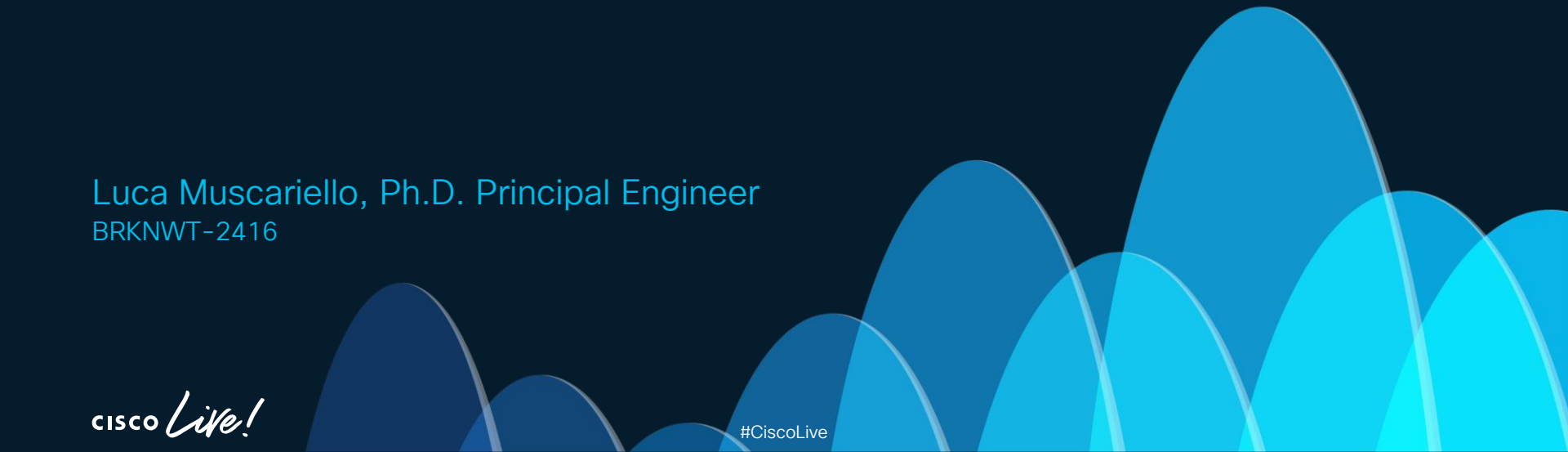
The bridge to possible

AI-Driven Cloud-Native Threat Detection in Realtime

Luca Muscariello, Ph.D. Principal Engineer

BRKNWT-2416



cisco Live!

#CiscoLive

Cisco Webex App

https://ciscolive.ciscoevents.com/ciscolivebot/#BRKNWT-2416

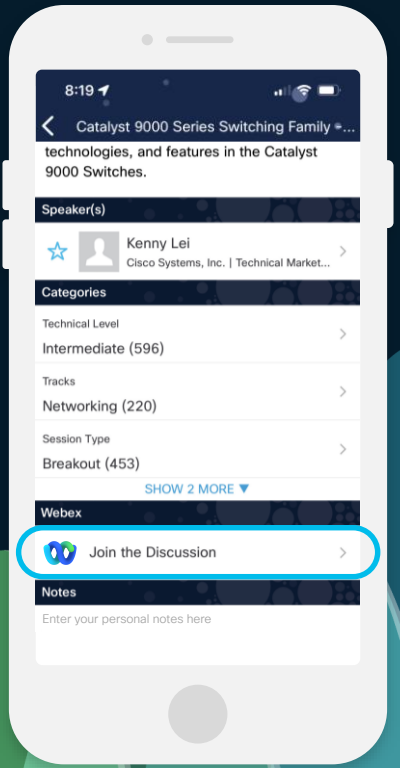
Questions?

Use Cisco Webex App to chat with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until June 7, 2024.



Agenda

- Cloud Detection & Response
- AI-Driven Purple Teaming
  - Formulate Threat Hypothesis
  - Observe Facts and Enrich Data
  - Adversary Emulation
  - Detect Cyber Threats
- Realtime CDR
- Conclusion

*“Cloud-based Cyber-attacks increased by **154%** in 2024 compared to 2023.”*

Checkpoint Research, 2024

Sysdig, AWS Reinvent 2023

“**10 minutes** – that’s all it takes to execute an attack in the cloud after discovering an exploitable target

“The global average cost of a data breach in 2023 was **$4.45 million**, +15% over three years, highlighting the growing financial burden on organizations

Cost of a Data Breach, IBM report, 2023

Mean Time to Detect a Breach today

204 days is the average time to identify a security breach in 2023 (MTTI)

73 days is the average time to contain a security breach in 2023 (MTTC)

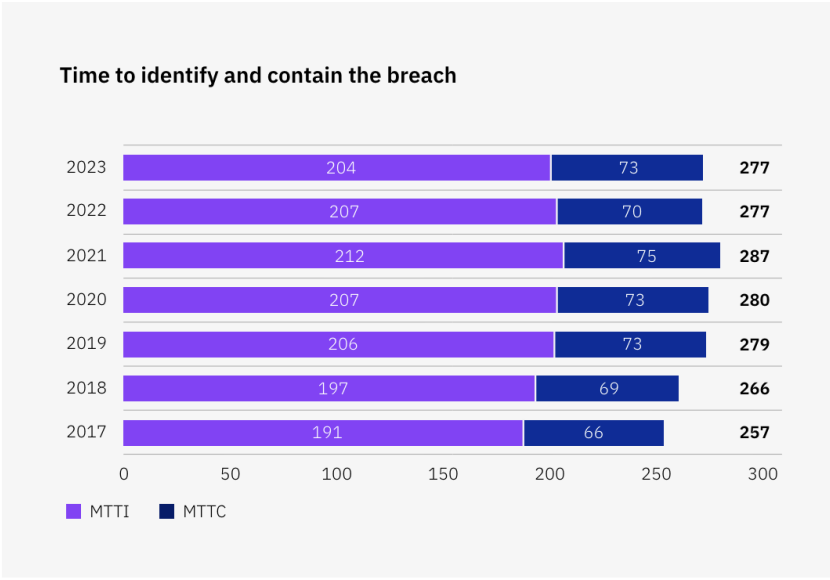
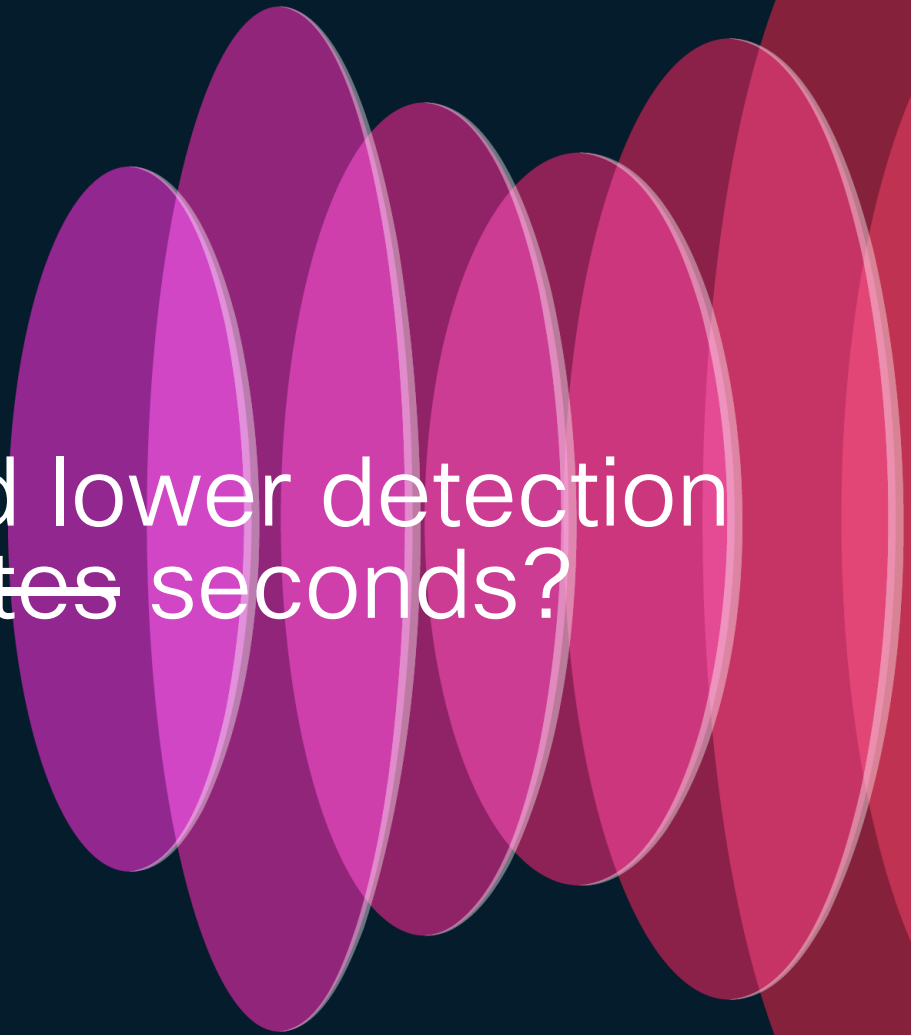
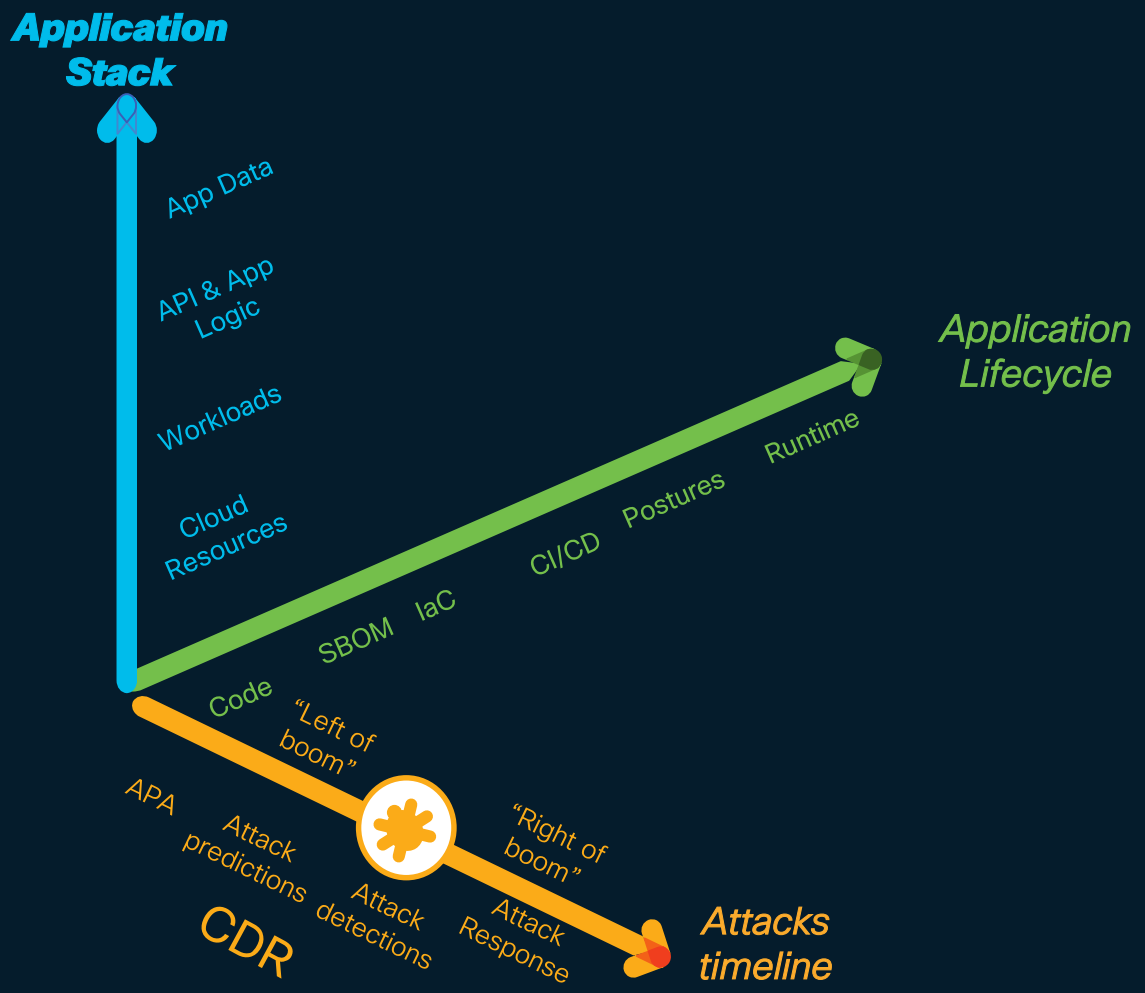


Figure 5. Measured in days

What if we could lower detection time to ~~minutes~~ seconds?



Application Security Dimensions



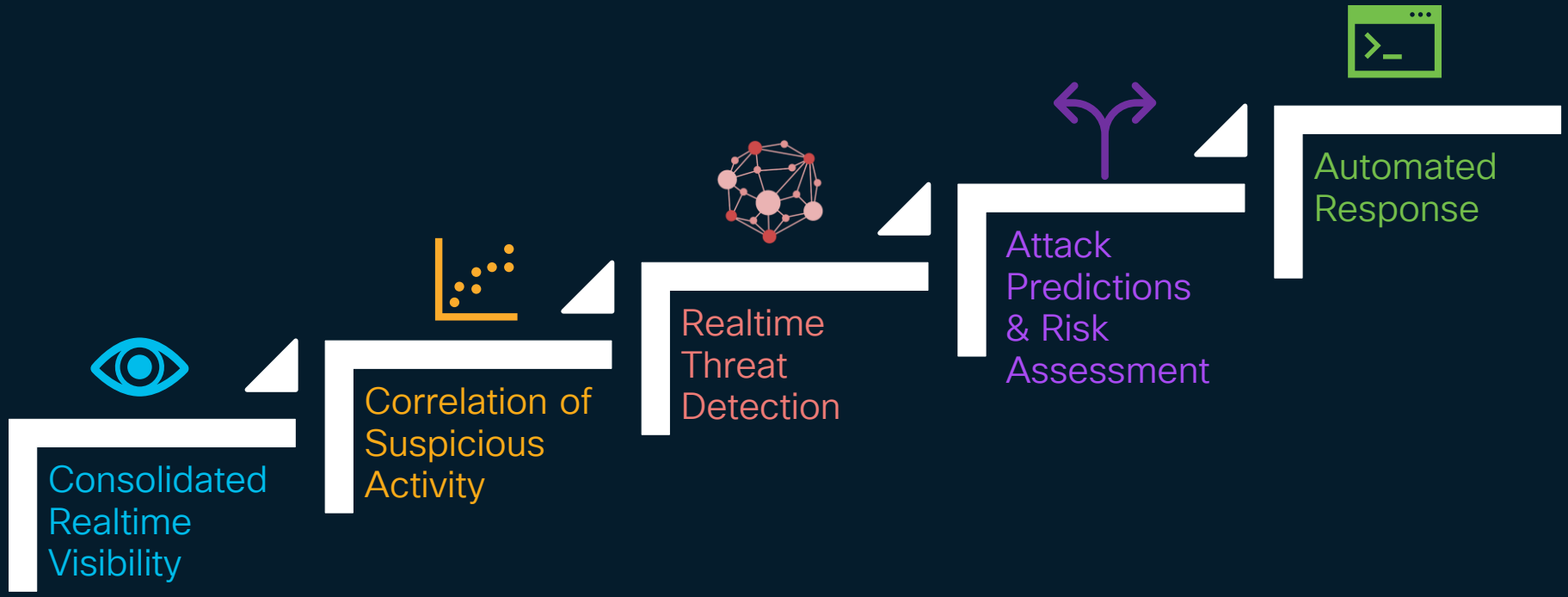
Why CDR

CDR does not replace traditional cloud application security:

