Keep pace or get left behind.

Today: the perfect storm.

With the adoption of DevOps and CI/CD pipelines, new application builds can be automated to go live daily, hourly, or even faster. Add to that the expanding attack surface and complexity of modern apps (multiple languages, 3rd party components, cloud services, containers, microservices, etc.), and now you have the **equation for the perfect storm:** faster delivery + more areas to attack + greater app complexity = a significant increase in security, privacy, and compliance risks.

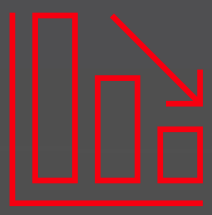
Not getting security right the first time not only results in immediate vulnerabilities but also mounting technical debt which causes significant issues as applications age. And hackers love to exploit vulnerabilities in legacy code. Once attackers get their foot in the door, they have visibility into even more potential vulnerabilities.

Compounding these challenges is the fact that remediating a vulnerability once it's released to production is exponentially more expensive than doing that same task during development!

Security or bust.

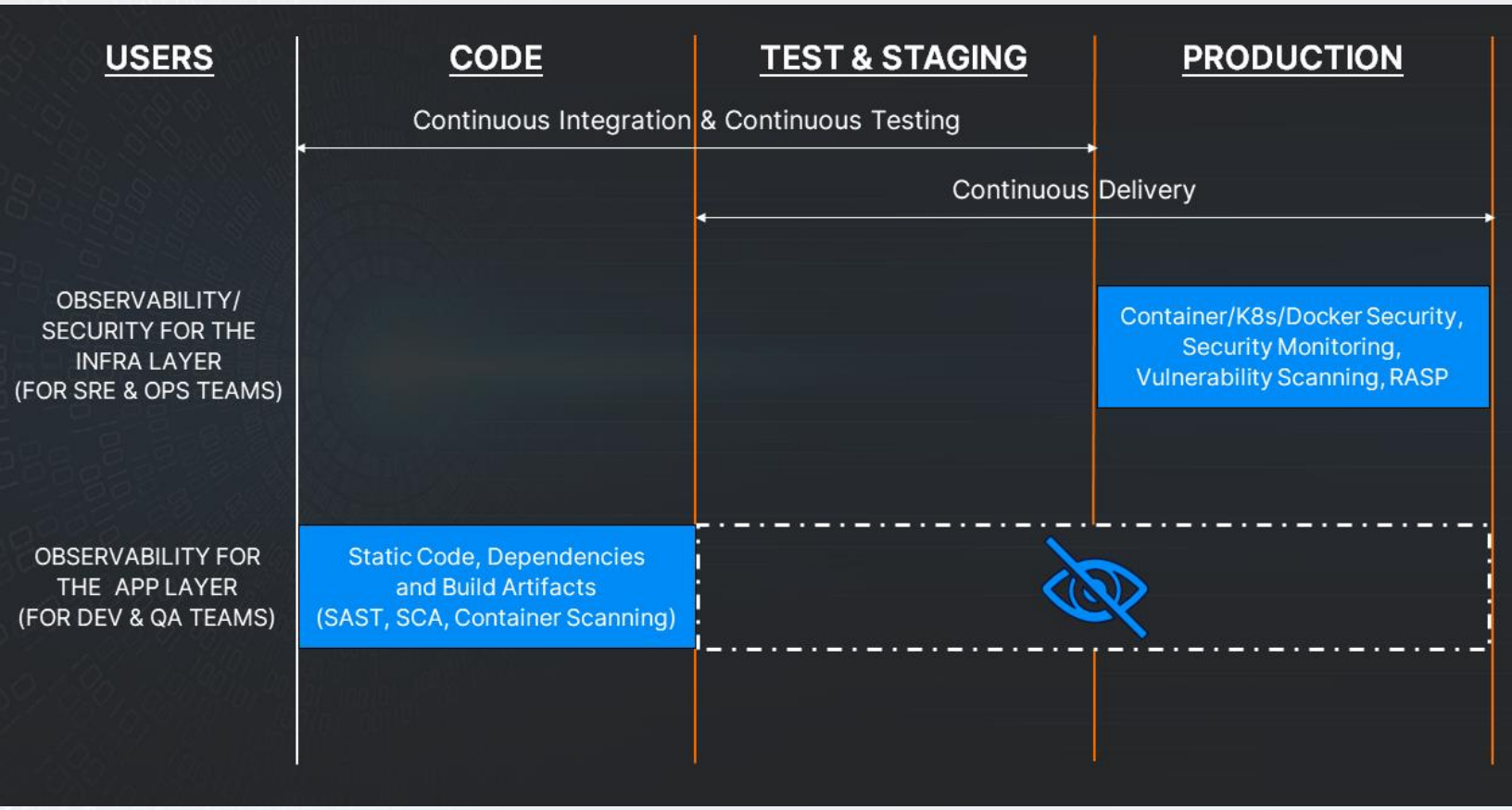
Today's Application Security teams are struggling to keep pace with modern development and the high volume and complexity of modern apps. Because of this, it has become humanly impossible for AppSec teams alone to identify all the security and compliance risks before deploying to production. And unfortunately, adding more application security experts is extremely difficult—these resources are hard to find and super expensive.

