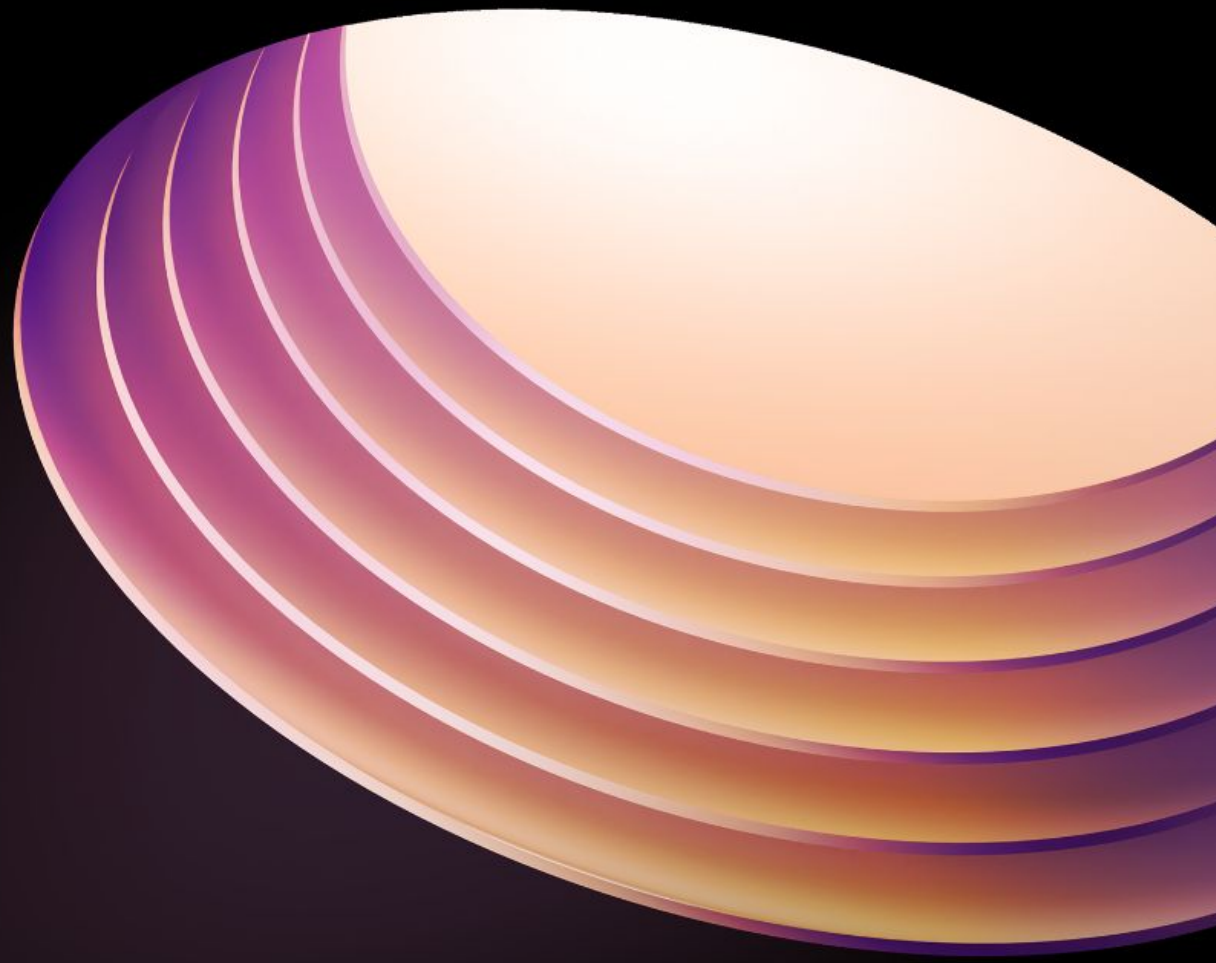
Computer Security In the AI Exploit Era

Patrick Ladd
Senior Technical Account Manager

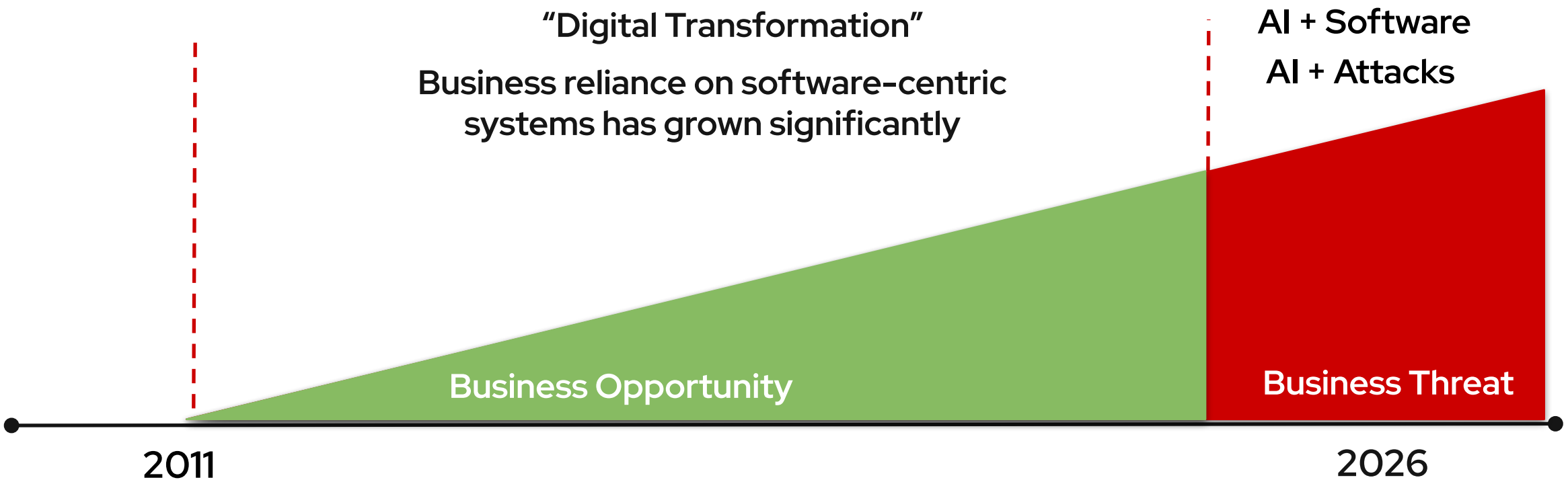
Ronny Lim
Senior Software Engineer



Red Hat | IBM

“Software is eating the world”

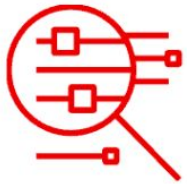
Your business is only as strong as your ability to manage software



Red Hat | IBM

The Mythos Moment

When discovery outpaces defense



AI discovers vulnerabilities at machine speed

Frontier models find critical zero-day vulnerabilities autonomously — including bugs hidden for 27 years — at under $50 each.



Vulnerability discovery now exceeds remediation capacity

Discovery is exponential. Remediation is linear.

Less than 1% of AI-discovered vulnerabilities have been patched.



Discovery is no longer the bottleneck. Response is.

4.8M Unfilled cyber roles. Growing 19% year over year.

The National Vulnerability Database is shifting to selective enrichment.