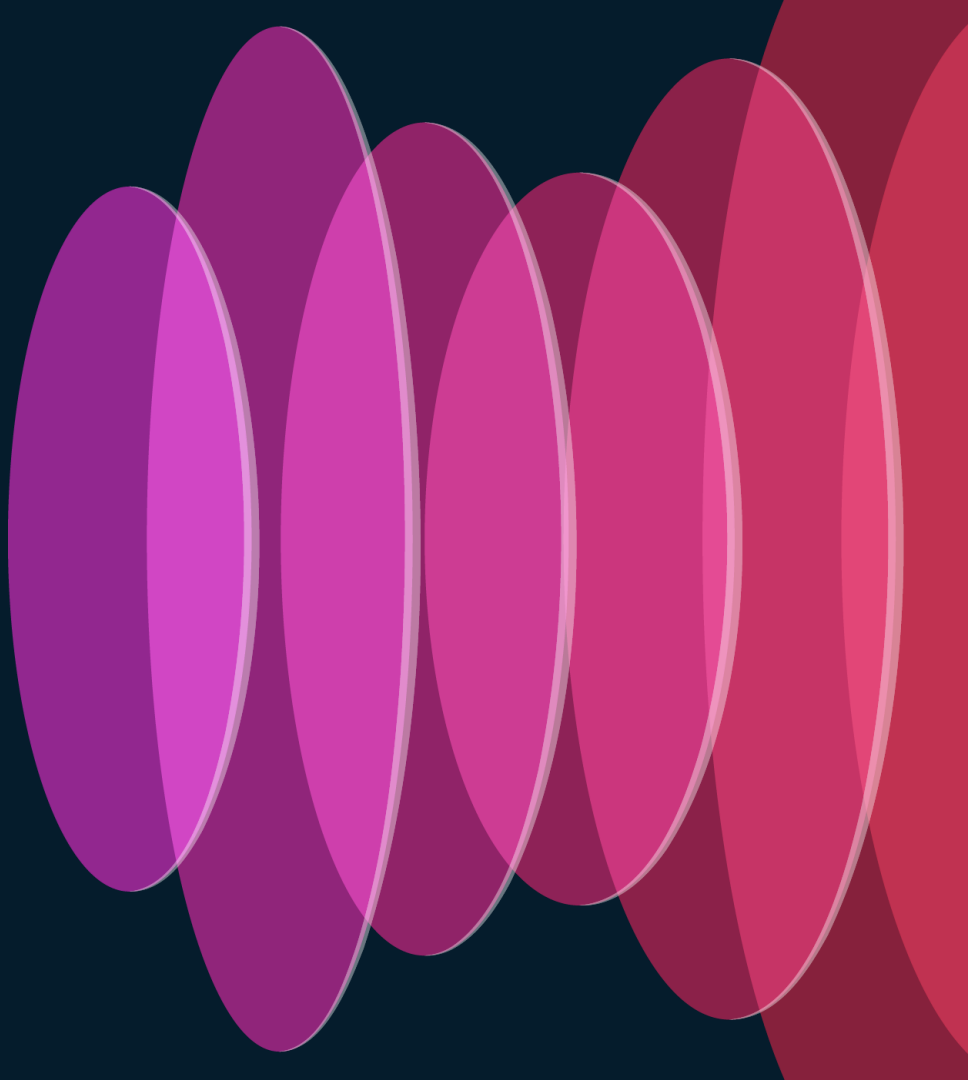
CIEM/CSPM are like locks on the doors ←→ CDR is just like video cameras inside the house providing a live view of what is happening inside, with additional intelligence to:

- Identify suspicious activity
- Provide a stream of security events as a forensic proof of threat detection
- Understand the context of what the attacker is doing in the environment
- Predict the next move

CDR Objective



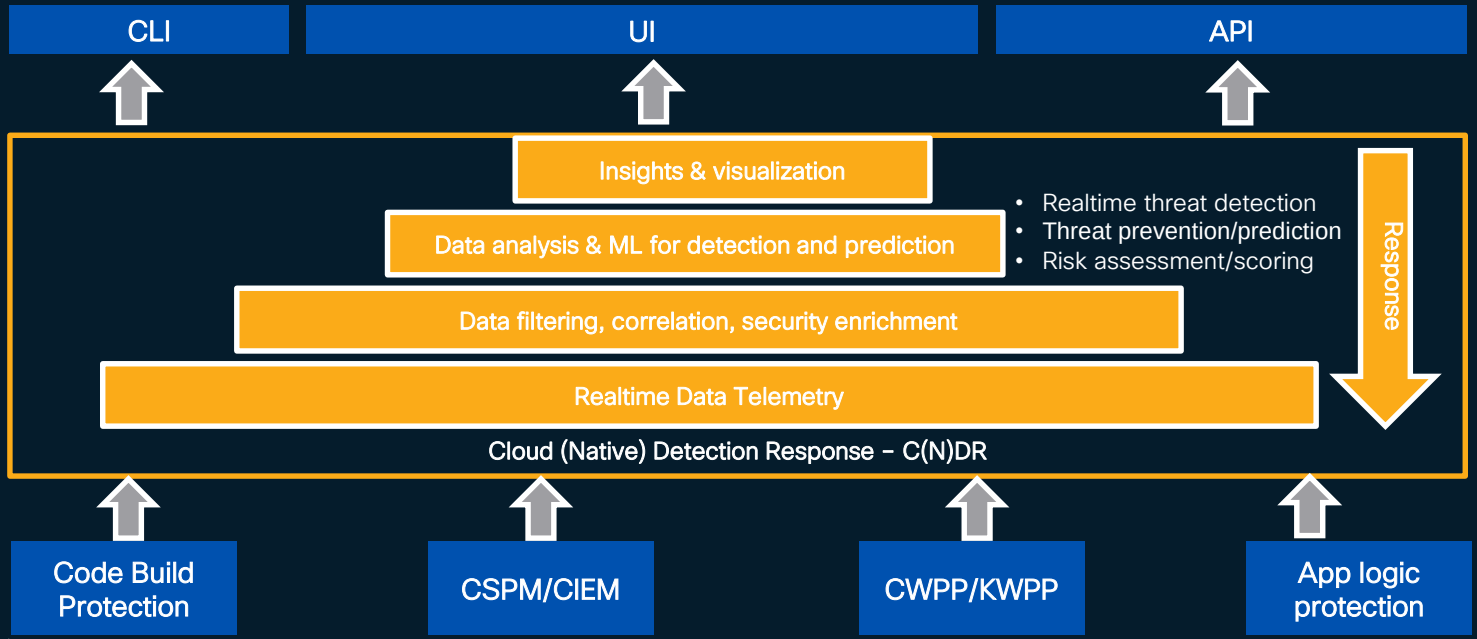
Realtime CDR



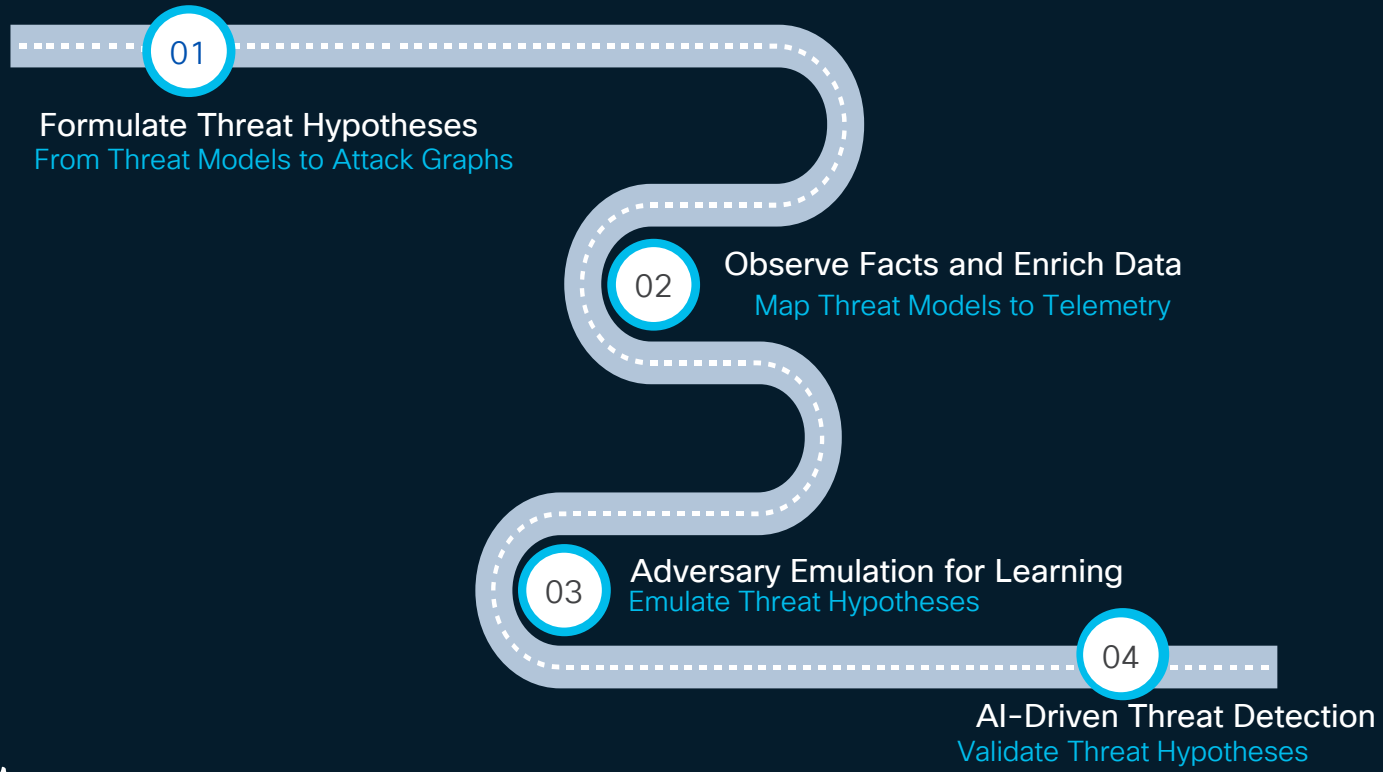
A CNAPP capability

"...many customers are starting to see detection and response as a first-class citizen within CNAPP. Their need is starting to expand beyond just workload runtime security and address the cloud control plane (via analysing cloud logs) to detect suspicious activity across users and services. "

