



Connect

Faster Than the Patch

Closing the Exposure Gap with Ansible
Automation Platform

Chris Renwick

Sr. Specialist Solution Architect (Ansible)





Chris Renwick

Sr. Specialist Solution Architect (Ansible)
Red Hat



The Technology Landscape Has Changed

IT and security teams are under pressure:

- ✓ AI can discover thousands of vulnerabilities in days—faster than humans.
- ✓ Exploit development time has compressed dramatically from public CVE to working exploit in hours not weeks.
- ✓ The bottleneck is no longer finding—it's controlling an exposure before a fix exists.

45%

of vulnerabilities in large organizations remain unpatched after 12 months.

Source: Centre for Emerging Technology and Security (Alan Turing Institute), 2025 — cited in CETAS analysis, April 2026.

Frontier AI models

Frontier models (like Mythos) that can identify and autonomously build exploits to 30-year-old zero-days in hours. When previously these vulnerabilities evaded human detection for decades.

Project Glasswing

While Glasswing accelerates discovery for Anthropic built models, its limitations mean it can't cover every model or threat surface.

Project Lightwell

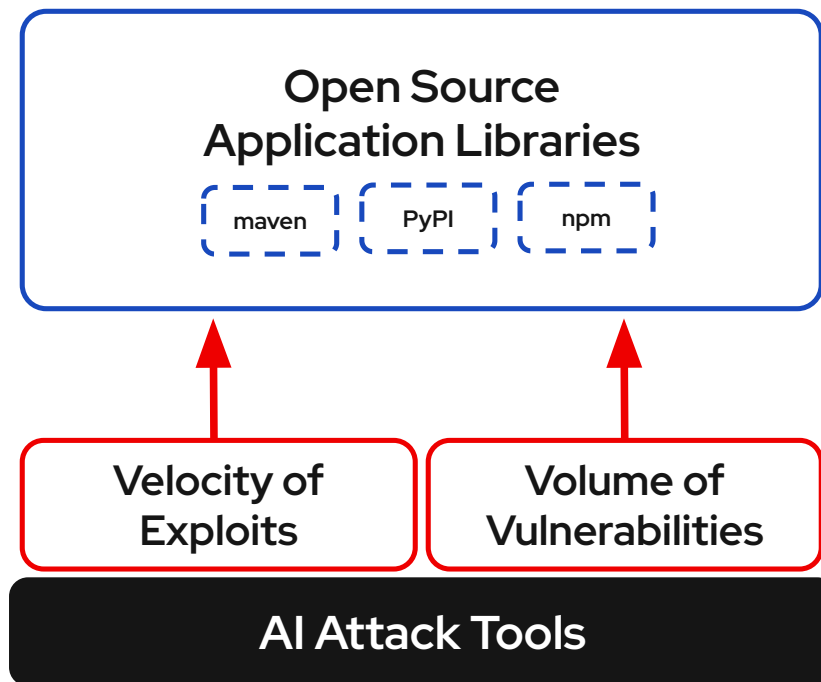
Project Lightwell will help counter AI-discovered vulnerabilities by decreasing the time between vulnerability discovery and patch availability.

The Mythos Moment

Accelerating Two Parallel Challenges - "The Vulnerability Gap"