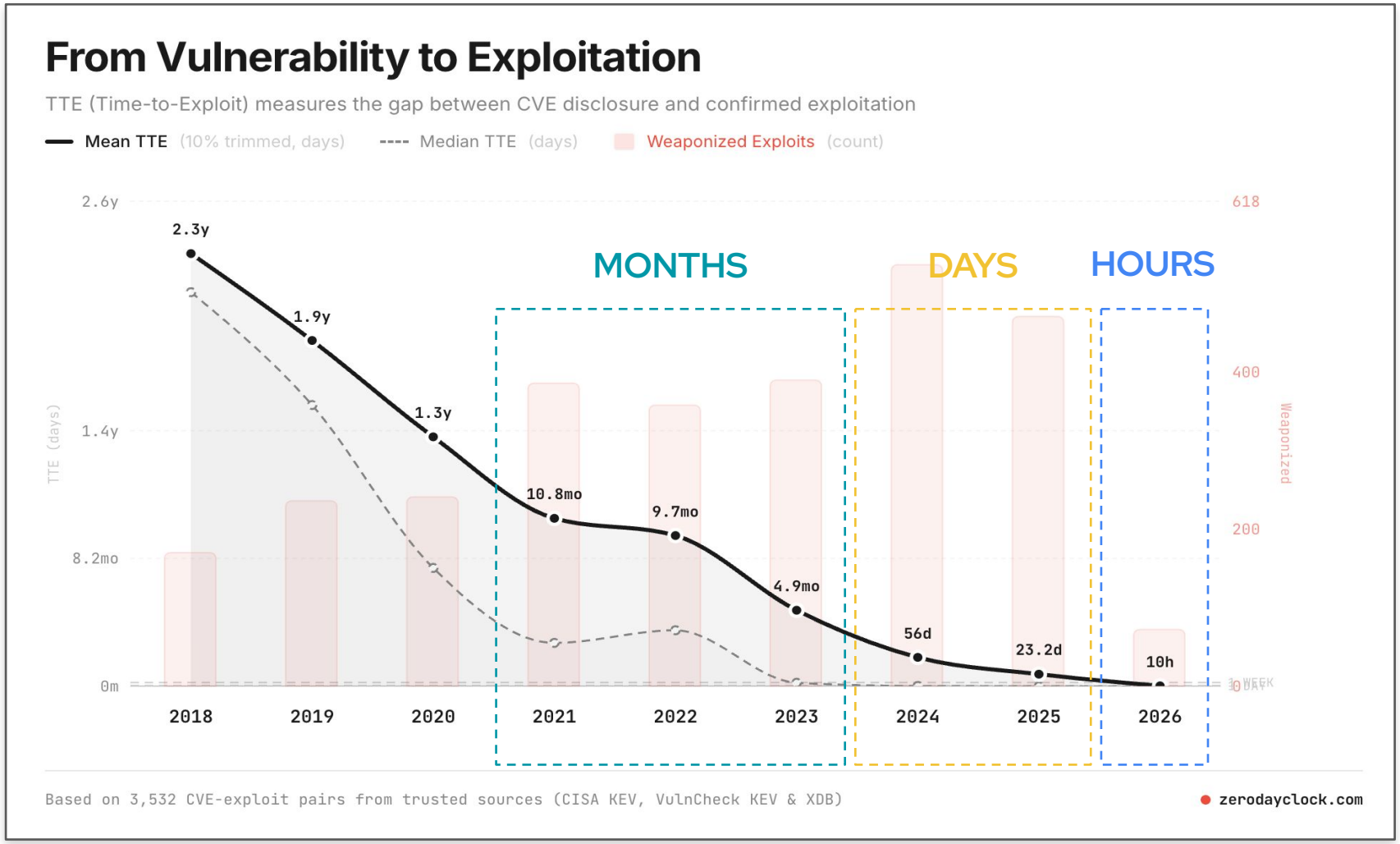
Shift from vulnerability management to exposure management.

The number of published CVEs has grown more than 520% since 2016.

Cost of attacks in 2025 was ~$60B. That's projected to grow to $138B by 2031.

Red Hat | IBM

AI is Accelerating the Security Vulnerability Landscape



Mythos Threats to Customers

- Automate Exploitations
- Chaining Vulnerabilities
- Expose Critical Infrastructure
- Increase Threat Volume
- Increase Threat Velocity
- Lower Barrier to Entry

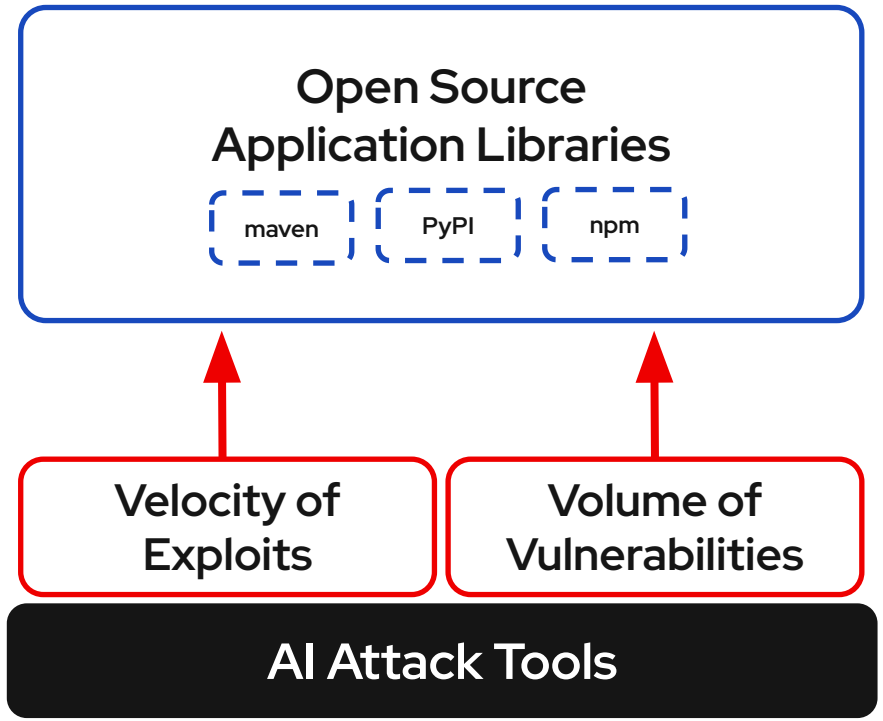
Source: [The "AI Vulnerability Storm": Building a Mythos-ready Security Program, Zero Day Clock]

The Mythos Moment

Exposing Two Parallel Challenges – "The Vulnerability Gap"