Sysdig blog, March 17, 2023

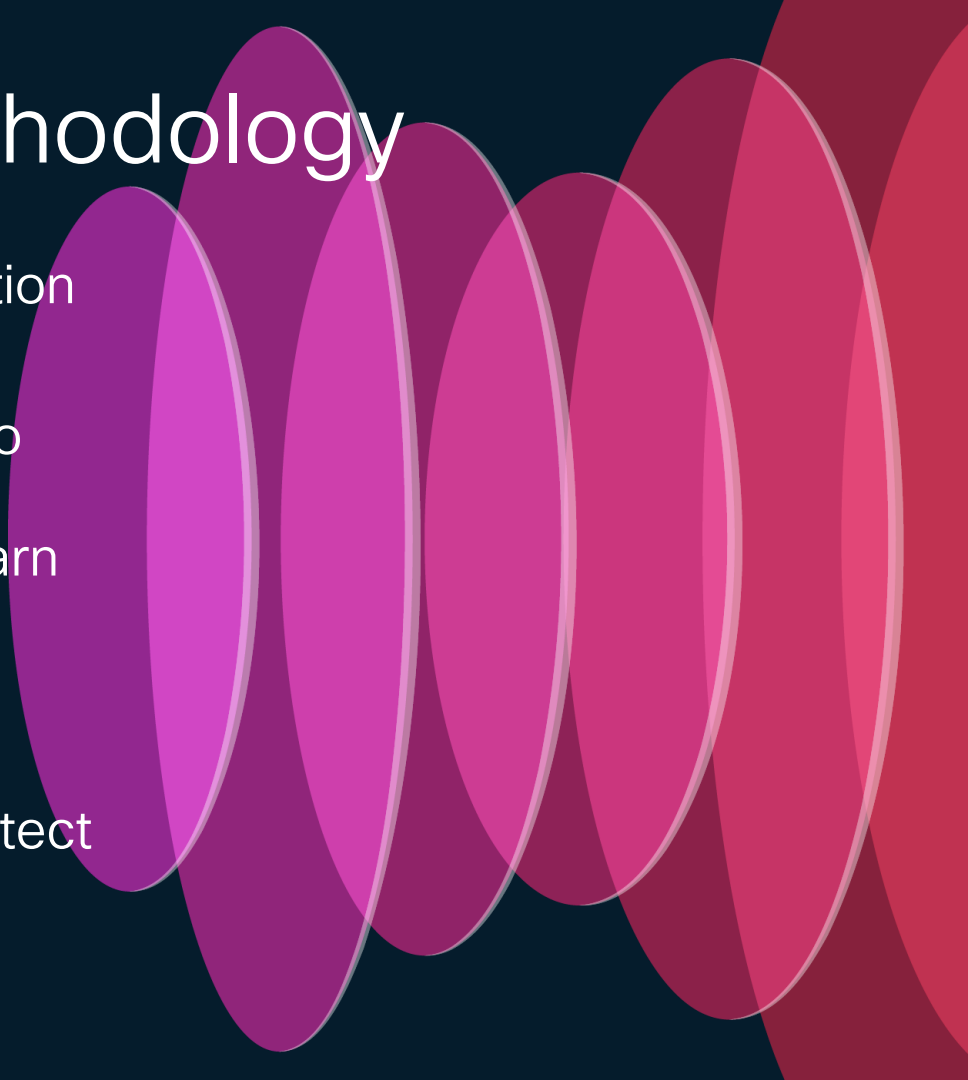


Purple Teaming Approach Overview

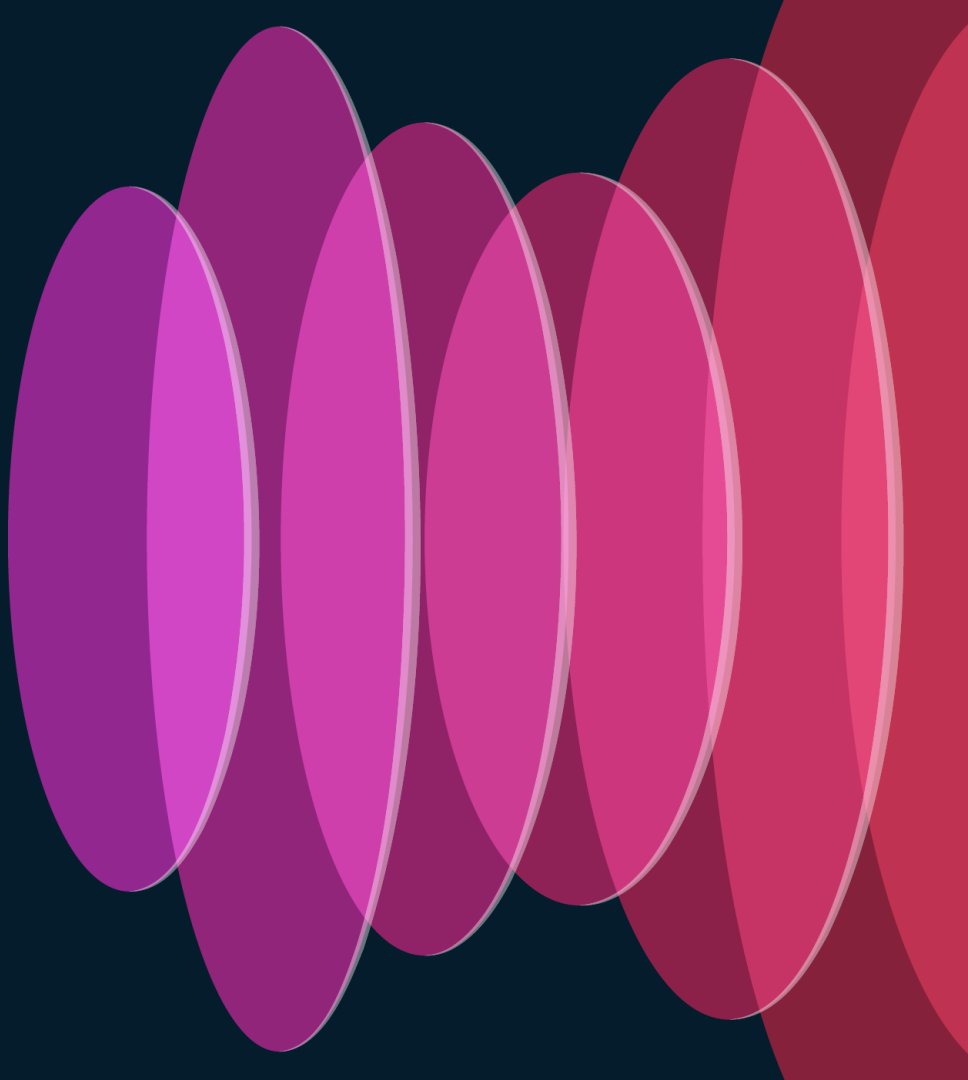


Purple teaming methodology

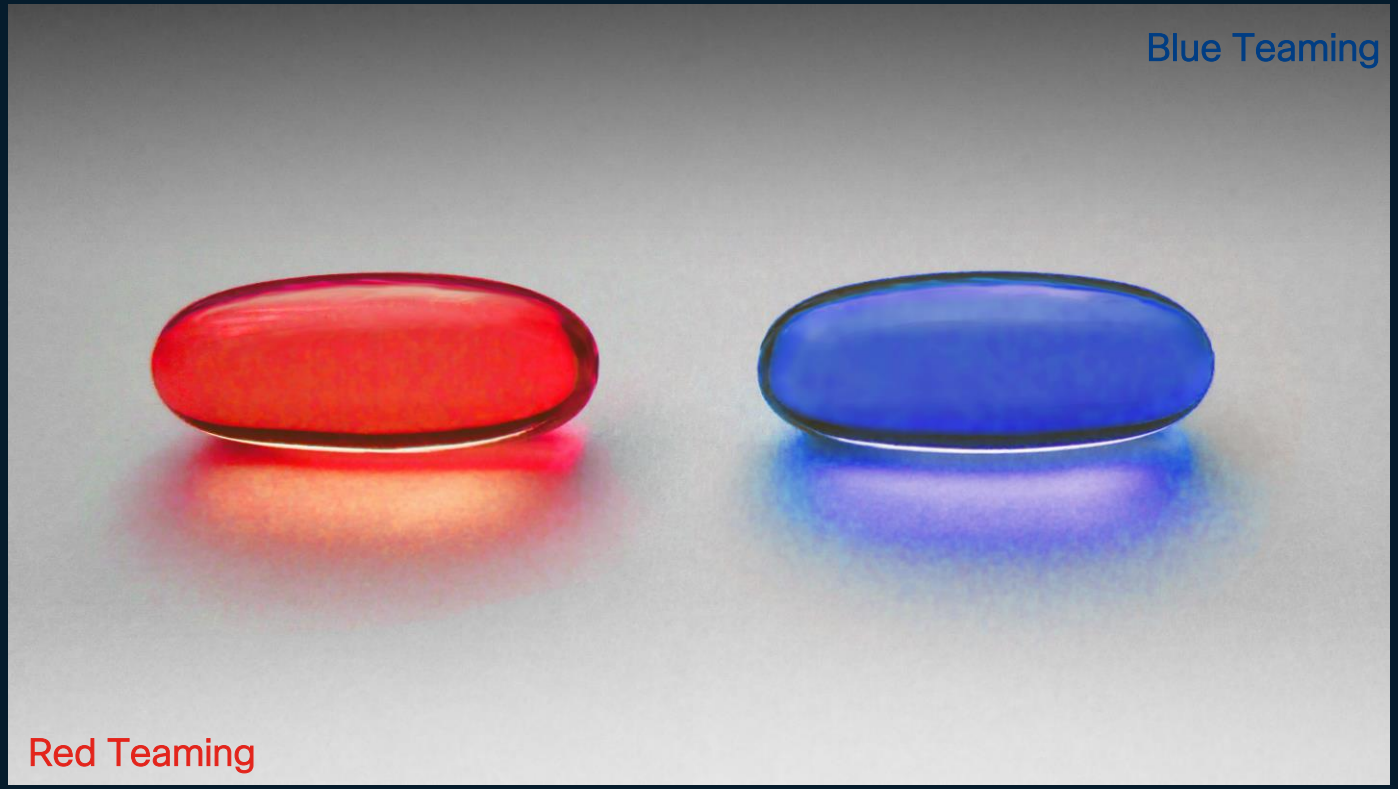
- Broadly speaking it's the collaboration of red and blue teams
- In this context it's a methodology to train an autonomous agent to formulate threat hypothesis and learn how to detect them based on observations
- Also used to optimize telemetry to include observables required to detect threats and attacks



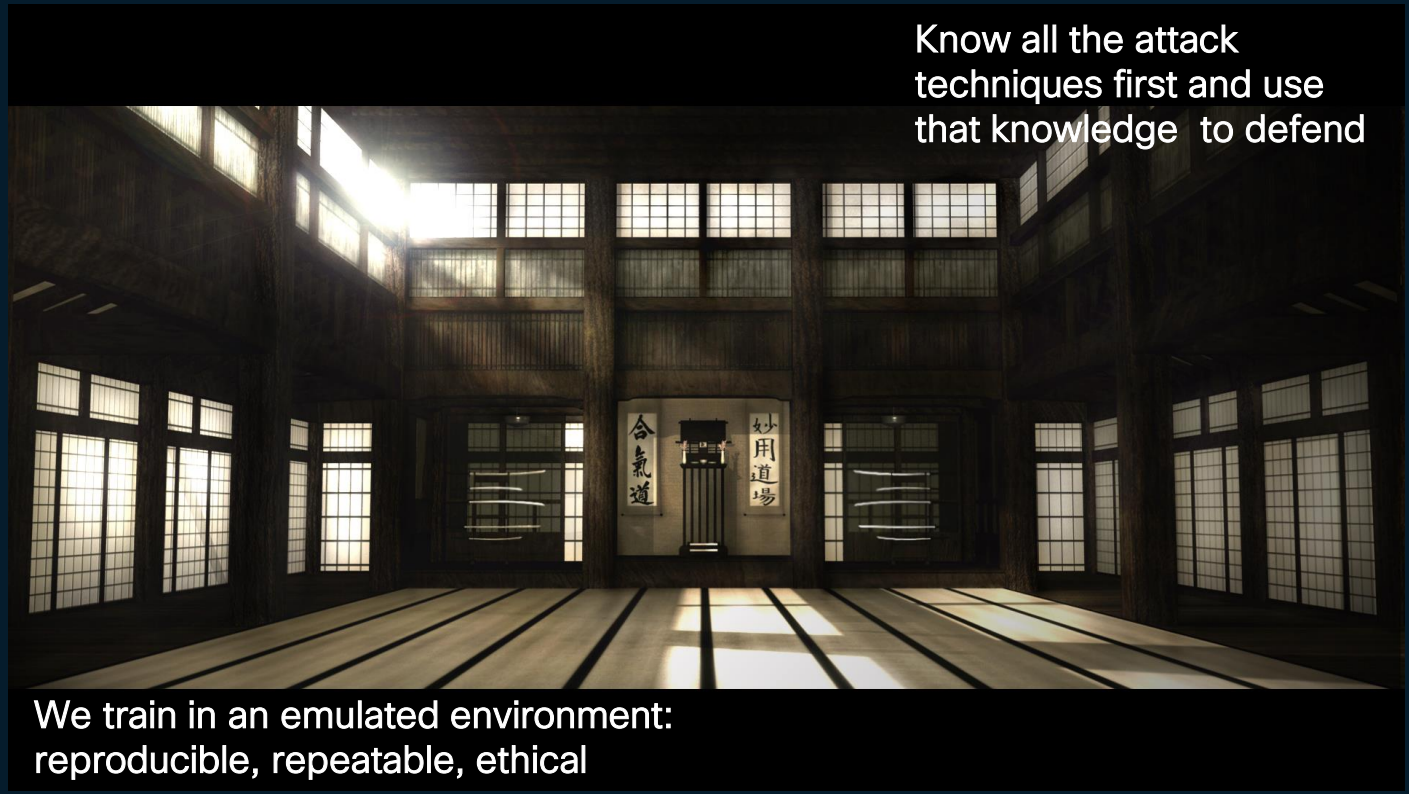
AI-Driven Purple Teaming



Purple teaming methodology



Training in an emulated environment



Note: Images generated by AI

Mapping Emulation to a Real System

- We use a digital twin of a realistic Cloud environment
- Improve accuracy of the emulated world
- Keep the emulation scalable

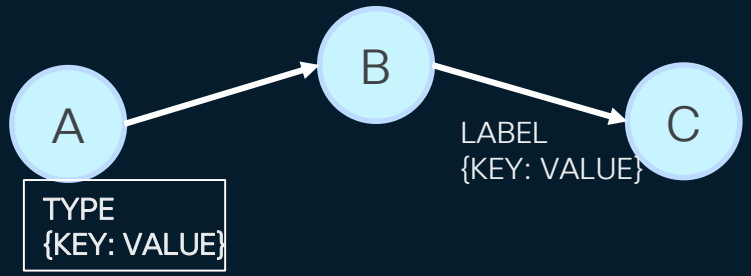
Note: Images generated by AI

Formulate Threat Hypotheses

Knowledge representation and reasoning

- Best hypotheses can be formulated if all the knowledge of the system is represented in efficient data structures to perform reasoning tasks
- Graphs can be used to model how entities, objects or events are semantically interrelated.
- Graphs are powerful models to store knowledge and compute tasks

A Property Graph of the System under Attack

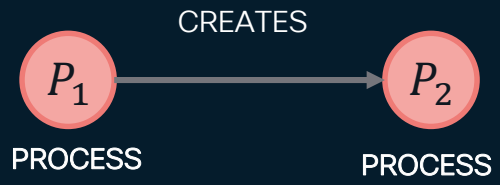


- Nodes
  - Entities with name and type
    - Network flows
    - Processes
    - Files
    - Containers
    - Hosts, Clusters etc.
  - With properties
    - Key-value pairs
- Edges
  - Relationships with a name and type
  - Are directed

Provenance and Context Graphs

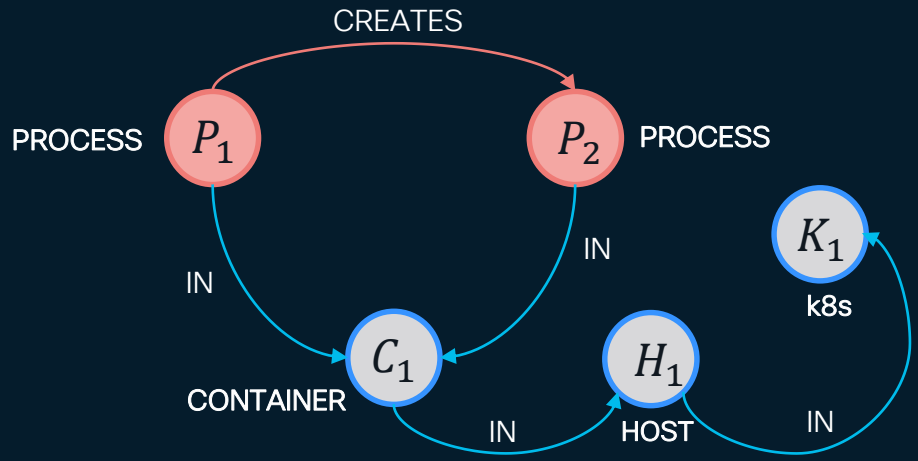
Provenance

- It models activities in the system
- Causality



Context

- System relationship among objects
- Models long term relationship



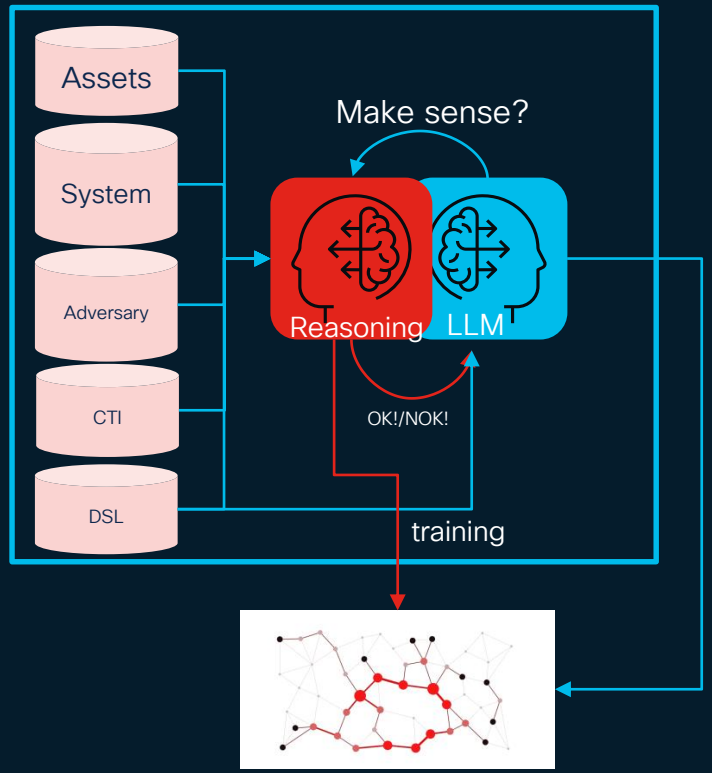
Attack Graphs Generation via DSL Language

- Formulate cyberthreat hypotheses for specific domains (public cloud accounts, private data centres or an enterprise network) and generate attack graphs
- We adopt a meta-attack language (MAL) as a meta domain specific language to specify the threat model of the system S_1
- The DSL language describes assets, events, adversaries of the specific MITRE ATT&CK matrix (tactics/techniques/procedures) and other cyberthreat intelligence.
- Nodes can be assets such as containers, applications etc.
- Edges can be adversaries such as TTPs

```
asset ApiKubectl {
    // get pods
    * discovery[1][uniform(20,1)]
    * execution[1][lognormvariate(2,1)]
        -> End
}

asset ContainerClientSide inherits Attacker {
    * discovery[1][lognormvariate(2,1)]
        -> End
}

asset Container {
    * discovery[1][uniform(20,1)]
}

asset JuiceShopContainer extends ContainerClientSide {
    // SSTi
    fe4445d2-e878-484f-8f85-980b39de0cb5[2][lognormvariate(3,1)]
}

asset Shell {
    * discovery[1][uniform(20,1)]
    * collection[1][uniform(20,1)]
    * execution[1][uniform(20,1)]
        -> End
}
```

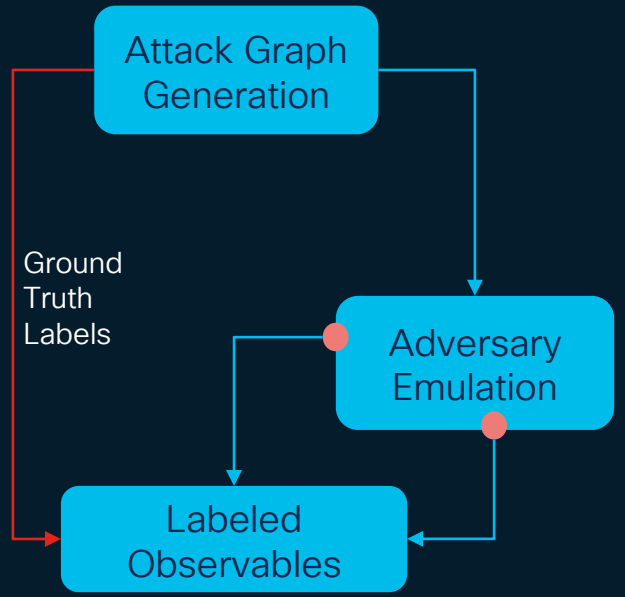
Formulate Threat Hypthesis with Probabilistic Attack Graphs Generation

- Hypotheses can be generated in two ways to produce attack graphs
  - Graph traversal based: it requires defining constraints and an optimization objective to generate focused hypotheses, e.g. “Data Destruction”.
  - LLM based: acts as meta-heuristic that can quickly provide hypothesis to validate.
  - Hypothesis can be verified efficiently.



