Welcome to the Post Patching Era

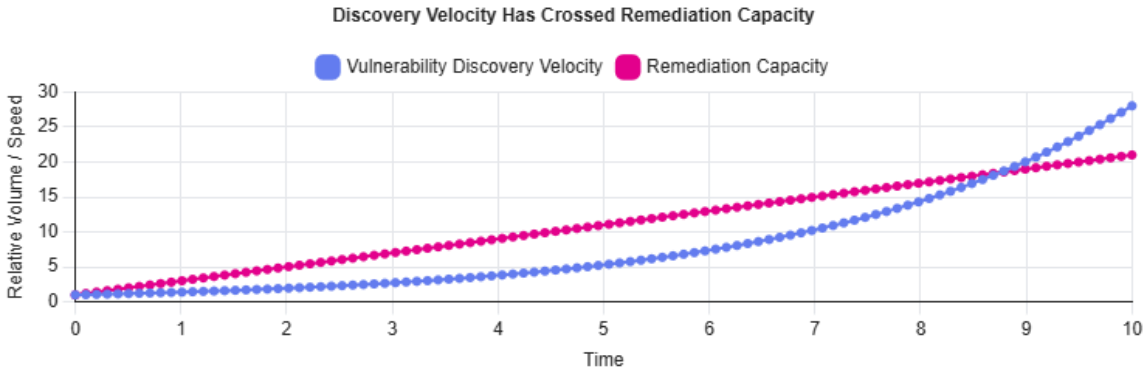
Why vulnerability at machine speed forces a structural shift in cyber defense

FS-ISAC Insurance Risk Council
April 15, 2026

Christopher Frenz

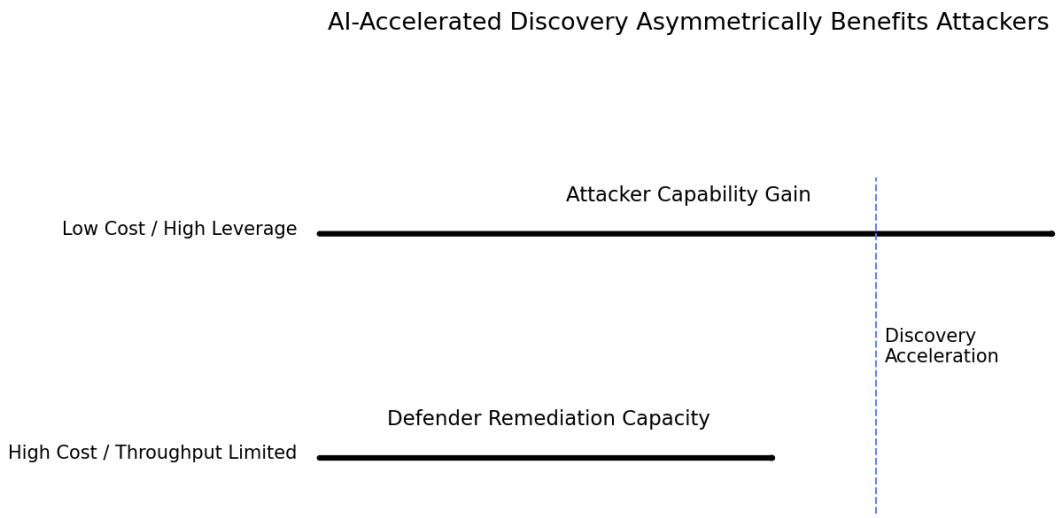
The Inflection Point

- AI systems can now discover and chain vulnerabilities faster than humans can review, test, and deploy patches
- Discovery velocity has crossed remediation capacity
- This is not theoretical – projects like Glasswing make it explicit
- This is now a “physics” problem and not a tooling gap



