{deepfactor}

Meetup

Uncovering Hidden Security Risks in C++ Apps with LD_PRELOAD and API Interception

TUESDAY, APRIL 11TH

KIRAN KAMITY
CO-FOUNDER, DEEPFACTOR

// Evolution of software development over the last decade

Monoliths to Containers

currently using or planning to use containers in production

Release Speed

release code weekly or more frequently

Testing Automation

of functional testing is automated

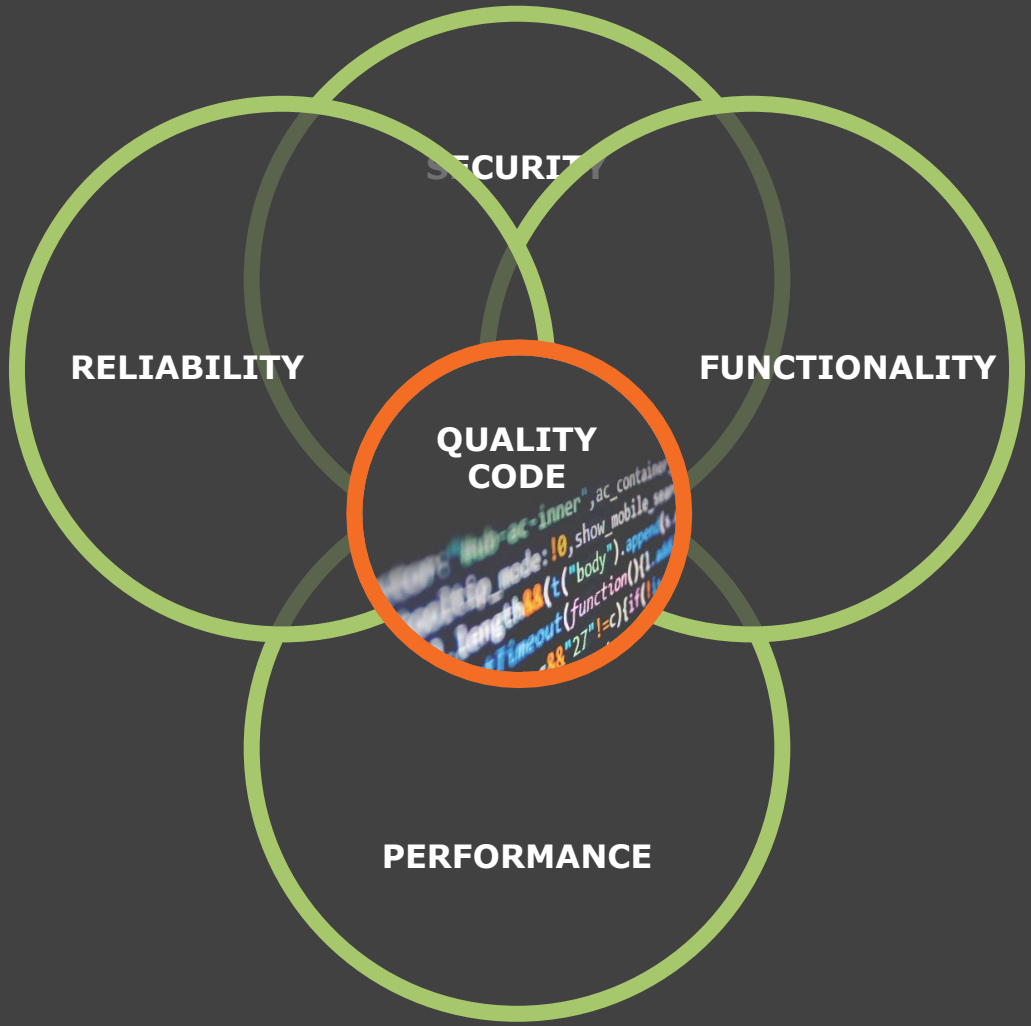
source: CNCF & The State of Testing Report 2022

"Quality" now entails user experience, quality of service, availability, performance, **security** and even business impact.

