





# Security Posture?



How to use the Application Vulnerability Report to draw actionable insights about the security of your own Cloud Foundry application landscape



Fabian Richter, SAP
September 17, 2026

# Agenda

**01**

## What Is the Application Vulnerability Report?

Vulnerability visibility for BAIP apps: continuous CVE detection, Findings API, per-application status.

**02**

## Shield Right, not just Shift Left

Why runtime scanning complements the view of the actual attack surface.

**03**

## How the scanner works and what a Finding looks like

OSV-based scanning without an agent, structured Findings, queryable via the OData API.

**04**

## In practice: Log4Shell and xz-utils

Two case studies showing how early signals would have made a risk visible.

**05**

## Consumption paths, regulation, and roadmap

API, Cockpit UI, CALM. Evidence for DORA, NIS2, ISO 27001. Outlook through 2027.

# Why the Application Vulnerability Report?

Before the service, there was no platform-side security signal for BAIP applications.

## The blind spot

NO SIGNAL FROM THE PLATFORM

**Blind spot**

BAIP provided no insight into the vulnerabilities of running applications.
Customers could not see what was actually vulnerable in Cloud Foundry.

## Everyone on their own, outside the platform

SHIFT LEFT ON YOUR OWN

**Does not scale**

Customers shifted left themselves, with their own tools outside BAIP.
That does not scale and does not correlate with what is running on the platform.

## Why the service is needed

> No signal from the platform.

> Shift Left on your own does not scale.

> Runtime posture for the worst case

# What Is the Application Vulnerability Report?

Vulnerabilities in BAIP apps made visible, provable, and manageable - the quick overview.

## What

VISIBLE

### API and Cockpit UI

Structured Findings per application via the OData v4 API.

As an overview in Workzone.

Findings per application at a glance.

## How

PROVABLE

### Continuous scans

Ongoing CVE detection in container images and build artifacts.

No agents, no configuration, timestamped history.

## Why

MANAGEABLE

### Security management

Vulnerability management for BAIP applications.

Evidence for ICT risk management: DORA, NIS2, ISO 27001, SOC 2, PCI DSS.

In Beta since December 2025, UI in Workzone via BAIP Cockpit since April 2026.

# The Application Vulnerability Report does Shield Right

Scans what actually runs - after deployment. It complements Shift Left, it does not replace it.

## Shift Left

THE CONTEXT

### Before deployment

SAST, SCA, and pre-commit hooks check the plan: code and manifest, before the build.

**Blind spot**: base image, transitive deps, and CVEs that only become known after the build.

## Shield Right = this is what the Application Vulnerability Report does

WHAT THE REPORT DOES

### Scans reality

Continuously evaluates running CF droplets, re-scored on every CVE.
Sees the real attack surface: what was shipped, including base image and transitive dependencies.

