Prevention and Mitigation

FedRAMP permits a finding to cease being considered IRV when a control verifiably prevents the triggering payload before the vulnerable resource processes it.[2] A WAF virtual patch, API-gateway schema, mail filter, or content-disarm boundary can qualify when it covers the actual trigger and every relevant path.

The stricter operational posture is still to preserve intrinsic reachability and record prevention separately. WAF rules are tuned, bypassed, moved into detection-only mode, and sometimes removed during incidents. Keeping the finding visible as intrinsically reachable while attaching a signed VEX or other mitigation assertion produces a record that degrades safely when the control fails. CycloneDX provides machine-readable impact-analysis states and justifications for such assertions.[6]

This does not conflict with P_0, which is refreshed through validated assessment and ConMon for scoped application and input-control classes. The paper keeps mutable, CVE-specific perimeter rules in the mitigation record unless the provider establishes the same scope, currency, and invalidation bar.

Known Exploited Vulnerabilities retain the remediation expectation in `VDR-TFR-KEV` even when fully mitigated.[3] The provider should therefore track reachability, prevention, mitigation, disposition, and the KEV clock as separate facts.

12 Limitations and Validation Requirements

This is a triage architecture, not a proof that all transitive triggers are known. Its principal limitation is deliberate: an indirect trigger whose semantics are absent from upstream metadata and not promoted by enrichment can remain outside $R_{\text{operational}}$. The method trades exhaustive recall for operational scale and makes that trade visible through the `unknown` state. Its second limitation is that P_0 is only as reliable as the validated scope, currency, coverage, remediation, and change-invalidation discipline of the assessment and ConMon program.

The claim that undocumented indirect-trigger vulnerabilities are exceptional should be measured rather than asserted. A reference implementation should be tested on a reviewed corpus containing direct protocol flaws, connection-bound flaws, local privilege vulnerabilities, file and media parsers, deserializers, logging and expression systems, and known transitive cases. At minimum it should report:

- false-negative and false-positive rates for indirect-trigger promotion;
- the proportion of CVEs left unknown;
- agreement across analysts, agents, model versions, and repeated runs;
- provenance coverage and the authority of cited sources;
- enrichment cost and latency per previously unseen CVE; and
- time from new threat intelligence to fleet-wide reclassification;
- DAST interface and trigger-class coverage, including authenticated paths; and
- time from a relevant finding or material change to invalidation of P_0.

When a provider elects to use the optional evidence graph to earn a narrower finding-level exclusion, that claim needs validation. Edge completeness, binding to deployed artifacts, change invalidation, and alternate-path discovery are security properties. A signed assertion is useful only if its scope is exact and its inputs remain true. None of this converts the optional reference form into a baseline requirement for providers using Equation 2.

13 Conclusion

FedRAMP's finding-level internet-reachability requirement exposes a metadata problem upstream of any individual provider. Direct reachability can be computed reasonably well from network, runtime, and scanner evidence. The primary obstacle to scalable transitive classification is that today's vulnerability sources and scanners do not reliably identify which CVEs have an indirect internet-content trigger. More detailed deployment evidence can refine where a known trigger applies, but a complete application-wide evidence graph is not required by the operational method recommended here.

The operational profile in this paper is intentionally modest. It computes the direct branch, uses CVSS attack vector only for that branch, promotes known indirect triggers once per CVE, and joins them to coarse content-receipt archetypes. It also reuses assessment- and ConMon-backed prevention assertions for the input-control classes already tested. The resulting operational set therefore converges toward the directly accessible surface, plus known indirect triggers whose path or prevention evidence is missing, stale, failed, or out of scope. That is a conditional operational convergence, not a semantic claim that internet reachability and internet accessibility are identical.

Two explicit presumptions make the method scalable. Unknown indirect triggers are not promoted until upstream intelligence or review identifies them, and a current P_0 assertion is trusted within its validated scope until contrary evidence or change revokes it. The first creates residual risk from missing trigger metadata. The second creates residual risk from incomplete testing or stale control evidence. Threat intelligence, KEV handling, recurring DAST, finding remediation and retest, change detection, agentic proposals, selective analyst review, and rapid reclassification control those risks.

Equation 2 is the paper's recommendation for that current environment. The full-information form is non-normative: it documents an interoperability option if vendors, CNAs, scanners, or provider-selected tools later supply richer inputs. It does not establish that FedRAMP expects providers to build a complete data-flow attestation system now, and its existence does not make the present classifier deficient for its declared triage purpose. Existing DAST and remediation evidence is sufficient for the paper's scoped standing-control presumption; the optional graph would improve precision, not create the premise from nothing. The durable near-term improvement is upstream: publish provenance-backed trigger profiles once and carry them through scanners so providers can promote exceptional indirect cases consistently. Agentic enrichment can bridge that gap, but it does not become the authority for facts no source has disclosed.

References

[1] FedRAMP, *FedRAMP Definitions: Internet-Reachable Vulnerability*, Consolidated Rules for 2026.
FedRAMP definitions

[2] FedRAMP, *Vulnerability Evaluation and Reporting*, Consolidated Rules for 2026, including `VER-EVA-EIR`, `VER-EVA-GRV`, and `VER-RPT-VDT`.
FedRAMP VER reference

[3] FedRAMP, *Vulnerability Detection and Response*, Consolidated Rules for 2026, including `VDR-TFR-PVR` and `VDR-TFR-KEV`.
FedRAMP VDR reference

[4] FedRAMP, *Continuous Monitoring Playbook*, Version 1.0, November 17, 2025, vulnerability scanning and assessor responsibilities.
FedRAMP ConMon Playbook

[5] Forum of Incident Response and Security Teams (FIRST), *CVSS v4.0 User Guide*, "Attack Vector Considerations."
FIRST CVSS v4.0 User Guide

[6] OWASP CycloneDX, *CycloneDX v1.7 Specification Reference*, impact analysis states and justifications.
CycloneDX v1.7 reference

[7] OWASP Foundation, *Web Security Testing Guide v4.2*, introduction and limitations of automated testing.
OWASP WSTG v4.2 introduction

Glossary of Abbreviations

`3PAO`	Third-Party Assessment Organization	`ALPN`	Application-Layer Protocol Negotiation
`CISA`	Cybersecurity and Infrastructure Security Agency	`CNA`	CVE Numbering Authority
`ConMon`	Continuous Monitoring	`CPE`	Common Platform Enumeration
`CVE`	Common Vulnerabilities and Exposures	`CVSS`	Common Vulnerability Scoring System
`CWE`	Common Weakness Enumeration	`DAST`	Dynamic Application Security Testing
`IRV`	Internet-Reachable Vulnerability	`KEV`	Known Exploited Vulnerability
`LEV`	Likely Exploitable Vulnerability	`NIRV`	Not Internet-reachable Vulnerability
`PAIN`	Potential Agency Impact N-rating	`SAST`	Static Application Security Testing
`VDR`	Vulnerability Detection and Response	`VER`	Vulnerability Evaluation and Reporting