Improve Software Quality by Building Digital Immunity

Gartner

// Security is now part of quality code



// Security hasn't adapted to modern development

Web App Vulnerability Breach

$2B loss

source: The Business Journals

Supply Chain Attacks

$90M loss

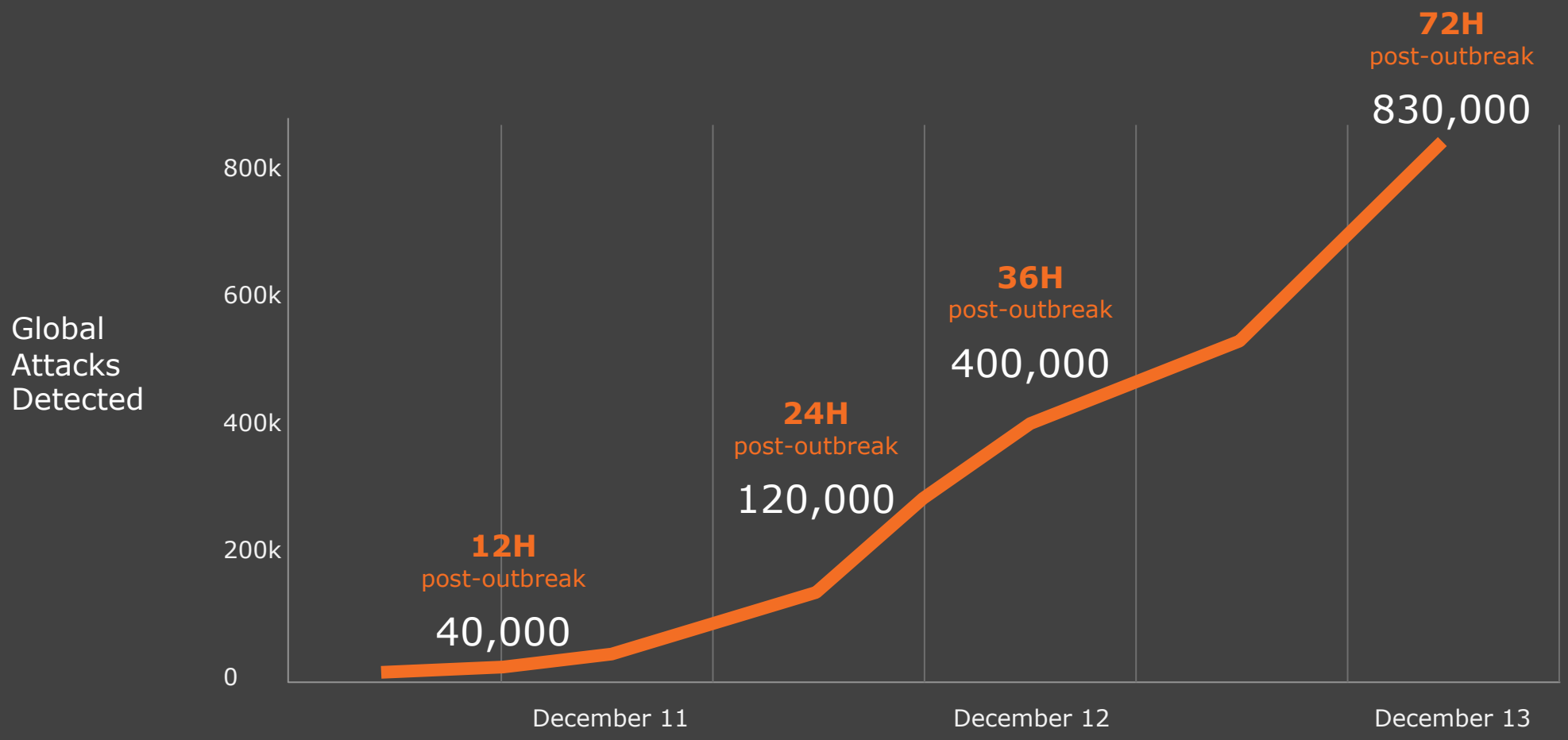
source: bitsight

Zero Day Open Source Vulns

830K attacks in 72 hours