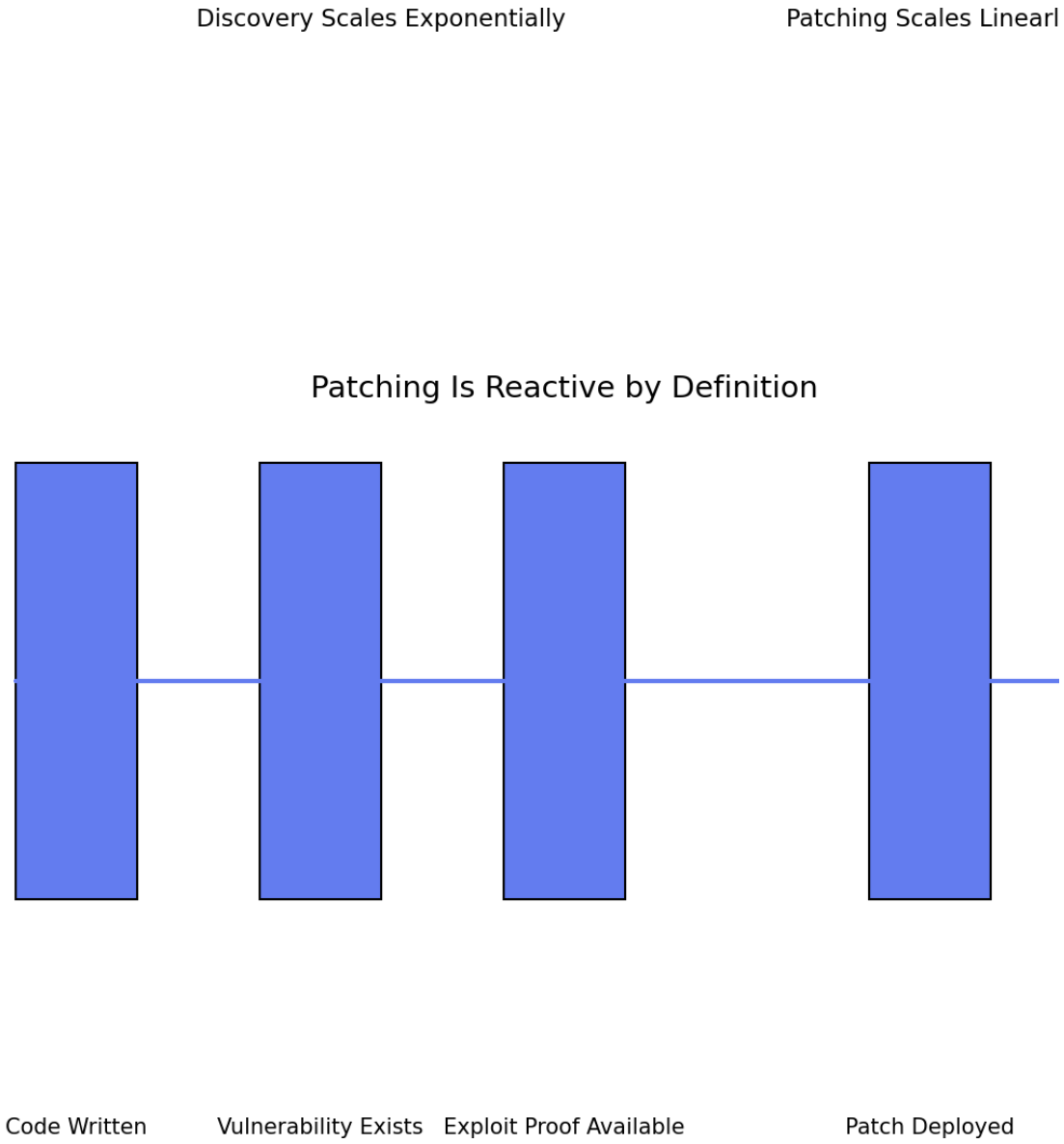
The Question No One Is Asking

- What good is finding 1,000 vulnerabilities if I can’t safely fix 1,000 vulnerabilities?
- Patch pipelines are throughput-limited by design
- Change risk grows superlinearly with volume
- Backlogs are not a management failure—they are a mathematical certainty
- Tools like Glasswing asymmetrically benefit attackers