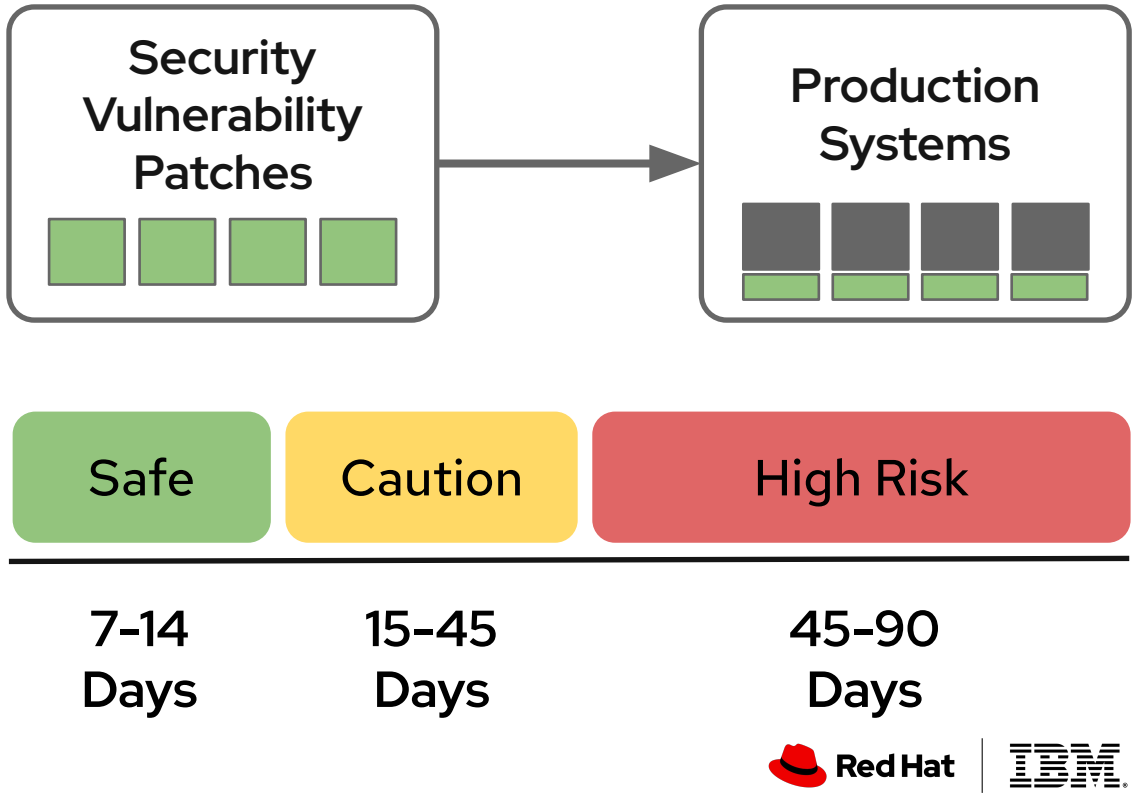
Massive CVE Growth

Problem: Time to Exploitation Dropping



Avg. Days Patch to Production: **40+**

Problem: Re-Build and Deployment Velocity



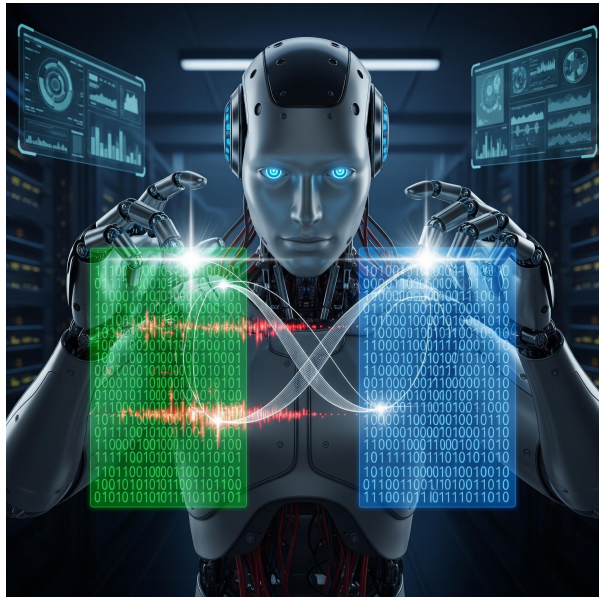
Red Hat | IBM

My Vendor Distributes Binaries!

AI Doesn't Care

- Exploiting Software Distributed as Binary:
  - Step 1: Disassemble Original Version
  - Step 2: Disassemble Patched Version
  - Step 3: AI analysis of patched "source code"
  - Step 4: Profit! (exploit…)

- Vendors distributing binaries are being exploited just as rapidly, because AI doesn't care what your source language is, and machine code is a language

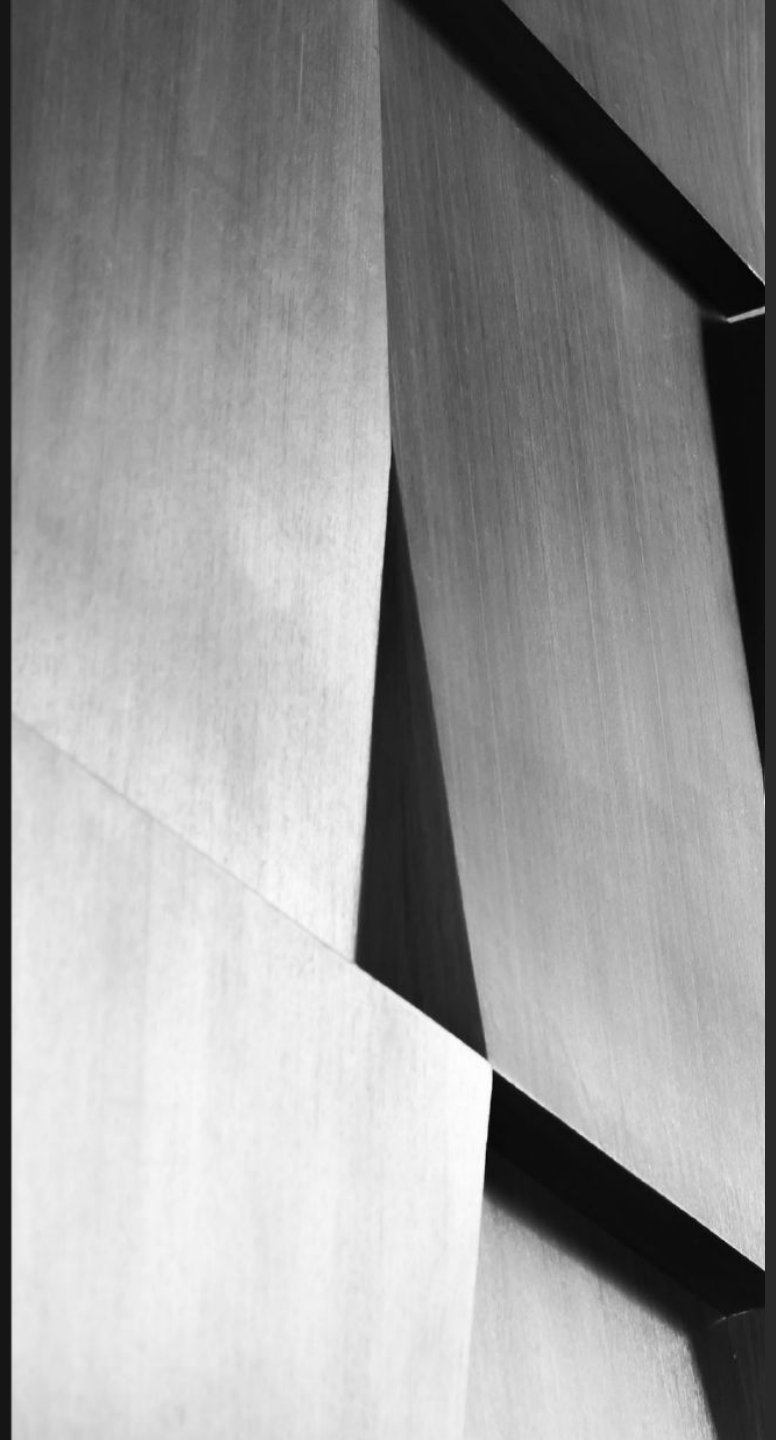


What's the solution?