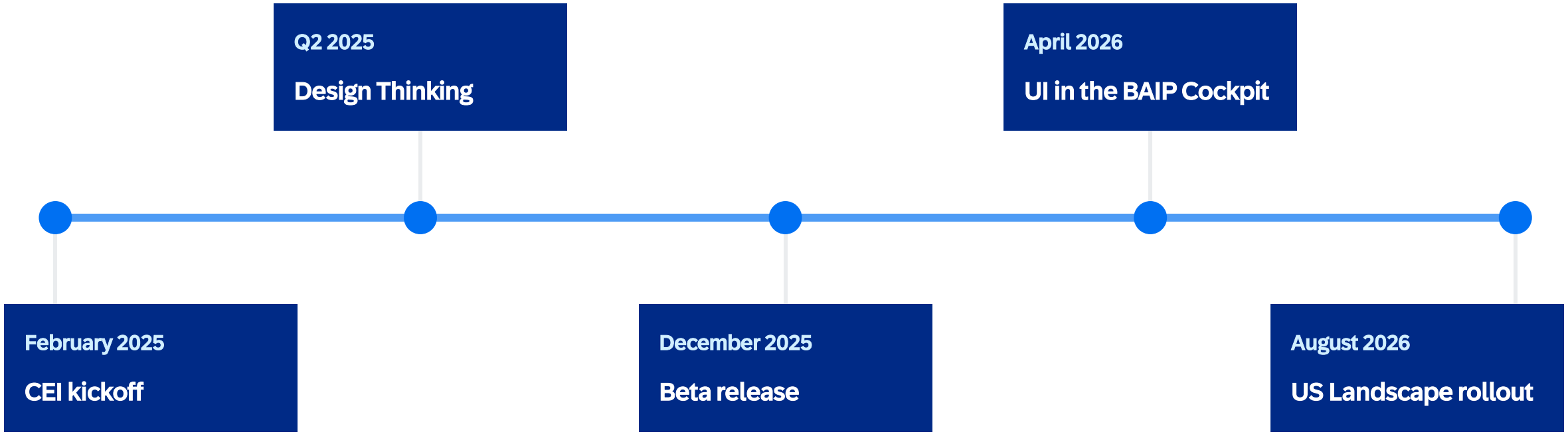
NO EFFORT

### No agent, no plugin

No workflow change. A platform-native signal from BAIP, correlated with what really runs in production.

# From Customer Engagement Initiative to productive operation

Five milestones. From the first workshop in Heidelberg to the US Landscape rollout.



# Who uses the Application Vulnerability Report today?

Application Vulnerability Report has been in Beta since December 2025. A lively adoption picture even before the GA release.