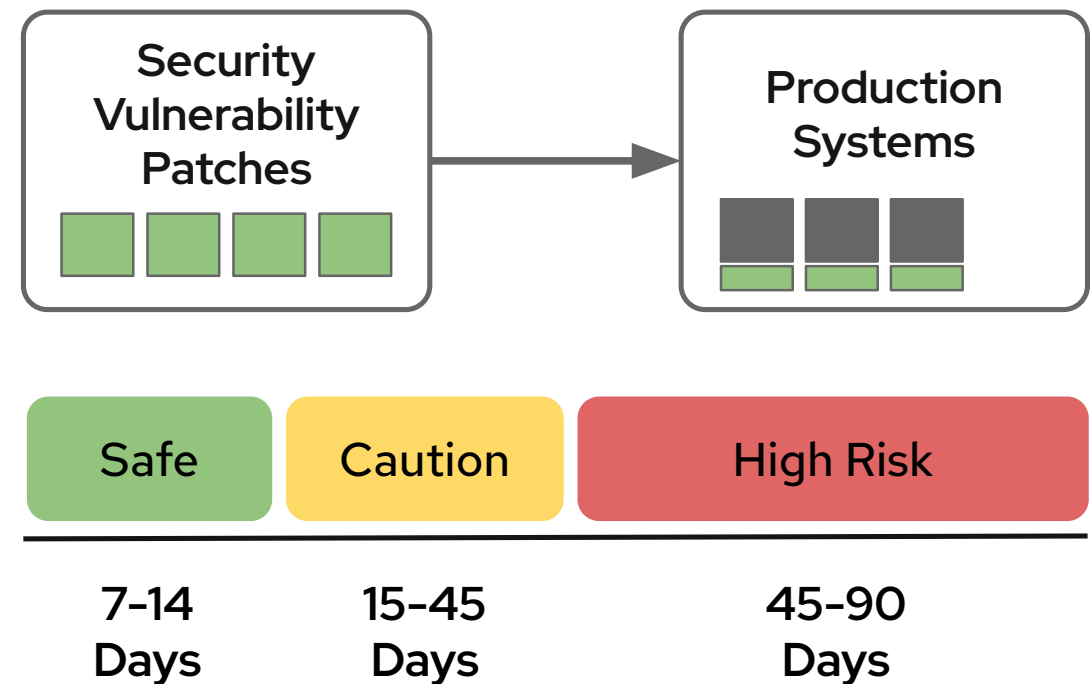
520% CVE Growth since 2026

Problem: Time to Exploitation Dropping



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

Problem: Re-Build and Deployment Velocity



The Path Forward – from Detection to Remediation

When a vulnerability drops, the window between discovery and remediation is where the risk lives. **Ansible closes that gap.**

Vulnerability lands. Ansible mitigates.

Respond before the patch is ready

- Rapidly identify affected systems for threat assessment.
- Enable workflows that isolate affected systems and disable exposed services automatically.
- Trigger increased alerting without human intervention.

Patch with confidence, when ready.

The full pipeline tested and validated.

- Patch vulnerabilities consistently when patches are ready using Ansible Automation Platform pipelines.
- Enable validations to ensure systems are back online before declaring success.
- Enable the ability to return to a known state with automated rollbacks.

Prove it worked. Every time.

Formalize governance before you scale

- Record every action in your ITSM (detection, remediation, validation, rollback)
- Know exactly what's automated, what needs human approval, and who owns what.



Red Hat Lightspeed (formally known as Insights)

Your intelligent advisor to navigate infrastructure complexity



Continuous Assessment

Constantly evaluates your infrastructure against known issues and vulnerabilities.



Vulnerability Detection

Identifies security risks before they can be exploited.



Remediation Recommendations

Provides specific guidance for resolving detected issues.