The new normal

- Patch as Rapidly as Possible
  - Throughout your infrastructure! 😲
  - What to patch?
    - Hardware (firmware is software!)
    - OS Infrastructure
    - Layered Infrastructure
    - Application Stack
- Trusted Sources
  - Ability to respond quickly
  - Private pre-publication where possible
  - Red Hat Trusted Libraries -> Lightwell





THE VULNERABILITY GAP · PART 2 - BUILD AND DEPLOYMENT

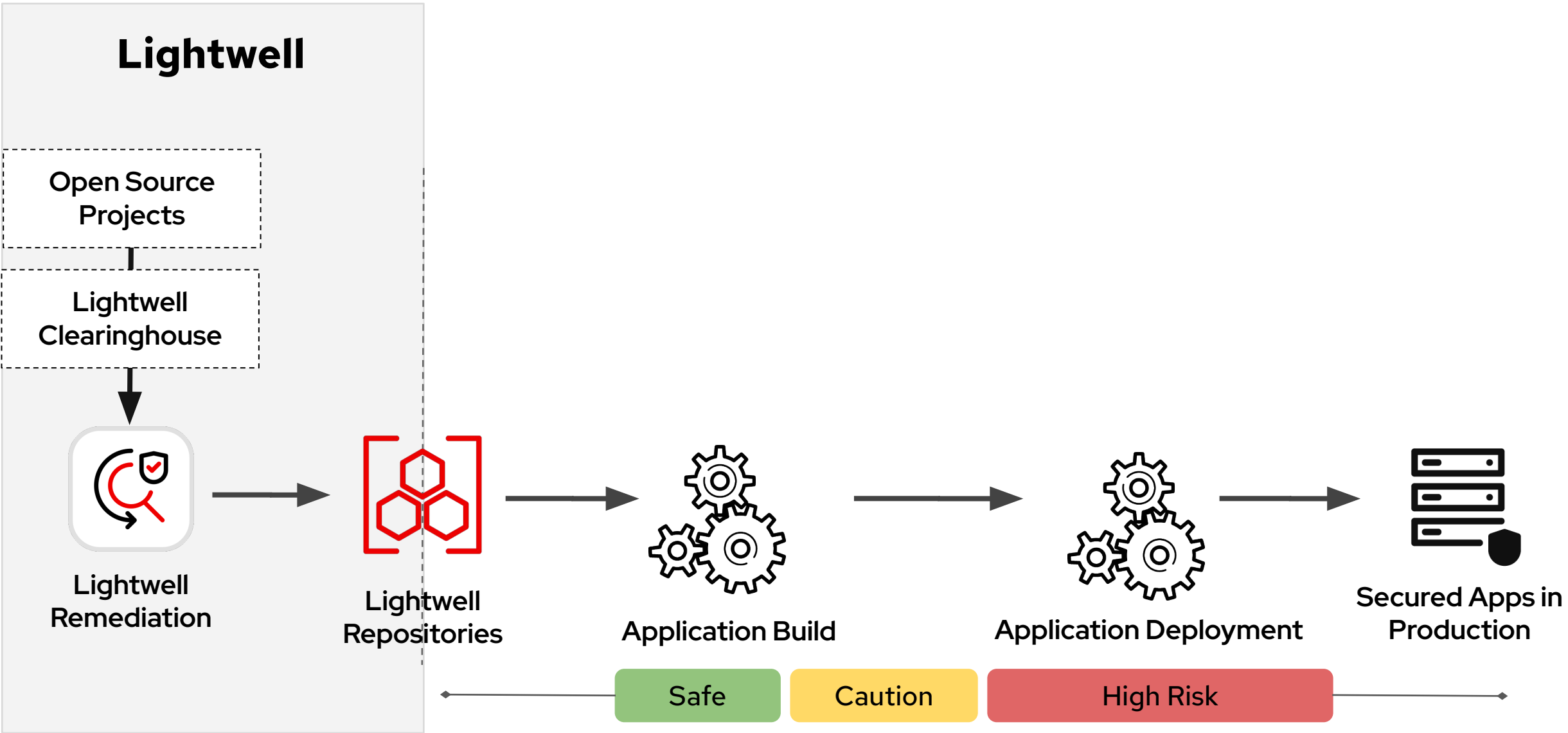
Patch to Production

Compressing the build, test, and deploy cycle from 45-90 days to 7-10 days

Red Hat | IBM

Lightwell is only part of the solution

Security Remediation + Rapid Deployment

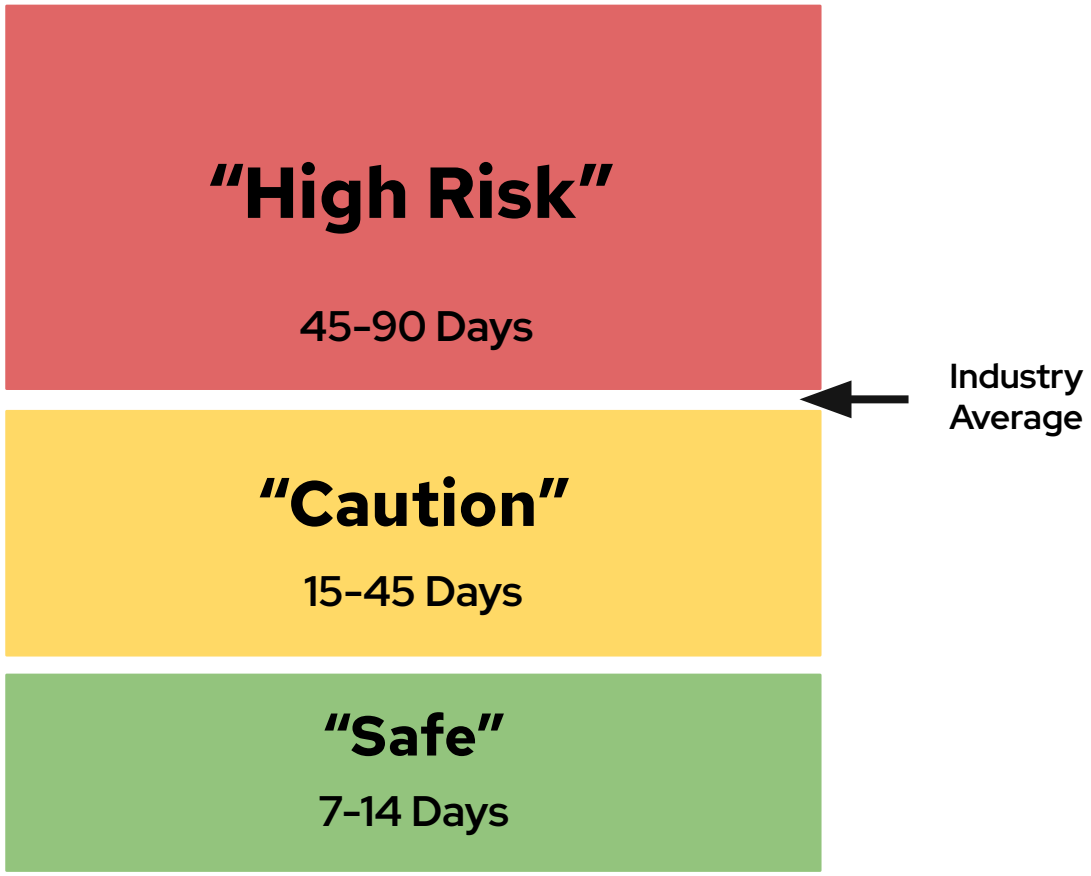


From Patch to Production?

Security Remediation is only part of the challenge

How long does it take to get from patch to production*?

(* not just Sev 1s)



Red Hat | IBM

AI cyber models found the bug in hours

Why does the fix still take months?

CVE Discovery

Hours - days (AI-assisted)

Remediation (Lightwell)

Days, not weeks

Patch-to-Production: build, test, sign, deploy

45-90 days today - this is today's bottleneck



The exposure window hasn't shrunk - it's moved.

As CVE discovery and remediation accelerate, the build-test-deploy cycle becomes the dominant share of total exposure time. Closing this gap is the only way the speed gains from Part 1 actually translate into a smaller attack surface.