source: checkpoint software

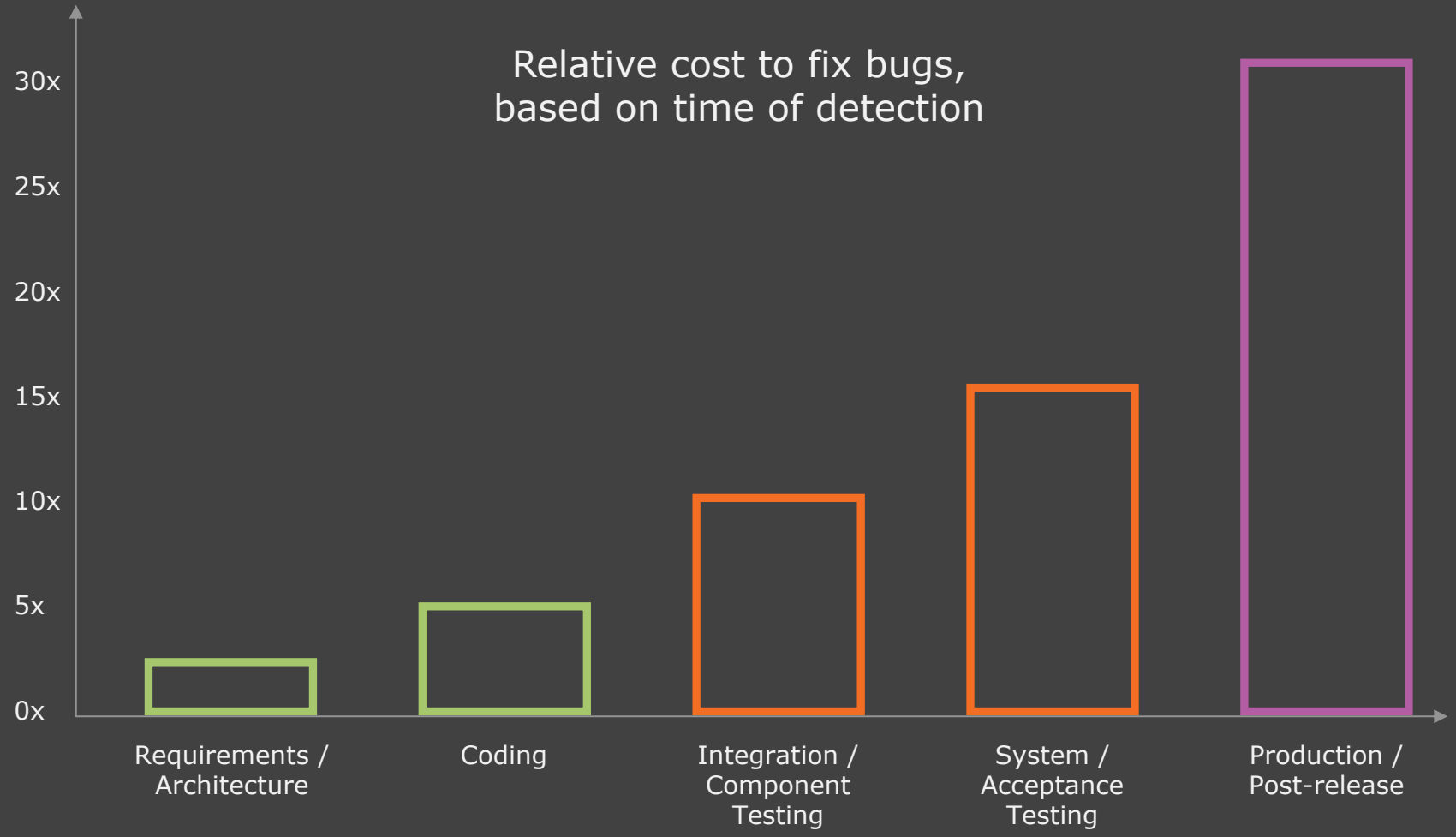
// Log4J Day Zero



source:checkpoint software

{df}

// The cost of fixing vulnerabilities grows exponentially the later you fix it



source:NIST

// Different Approaches to Uncovering Security Risks

Static Application Security Testing (SAST)

Scan source code for known vulnerabilities.