Compliance Monitoring

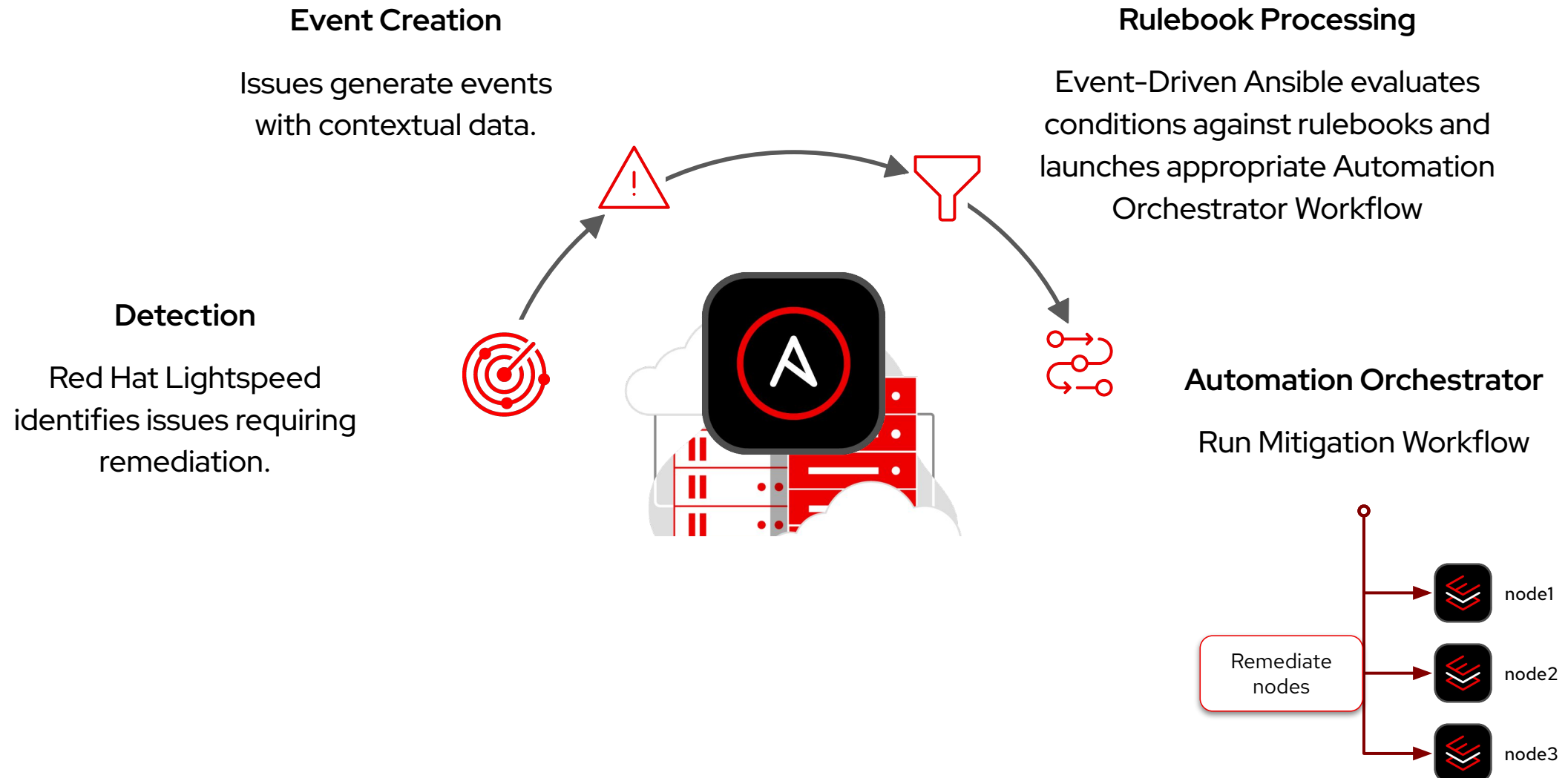
Tracks adherence to industry standards and internal policies.

With integrations into the complete Red Hat portfolio, Red Hat Insights uses **predictive analytics** and decades of **deep domain expertise** to help you reduce complex manual management activities.



Integration Architecture

Red Hat Insights (Lightspeed) + Event-Driven Ansible + Ansible Automation Platform



Story time

Any local user can become **root**. On every major Linux distro. In seconds.

A 12-year-old bug hiding in plain sight in polkit's pkexec, installed by default on every major Linux distribution since 2009.

CVSS SCORE

7.8 High

Red Hat severity:
Important

HIDDEN FOR

12+ years

In pkexec since May 2009

AFFECTED

Every major distro

RHEL, Ubuntu, Debian, Fedora, CentOS,
SUSE

COMPLEXITY

Trivial

Public exploit in under 3
hours



Unprivileged user

An intern with an SSH login



pkexec, no arguments

Treats an environment variable
as its argument



Root access

Full control of the entire system

pkexec doesn't validate its arguments. Pass it nothing, and it reads environment variables as code, running whatever the attacker planted.

Company A: "Not worth automating"

An illustrative story. No automation for vulnerability response.

“The cost of automating against vulnerabilities isn’t worth it. We’ll handle them as they come.”



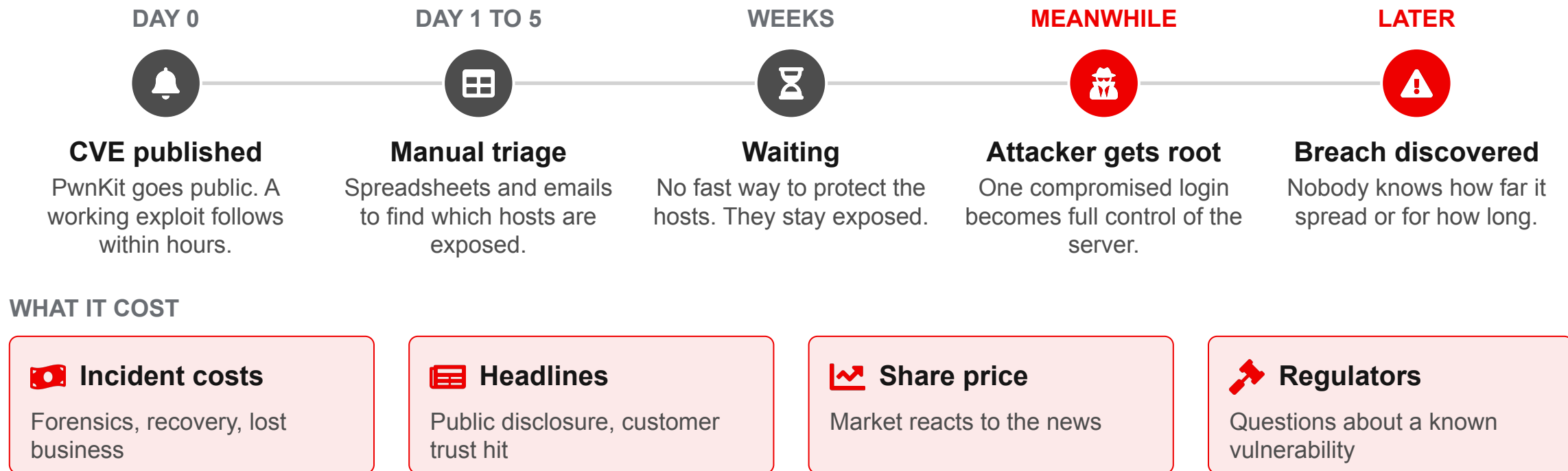
Company A

```
[ec2-user@companyA ~]$ _
```