Instead, AppSec teams need help from the Engineering teams to find app security and compliance vulnerabilities early in dev and to make security part of the definition of 'done' before shipping to production. It's more than just shifting left...it's **STARTING LEFT**.



We are 'Runtime Blind'.

Today's Developers and AppSec teams lack continuous RUNTIME visibility into the impact of their code, their 3rd party components, and the interpreter itself on the application's security, privacy, and compliance. In effect, they are 'Runtime Blind'.

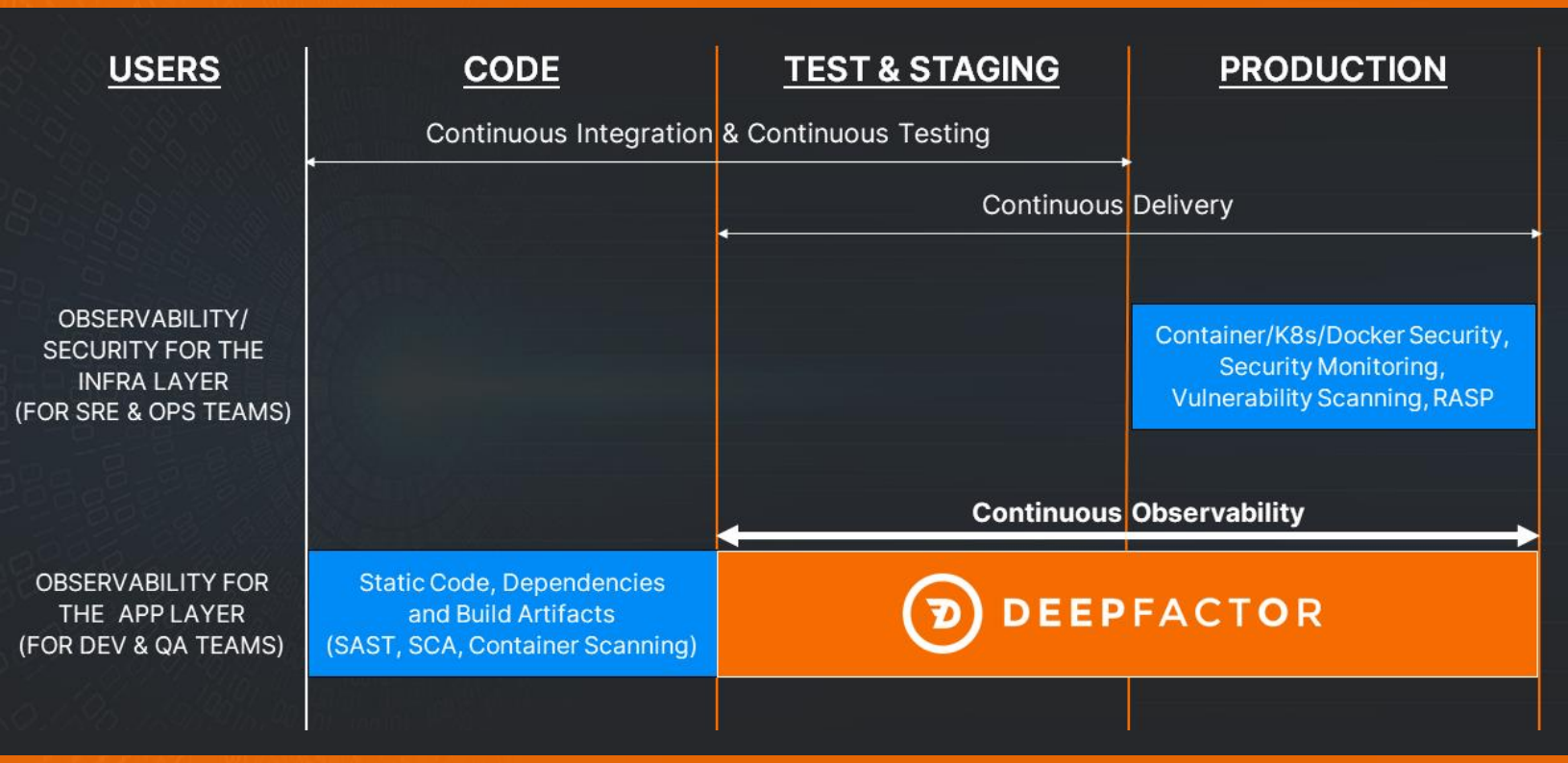


Unfortunately, existing AppSec tools for Dev and QA teams are insufficient and disjointed.

- **SAST and SCA** tools scan code or find risks in build images, but they don't observe the app while it's running; have no visibility into 3rd party and open source behaviors; have no capability to find and triage compliance risks; and generate too much noise—aka false-positives—which hinders productivity. These tools are needed, but not all by themselves.
- **DAST** tools treat your app like a black box and don't look 'inside' the application processes, and, only cover one aspect of runtime behavior (web and API). So, while it's essential to have DAST, its limited scope means DAST alone is not enough.
- **IAST** tools were designed almost a decade ago and are pre-devops/containers/Kubernetes/ Docker; are language-specific; and are fundamentally not designed for developers. If you have 20 different containers, it's very cumbersome to understand what language each of the apps is instrumented in and follow custom steps for each container.

We need to be 'Runtime Ready'.

Engineers need an all-in-one, purpose-built tool that looks inside every thread/process/container WHILE THE APP IS RUNNING in test/staging/prod and automatically identifies security and privacy **vulnerabilities that only manifest at runtime**—such as system call risks, behavior violations, and