Software Composition Analysis (SCA)

Scan for know vulnerabilities in OS packages, dependencies, and OSS

Runtime Security Dev & Test

Identify unknown vulnerabilities that can't be detected by SAST/SCA by analyzing running apps

Runtime Security In Production

Detect insecure file, network & memory behaviors for security & compliance with PCI/SOC2/..

Dev

Prod

{df}

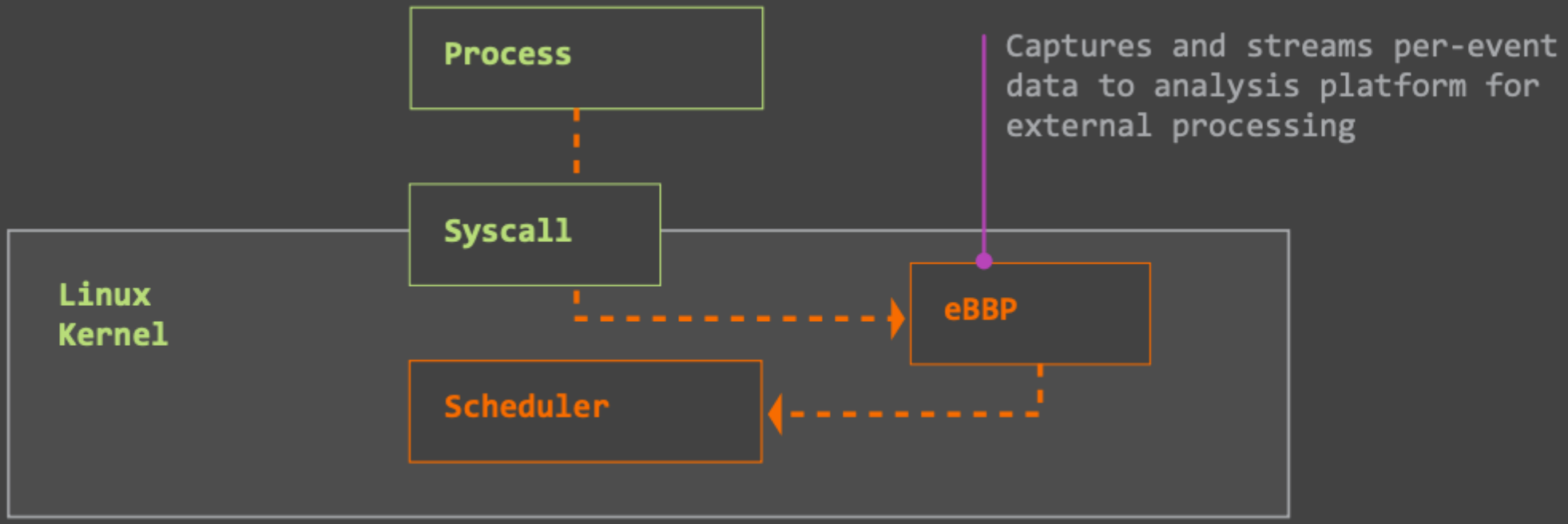
// Comparison of API Interception Techniques

	Inline Hooking	Import Table Rewriting (LD_PRELOAD)	Callbacks (eBPF)	Method Rewriting
Performance				
Deployment and Maintenance				
Stability				
Compatibility				
API Breadth				

// Example of Import Table Rewriting with LD_PRELOAD


```
void *malloc(...) { // perform any additional desored malloc functionality }
```