Adversary Emulation and Observables

- Attack graphs are not observable as they are instances of attack hypotheses
- They can be used to execute an attack on an emulated system
- Observables can be extracted from the emulated systems with different telemetry sensors
- Emulation is powerful and yet expensive and hard to scale.

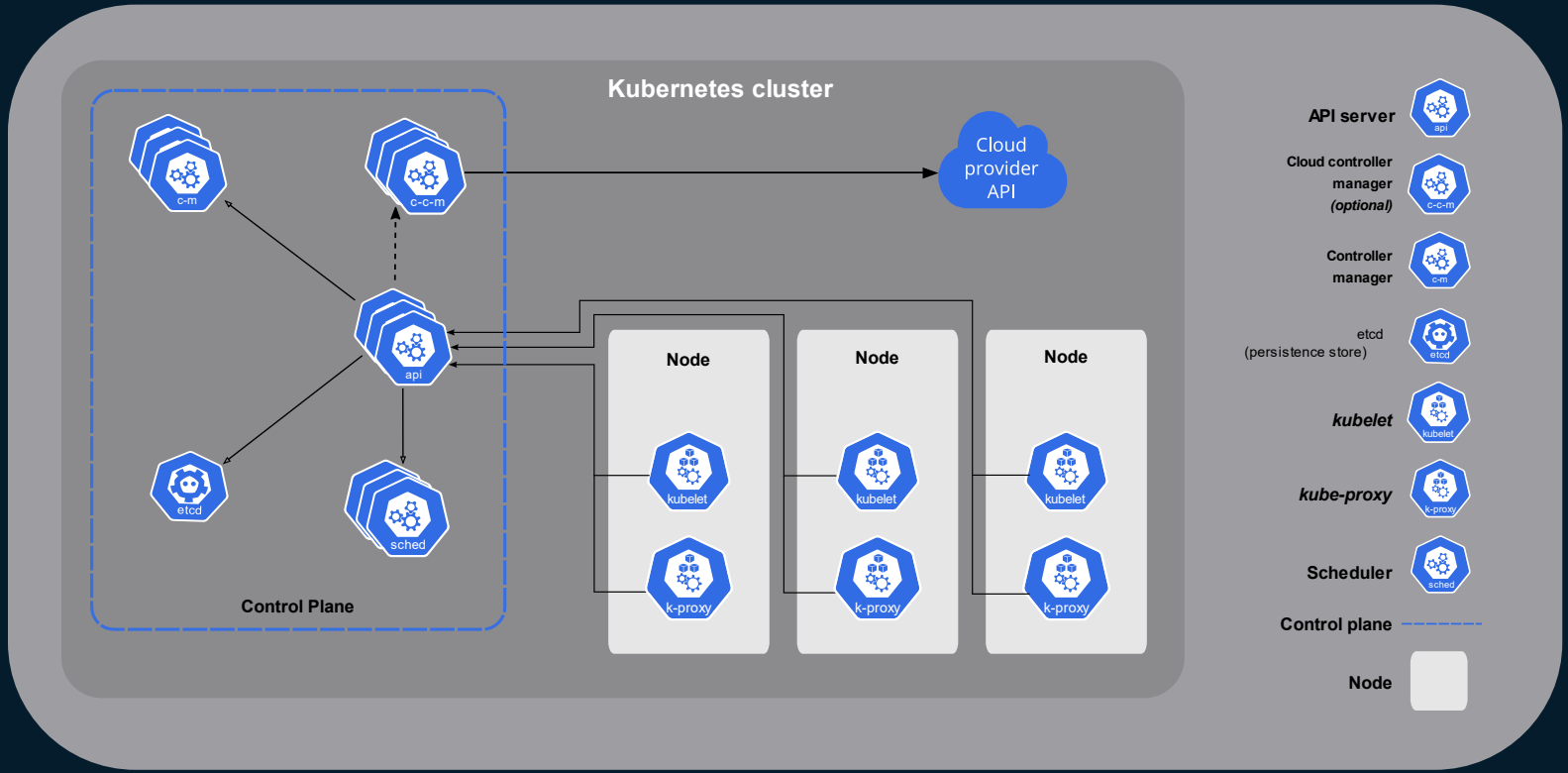


Current Focus
MITRE ATTA&CK Threat Matrix for Kubernetes

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Impact
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access the K8S API server	Access cloud resources	Data Destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account	Resource Hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access container service account	Network mapping	Cluster internal networking	Denial of service
Application vulnerability	Application exploit (RCE)		Access cloud resources	Connect from Proxy server	Applications credentials in configuration files	Access Kubernetes dashboard	Applications credentials in configuration files	
Exposed Dashboard	SSH server running inside container					Instance Metadata API	Writable volume mounts on the host	
							Access Kubernetes dashboard	
							Access tiller endpoint	

Observe Facts and Enrich Data

Observability for Cloud-Native Applications



Realtime observables in Kubernetes

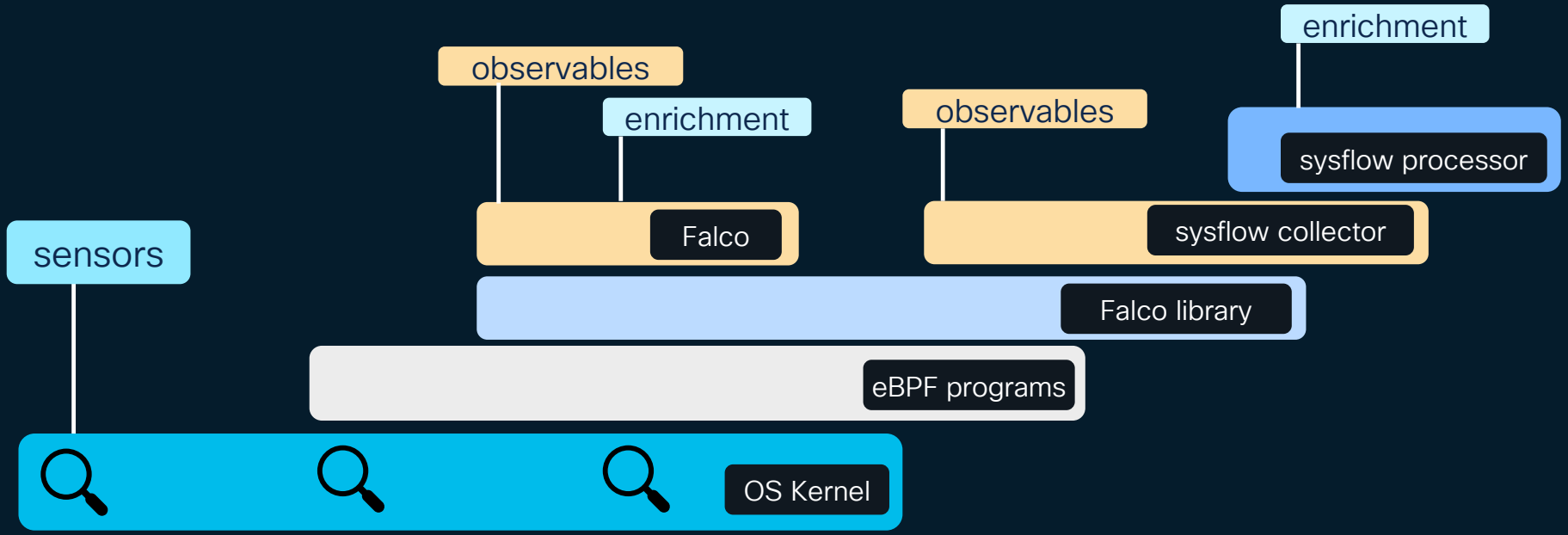
- Kubernetes can belong to a public cloud account (EKS, GKS, AKS etc) or to a private DC.
- We focus on system observables that can be collected from eBPF sensors from the host nodes of the cluster
- We focus on suspicious activities that can be recognized using a state-machine-based rule engine (e.g. falco) or a grammar-based engine (sysflow)
- Falco and Sysflow kernel system-calls based observables
- Kubernetes audits logs and API

Extended OCSF – Open Cybersecurity Schema Framework

- Created by AWS and Splunk to optimize ETL data and cybersecurity events
- A flexible and extensible framework
- We have adopted and extended OCSF to model knowledge graphs for reasoning using ETL data events and system data

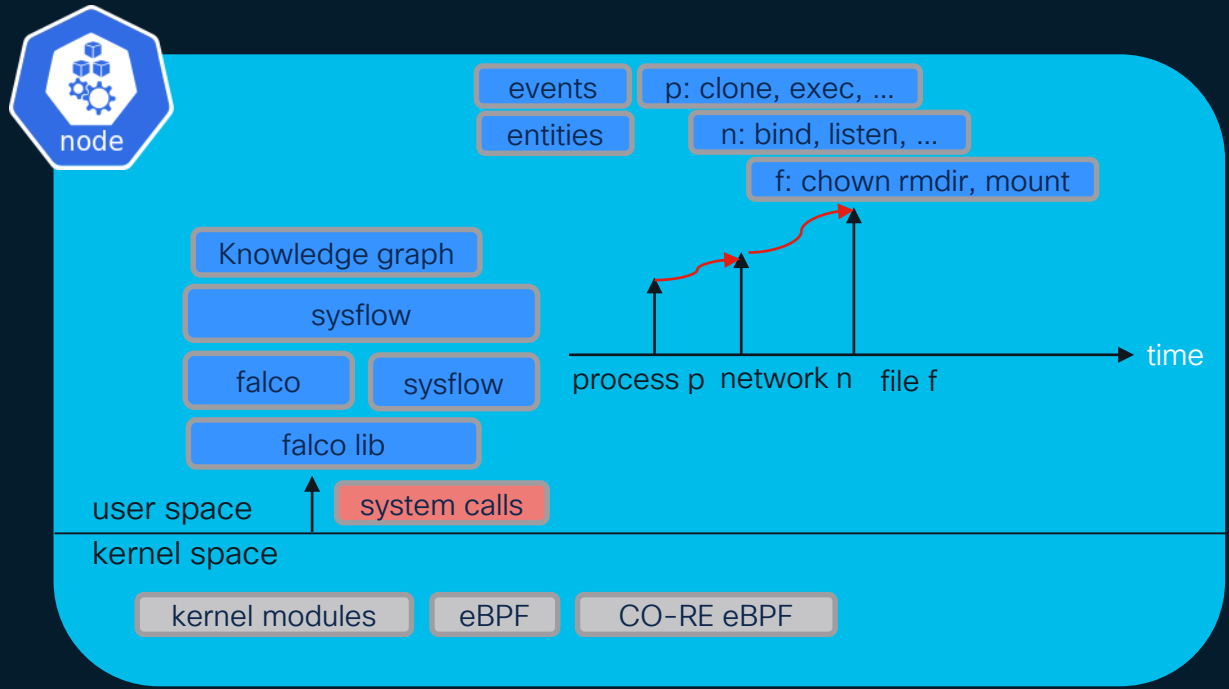
Open Cybersecurity Schema Framework

Linux Foundation CNCF/sysdig Falco and IBM sysflow

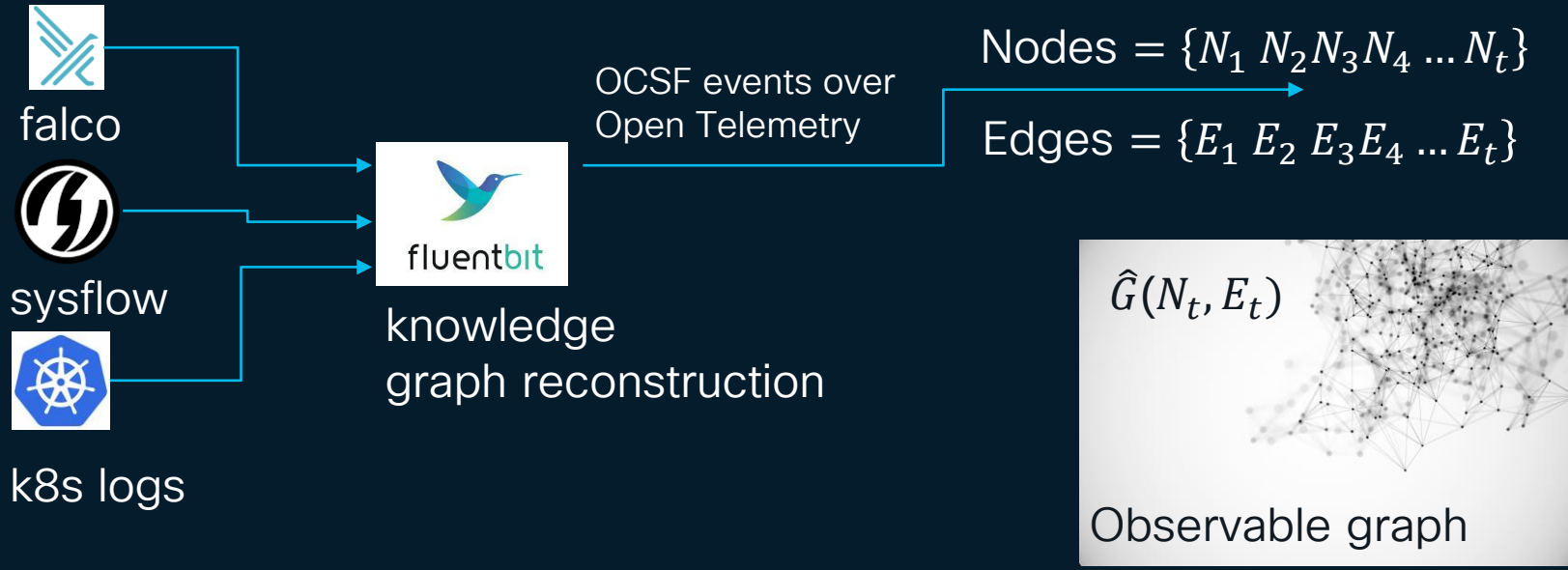


System Based Observability

- Falco extracts efficiently system calls from the node kernel
- Falco models the system activity via a state machine w/ entities and events such as process, network, file.
- sysflow models the system as a DSL language which recognizes events and activities to enrich data to create the provenance graph
- Fluent bit filters and merges information as graph nodes and edges to feed an ETL pipeline.