runtime use of vulnerable dependencies—throughout the CI/CD pipeline. Providing developers with this runtime observability during development empowers them to **'secure at the source'** and identify potential issues before they reach production. This is where DeepFactor's Continuous Observability for Security & Compliance platform steps in.



Welcome to DeepFactor: Continuous Observability for Security & Compliance.

DeepFactor is the industry's first Continuous Observability platform enabling Engineering and AppSec teams to find and triage RUNTIME security, privacy, and compliance risks in your applications—including 3rd party components—within the DevOps pipeline. With zero code changes, DeepFactor automatically observes billions of live telemetry events in every thread/process/container to detect anomalies during test, staging, and production. Deep Insights cover system call risks, data risks, behavior risks, DAST scans, a software bill of materials (SBOM), and vulnerable dependencies to create high-fidelity alerts with actionable evidence. Reduce MTTR, accelerate release velocity, and 'start left' to create and maintain secure and compliant apps. **DeepFactor is created for developers by developers.**

Huge Benefits With Deep Insights.

DeepFactor identifies risks that only manifest at runtime. When you navigate to an alert, you are shown the list of all occurrences of that alert. Then, you can triage each occurrence separately.

1. Comprehensive Runtime Insights

SYSTEM CALL RISKS

Risks in process, memory, filesystem, and network behaviors determined by observing system and library calls

BEHAVIOR VIOLATIONS

Alert developers during CI if in-house or 3rd party app deviates from expected process, memory, filesystem, and network behaviors defined by policies

DATA RISKS*

Identity & credential tracking, weak encryption, unencrypted PII in DB or object storage, keys in env vars, data audit logs, unencrypted data in flight, etc.