Red Hat | IBM

45-90 days isn't one slow step

it's five compounding delays

No live inventory of affected systems

Teams spend days just figuring out which images and apps contain the vulnerable component - often manually grepping SBOMs that don't exist.

Manual rebuild & test cycles

Rebuilding an image and re-running the full test suite is a manual, often ad-hoc process repeated separately for every affected app.

Compliance and change-approval queues

Security and change-management review happens after the fact, often as a scheduled weekly or biweekly board meeting.

No automated trust chain

Signing and provenance, if done at all, are manual steps bolted onto the end - adding days and creating audit gaps.

Fragmented deployment tooling

Bare metal, VMs, and containers are patched with different tools and schedules - so one CVE means N separate rollout projects.

Red Hat | IBM

From "which apps are affected?" to an instant, scored answer

BEFORE: 5-10 days

- Security team circulates the CVE via email/ticket to every app team
- Each team manually checks Dockerfiles and dependency manifests
- Spreadsheets are compiled to track which images are affected
- No way to prioritize - every hit treated as equally urgent

COMPRESSED: Hours

- Apps built on **Red Hat Hardened|Trusted Images** start with a smaller, curated CVE footprint
- **Trusted Profile Analyzer**'s live SBOMs are queried the moment the CVE is published
- Every affected image, app, and environment is listed automatically, org-wide
- Exploitability and blast-radius scoring lets teams prioritize the handful that matter most

Red Hat | IBM

From a manual rebuild-and-hope cycle to an automated, signed rebuild

BEFORE: 10-20 days

- Developer manually bumps the dependency version in each repo
- Image is rebuilt by hand and pushed to a registry
- Signing, if performed, is a manual step someone has to remember to do
- No consistent provenance record - hard to prove what changed and why

COMPRESSED: Minutes to Hours

- The trusted software factory detects the new patched component and triggers an automated rebuild
- A pull request with the rebuilt image is generated for review - no manual repo edits
- **Red Hat Trusted Artifact Signer** automatically signs the artifact and attaches attestations
- Provenance and SLSA-aligned chain-of-custody are generated as a by-product of the build, not an afterthought

Red Hat | IBM

From scheduled test cycles and weekly change boards to continuous validation

BEFORE: 15-25 Days

- QA team queues the rebuilt image for the next scheduled test cycle
- Test environments are provisioned by hand and often drift from production
- Security/compliance review is a separate meeting on a weekly or biweekly cadence
- Any failure restarts the queue from the back

COMPRESSED: 1-2 Days

- **OpenShift Pipelines** triggers the test suite automatically on every signed, rebuilt image
- **Ansible** playbooks provision identical, production-like test environments on demand
- **Policy-as-code** evaluates compliance/security gates inline - no separate meeting
- Failures surface immediately with full context, so fixes loop back same-day

Red Hat | IBM

From manual promotion approvals to policy-driven, auditable promotion

BEFORE: 5-10 Days

- Each environment promotion (dev -> test -> stage -> prod) needs a separate sign-off
- Approvers re-verify the same compliance checks by hand at every stage
- Promotion status is tracked in spreadsheets or chat threads
- Rollbacks, if needed, are manual and poorly documented

COMPRESSED: Hours

- **GitOps**-based promotion automatically re-verifies signature, SBOM, and policy status at every gate
- **Trusted Profile Analyzer** continuously re-checks the image against current CVE and policy data
- A single declarative pipeline shows exactly where the artifact is and why it passed or failed each gate
- Rollback is a Git revert - fast, auditable, and consistent across environments

Red Hat | IBM

From three separate rollout projects to one coordinated deployment

BEFORE: 10-25 Days

- Bare metal patching scheduled separately via change windows and manual runbooks
- VM patching follows its own schedule, often with different tooling and owners
- Containerized workloads redeployed manually, app by app, cluster by cluster
- Each track reports status differently - no single view of overall rollout progress

COMPRESSED: 1-3 Days

- **Red Hat Satellite** manages content and patch lifecycle across bare metal and VM fleets from one console
- **Ansible Automation Platform** orchestrates the rollout - same playbooks, same approval flow, every tier
- **OpenShift** rolls out updated container images via automated, health-checked rolling updates
- One coordinated rollout, one status view, one timeline - across every footprint

Red Hat | IBM

Five compressions, one outcome:

7-10 day patch-to-production

Identify Exposure

Hardened/Trusted Images + Trusted Profile Analyzer



Rebuild, Sign & Attest

Trusted Software Factory + Trusted Artifact Signer



Test, Validate & Approve

OpenShift Pipelines + Ansible Automation Platform



Promote Through CI/CD

OpenShift GitOps + Trusted Profile Analyzer



Deploy at Scale

Satellite + Ansible + OpenShift



Before (manual) After (automated)

TOTAL CYCLE TIME (TARGET DAYS)