Knowledge Graph Reconstruction in Real-time



Training and Inference

The sampled graph $\hat{G}(N_t, E_t)$ constitutes the data that can be used for training the detection architecture

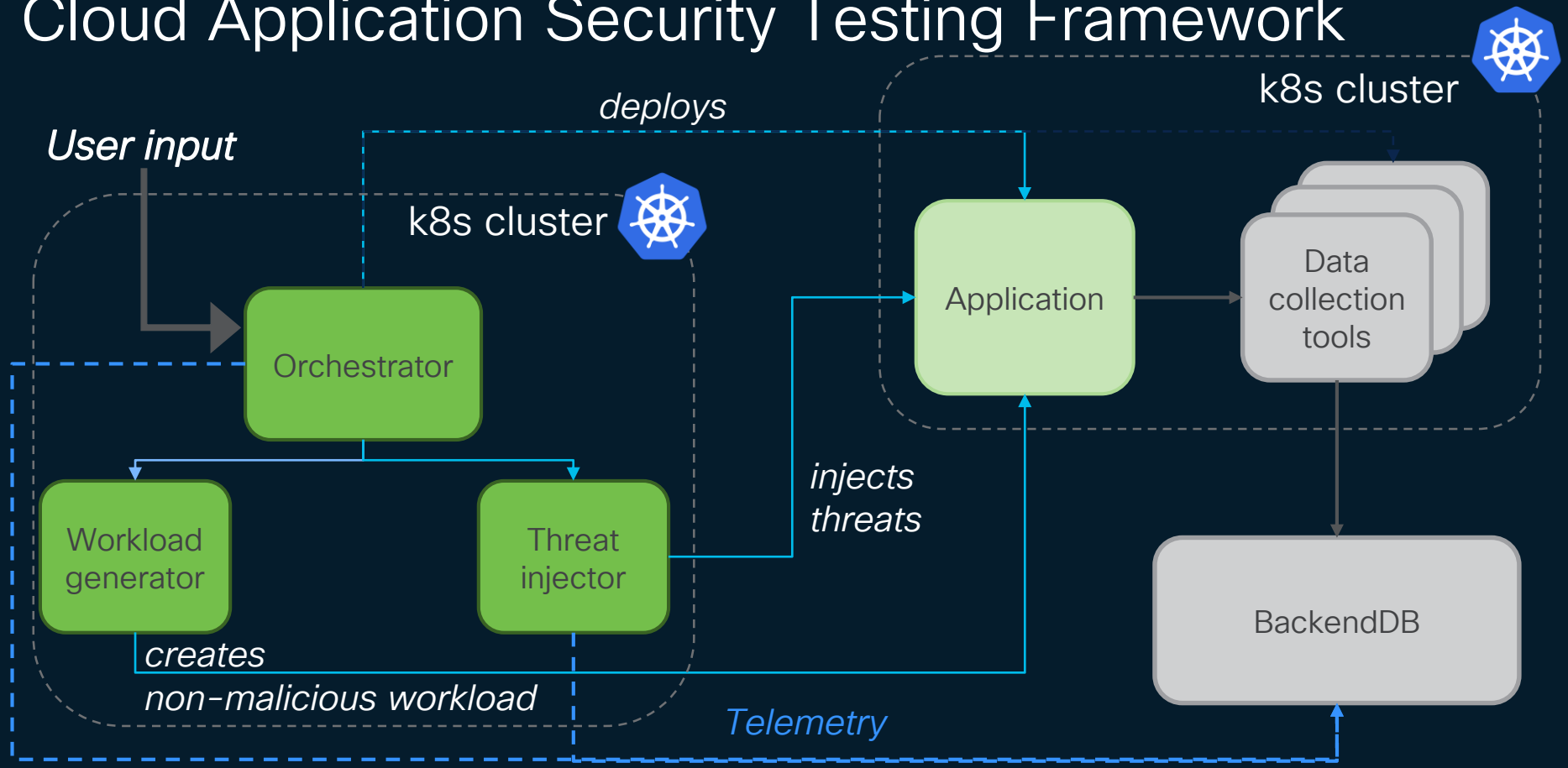
The richer the data the more efficient the detection will be

However, there are trade-offs to consider at inference time as the cost of extracting observable data is not zero

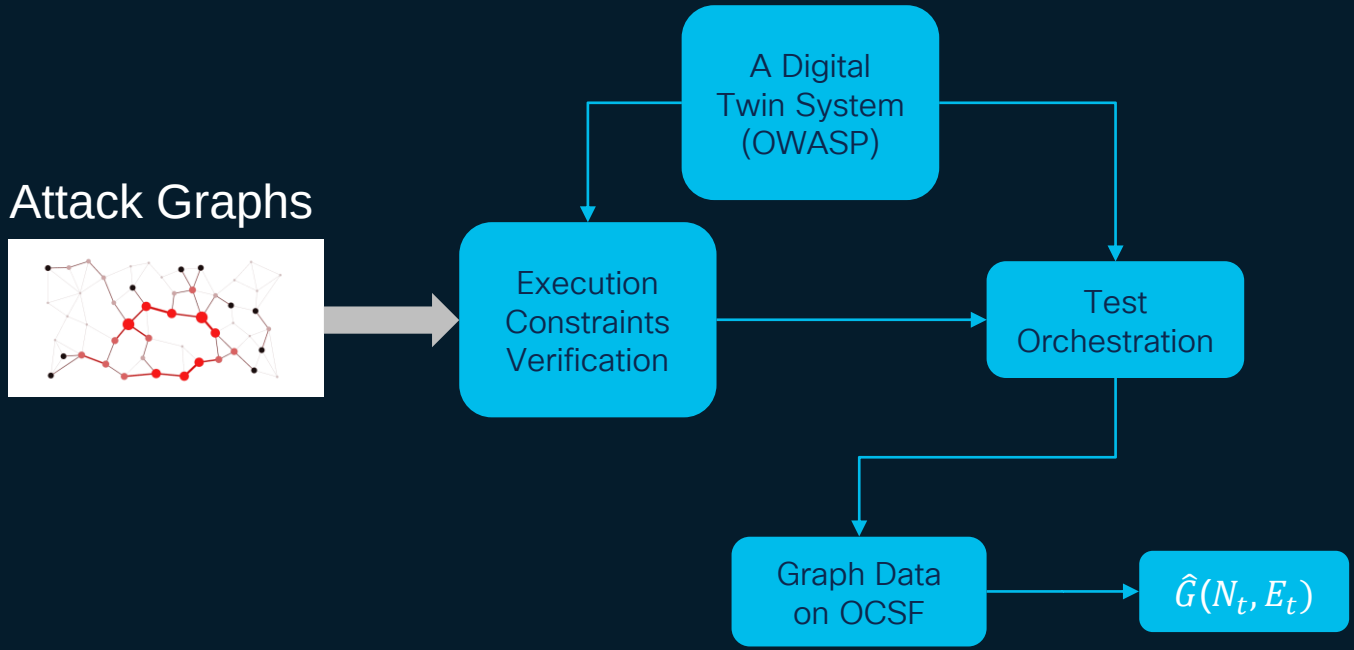
Optimal choices about the way $\hat{G}(N_t, E_t)$ is sampled requires extensive training and fine tuning

Adversary Emulation

Cloud Application Security Testing Framework



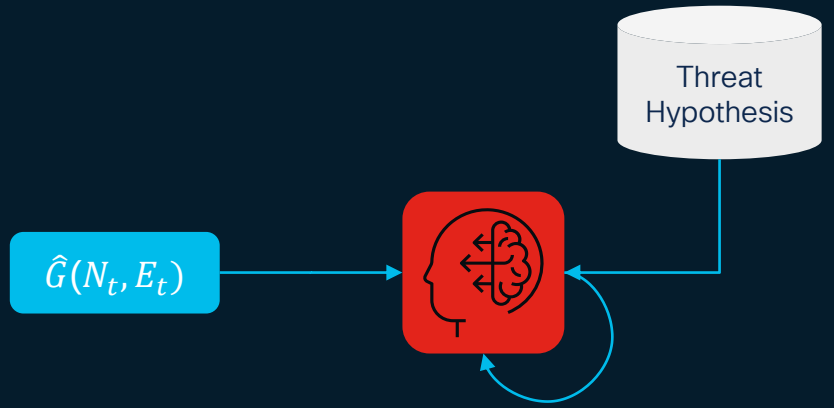
Executing Threat Hypotheses w/ Adversary Emulation



Detect Cyber Threats

Threat Hunting

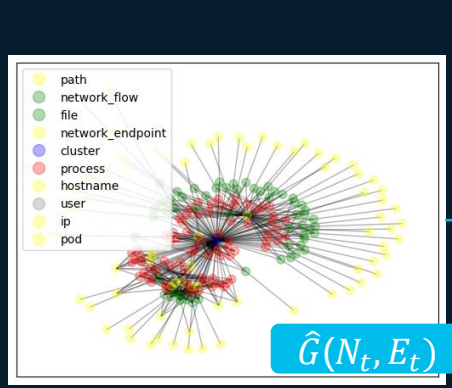
- Threat hunting is performed over the graph data $\hat{G}(N_t, E_t)$ in OCSF format that is sampled from the sensors in real-time
- We use ArangoDB as graph DB to perform threat hunting queries to verify threat hypothesis that the system has learned to perform off-line

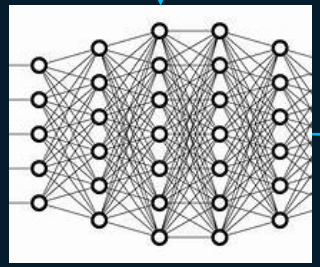


Verification of the Hypothesis via Operations over Graph data in realtime

Scalable Detection via Graph Neural Networks

- Automatic Threat hunting remains the formal framework we have developed to verify detections
- Verification of 100% of the sampled graphs in real-time can be costly or generate delay in the response
- We leverage graph embeddings as representations of the sampled graphs to scale up detection computations in real-time
- The core of the detection architecture is based on Graph Matching Network (GMN) embeddings
- Accurate embeddings can be used to perform prediction on detection, risk scoring and more.