PRIVACY & COMPLIANCE RISKS*

Risks mapped to GDPR, PCI, ISO27001, and other compliance frameworks

CHANGES BETWEEN RELEASES & ENVIRONMENTS*

Deviations in ports, processes, metrics and configurations between versions and between environments

2. Visibility Into Your Software Supply Chain

SOFTWARE BILL OF MATERIALS (SBOM)

Catalog of all dependencies—including open source and 3rd party—and OS packages used by the app, along with licensing information and runtime metrics such as processes, ports, files, and network connections; value-add for SOC2/other compliance processes

3. Prioritized Vulnerabilities & Reduced SCA Alert Volume

DYNAMIC DEPENDENCY ANALYSIS

Prioritized list of vulnerable dependencies based on actual runtime usage, touchpoints & actionability—augments SCA tools & reduces alert fatigue

VULNERABLE OS PACKAGES

Find vulnerabilities in the OS packages on VM or container that the app actually loaded along with usage information which helps prioritize and easily fix alerts

4. Enriched DAST Insights

OWASP ZAP SCAN RESULTS (WEB)

Results of built-in headless OWASP ZAP DAST Scanner

API SCAN

Scan your API interfaces for OWASP vulnerabilities using Swagger/OpenAPI

*Coming Soon

Continuous Observability for all.

To date, 'observability' has been applied to performance-related tracing and troubleshooting. Now it's time to expand the definition to include observability in the context of security and compliance.

Build Fast → Continuous Integration

Deploy Fast → Continuous Delivery

Detect Functionality Bugs Fast → Continuous Testing

Detect **Security, Compliance** & Performance Risks Fast → **Continuous Observability**

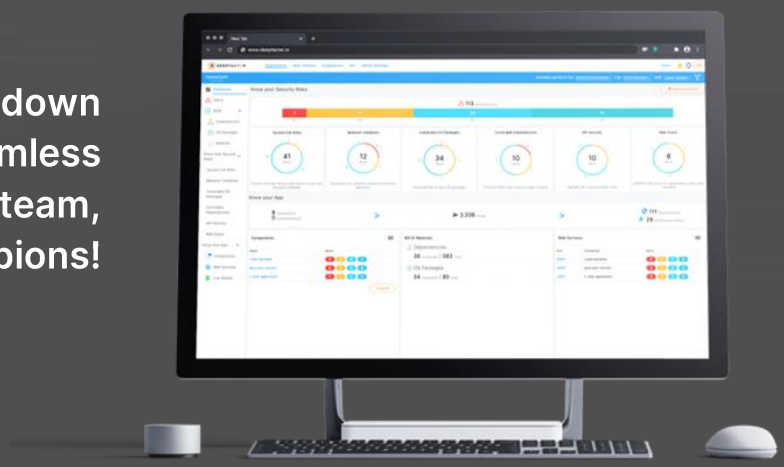
DeepFactor is a pioneer using the concept of Continuous Observability to identify runtime security and compliance risks in an app to enable **Engineering** to ship secure code to production as part of their day-to-day tasks and without drowning in alert fatigue.

With DeepFactor, **Developers** can automatically observe BILLIONS of live application telemetry events in every thread/process/container to identify and triage security and compliance risks across various layers of the application stack—system calls, library calls, and network, web, API, and configuration layers.

The **Application Security team** can establish guardrails, prioritize alerts, and empower Engineering teams to abate security risks before production using automated, continuous visibility into the actual RUNTIME behavior of every build.

Without compromising release velocity, **Engineering Leadership** can increase productivity and decrease mean-time-to-remediate (MTTR) security and compliance risks pre-production as their teams ship secure releases on schedule using a developer-centric, purpose-built, continuous observability tool.

DeepFactor enables Dev to break down silos, reduce friction, and have seamless collaboration with the AppSec team, turning Dev into AppSec champions!



Continuous Observability in 5 steps.