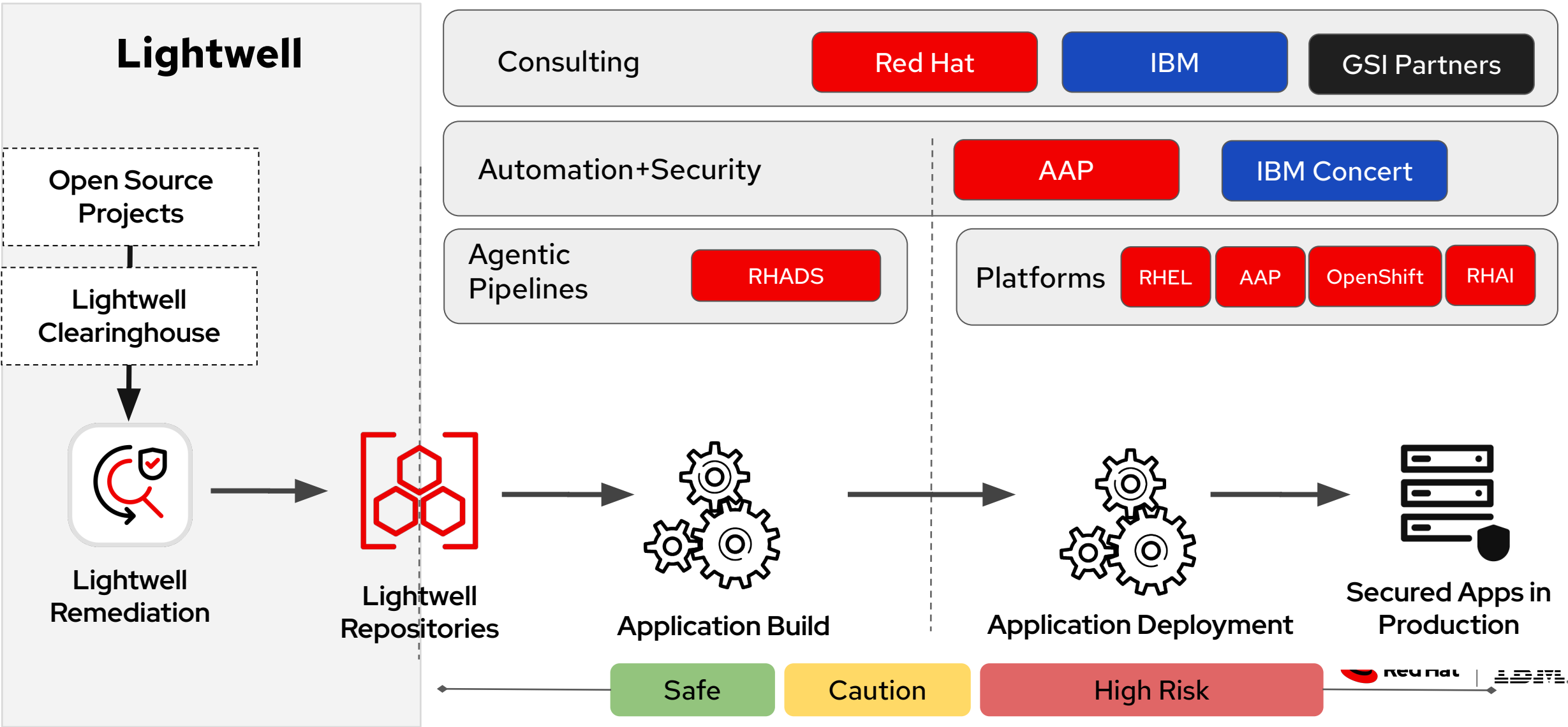
45-90

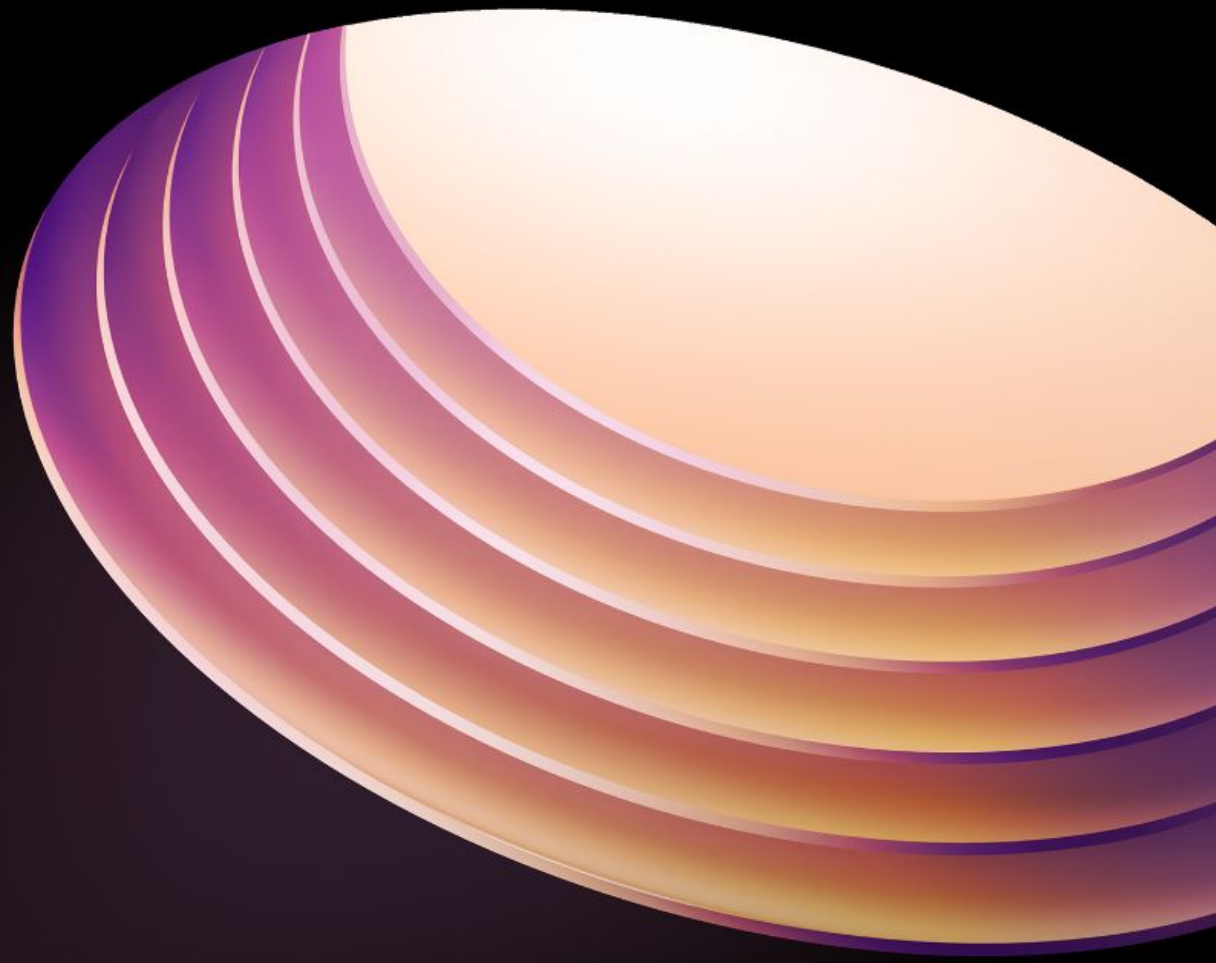
7-10

Red Hat | IBM

Reducing the Vulnerability Gap

Remediation + Improved Builds + Automated Deployments





Lightwell

THE VULNERABILITY GAP · PART 1
VULNERABILITY REMEDIATION

Red Hat | IBM

The World Runs on Open Source

Everyone depends on open source. Can you trust it?

96%

of codebases contain open source

90%

of AI/ML workloads are built on Python

500K+

packages on PyPI - the Python Package Index

The Threat is Real and Growing

Supply chain attacks grew 742% in the last 3 years.

⚠️ PyPI Malware

- ⚠️ Typosquatting
- ⚠️ Dependency confusion
- ⚠️ Malicious binaries
- ⚠️ Account takeover



The Python Problem

When you run "pip install", what are you actually getting?

Anyone Can Publish

PyPI has no vetting process. Anyone with an email can upload a package. There is no code review, no build verification, no provenance.

Pre-built Binaries

Most packages ship as pre-compiled wheels. You're running binaries built on someone's laptop, not from auditable source.

No Provenance

There's no way to verify who built a package, when, or from what source code. You're trusting strangers with your infrastructure.

Dependency Chains

Installing one package can pull in dozens of transitive dependencies. Each one is an attack surface.

Introducing Lightwell

Update **confidential** designator here

Same packages. Same pip install. Radically different trust model

	Public PyPI	Red Hat Trusted Libraries
Build Origin	✖ Pre-compiled on unknown machines	✔ **Built from source in trusted pipelines**
Transparency	✖ No SBOM, no provenance	✔ **SPDX SBOM + SLSA attestation in every package**
Security Scan	✖ No mandatory scanning	✔ **Snyk, Coverity, ClamAV on every build**
Verification	✖ Trust the uploader	✔ **Cryptographically signed + auditable build logs**
Support	✖ Community best-effort	✔ **Red Hat enterprise support**

The Red Hat Difference

SLSA (Supply-chain Levels for Software Artifacts)