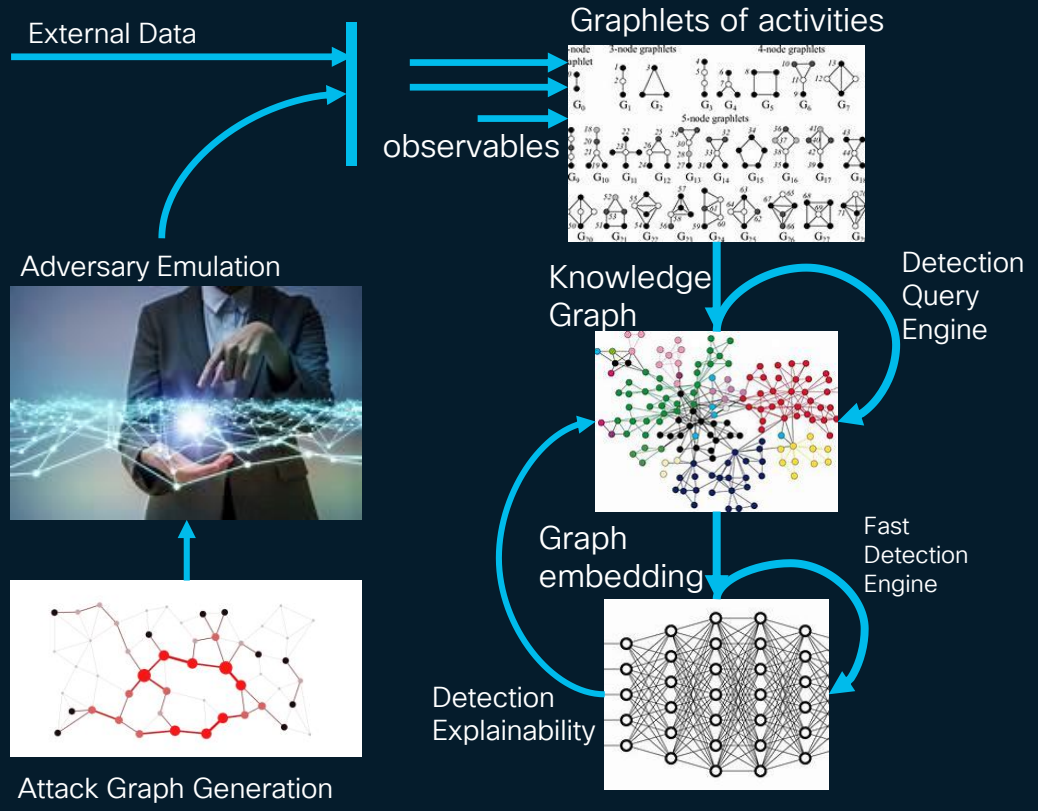


$$\phi \cdot \hat{G}_1(N_t, E_t) \sim \phi \cdot \hat{G}_2(N_t, E_t)$$

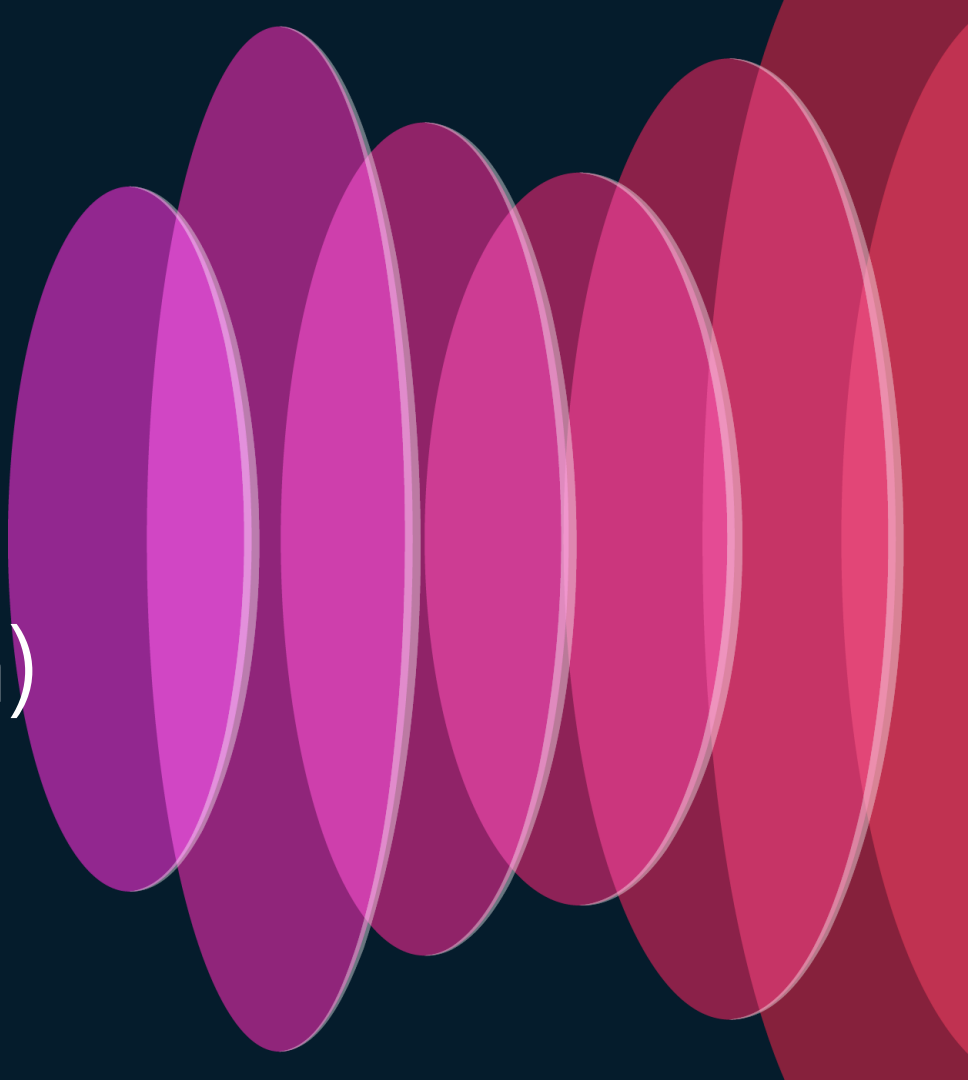
Architecture Summary

AI-driven purple teaming framework

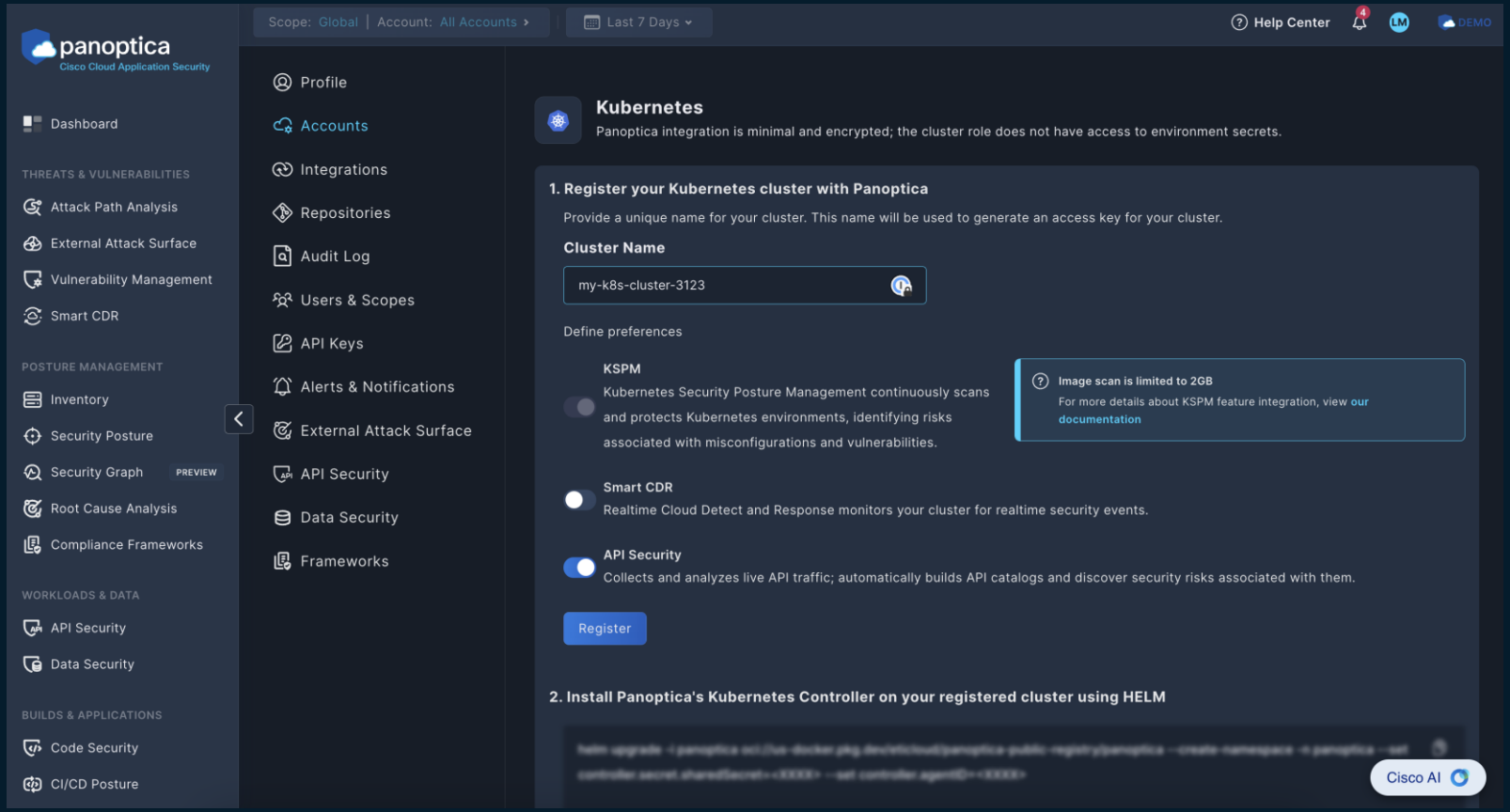
- **A purple team** simulates malicious attacks and penetration testing to identify vulnerabilities and recommend remediation (combination of red and blue teams)
- **In realtime** We want to empower purple teams with tools that can accelerate testing and extend security test coverage and do it in real-time.
- **Based on graphs:** We have created a framework where complex activities are modeled as graph data as the most suitable representation.
- **AI/ML:** reasoning based on threat intelligence computing scaled up with graph embedding for pattern matching and root cause analysis triggering explainability and real-time forensics.



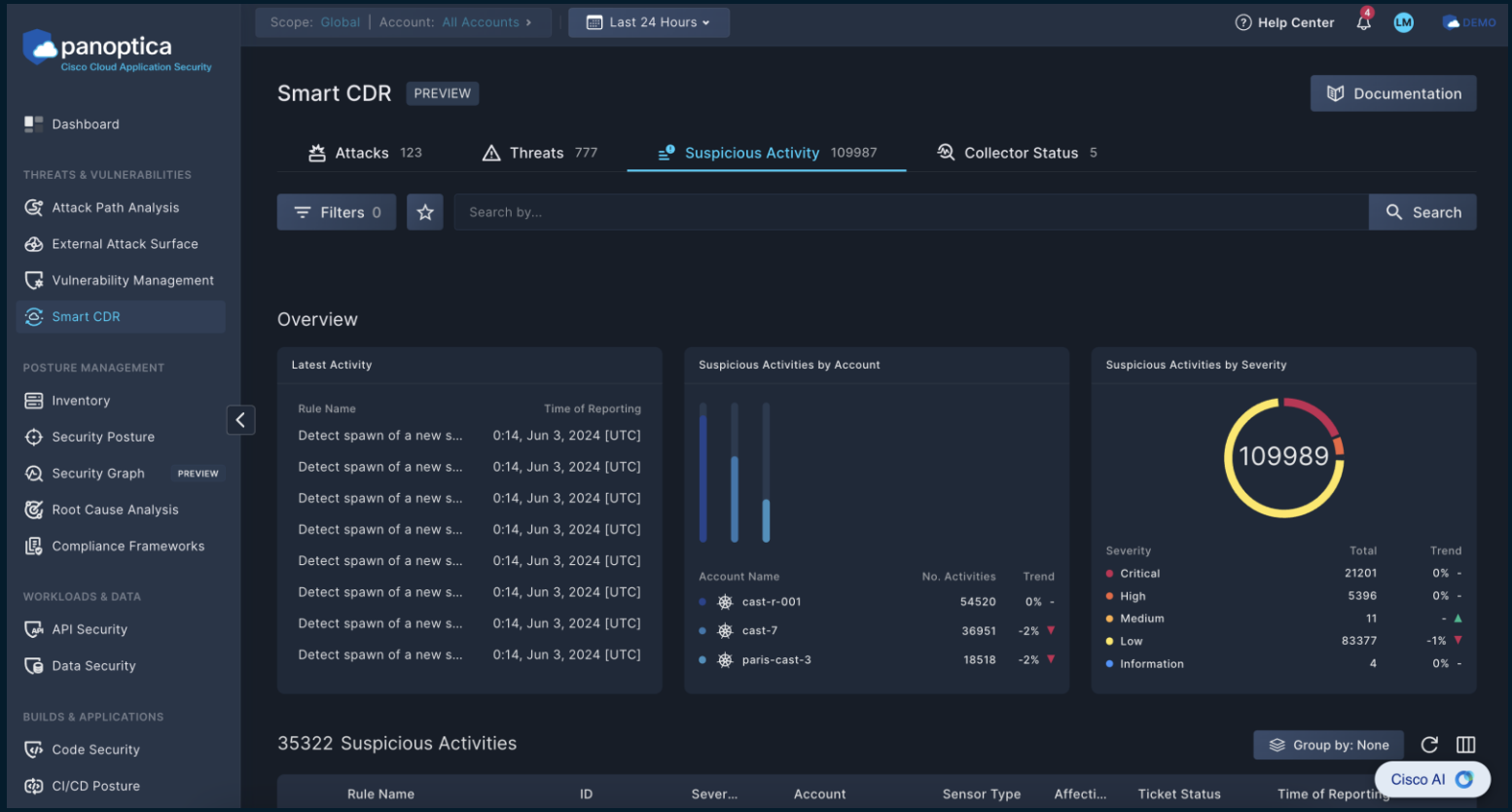
Cisco Cloud Application Security (Panoptica) CDR