1 RUN ANY WORKLOAD
Observe Billions of Events

2 DETECT ALERTS
Identify Needles in the Haystack

3 ACTIONALBE INSIGHTS
Stack Traces, Metrics, and Actionable Evidence

4 TRIAGE ISSUES
Prioritize, File Tickets, Notifications

5 CI PIPELINE INTEGRATION
Observability-as-Code API

- **Run Your App with DeepFactor:** Our patent-pending Deep Passive Observability technology collects BILLIONS of telemetry events from every software component it is deployed with, observing behavior, configuration, connections, dependencies, function calls, system calls and more. Deep Passive Observability has minimal performance impact in staging, creates no additional security risk, and is transparent to production environments.

- **Get Insights:** dfctl sends the billions of app events to the DeepFactor portal. This telemetry is analyzed, metrics are identified, and anomalies detected. These security and compliance insights are presented, with actionable evidence such as stack traces, metrics, and more. Insights are grouped into 4 modules, as previously noted.

- **Start OWASP ZAP Scans:** DeepFactor's portal has a fully integrated headless OWASP ZAP scanner, which is a great complement to any observability platform. Scans can be kicked off with zero setup and can greatly enhance applications' code coverage, and augment DeepFactor's telemetry and insights. In addition, DeepFactor also passes in the observed URIs back to the ZAP scanner and increases scan coverage.

- **Integrate With Your Favorite Tools:** The DeepFactor portal provides a centralized management and reporting interface to your SaaS or self-hosted deployment. DeepFactor comes with pre-packaged integrations with popular developer tools such as Jira, Jenkins, Slack, GitHub and more, so you can start integrating your favorite tools right away.

Jenkins JIRA slack GitHub

- **Integrate With CI/CD pipeline**

aws Code Services Azure Pipelines circleci CloudBees. Swagger GitHub Actions GitLab Jenkins Travis CI

- **Use Observability-as-Code API:** Integrate with CI/CD pipelines, gate builds, and more! Read the blog.

What Our Customers Are Saying.

"From a security perspective, [DeepFactor] filled a hole in Jobvite's current observability fabric. [DeepFactor] sets you up for a lot of future wins because you have the whole piece."

– Ron Teeter
VP & Chief Architect
Jobvite

"DeepFactor helps us easily visualize blind spots for every component and every version of our application **before** we deploy to production."

– Mohit Dhawan
SVP Engineering and Operations
Komprise

"We inundate our developers with a list of vulnerabilities they need to go attack. Having DeepFactor gives us...a prioritized list of vulnerabilities or issues that we need our engineers to focus on."

– David Huang
VP Global Tech Operations
Cadent

DeepFactor is as easy as A, B, C.

DeepFactor is unique.

- Requires zero code changes to the app
- Is agnostic to the language in which the app is written
- Uses one, simple dfctl command
- Works with any workload (container/Kubernetes/ Docker or even traditional apps) and any cloud
- Has low—single-digit—performance overhead
- Plugs into any CI/CD platform.

A. dfctl Command

The dfctl command is used to observe any workload without changing the code or build scripts. Simply run your app with this command and start seeing telemetry. dfctl uses Deep Passive Observability technology to observe the billions of events occurring in **every thread/process/ container** of the application in traditional apps or containers/Kubernetes/ Docker apps. You can run dfctl during dev, test, staging, pre-prod or even prod.

Kubernetes: `kubectl apply -f deepfactor-adm-webhook.yaml`

Docker: `dfctl run -a MyApp -c MyWebServer --image nginx:latest -name mynginx1 -p 80:80 -d`

Traditional/ Non-container: `dfctl run -a MyApp -c MyComponent --cmd java -jar DfDemo-0.0.1-SNAPSHOT.jar`

B. DeepFactor Portal

The DeepFactor portal includes the backend for collecting and analyzing telemetry, as well as the management portal UI. This can be setup in both cloud or on-premises. AWS and VMware environments are supported today.

C. Observability-as-Code API

Similar to how 'Infrastructure-as-Code' enables DevOps engineers to orchestrate infrastructure using scripts, DeepFactor's Observability-as-Code API enables DevSecOps Engineers to leverage observability functionality in their CI/CD pipeline and gate builds based on the security and compliance insights gathered by DeepFactor's Continuous Observability platform.

DeepFactor's Observability-as-Code API is available as a Swagger doc. It enables customers to do the following:

- Run your app with DeepFactor using the dfctl command
- Get the list of insights determined by DeepFactor
- Gate releases based on DeepFactor's insights
- Trigger headless OWASP ZAP scans

Thoughtful Design.