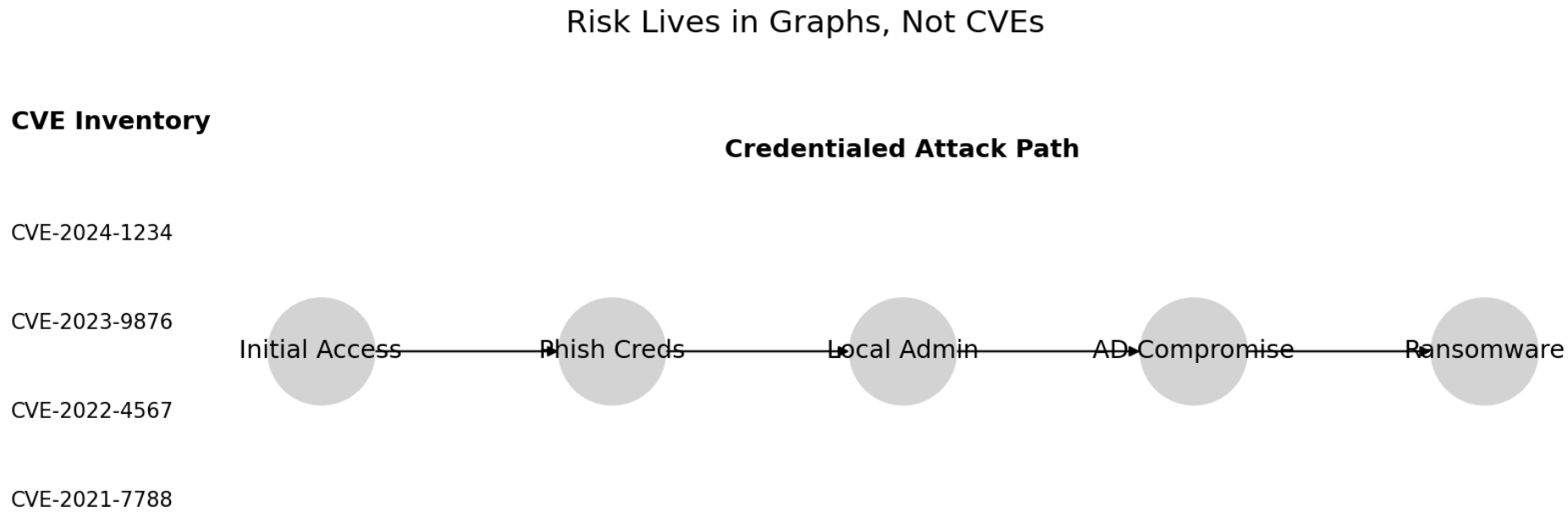
Why the Patch-Centric Model Breaks

- Patching Is Reactive by Definition
  - Occurs after code is written
  - Occurs after the vulnerability exists
  - Often occurs after exploit proofs exist
  - Scales linearly while discovery scales exponentially
  - You cannot “prioritize” your way out of a scaling mismatch
  - While patches will still matter they can no longer be a primary defense



The Hidden Assumption We've Been Making

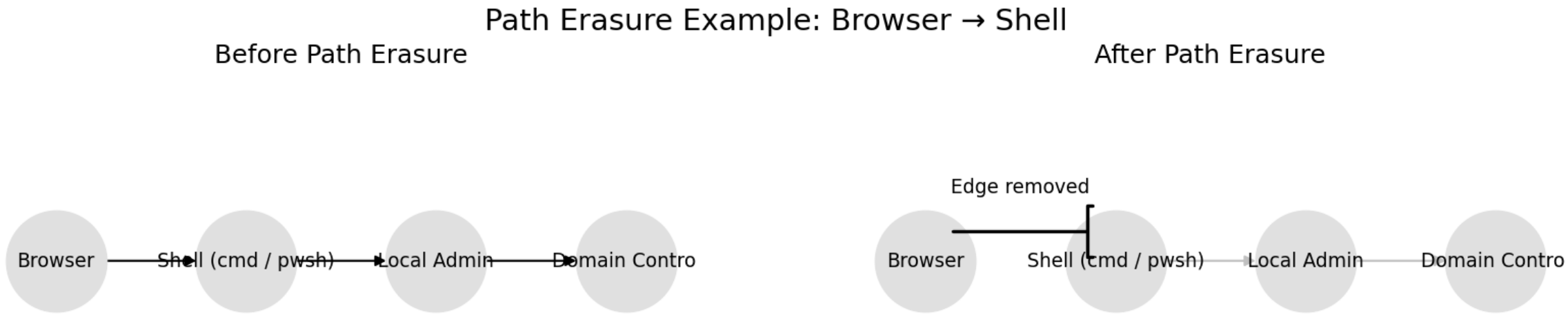
- We Act Like Vulnerabilities = Risk
- But in reality:
  - Vulnerabilities only matter if they can compose into control, movement, or impact
  - A defect without a path is noise
  - Risk lives in graphs, not CVEs



