eBPF-Centric Security Observability and Investigation in Kubernetes

Mihai Alexandru ANDREI · Alexandru CHIS · Titus Constantin BALAN

Universitatea Transilvania din Brașov

Configuration is not the same as behavior

A successful deployment does not prove that an application behaves as intended.

INTENDED STATE

Policy and workload configuration

What we expect the system to do

EXECUTION EVIDENCE

Network, process and admission observations

What the cluster actually did

The investigation challenge is explaining the gap between intent and evidence.

What is Isolens?

A Terraform-driven Kubernetes laboratory for controlled application investigation.

ISOLENS / PURPOSE

Create a known environment to investigate application behavior.

Operator control plane + Terraform-managed Kubernetes resources

WHAT OPERATORS CONFIGURE

Wards, applications, exposure and policies

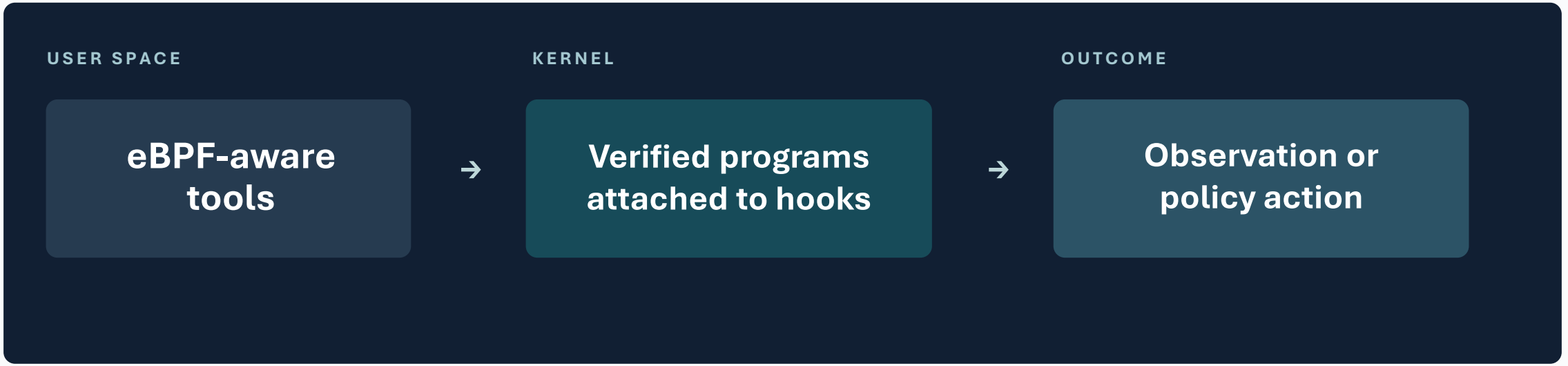
WHAT OPERATORS INVESTIGATE

Network, process and admission evidence

The platform integrates existing tools rather than implementing a new eBPF sensor.

Why eBPF matters to Isolens

eBPF-based tools observe and act on events through Linux kernel hooks.



Hooks may include network events, tracepoints and process-related execution paths.

The tools Isolens integrates

NETWORK / EBPF

Cilium + Hubble

Enforcement + flow visibility

Cilium applies the networking policies. Hubble exposes flow identities and forwarding or drop verdicts.

FORWARDED / DROPPED

RUNTIME / EBPF

Tetragon

Process execution visibility

Runtime events identify executed binaries and their Kubernetes workload context.

process_exec

ADMISSION / NON-EBPF

Kyverno

Deployment constraints

Evaluates Kubernetes admission requests before non-compliant resources are admitted.

ALLOW / DENY

Together, they expose complementary evidence about a configured workload.

Isolens creates a controlled investigation space

Configure a workload, apply constraints, then inspect evidence from complementary layers.

01

Define

Wards, applications, topology and constraints

→

02

Materialize

Terraform stages generate the Kubernetes resources

→

03

Exercise

Run a known application action under controlled rules

→

04

Interpret

Compare configuration and observed evidence

The platform coordinates configuration and execution; evidence correlation remains operator-driven.