DEEPFACTOR | Applications | Alert Policies | Integrations | API | Admin Settings | Docs | VW

FINANCEAPP (APPLICATION) — SHOWING METRICS FOR Default Environment FOR Entire Duration AND Latest Version

Dashboard
Alerts
BOM
Dependencies
OS Packages
Network
Know Your Security Risks
System Call Risks
Behavior Violations
Vulnerable OS Packages
Vulnerable Dependencies
API Security
Web Scans
Know Your App
Components
Web Services
Live Stream

Know your Security Risks

Download Report

113 Security Alerts

P1	P2	P3	P4
7	31	34	41

System Call Risks	Behavior Violations	Vulnerable OS Packages	Vulnerable Dependencies	API Security	Web Scans
41 Alerts (1, 6, 15, 19)	12 Alerts (1, 3, 2, 6)	34 Alerts (1, 15, 1, 17)	10 Alerts (2, 3, 5)	10 Alerts (2, 3, 5)	6 Alerts (2, 2, 2)
System call risks hiding deep down in your and 3rd party software	Suspicious file, network, process & memory behaviors	Vulnerabilities in your OS packages	Priortize CVEs with runtime usage analysis	OWSAP ZAP scans for REST APIs	OWSAP ZAP scans for applications with a web frontend

Know your App

8 Components
0 Active Instances

3.20B Events

111 Security Alerts
29 Performance Alerts

Components

Name	Alerts
node-backend	2 4 8 4
java-user-service	0 3 2 8
c-web-application	1 1 2 6
java-tomcat	1 6 4 9

Bill Of Materials

Dependencies
36 Vulnerable / 583 Total

OS Packages
34 Vulnerable / 80 Total

Web Services

Port	Component	Alerts
9090	node-backend	0 0 0 0
4000	java-user-service	0 0 0 0
3000	c-web-application	0 0 0 0
8080	java-tomcat	0 0 0 0

DEEPFACTOR | Applications | Alert Policies | Integrations | API | Admin Settings | Docs | VW

FINANCEAPP (APPLICATION) — SHOWING METRICS FOR Default Environment FOR Entire Duration

Dashboard
Alerts
BOM
Know Your Security Risks
System Call Risks
Behavior Violations
Vulnerable OS Packages
Vulnerable Dependencies
DAST
Know Your App
Components
Web Services
Live Stream

Observed OS Packages

Enter package name

Filter By: Vulnerable: ALL — Component: ALL

	Package	version	License	Component	Usage (Shared Libraries)	Vulnerabilities	Alert
P1	glibc	2.19-18+deb8u10		node-backend	6	1 6 5 9 12	FINA-9
P2	glibc	2.27-3ubuntu1.4		java-user-service	5	0 0 6 26 6	FINA-35
P2	gcc-4.9	4.9.2-10+deb8u1		node-backend	1	0 0 2 1 0	FINA-10
P4	openjdk-lts	11.0.9.1+1-0ubuntu1~18.04		java-user-service	2	0 0 4 0 0	FINA-34
P4	gcc-8	8.4.0-1ubuntu1~18.04		java-user-service	1	0 0 2 0 0	FINA-36
	glibc	2.27-3ubuntu1		c-web-application	7	None	None
	gnutls28	3.5.18-1ubuntu1		c-web-application	1	None	None
	krb5	1.16-2build1		c-web-application	4	None	None
	libpsl	0.19.1-5build1		c-web-application	1	None	None
	rtmpdump	2.4+20151223.gitfa8646d.1-1		c-web-application	1	None	None