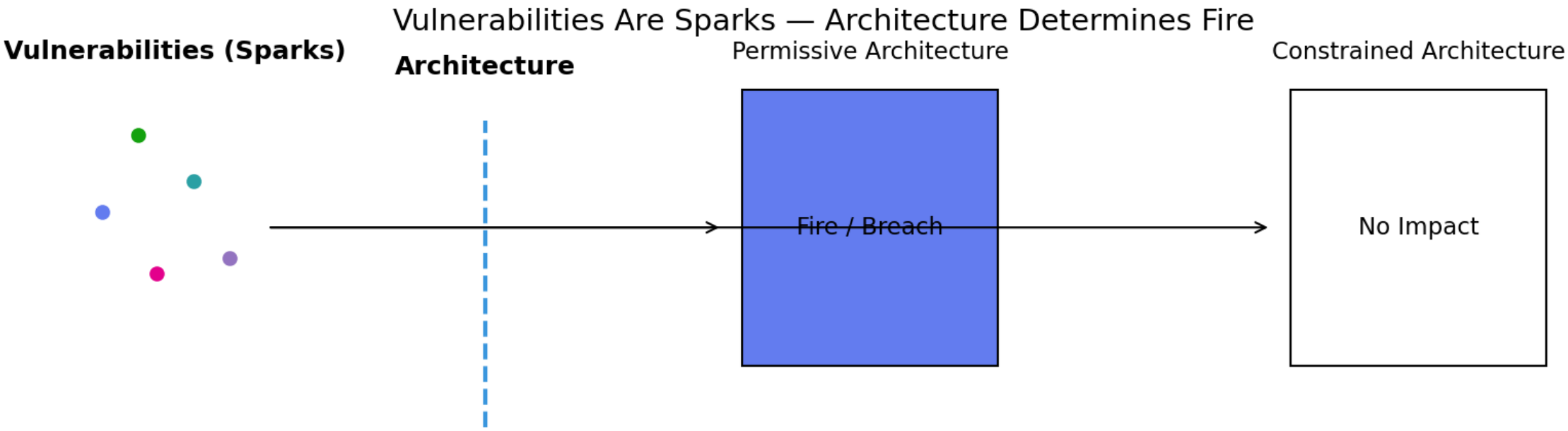
CVEs are potential inputs; attack paths are mandatory

Introducing Path Erasure

- Path Erasure: A Subtractive Model
- Path erasure is the deliberate removal of:
  - Trust relationships
  - Implicit privilege transfers
  - Lateral movement opportunities
  - Exploit chaining surfaces
- For example, there is almost never a legitimate need for a browser to launch a command prompt, Powershell, WMI, etc
- Removing the possibility of this occurring eliminates entire attack paths whether the browser is fully patched or not



Prevent vulnerabilities from becoming attacks—even when exploited



“Non-Conductive” Systems

- Vulnerabilities are sparks
- Architectures define whether sparks become fires

Compromised Workstation ≠ Credential Theft

What usually goes wrong

In most environments:

- A compromised workstation → memory access
- Memory access → LSASS extraction
- LSASS → cached hashes / tokens
- Credentials → lateral movement everywhere
- That implicit chain is not inevitable. It exists because of design choices.

Key Insight:
With path erasure the machine is lost. The credentials are not.

Path erasure example: Hardened LSASS

- Run LSASS as a protected process (PPL)
- Disable legacy credential caching
- Enforce Credential Guard / VBS isolation
- Remove the ability for non-trusted binaries to access credential memory
- Block unsigned code from interacting with authentication subsystems
- The workstation can be fully compromised but credential material never becomes available

Code Execution ≠ Lateral Movement

What usually goes wrong

Traditional security assumes:

- Code execution = movement opportunity
- Process execution = pivot
- Shell = network reach

