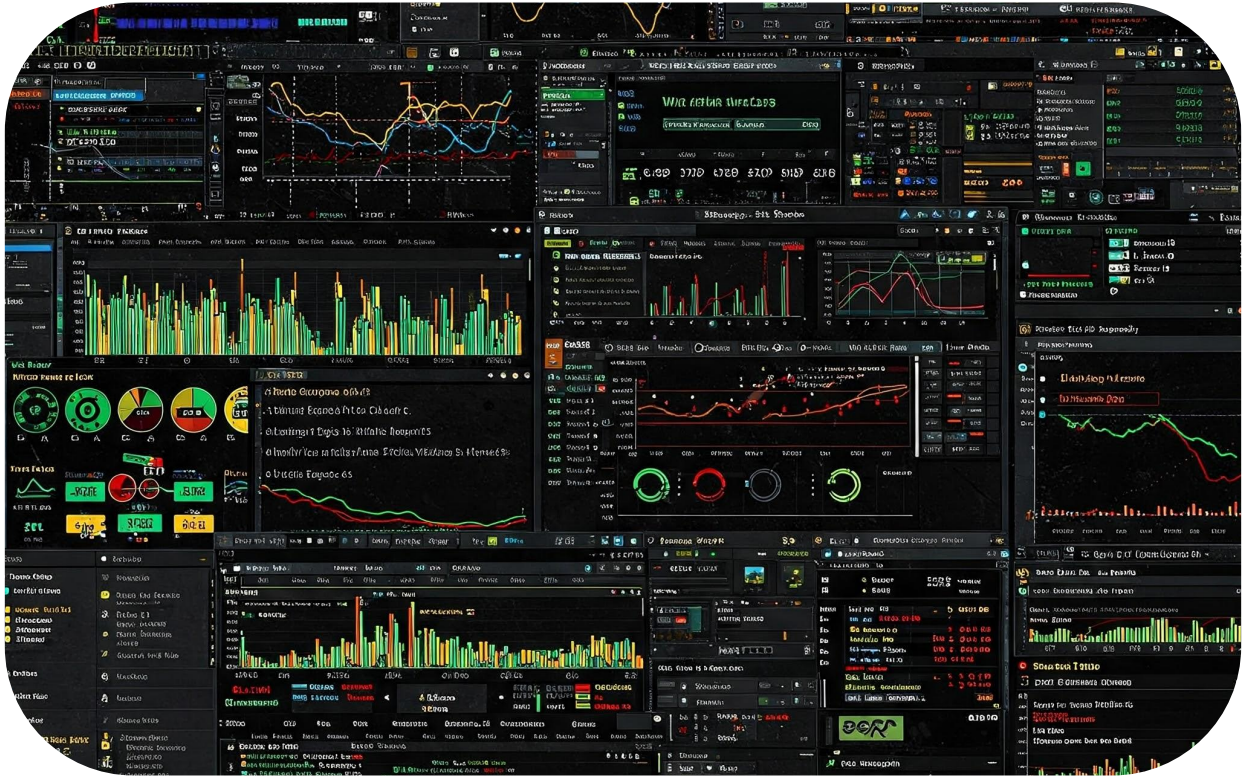
The First Age of AppSec has Ended

The challenge and opportunity ahead is without scale and precedent.

The current AppSec model is unsustainable

We need to re-imagine our approach

Too much noise - not enough signal



AppSec’s Challenge is Context

Accuracy

Relevance

Clarity

When Context Saves Your Life

AppSec’s Context Problem

Accuracy

Relevance

Clarity

Every Year, the same story

What Happens When You Flip the Model?

What if you didn’t scan from the outside but listened from the inside?

What if your Application could tell you when it’s under attack - not in theory but in real time?

That is what we have built.

Securing an entire city…

Too much noise, too little signal…

- Fragmented tools
- Analyzing pieces
- Inaccurate tools (FP & FN)
- Lack of context
- Unrealistic test environment
- No threat intelligence
- No behavioral analysis
- Only ~5% truly exploitable

Clues…

- Across the rest of the stack, security has moved away from scanners and perimeter to EDR/XDR agents
- AppSec is clamoring for KEVs, reachability, exploitability, blast radius and contextual risk ratings

The answers are in production!