Wards connect investigation intent to Kubernetes resources

Analysis subjects, ward namespaces and applications form the experiment model.

ANALYSIS SUBJECT

Application or system under study

WARD · KUBERNETES NAMESPACE

Application A

Policies + resource rules

Templates

Predefined workload starting points.

Scenario blueprints

Repeatable examples assembled from editable ward and application configuration.

Templates and blueprints are starting points; configurations remain editable.

Platform Infrastructure and Deployment

Stages separate cloud foundations, shared Kubernetes services and experiment resources.

AWS FOUNDATION

01 Bootstrap

02 Core

SHARED KUBERNETES PLATFORM

03 Platform prerequisites

04 Platform

EXPERIMENT LAYER

05 Policies

06 Applications

AWS foundation

Remote state, registry, VPC, EKS and node infrastructure.

Shared platform

CRDs, networking, observability, admission engine and control-plane services.

Experiment layer

Ward namespaces, workloads, connectivity rules and managed policies.

The web operator manages policies and applications — not the foundational stages.

Bootstrap and Core establish the cluster

Infrastructure-owned stages prepare remote state and the Amazon EKS foundation.

AWS FOUNDATION

- 01 **Bootstrap**
- 02 **Core**

SHARED KUBERNETES PLATFORM

- 03 Platform prerequisites
- 04 Platform

EXPERIMENT LAYER

- 05 Policies
- 06 Applications

STAGE 01 / BOOTSTRAP

Remote-state and registry prerequisites

S3 remote-state resources and shared Amazon ECR repositories.

→

STAGE 02 / CORE

AWS network and EKS foundation

VPC networking, subnets, IAM integration, EKS control plane and worker nodes.

Result: a provisioned Kubernetes cluster ready to host shared platform services.

Prerequisites and Platform install the shared services

The cluster becomes an environment that can enforce policies and produce evidence.

AWS FOUNDATION

01 Bootstrap

02 Core

SHARED KUBERNETES PLATFORM

03 **Platform prerequisites**

04 **Platform**

EXPERIMENT LAYER

05 Policies

06 Applications

STAGE 03

Platform prerequisites

Required CRDs before platform planning.

STAGE 04

Platform

Cilium + Hubble · Tetragon · Kyverno

Gateway API · Isolens services · Keycloak · PostgreSQL

Shared services are provisioned before workloads and experiment policies.

Policies and Applications define the experiment

These are the only two stages managed through the operator interface.

AWS FOUNDATION

- 01 Bootstrap
- 02 Core

SHARED KUBERNETES PLATFORM

- 03 Platform prerequisites
- 04 Platform

EXPERIMENT LAYER

- 05 **Policies**
- 06 **Applications**

STAGE 05 / POLICIES

Security and runtime constraints

Kyverno admission policies and optional custom Tetragon TracingPolicies.

STAGE 06 / APPLICATIONS

Workloads inside wards

Namespaces, Deployments, Services, ConfigMaps, NetworkPolicies and optional HTTPRoutes.

Ward boundaries are namespace-based controls, not hostile multi-tenant isolation.

The operator control plane manages experiment changes

Operator actions flow through the backend to a dedicated Terraform runner.



IDENTITY

Keycloak (OIDC + PKCE) → backend session

PERSISTENCE

PostgreSQL: configuration, runs, logs and audit

Plan · Apply · Destroy · View Runs & Logs · Cancel · State Unlock

Complementary evidence across the workload lifecycle

One controlled action can be explained through distinct evidence sources.

BEFORE EXECUTION	**Kyverno**	Admission request → validation decision
DURING EXECUTION	**Tetragon**	Process / command → Kubernetes identity
NETWORK PATH	**Cilium + Hubble**	Flow and policy verdict → forwarded / dropped

The operator matches records using Pod, namespace, timestamps and the known action.

A rejected Pod has no workload-specific runtime or network evidence.

CONCLUSIONS

Isolens turns Kubernetes into a controlled space for investigation.

CONFIGURE

Wards, applications and policies define the intended behavior.

OBSERVE

Cilium/Hubble and Tetragon provide complementary evidence.

INTERPRET

The operator relates results to actions and admission decisions.

github.com/Mihaidono/security-observability-cluster