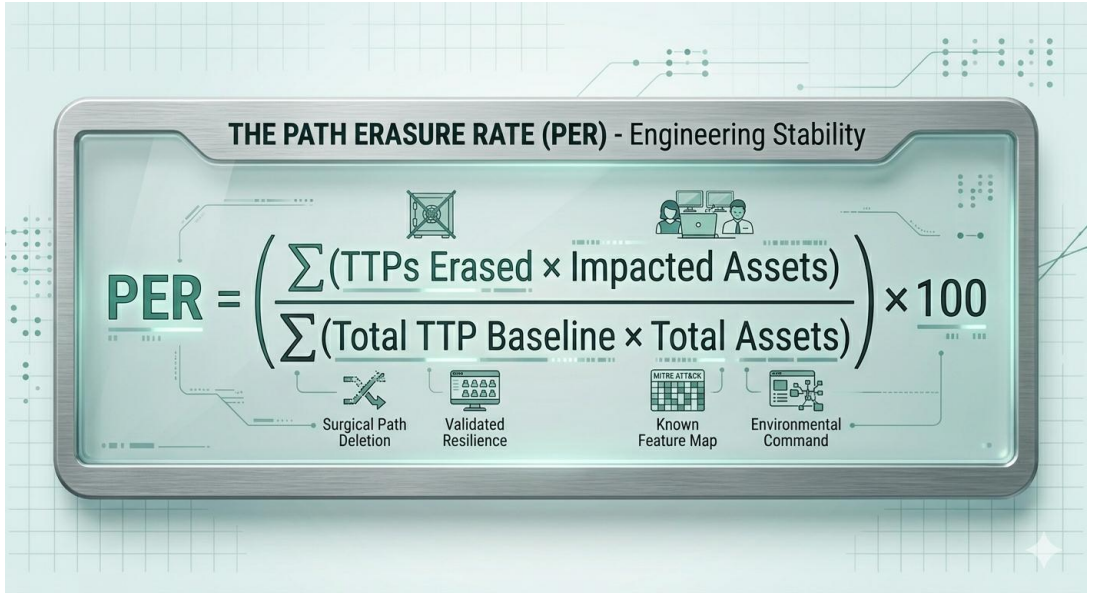
Key Insight
Execution is an event. Movement is a privilege.
Path Erasure removed the edge, even if the exploit could not be patched

Path erasure example:
Process-level execution constraints

- Prevent user-facing apps (browser, email client, PDF reader) from spawning:
  - shells (cmd, PowerShell, bash, zsh)
  - management tooling (wmic, sc.exe, net.exe)
- Enforce child-process allow-lists
- Enforce per-process network policy (no inbound, no east-west)
- Remove implicit trust between execution context and network movement

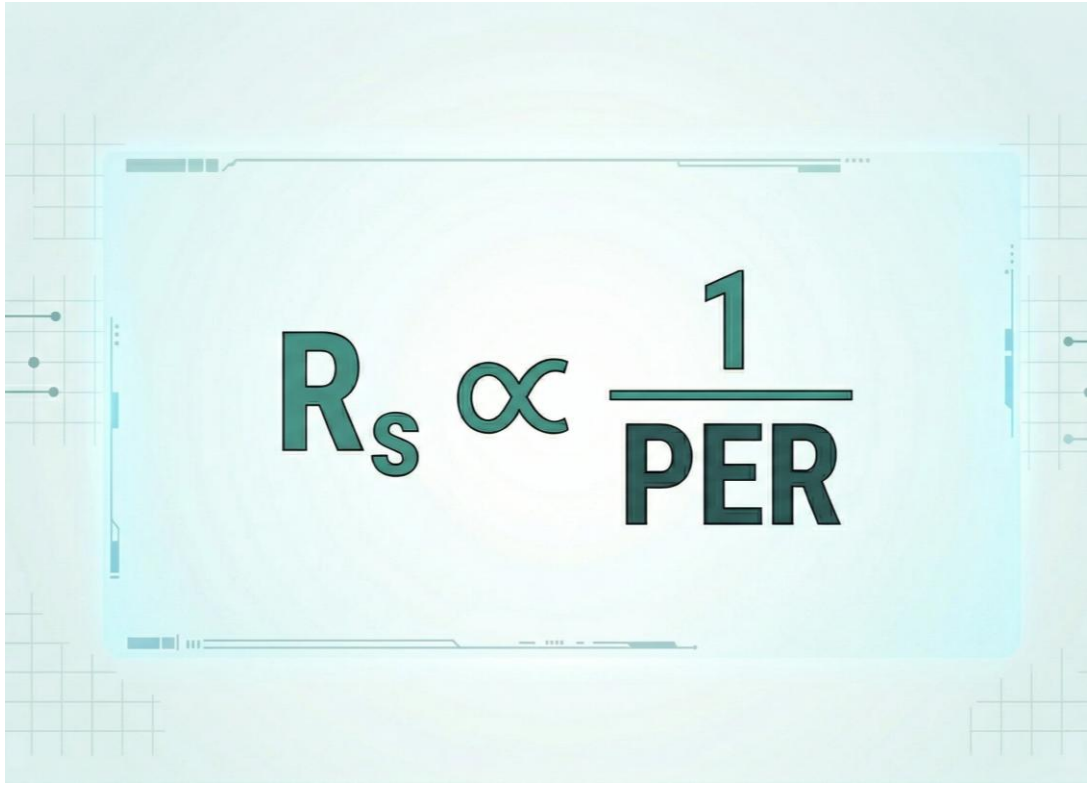
Path Erasure Rate (PER): Measuring Real Risk Reduction