10 rows — 1-10 of 34

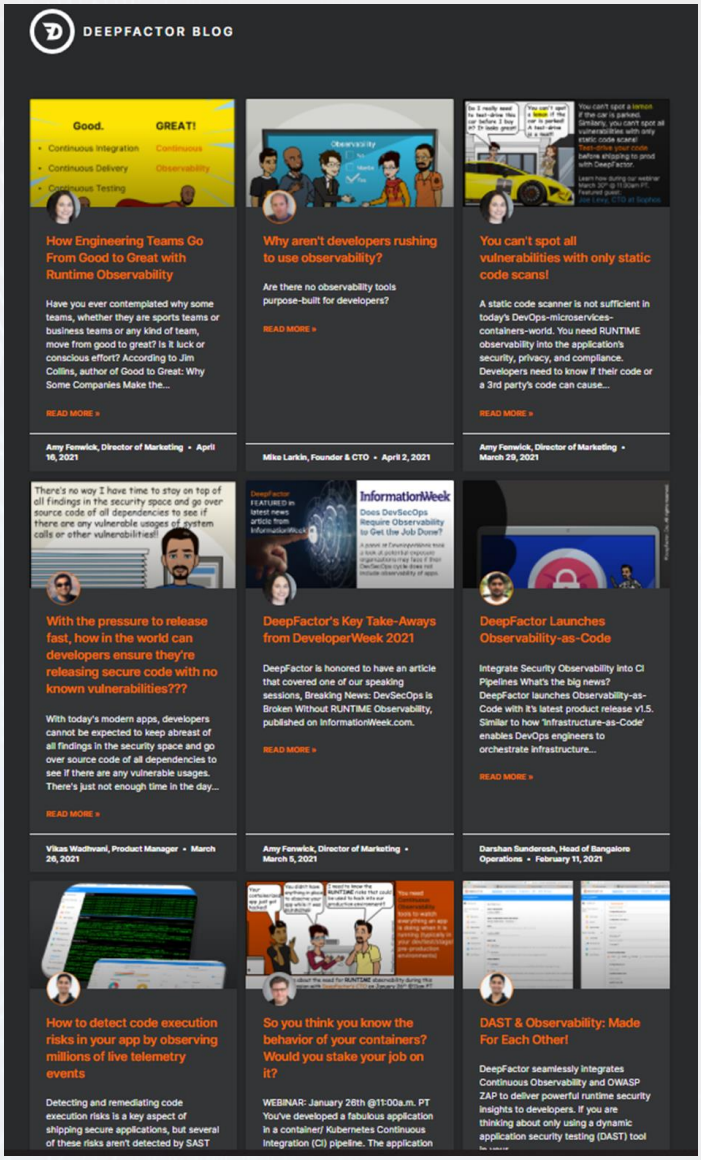
Outgoing Connections

Enter IP or Port

Remote IP	Remote Port	Protocol	Process Name	pid	ppid	Count
104.16.26.35	443	TCP	node	42	1	1
104.16.23.35	443	TCP	node	42	1	1

Stay Connected.

Read our weekly blogs



https://www.deepfactor.io/blog

Follow us on Social Media

https://www.linkedin.com/company/deepfactor

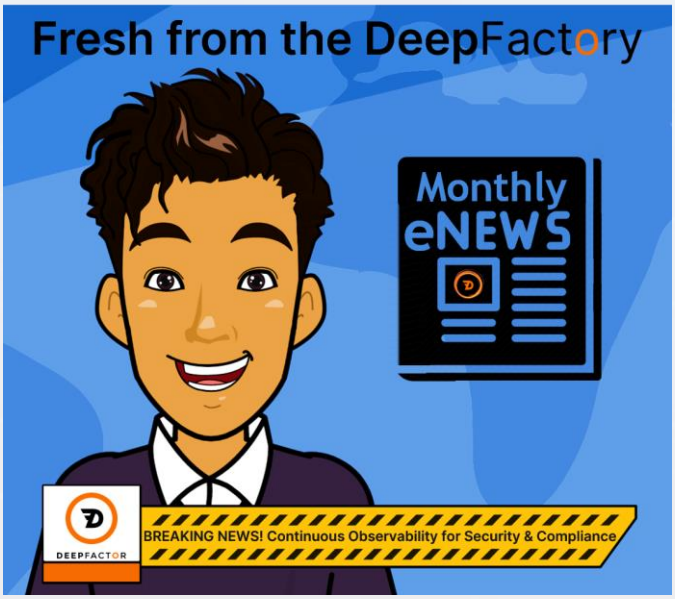
https://twitter.com/DeepFactor_Inc

https://www.facebook.com/DeepFactor.io

Subscribe to our monthly eNews: "Fresh from the DeepFactory"

In addition to our weekly blogs, we're providing another way for you to stay connected with and informed about DeepFactor. This monthly eNewsletter will summarize:

- Continuous observability news for Engineering teams, AppSec teams, and Engineering Leadership
- Industry events & webinars
- New product features and enhancements
- Customer stories
- Recent blogs
- And more!