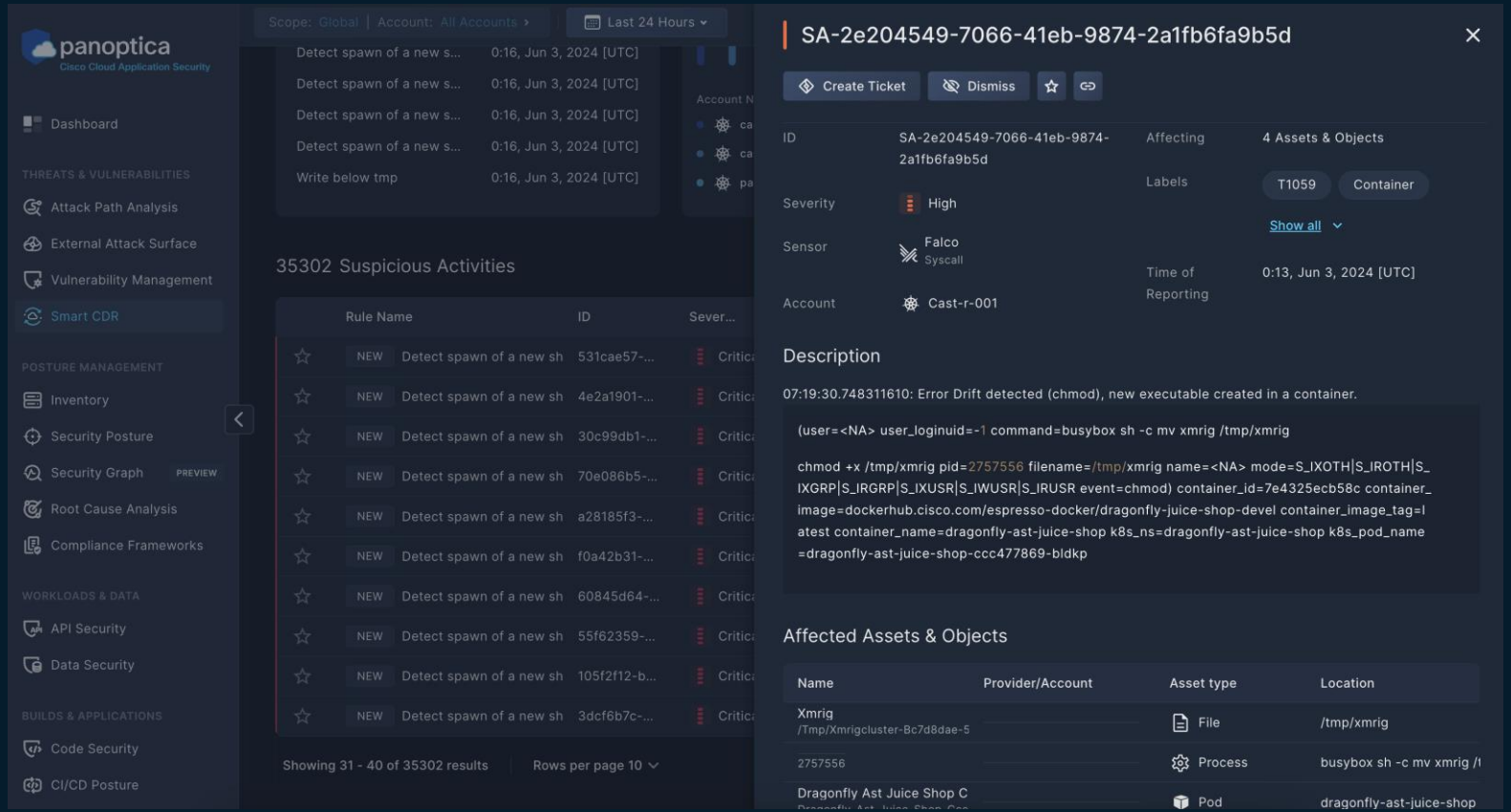
On boarding CDR



Suspicious Activity Overview

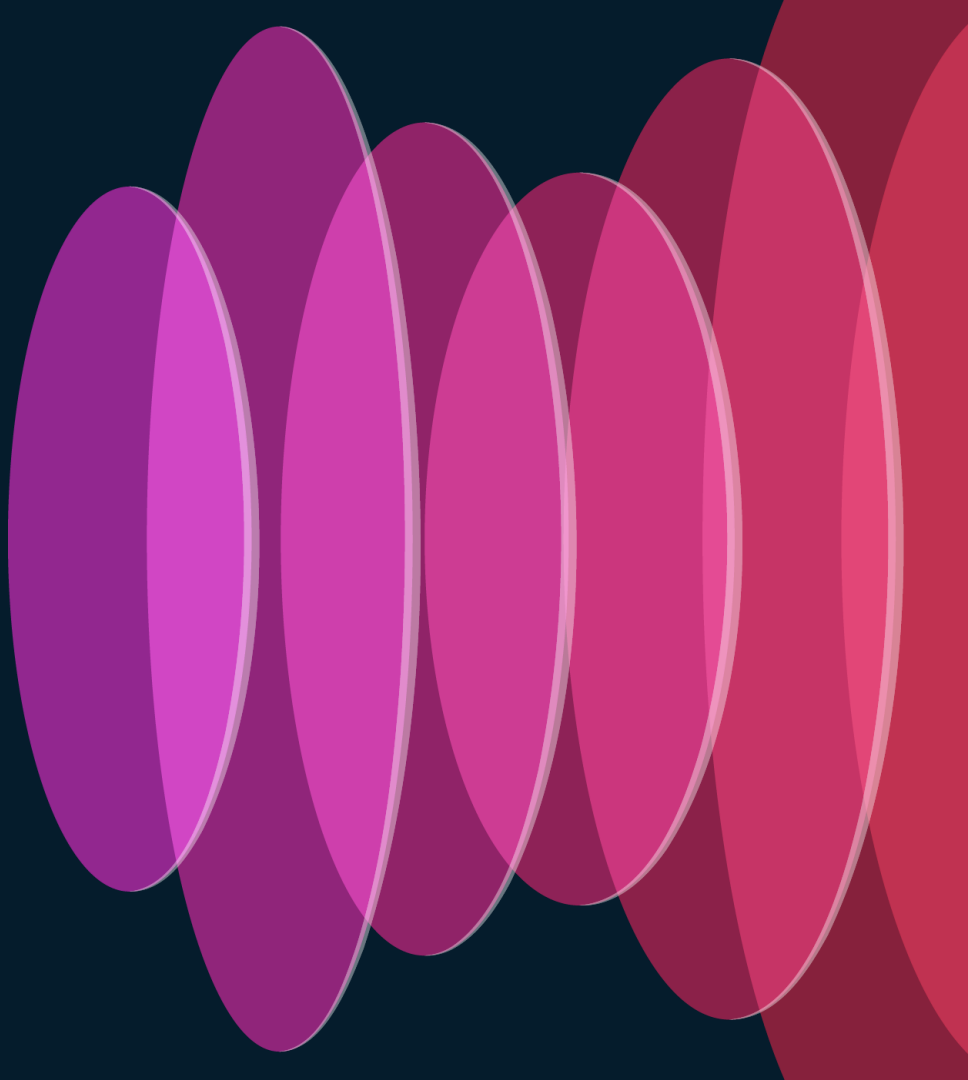


Suspicious Activity Details



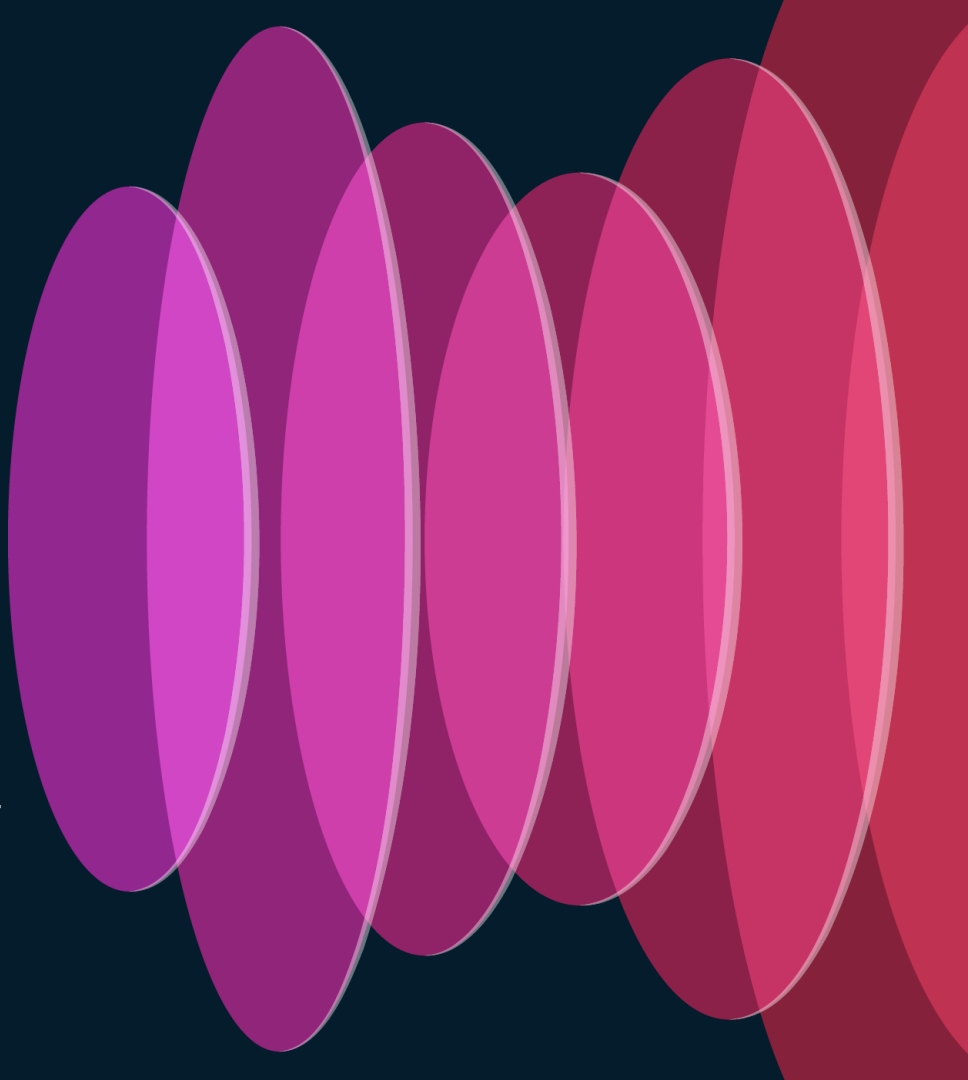
What is suspicious behavior?

Suspicious behaviors are events that have been identified based on rules defined within CDR (see next slide).



Examples: Rules that are used to identify suspicious activity

- Attacker may attempt to obtain account login and credential material
- Detect dumping of process memory
- Scan ports to check for listening ports
- Detect read & writes to file system
- Detect delete system and audit logs
- Connection attempts to Kubernetes cluster local domain (internal services in the cluster)



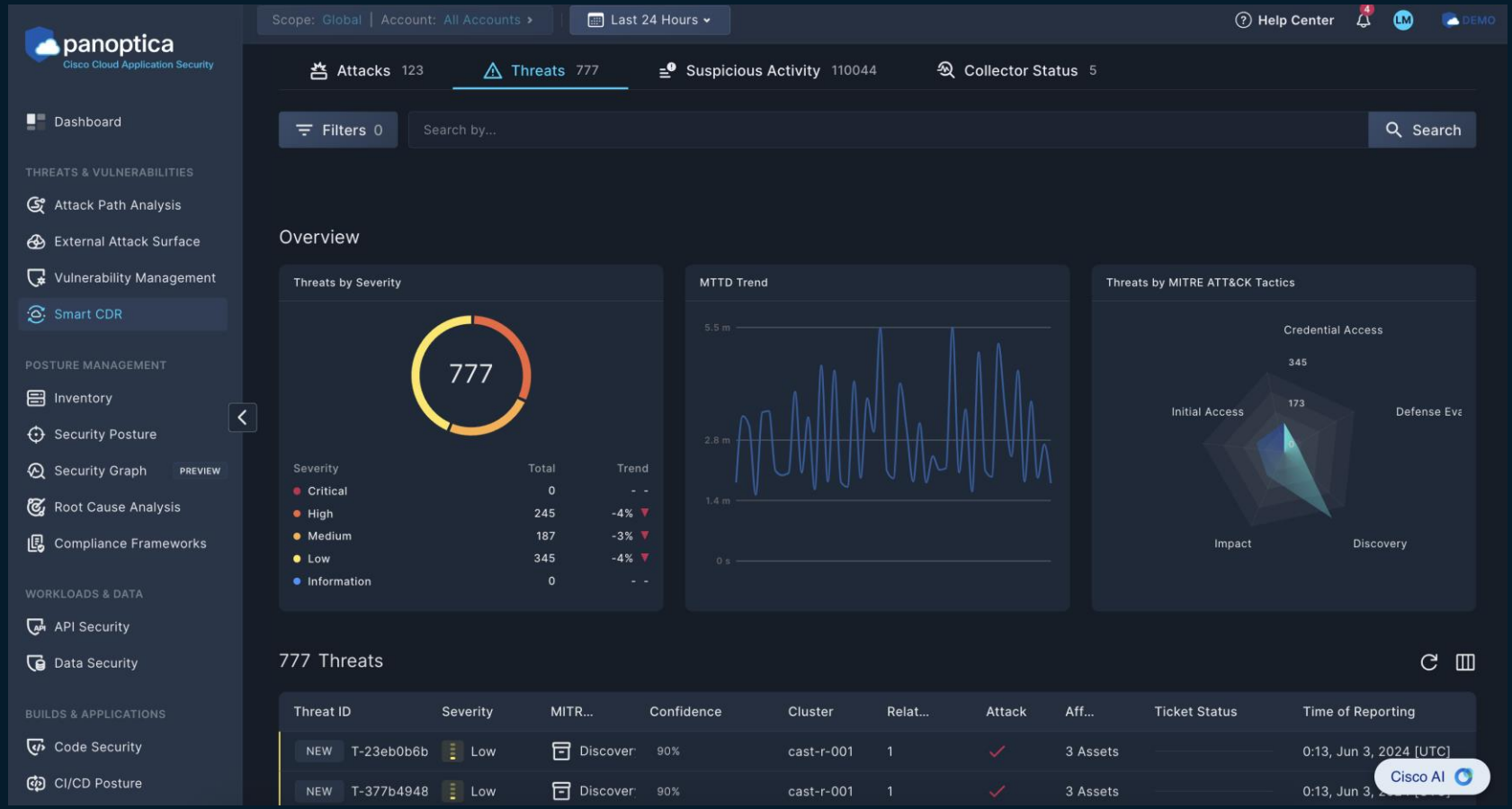
Notifications (WebEx) Example

Dragonfly Notification

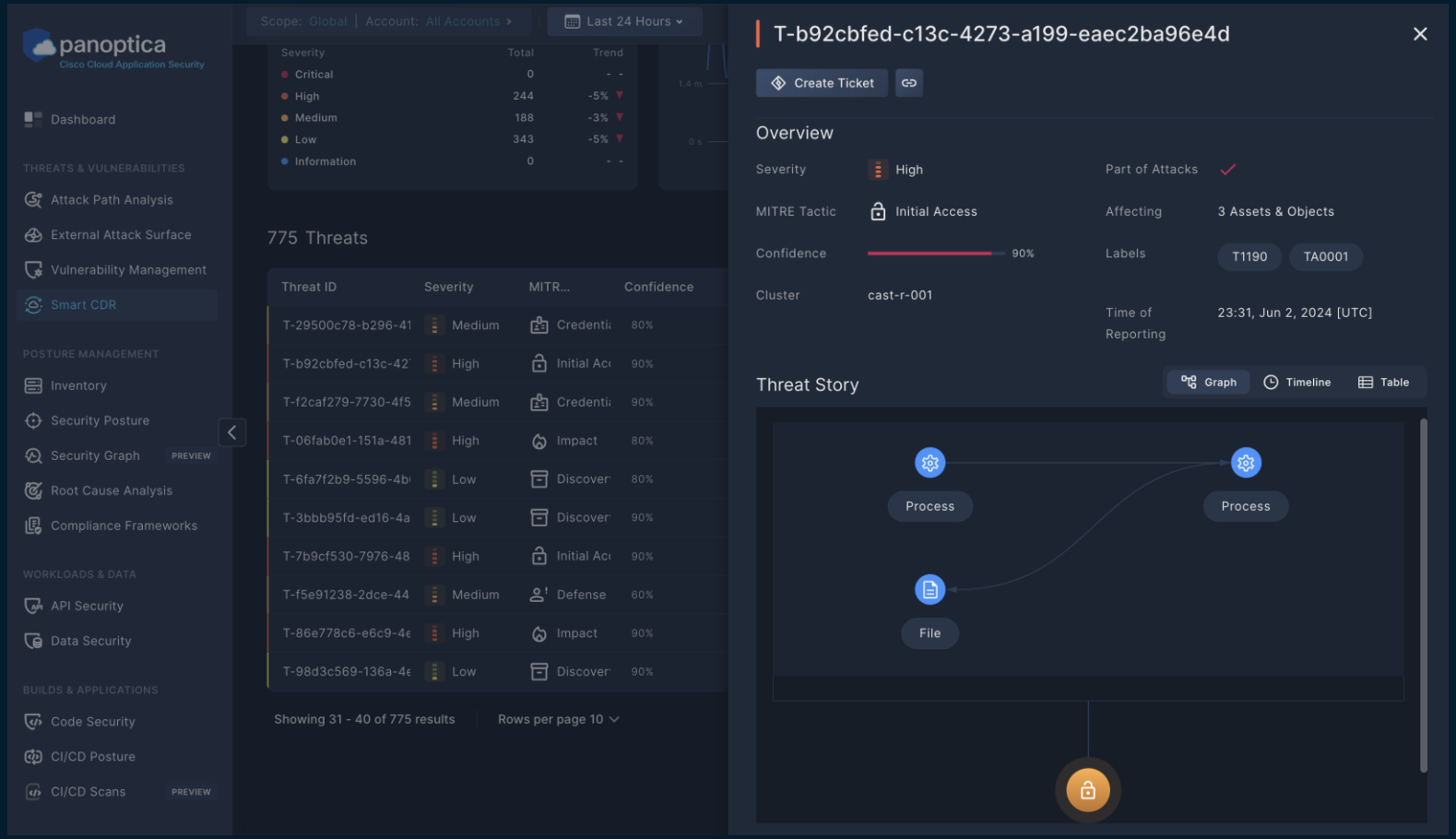
RuleID: 432a61cd-056c-4d02-822a-0e87b33e3833
Cluster uuid: cluster-test-demo
Severity: CRITICAL
Namespace: host
Pod name: undefined
Container name: host
Description: 09:13:17.764680555: Alert Attempts to obtain the mapping between cluster local domain and Cluster Ip (command=nslookup kubernetes.default.svc.cluster.local pid=603424 container_id=host name_space= pod_name=) k8s.ns= k8s.pod= container=host