- What PER Measures
  - Structural risk removal
  - Reduction in attacker optionality
  - Architectural efficacy — not activity
- What PER Ignores
  - CVE counts
  - Patch velocity
  - Alert volume
- A leading metric, not a lagging one
- Progress is fewer reachable paths, not fewer CVE findings
- PER does not require perfect enumeration — it tracks relative collapse of reachable paths over time.
- Submitted to OWASP for evaluation and standardization



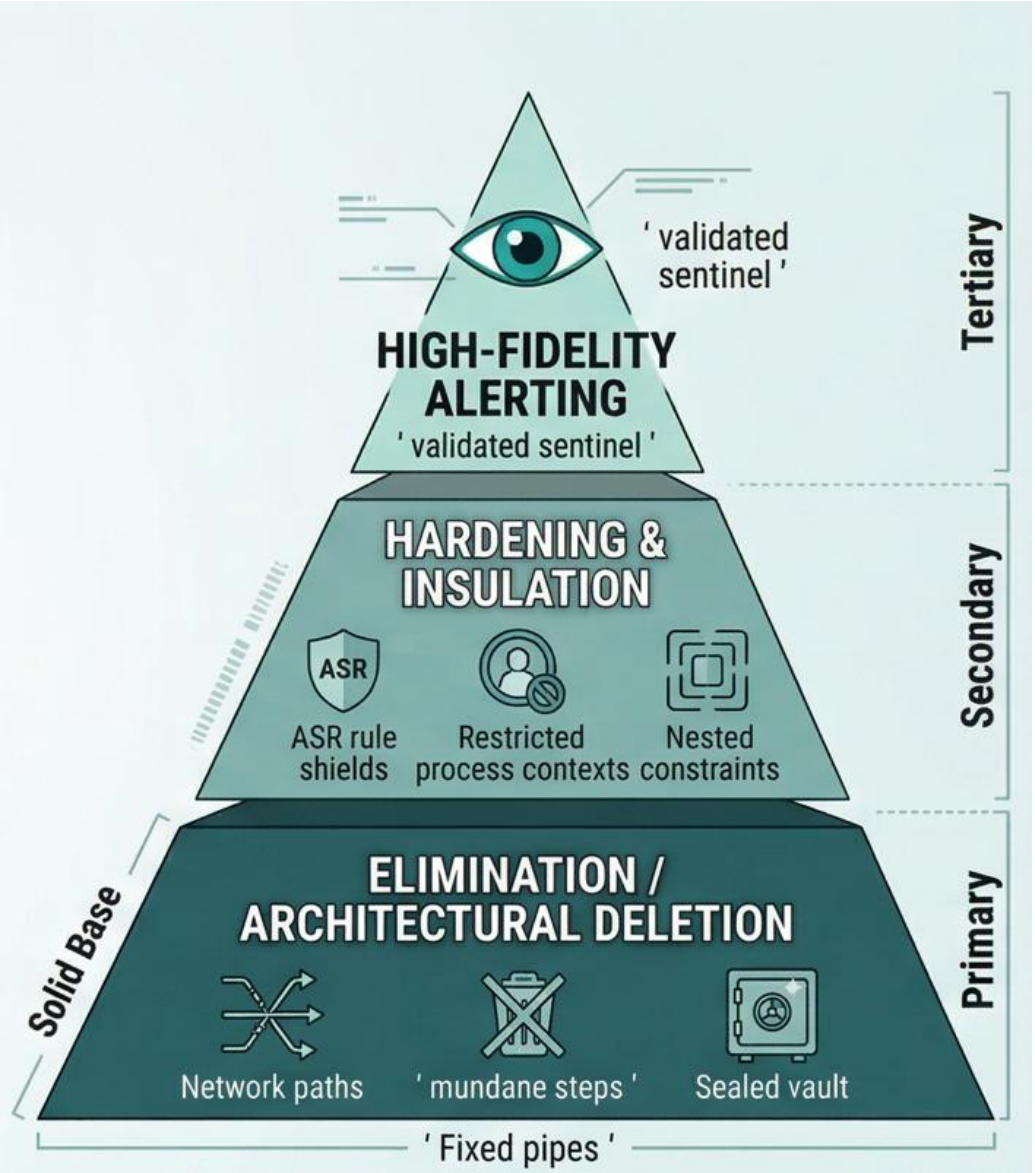
The Law of Subtractive Risk

- Law
  - Security improves fastest by removing attack paths,
  - not by adding reactive controls.
- Implications
  - Additive security increases complexity and failure modes
  - Subtractive security reduces system degrees of freedom
  - Fewer paths beats faster reaction
- Conclusion
  - The quietest environments are the safest
  - Silence is verified evidence, not neglect