Development

- Hundreds of repos/libs
- Simulated environment
- Point in time testing
- Weak code coverage
- No threat intelligence

Production

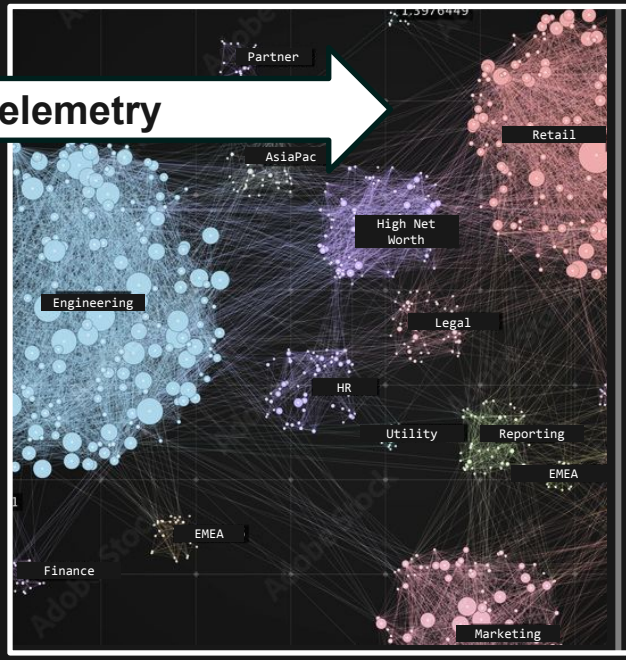
- Fully assembled apps/APIs
- Real users, real data
- Real environment, connections
- Real code behavior
- Real attackers, exploits