**100+**

**Customers**

actively use the Beta

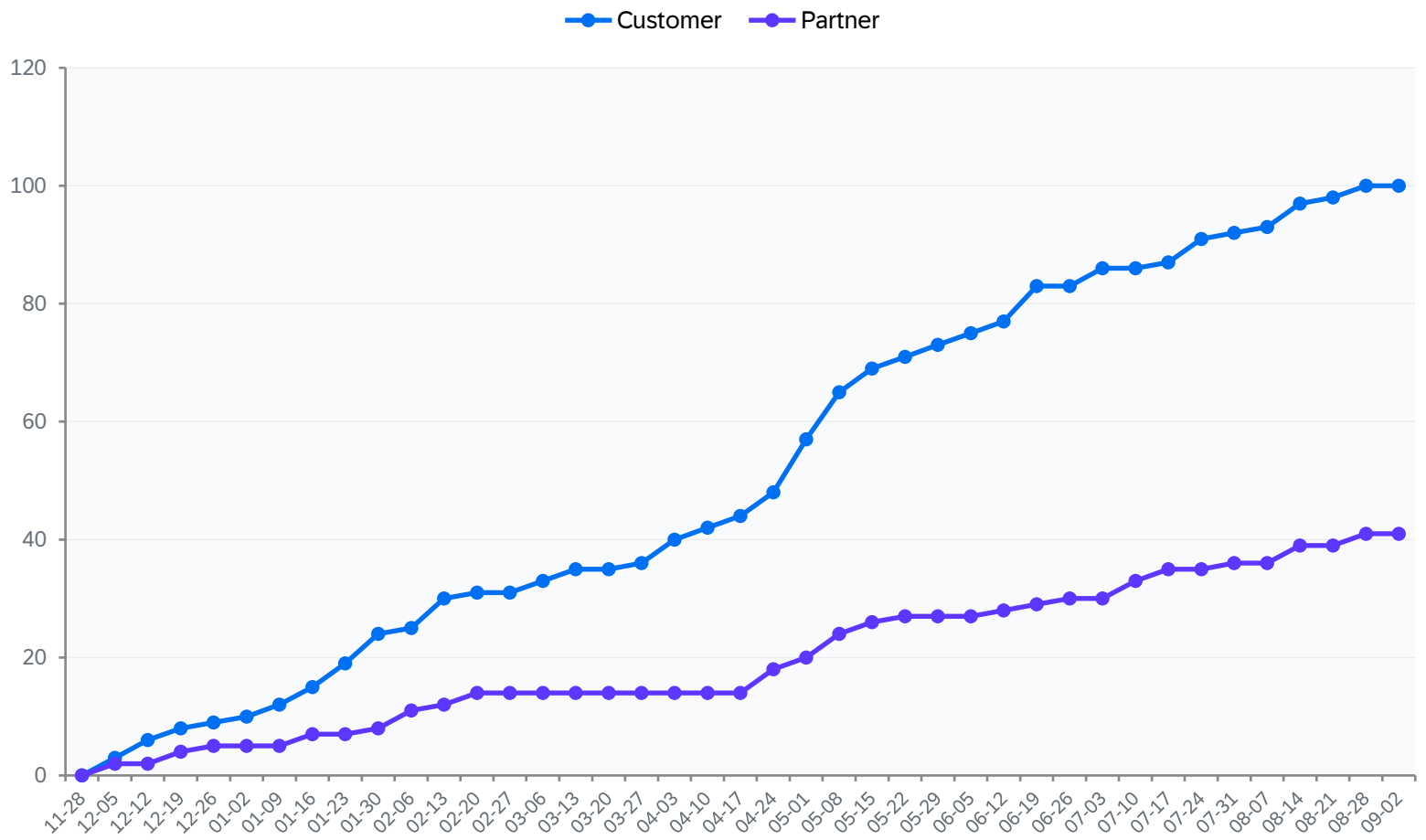
**41**

**Partners**

in the ecosystem integration

## Subscription Growth

Cumulative subscriptions since the Beta launch in December 2025



# UNDER THE HOOD

# How the Scanner Works

Continuous OSV-based scanning across the BAIP fleet. No agent, no sidecar, no configuration.

## Input

SOURCE

### CF App Droplet

OCI image layers and build manifest of every running Cloud Foundry droplet.
No agent, no sidecar, no configuration.

## MicroScanner Pipeline

OSV SCANNER

**40.000+**
OSV Advisories

**8**
Ecosystems

npm · PyPI · Maven · Go · NuGet · Cargo · RubyGems · Composer

SBOM extraction, transitive dependencies, Google-maintained OSV database.

## Output Channels

API AND UI

### OData v4

Findings Store per app, accessible via OData v4 and BAIP Cockpit.
OSV coverage applies automatically to every CF droplet.

# What a Finding Looks Like

Every vulnerability is a structured Finding object. Queryable by app, package, severity, CVE.

## Structure of a Finding

**IDENTITY:**

### findingId

Stable identifier joining CVE plus package across scans. Format FIND-<12 hex>. sourceId and activeCves map the scanner and CVEs.

**COORDINATE:**

### packageNames

Full ecosystem coordinate: group:artifact:version (Maven) or equivalent. **severity** drives the SLA class, **maxCvss** is the numeric score.

**RETRIEVAL:**

### Findings API

GET /api/v1/provider/Findings, OData v4. OAuth2 via BAIP service binding. Documented in the SAP Business Accelerator Hub.

## finding.json

```
{
  "findingId":     "FIND-7c3a0e91b842",
  "sourceId":      "osv-scanner",
  "severity":      "CRITICAL",
  "maxCvss":       10.0,
  "activeCves":    "CVE-2021-44228",
  "packageNames":  "org.apache.logging
                    .log4j:log4j-core:2.14.1",
  "recommendation":"Upgrade to 2.17.1.",
  "appName":       "payment-service",
  "spaceName":     "prod",
  "lastFound":     "2026-06-12T08:49:13Z"
}
```

# RISK FROM FOSS???

# Case Study 1: Log4Shell - Creeping Neglect

CVE-2021-44228, December 2021. Damage potential: several billion USD.

## What Was the Problem?

REACH

**400 million**

installations worldwide, in nearly every Java app.

GOVERNANCE

**Volunteer-run**

Tiny volunteer team, no full-time reviewer.

## Warning signals for years

| | | |
|---|---|---|
| **Bus Factor** | 1 maintainer, 50% of commits | 10/10 |
| **Release Frequency** | long release gaps | 8/10 |
| **Time to Close** | > 180 days | 9/10 |
| **Inactive Contributors** | >= 90% inactive | 9/10 |

## Result

**CVE-2021-44228**

Several billion USD in damage, patch time in weeks.

KEY MESSAGE

**Risk in plain sight**

All signals had been public for years.