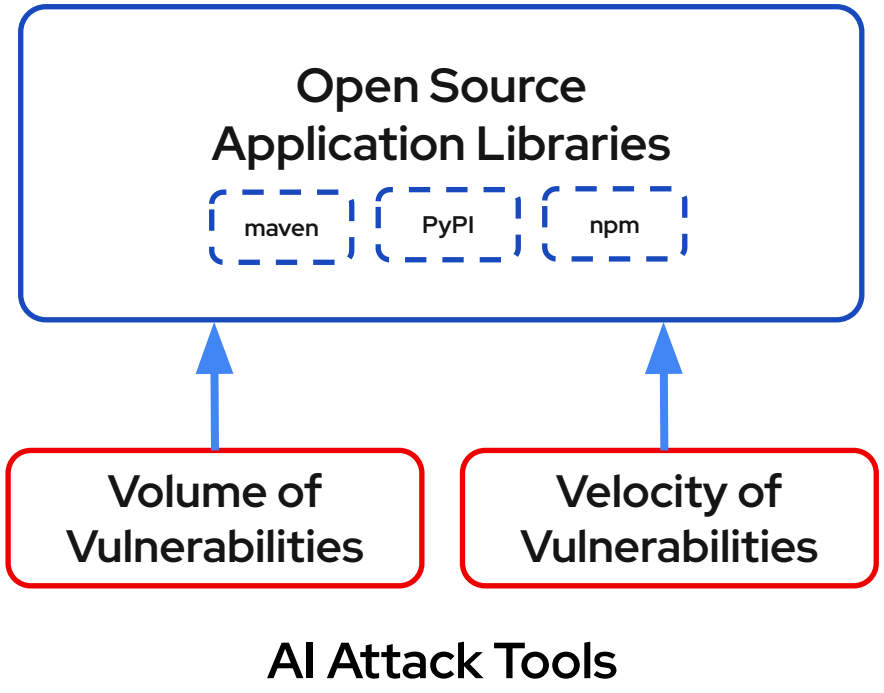
Feature	Standard Public Repos	Commercial "Secure" Vendors	Trusted Libraries
Build Standard	Unknown / Opaque	SLSA Level 2 (Often)	**SLSA Level 3 (Highest)**
Provenance	None	Limited Metadata	**Full Build-from-Source**
Security Response	Reactive / Manual	Varies	**Proactive (<24hr builds)**
Ecosystem	Isolated	Standalone Product	**Integrated & Native to RHADS, RHEL, and OpenShift**

Red Hat | IBM

Isolated Remediation is Costly and Unscalable

MASSIVE CVE GROWTH

Problem: Security Vulnerabilities



Remediating alone:

Your team triages the Mythos backlog internally. Three bad options for each finding:

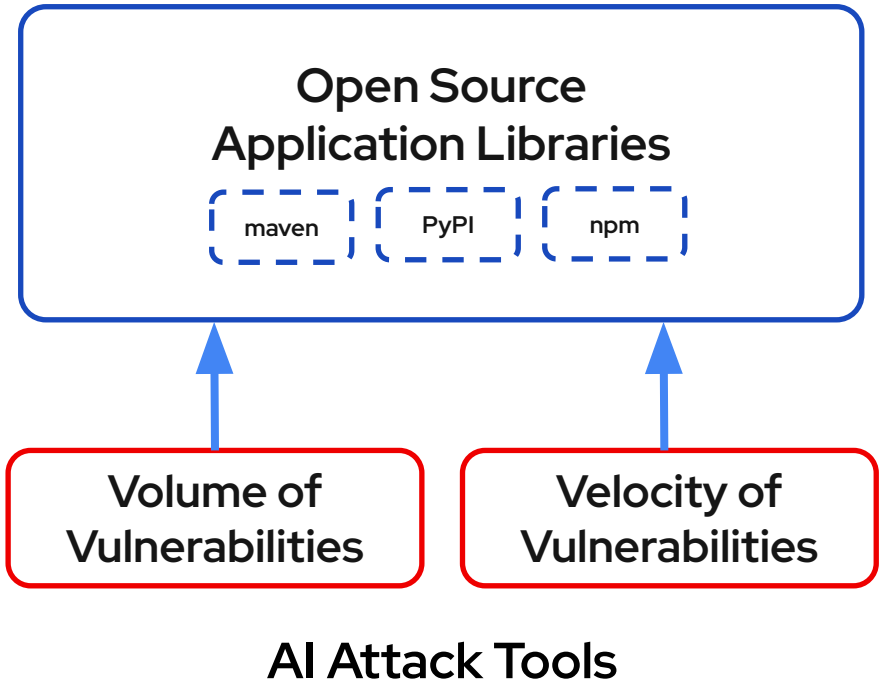
- Upgrade and break production.
- Patch internally and carry a permanent fork.
- Accept the risk of a confirmed **exploit.**

Red Hat + Lightwell

Trusted Security Remediation Partner

MASSIVE CVE GROWTH

Problem: Security Vulnerabilities



Remediating with Red Hat :

We backport fixes to the exact versions you run in production:

- No upgrades required.
- SLA-backed delivery.
- Signed artifacts.
- Every patch contributed upstream so you're never stuck on a fork.

Red Hat | IBM

What is Lightwell

Lightwell is a joint initiative between Red Hat and IBM focused on securing the open source software supply chain. It serves as a trusted clearinghouse for vulnerability management, coordinating scanning, triage, patching, backporting, and upstream collaboration.

Lightwell also provides curated and certified package registries for ecosystems

- Java
- Python
- JavaScript

Membership Tier	What's Included
Lightwell Network	Access to libraries, remediations, and patches as they become available. ▸ **Remediation** for serviced packages ▸ Digitally signed binaries and source ▸ Compliance artifacts ▸ Lightwell carries patches not yet accepted upstream
Lightwell Clearinghouse Premier	**Requirement:** Customer MUST have direct access to Mythos (or similar scanning tools) and willing to share vulnerability reports with Red Hat. In addition to all benefits of Lightwell Network, customers receive access to: ▸ **Vulnerability reporting and remediation** for member specific package versions ▸ Novel vulnerability verification and disclosure handling ▸ Requests may be anonymized and shared with other vertical members ▸ Security TAM ▸ All fixes released to: ○ Other vertical members ○ Lightwell Network ○ Upstream open source projects

Lightwell: What it covers

Platforms & Infrastructure:
- Red Hat Enterprise Linux (RHEL)
- OpenShift
- Ansible Automation Platform

Enterprise Open Source Products:
- Red Hat Middleware & Runtimes
- Red Hat Trusted Libraries
- Red Hat Universal Base Images (UBI)
- Red Hat Hardened Images

IBM & Red Hat Product Subscriptions:
- Existing security patching
- Product support
- Lifecycle management

Covered by Lightwell

Application Dependencies:
- Java / Maven libraries
- Python / PyPI packages
- JavaScript / npm packages (coming soon)

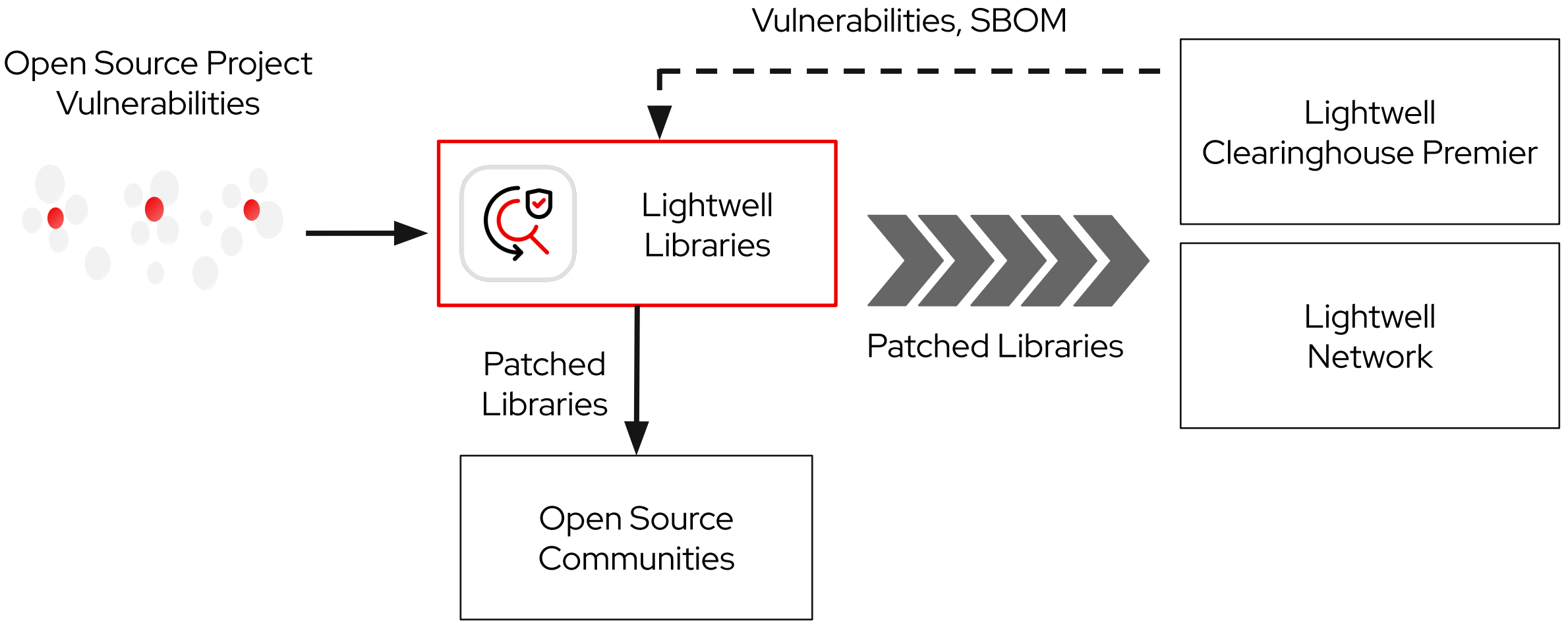
AI & Data Ecosystem (coming soon):
- AI frameworks
- Build tools
- Data and streaming libraries
- Transitive dependencies