Observing production…

PRODUCTION

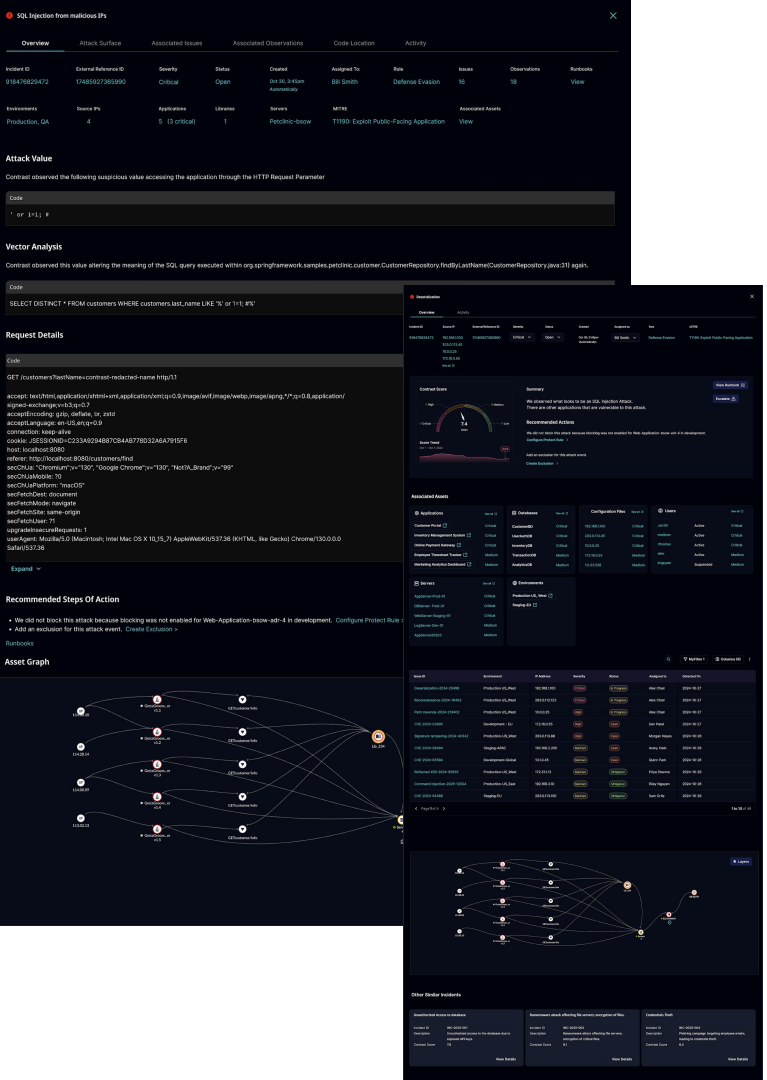
runtime telemetry

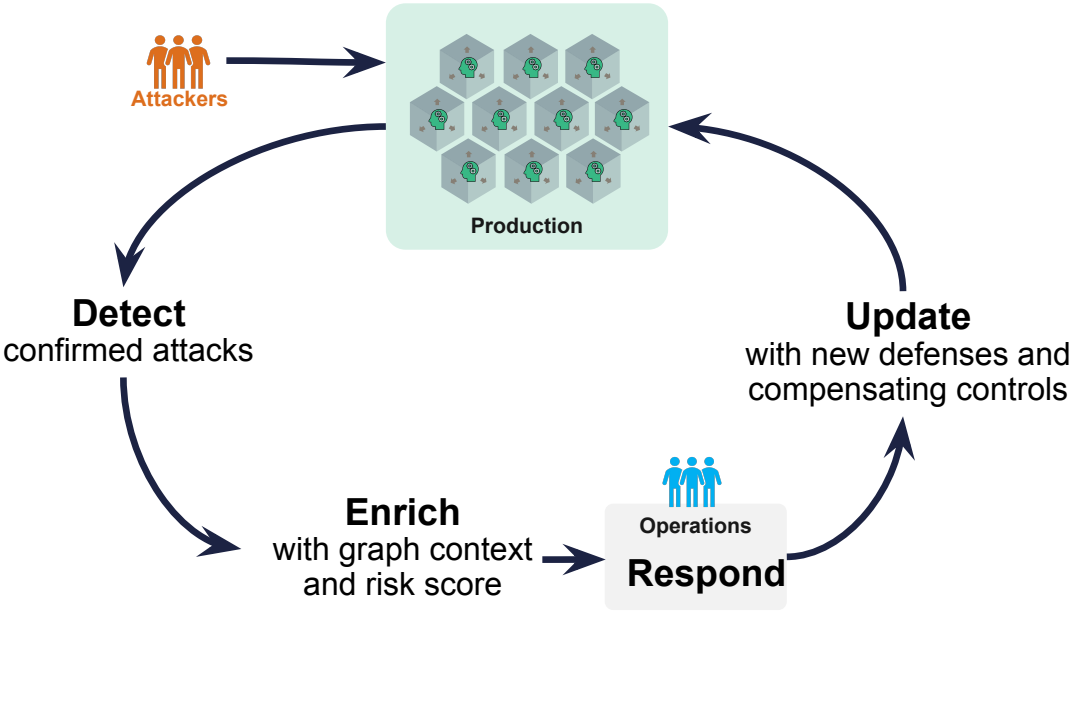
APPSEC GRAPH

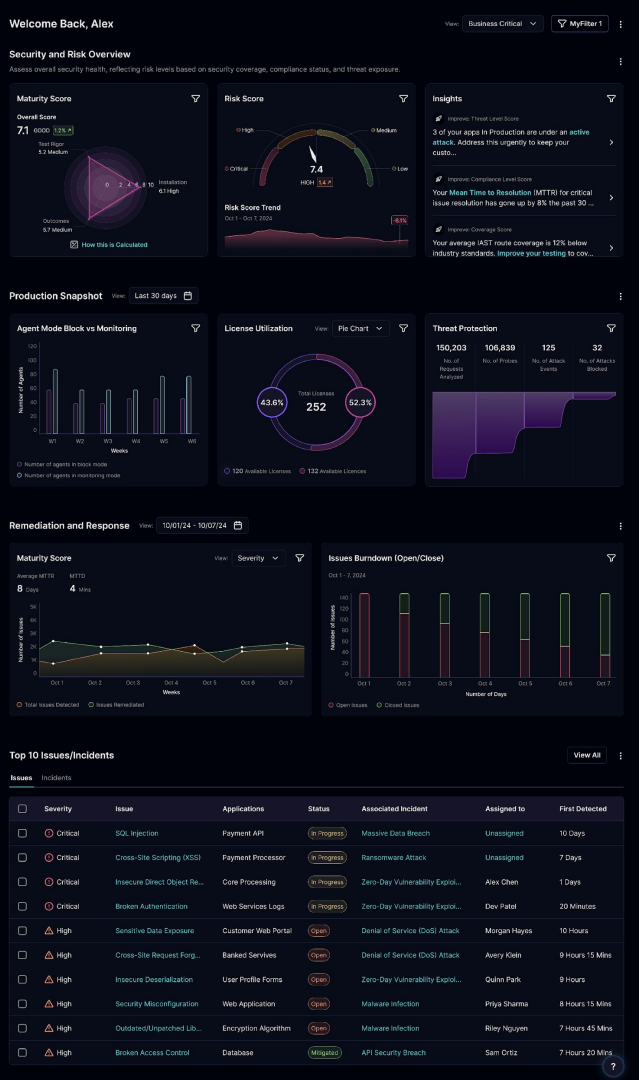


- ☐ Apps/APIs
- ☐ Servers
- ☐ Assets
- ☐ Attack Surface
- ☐ Attacks
- ☐ Blast Radius
- ☐ Cloud
- ☐ Connections
- ☐ Containers
- ☐ Control Flow
- ☐ Criticality
- ☐ CVEs
- ☐ Data Flow
- ☐ Defenses
- ☐ Incidents
- ☐ Libraries
- ☐ Probes
- ☐ Queues
- ☐ Repos
- ☐ Risky Functions
- ☐ Routes
- ☐ Services
- ☐ Teams

