The Subtractive Vulnerability & Path Erasure Standard

A framework for prioritizing remediation based on **Deterministic Solvability**.

Purpose

The goal of this standard is to maximize the **Path Erasure Rate (PER)**. We prioritize fixes based on how much attacker capability disappears once the issue is resolved. The goal is **not to primarily fix the most findings**, but to **remove the attack paths that attackers actually rely on**.

This guideline defines how we prioritize remediation work based on **attack-path elimination**, not vendor severity labels alone.

Core Principle

We prioritize fixes based on how much attacker capability disappears once the issue is resolved.

A finding matters if fixing it makes real attacks **impossible**, regardless of severity.

How We Rank Findings

Tier 1 – Attack-Path Eliminating (Highest Priority)

Fix immediately, regardless of vendor severity.

These findings:

- Enable **credential theft**
- Enable **lateral movement**
- Enable **persistent or stealthy access**
- Enable **control-plane or identity escalation**

- Provide **direct entry into critical systems**

Examples

- IMDSv1 enabled on cloud instances
- Over-privileged service or instance roles
- Unrestricted outbound egress
- Legacy authentication protocols (NTLM, legacy OAuth, etc.)
- Personal or unmanaged CI/CD repositories for company code

Fixing these removes entire classes of attacks.

Tier 2 – Blast-Radius Limiting

High priority once Tier 1 paths are addressed.

These findings:

- Reduce attacker reach **after initial access**
- Limit how far an attacker can move
- Reduce data exposure or persistence options

Examples

- Excessive east-west network access
- Broad administrative scopes in SaaS / cloud
- Backup or snapshot deletion permissions
- Weak logging protections

Fixing these limits impact if something goes wrong.

Tier 3 – Execution Hardening / Hygiene

Important, but lower priority if no major paths remain.

These findings:

- Improve baseline security

- Reduce exploitability
- Do **not** by themselves stop end-to-end attacks

Examples

- Individual host vulnerabilities without chaining value
- Patch-only issues that require prior access
- Configuration hygiene with no identity or pivot impact

Useful, but rarely decisive on their own.

How Vendor Severity Is Used

Vendor severity (Critical / High / Medium / Low) is:

- A **triage input**
- **Not** a priority decision by itself

If a “Medium” finding:

- Enables credential theft
- Enables lateral movement
- Enables persistence → It is treated as **high priority**

If a “Critical” finding:

- Requires deep prior access
- Removes no attacker capability → It may be deprioritized

Key Question to Ask for Every Finding:

“What can an attacker no longer do if we fix this?”

If the honest answer is:

- “Steal credentials”
- “Move laterally”

- “Persist”
- “Access cloud or identity control planes”

Then the fix is **important**, regardless of severity label.

If the answer is:

- “Not much changes”

Then it is **lower priority**.

Severity-Based SLA Mapping

Vendor-provided severity ratings (Critical / High / Medium / Low) are used as **inputs** to the risk assessment process, but final remediation priority is determined by **attack-path impact**.

For audit and compliance purposes, findings are mapped into remediation timelines as follows:

Effective Severity Classification

A finding is assigned an *effective* severity based on its attack-path impact:

- Any finding that **enables credential theft, lateral movement, persistence, or control-plane escalation** is treated as **High or Critical**, **regardless of vendor severity label**, and remediated within the High/Critical SLA.
- Findings that **do not enable attack paths** but reduce blast radius or impact are treated as **Medium**, even if vendor severity is higher.
- Findings that represent hygiene or baseline hardening without attack-path impact are treated as **Low**, unless otherwise justified.

This approach ensures that remediation timelines are based on **actual risk**, not theoretical severity scores.

All re-classifications are documented and consistently applied.

Minimal Functional State

Remediation should prioritize returning the system to its **Minimum Functional State**—any vulnerability that represents 'Feature Bloat' or unnecessary connectivity should be solved via **Subtraction** (removal) rather than **Additive Hardening** (patching).

Asset Decommissioning

Sometimes the safest fix is to **remove the asset entirely**.

- Decommissioning a high-risk, low-value system eliminates **100% of its attack paths**
- This is always considered a **security win**
- Asset removal is treated as **risk elimination**, not metric gaming

---

Measuring Success

Success is measured by:

- Fewer viable attack paths
- Fewer recurring security events
- Reduced alert volume
- Systems that behave predictably under failure

Success is **not** measured by:

- Number of findings closed
- Amount of scanning performed
- Alert volume or dashboard coverage

Summary

- Severity labels help triage, **attack-path impact decides priority**
- Fix what removes entire classes of attacks first
- Design systems where attacks fail quietly
- Fewer paths = less risk = quieter operations

A finding we have to repeatedly detect is a path we failed to eliminate.