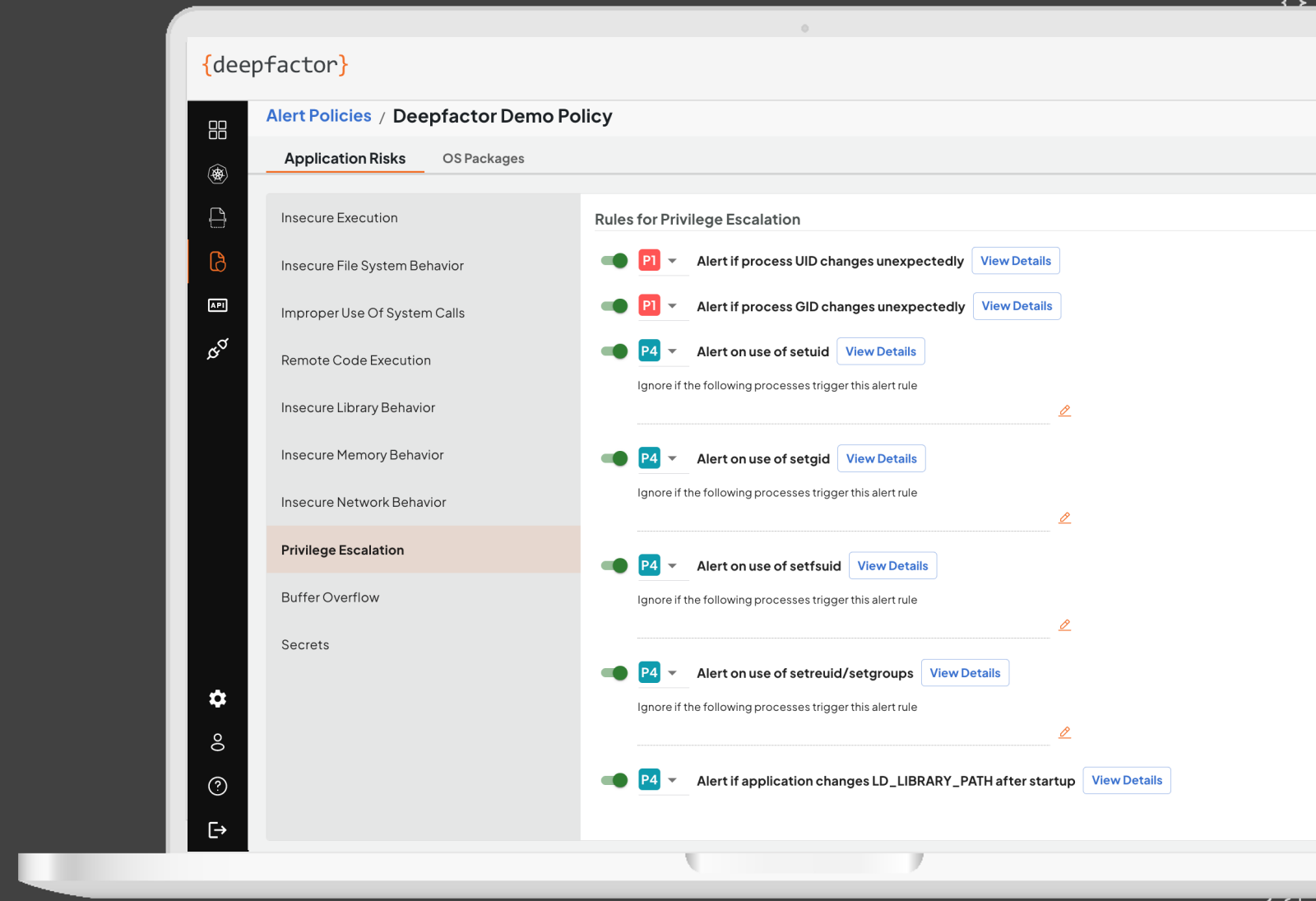
// Example of Callbacks with eBPF



// Runtime Analysis

Runtime Security

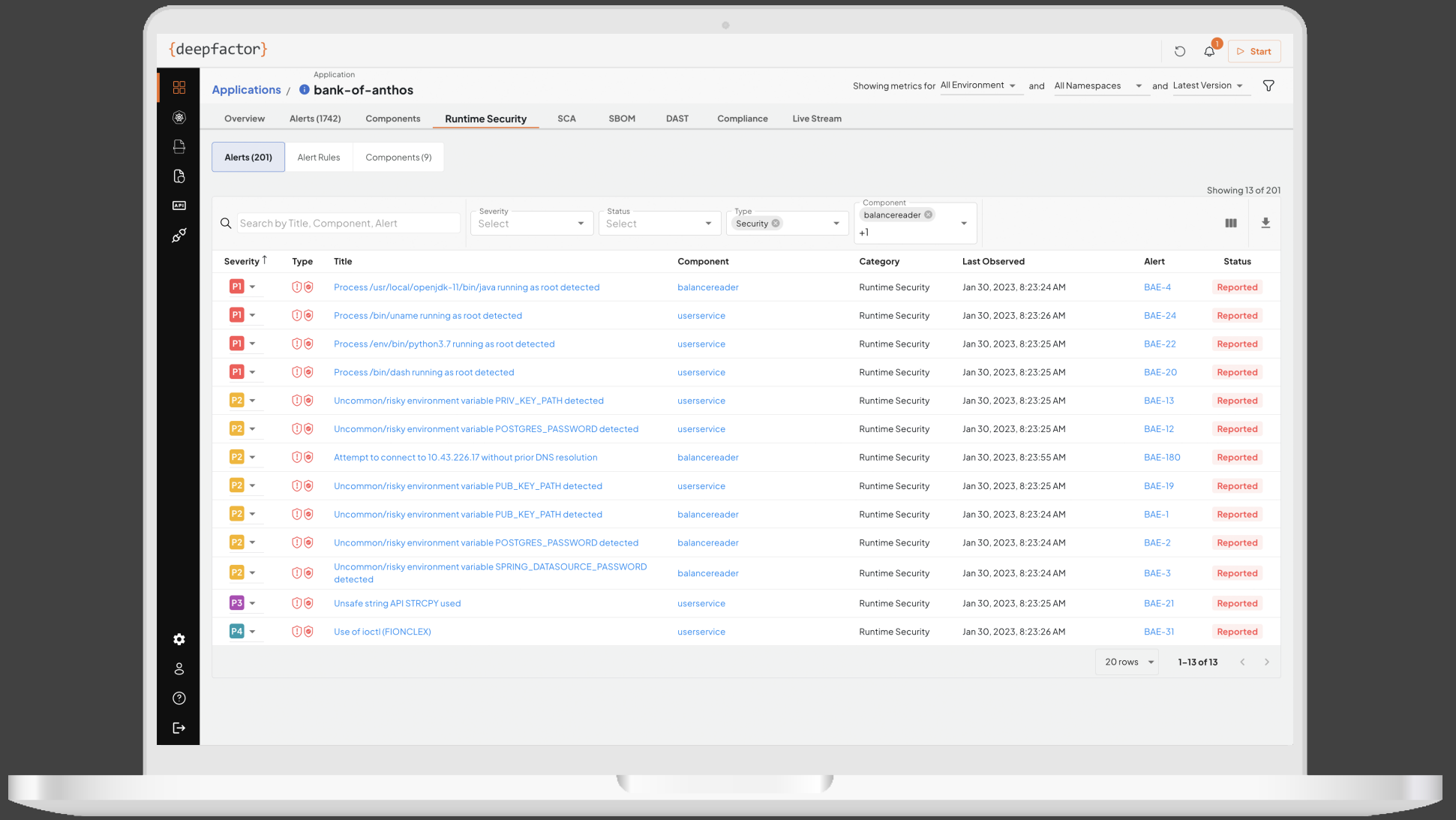
- Execution
- File system behavior
- System calls
- Remote code execution
- Library behavior
- Memory behavior
- Network behavior
- Privilege escalation
- Buffer overflow
- Secrets