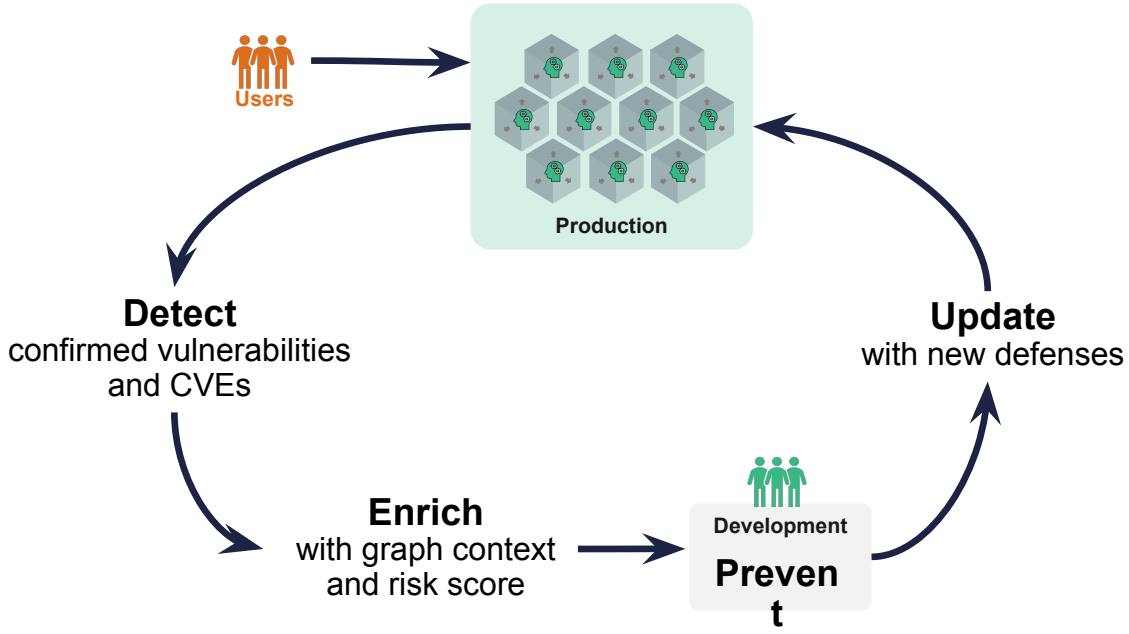
Handle application incidents

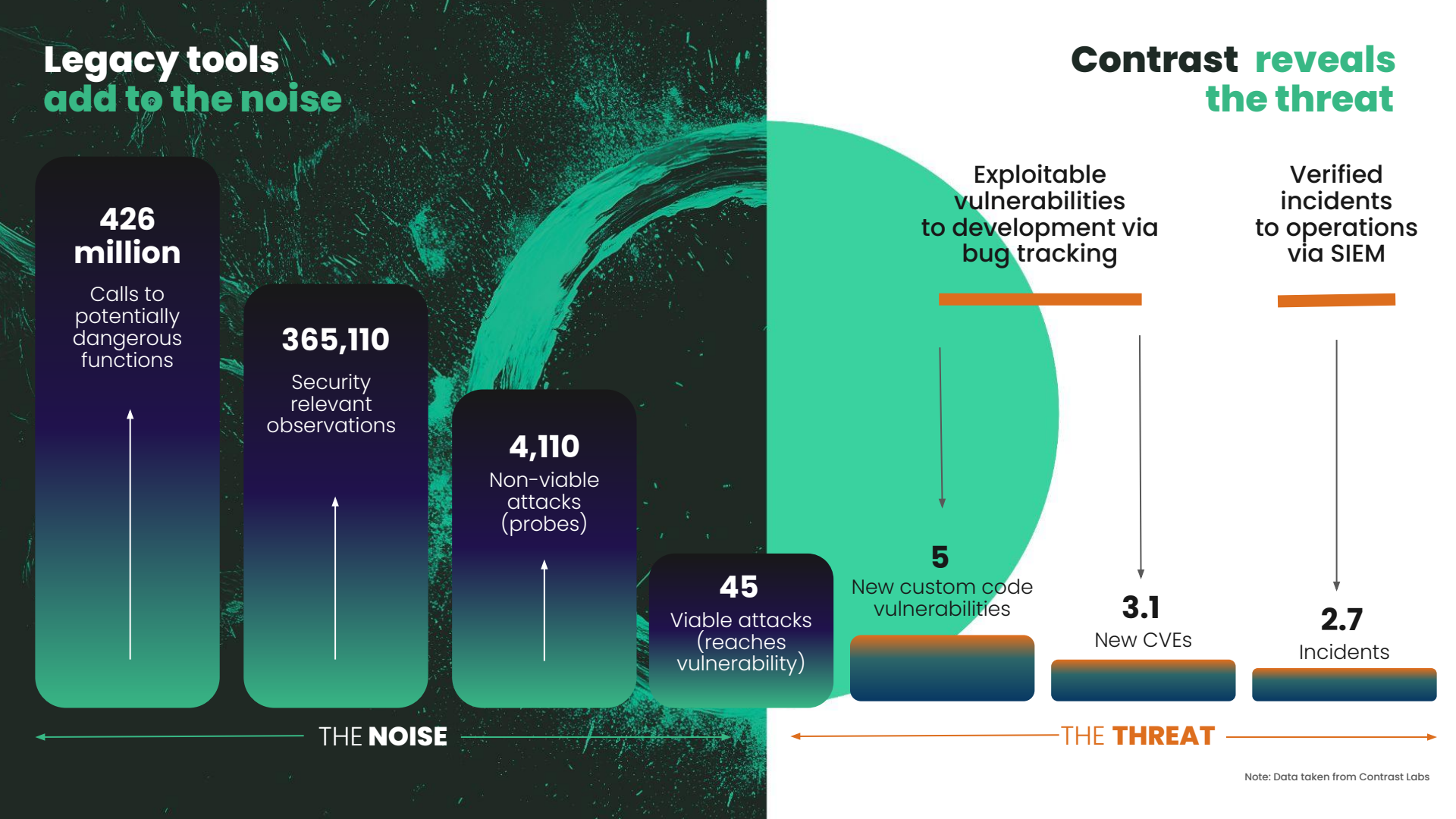






Streamline application security testing of code and libraries




Legacy tools
add to the noise
426 million
Calls to potentially dangerous functions
365,110
Security relevant observations
4,110
Non-viable attacks (probes)
45
Viable attacks (reaches vulnerability)
THE NOISE
Contrast reveals the threat
Exploitable vulnerabilities to development via bug tracking
Verified incidents to operations via SIEM
5
New custom code vulnerabilities
3.1
New CVEs
2.7
Incidents
THE THREAT
Note: Data taken from Contrast Labs

If you're afraid to test in production, remember that your customers are always testing in production.”

— **Jeff Sussna,** Manager Platform Engineering at **Infinite Campus**

Thank you