## Key Takeaways

- Four scores at 8-10/10 for years.
- Signals public, but never aggregated.
- FOSSY bundles them, visible early.

# Case Study 2: xz-utils Backdoor - Active Maintainer Takeover

CVE-2024-3094, March 2024. Discovered by chance - just before the global rollout.

## What Was the Problem?

**SITUATION**

**15 years alone**

1 developer maintained xz-utils since 2009, alone.

**INFILTRATION**

**2 years**

“Helper“ built trust, then a backdoor in the build.

## Governance signals

| | |
|---|---|
| **Bus Factor** 1 maintainer, 15 years | 10/10 |
| **Self-Merge Rate** > 50% without review | 9/10 |
| **Burstiness** suspicious pattern | 8/10 |
| **OpenSSF Badge** no Best Practices badge | 10/10 |

## Result

**CVE-2024-3094**

Discovered by chance via an SSH latency anomaly.

**KEY MESSAGE**

**More than CVEs**

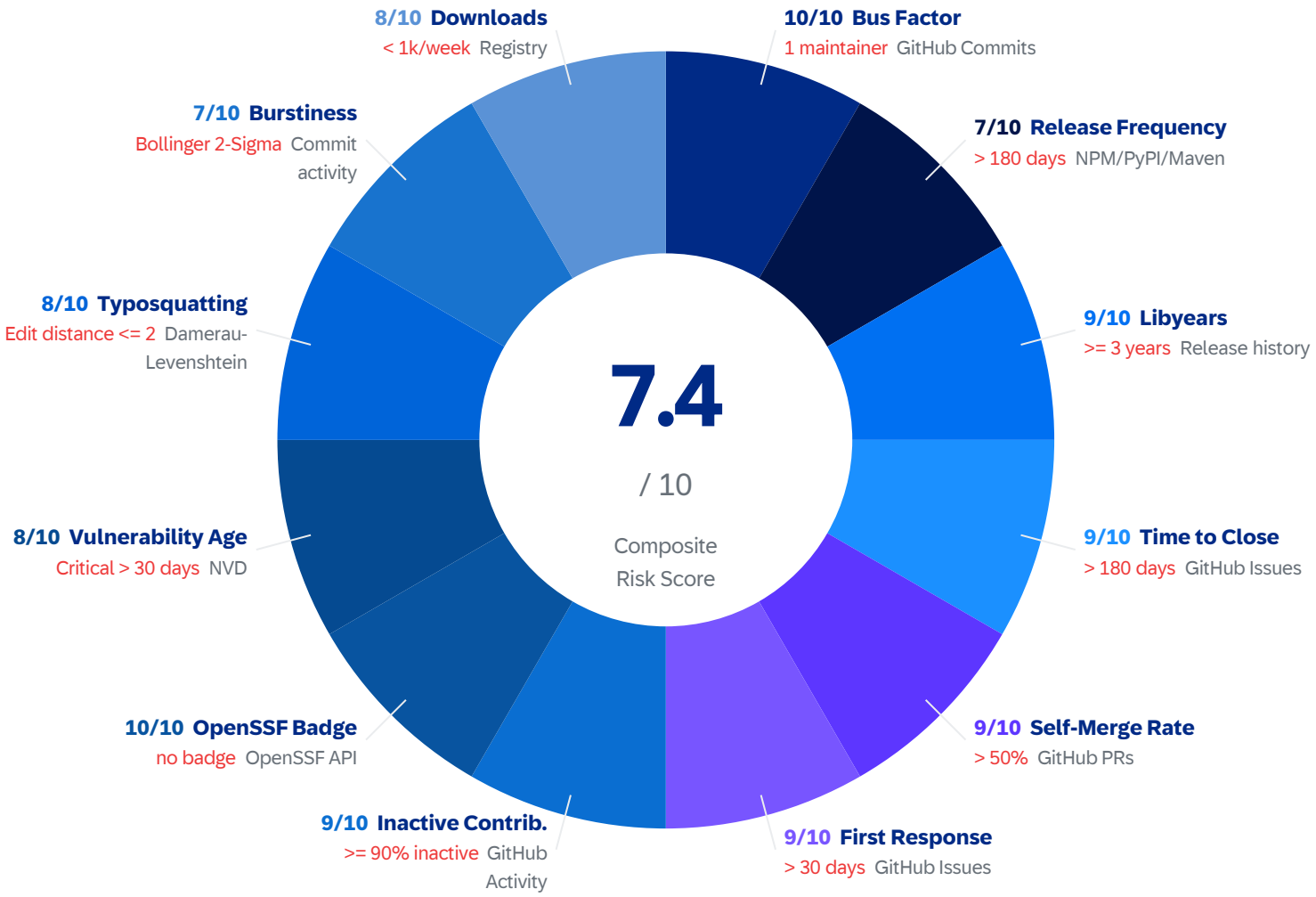
Governance patterns are risk signals too.