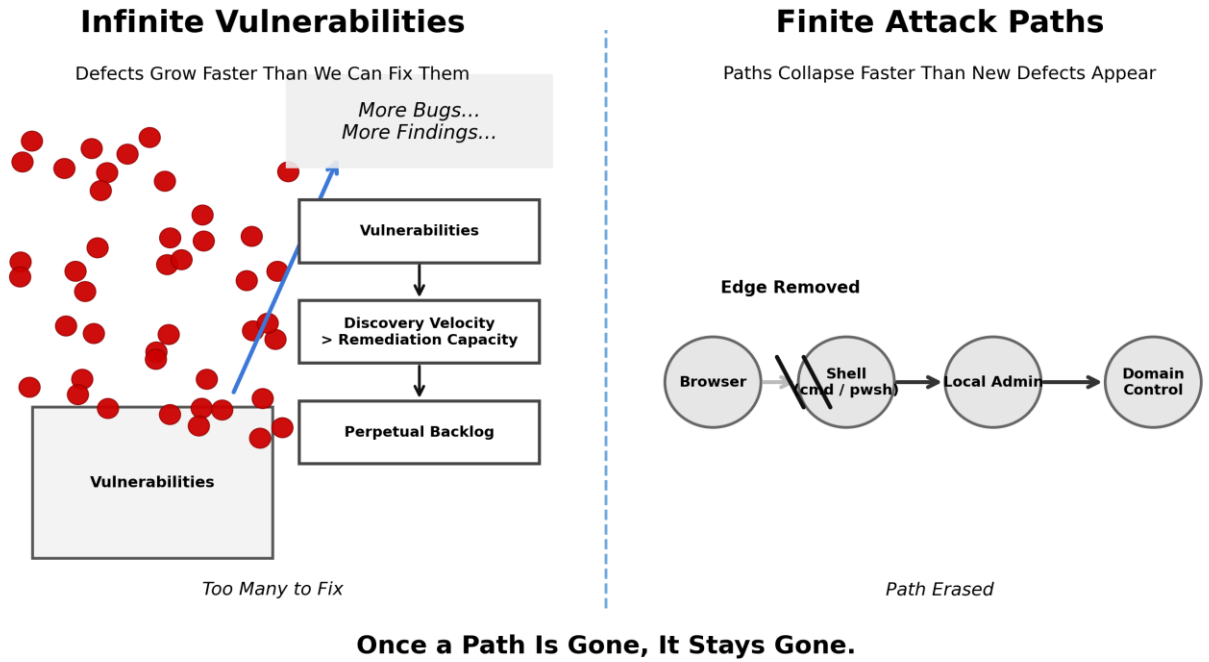
What Changes in a Post-Patching Era

- **Patches Still Matter—Just Not First**
- Primary defenses:
  - Path elimination
  - Privilege minimization
  - Deterministic network and identity boundaries
- Secondary controls:
  - Patching
  - Detection
  - Response