- If you have a subscription for IBM or Red Hat software, security patching and support continues as usual.
- If you have community versions of widely adopted open source libraries, now is the time to evaluate the benefits of enterprise-supported distributions.

Lightwell secures the broader universe of open source dependencies that sit outside traditional product support boundaries.

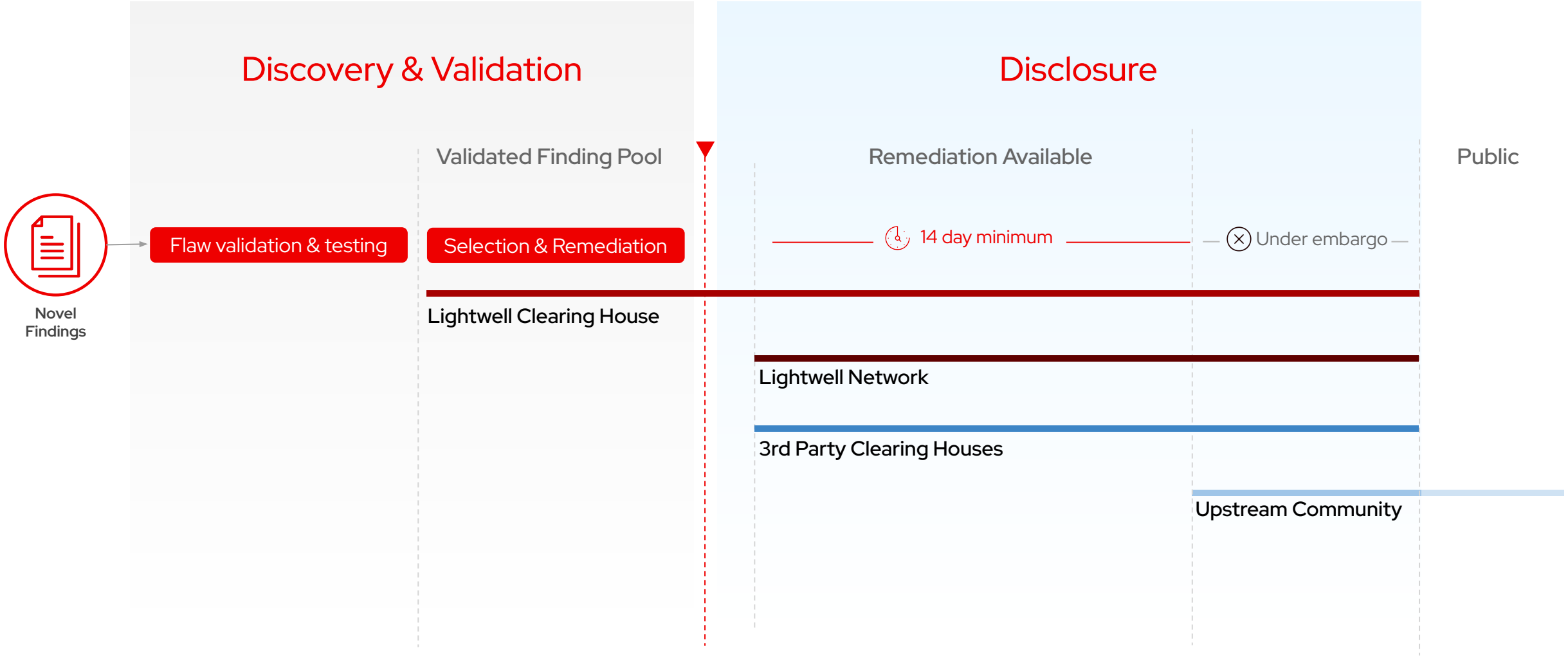
Red Hat | IBM

Lightwell: How it works

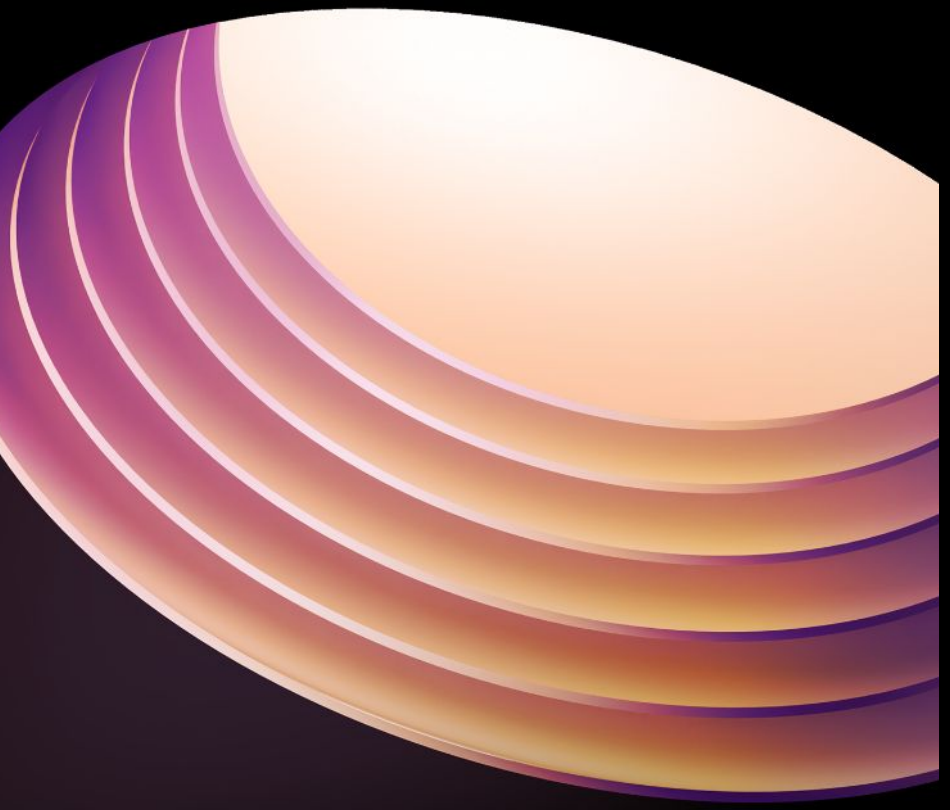


Red Hat | IBM

Lightwell Discovery & Disclosure



Red Hat | IBM



Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat | IBM