## Key Takeaways

- “ 15 years alone, 2 years infiltrated.
- “ Scores 8-10/10, the whole time.
- “ Not just CVEs, behavior counts too.

# FOSSY: 12 Scorers, One Composite Score.

Every metric 0-10. Lower = better. Data sources: GitHub, NPM/PyPI/Maven, NVD, OpenSSF Best Practices.



# USAGE & OUTLOOK

# Three Ways to Consume Findings

The right access for every role: automated integration, self-service, and a security view.

## API

FOR AUTOMATED INTEGRATION

### OData v4

OData v4 Findings API, OAuth2 via BAIP service binding.
Integration with pipelines and ITSM.

## BAIP Cockpit / Work Zone

FOR DEVOPS

### Self-service

Self-service overview, Findings search and filters.
Drilldown from app to CVE.

## CALM from 2027

FOR SECURITY EXPERTS

### SAP Cloud ALM

Findings in SAP Cloud ALM, a consolidated risk view.
Integration with Security Recommendations.

# From Regulation to Evidence

A framework demands evidence. The Application Vulnerability Report delivers it via API.

## EU regulation

DORA ART. 8

### Real time

ICT risk management demands continuous vulnerability monitoring.
Continuous CVE scan across all CF apps.

NIS2

### < 48h

Critical infrastructure must report incidents quickly.

## Supervision and audit

PCI DSS

### Auditable

Proof of control; an assurance is not enough.
Timestamped scan history, retrievable via API.

ISO 27001 A.12.6

### Chain of evidence

Requires a traceable chain of evidence.
Complete Findings history, auditable via API.

## Process and red teaming

ITIL

### Integrable

Change and incident management need reliable data sources.
Findings via API in change/incident workflows.

TIBER-EU

### API-first

Threat-led pentesting requires a known vulnerability baseline.
Integrable into your own risk dashboard.

DORA = Digital Operational Resilience Act · NIS2 = Network and Information Systems Directive 2 · TIBER-EU = Threat Intelligence-Based Ethical Red-Teaming

# Roadmap: Q3 2026 to 2027

From the global Landscape rollout to Findings in CALM to FOSSY and BAIP Security Insights.

## Q3 2026

US ROLLOUT

### US live

US Landscape live since August 2026.
First productive non-EU region.
More landscapes follow with GA.

## Q4 2026

FINDINGS IN CALM

### Cloud ALM

Findings in the operations workflow.
Integration with Security Recommendations.
A single risk view in SAP Cloud ALM.

## H1 2027

COVERAGE EXPANSION

### More scanners

FOSSY integration.
Trivy and OWASP scans.
Broader ecosystem coverage.

## H2 2027

VISION AND MATURITY

### GA

General Availability.
More landscapes worldwide.
Parts merge into Joule Studio.

**We look forward to your feedback - what would you like from the Application Vulnerability Report next?**

# DEMO?

# IT'S Q&A TIME!

# Thank You!

## Questions and Discussion

**We are interested in your experiences with the Application Vulnerability Report.**

Application Vulnerability Report - Docs →

Application Vulnerability Report on SAP Discovery Center →

**Fabian Richter**
SAP BAIP Cyber Defense
fabian.richter@sap.com

Application Vulnerability Report Docs