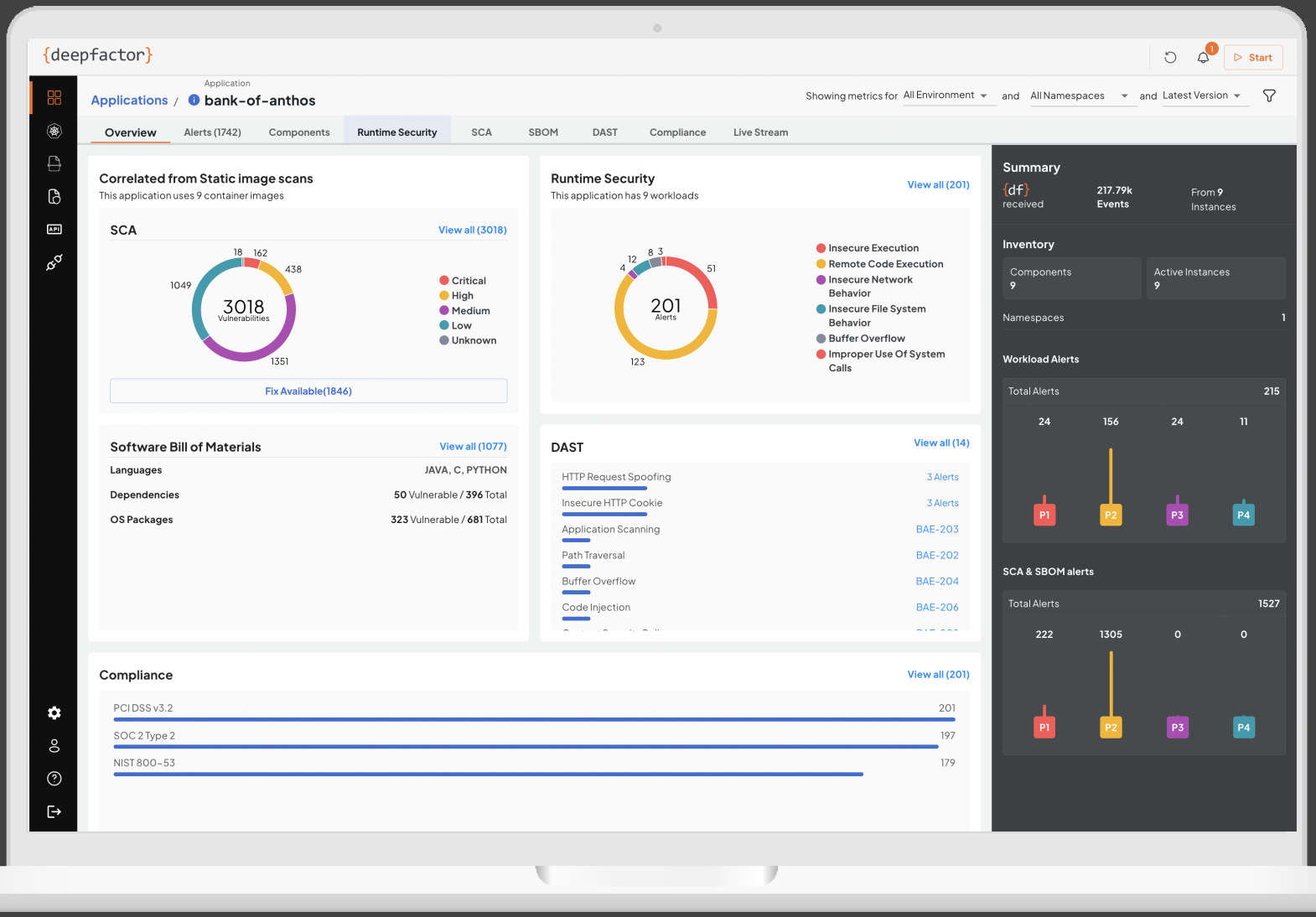
// Deployment Modes

- Monolithic applications
  - dfctl run -a "my application" --cmd COMMAND_WITH_ARGUMENTS
- Containers
  - dfctl run -a "my application" --docker-run DOCKER_RUN_OPTIONS --image IMAGE
- Kubernetes
  - helm --install df-webhook-stable -n df-webhook deepfactor/webhook --set clusterName=CLUSTER_NAME_OF_YOUR_CHOICE --create-namespace -f webhook-override.yaml
- Docker Swarm, AWS ECS, Lambda w/ container …
  - docker build -f Dockerfile.df --build-arg "APP_IMAGE=${APP_IMAGE}" --build-arg "APP_IMAGE_ID=`docker inspect ${APP_IMAGE} --format '{{.Id}}'`" --build-arg "DF_APP_NAME=my application" .

// Demonstration



// Free 60-minute Risk Assessment Report



hello@deepfactor.io

www.deepfactor.io