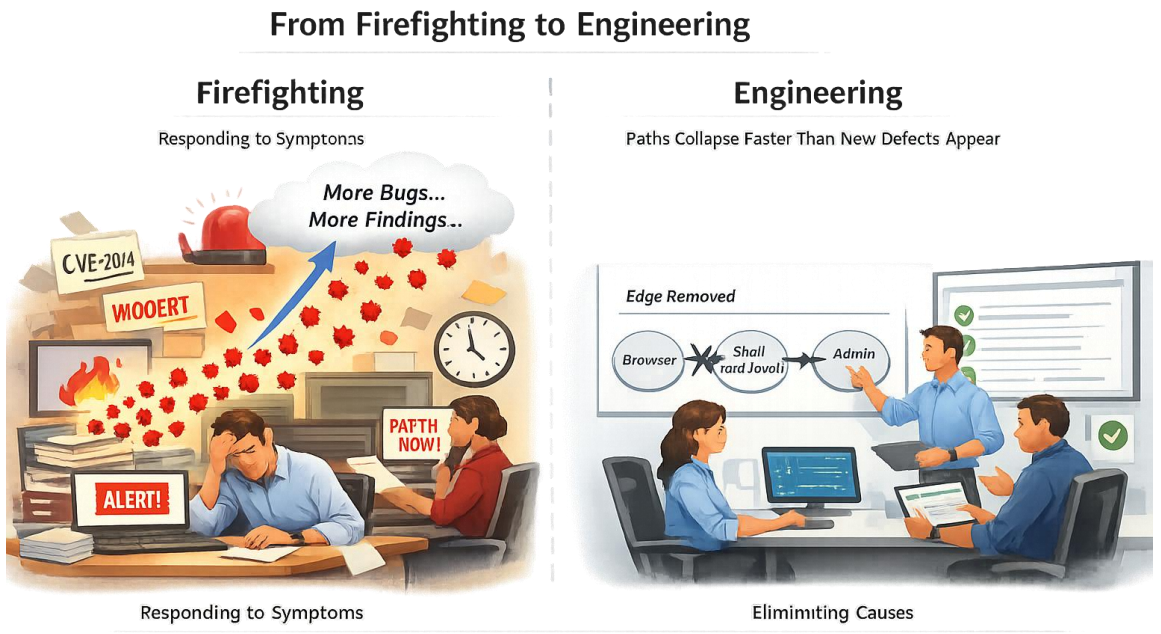
Why This Scales Better Than “Find & Fix”

- One architectural change can remove thousands of exploit possibilities
- Control efficacy improves even as defect count increases
- Security effort shifts from backlog management to design optimization
- Path erasure works in a world of infinite vulnerabilities. Defects grow faster than we can fix them, but paths collapse faster than new defects appear. Once a path is gone, it stays gone.



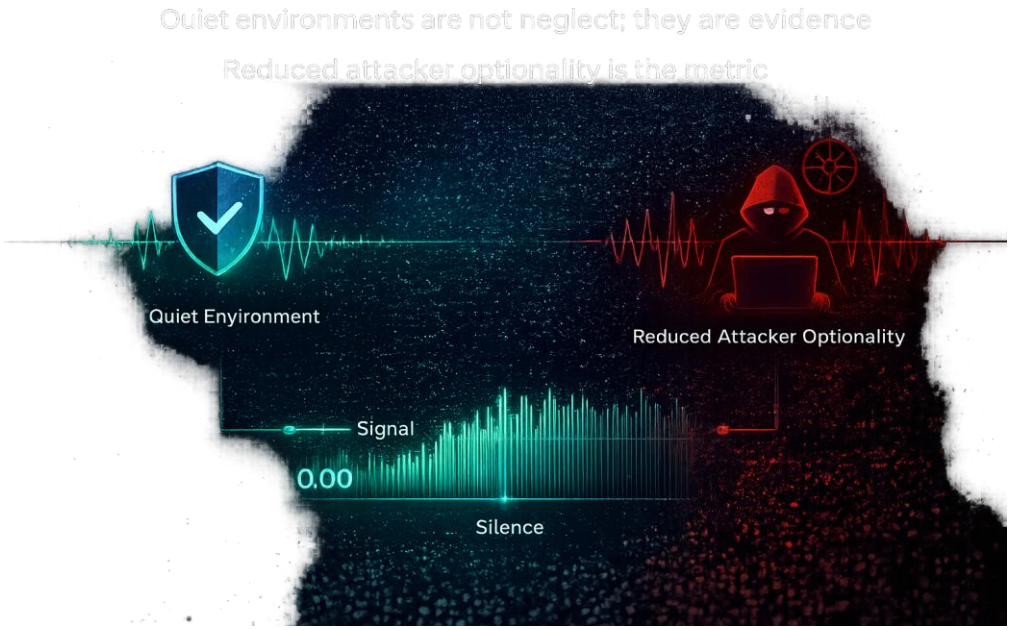
Implications for Financial Institutions

- Regulatory resilience improves (less dependence on patch emergencies)
- Ransomware blast radius collapses
- Third-party and zero-day exposure becomes tolerable, not catastrophic
- Security teams shift from firefighting to engineering
- Attacker economics are changed to make an organization a less desirable target



Progress Is Not Fewer Findings — It’s Fewer Paths

- Quiet environments are not neglect; they are evidence
- Reduced attacker optionality is the metric
- Silence becomes measurable signal



Post Patching is NOT No Patching

- It is:
  - Architecture first
  - Subtraction over accumulation
  - Systems thinking over alert consumption
- AI didn’t break security.
- It exposed which parts never scaled.