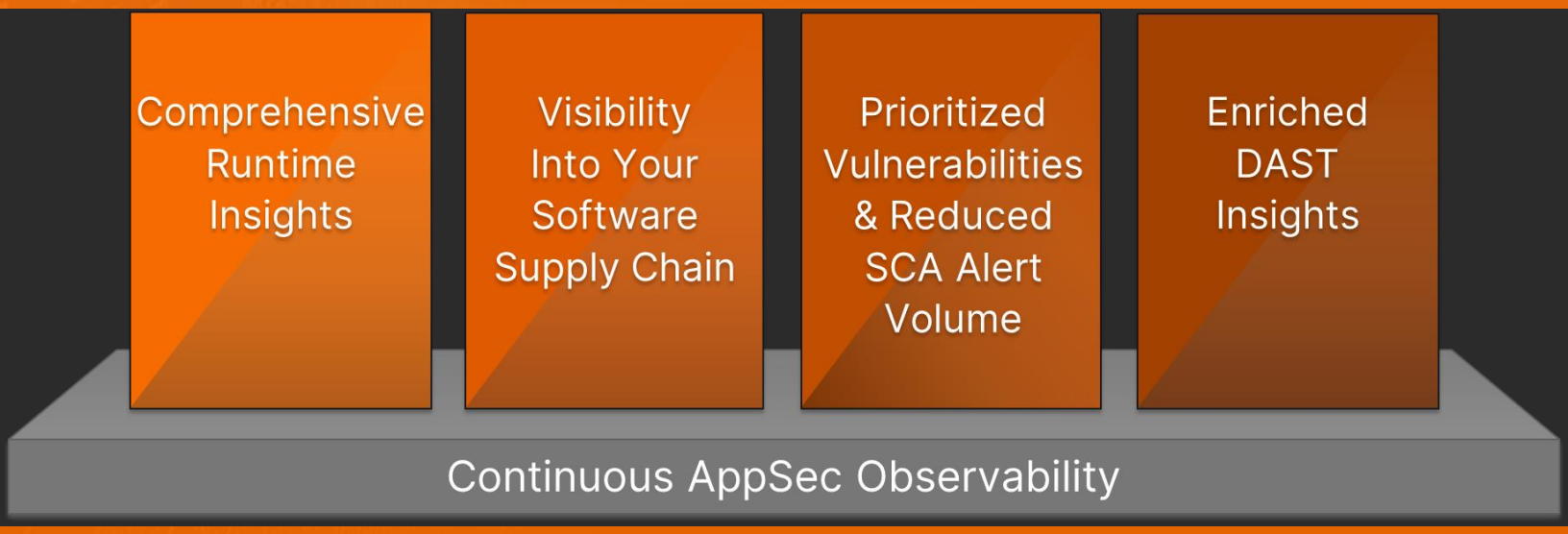


https://www.deepfactor.io/subscribe-to-deep-factory-e-news

About DeepFactor.

You no longer need to choose between shipping fast versus secure to production—DeepFactor empowers you to deliver both with confidence.

- DeepFactor was **created for developers by developers**
  - 100+ years of combined software development, security, and DevOps experience
  - Formerly key players for Citrix, Cisco, IBM, Qualys, and HPE
  - Offices located in the U.S. and India
- Observe billions of application events at runtime
- Detect anomalies to identify security and compliance risks
- Enable engineering teams to create secure and compliant apps



hello@DeepFactor.io

DeepFactor enables Engineering to break down silos, reduce friction, and have seamless collaboration with the AppSec team, turning Dev into AppSec champions.

CONTINUOUS OBSERVABILITY FOR SECURITY & COMPLIANCE

Request your demo today!

demo@deepfactor.io

www.DeepFactor.io