Company A: "Not worth automating"

An illustrative story. No automation for vulnerability response.

"The cost of automating against vulnerabilities isn't worth it. We'll handle them as they come."



The exposure gap is what hurt them. Nothing was in place to close it.

Company B: automation as a culture

An illustrative story. Ansible Automation Platform at the centre.



“If we do it more than once, we automate it. Security response is no exception.”



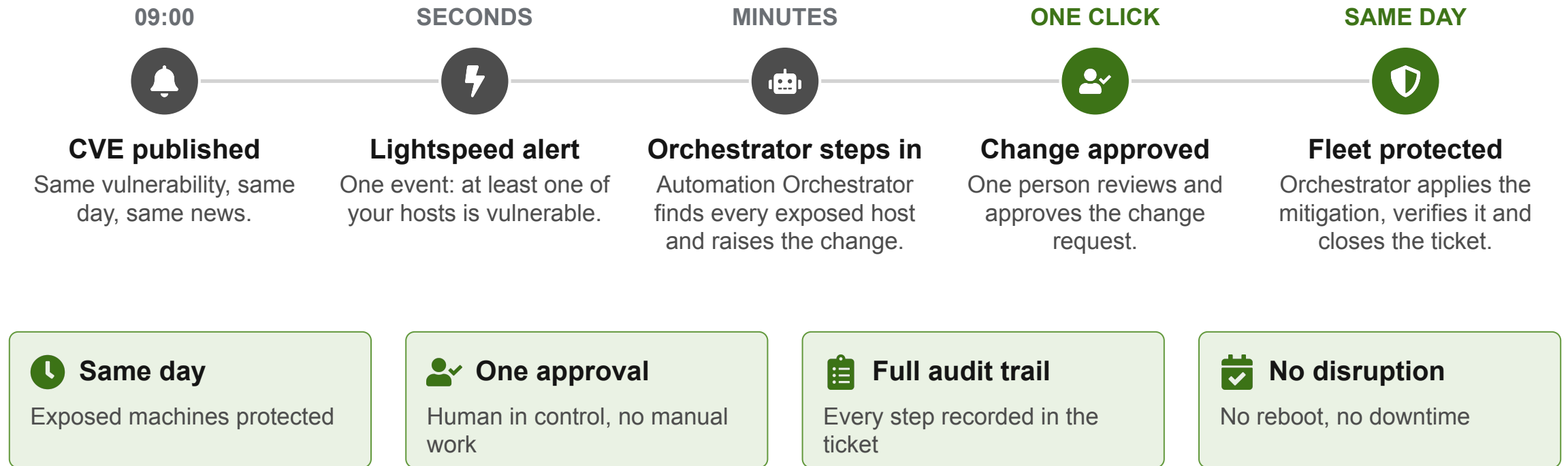
Company B

```
[ec2-user@companyB ~]$
```

Company B: automation as a culture

An illustrative story. Ansible Automation Platform at the centre.

“If we do it more than once, we automate it. Security response is no exception.”



Same vulnerability. Mitigated the same day. Nothing for the news to report.

So what exactly happened with Company B?

How did they protect themselves?

What did they do?

Demo





Connect

Thank you



linkedin.com/company/red-hat



facebook.com/redhatinc



youtube.com/user/RedHatVideos



x.com/RedHat