Investigate in Panoptica

Threats Overview

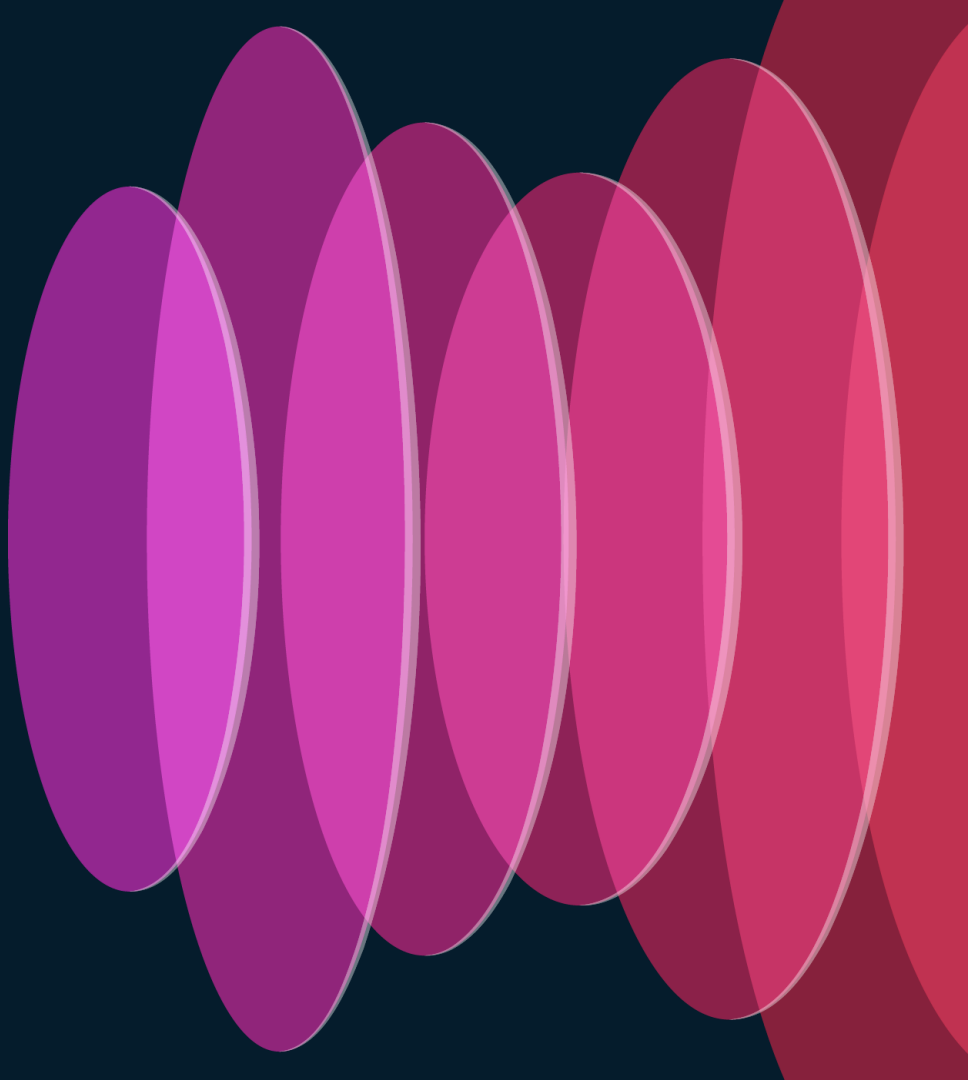


Threat Details

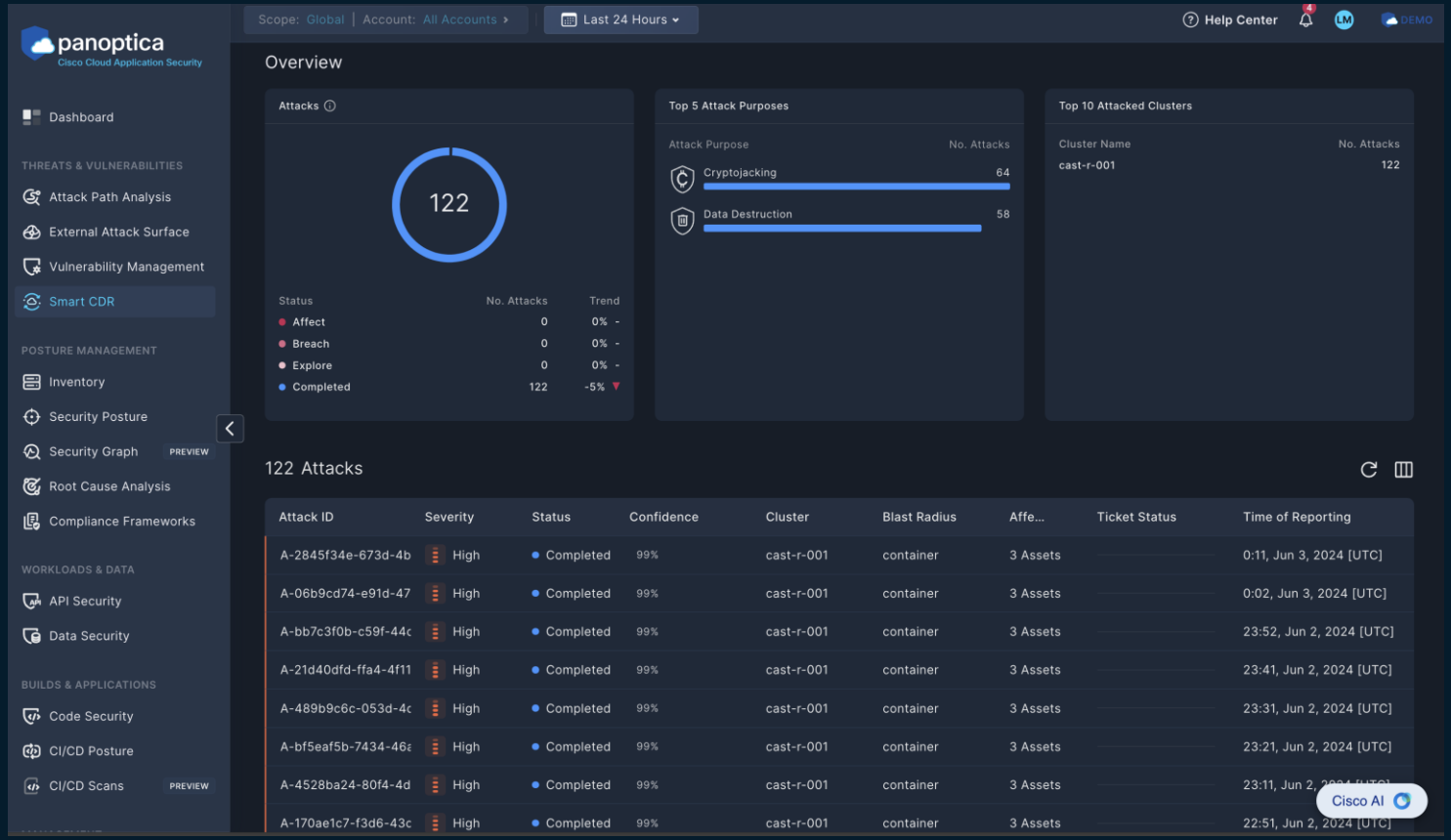


What is a threat?

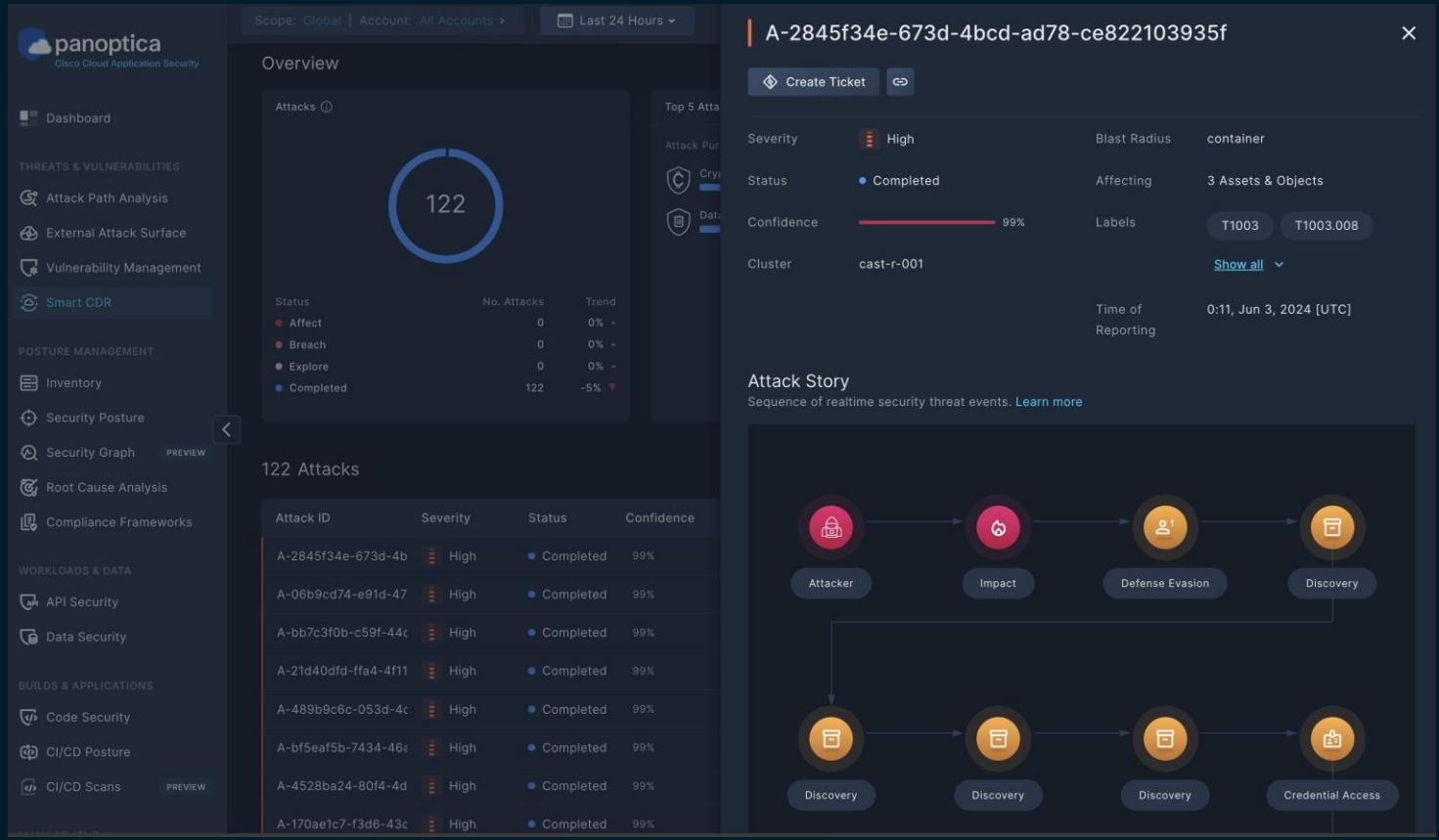
A threat is a series of suspicious activities that are related to each other and are intended to cause harm to the system. Each threat is then mapped to a MITRE TTP (Technique, Tactic, or Procedure).



Attack Overview

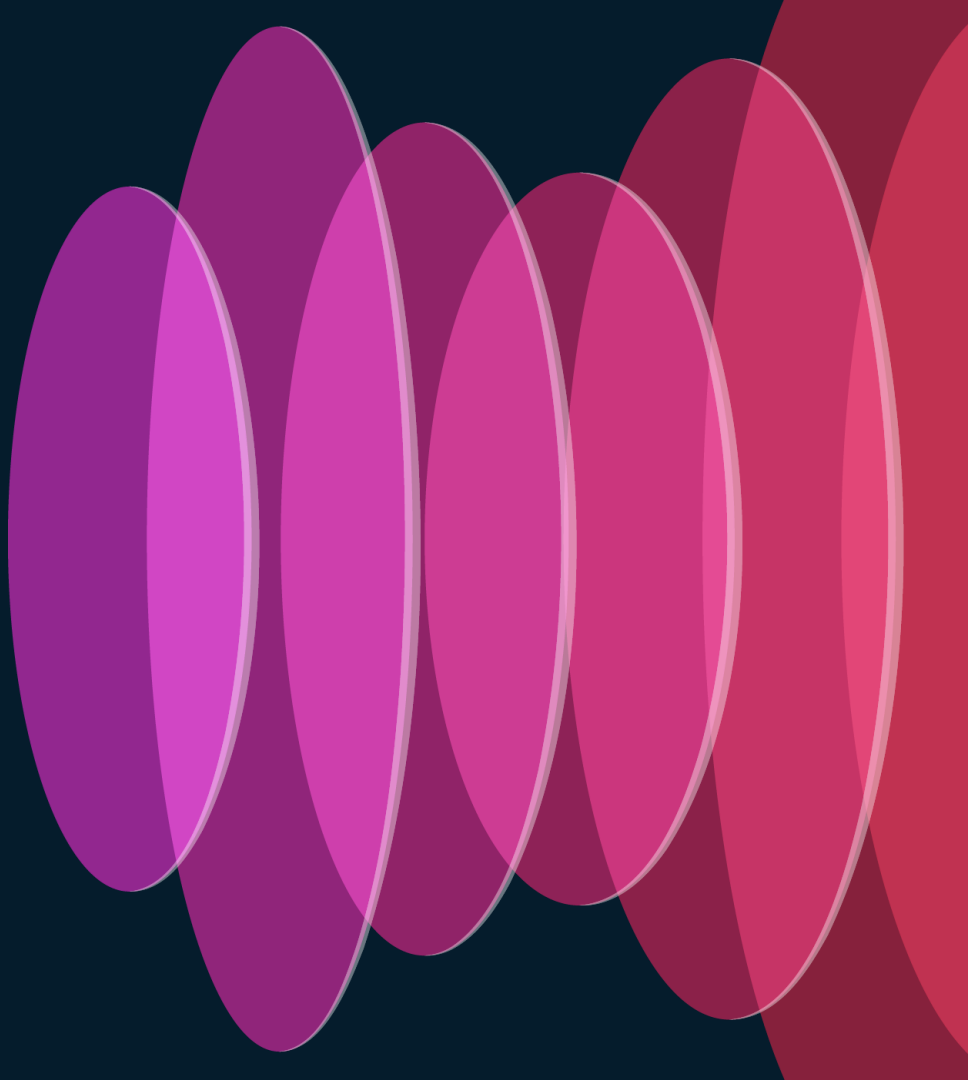


Attack Details



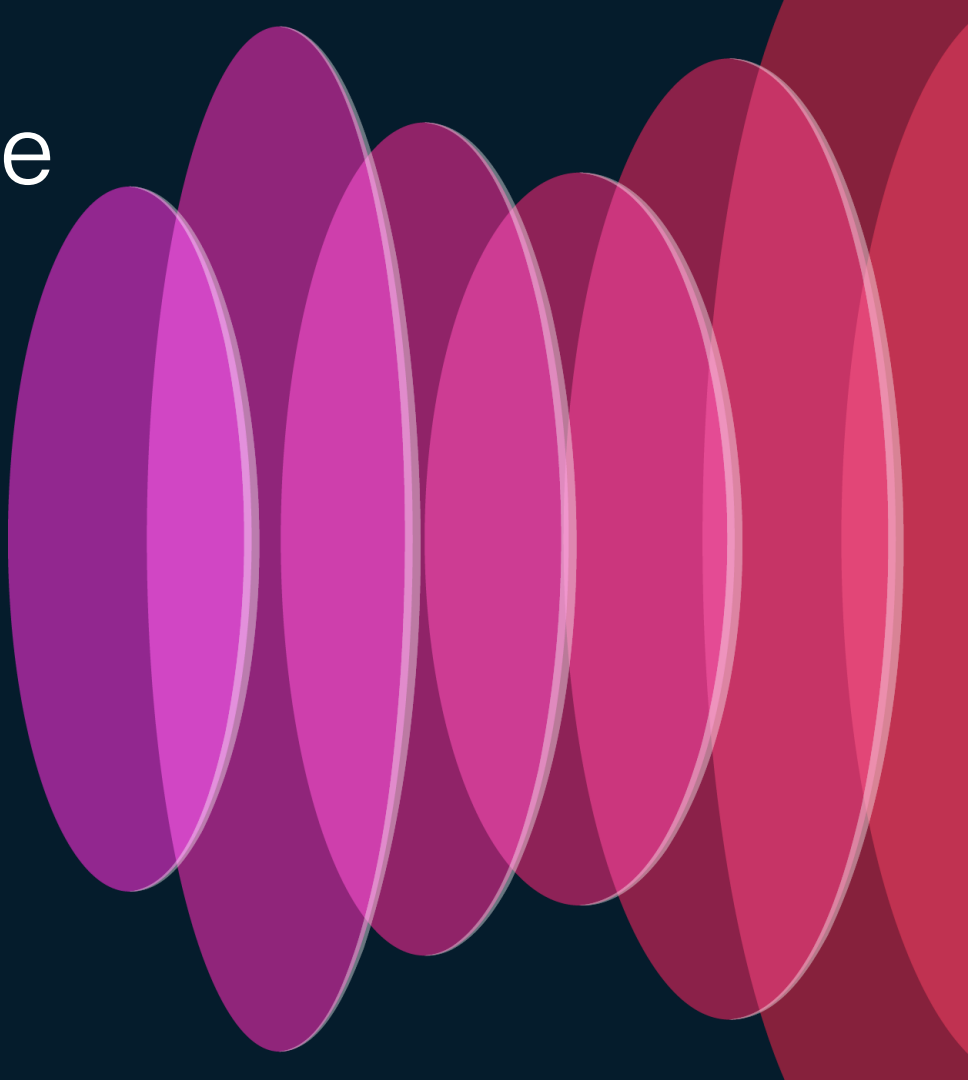
What is an attack?

An attack is an attempt to collect, disrupt, deny, degrade, or destroy the system. CDR focuses on identifying 5 categories of attacks (see next slide) based on a series of threats.



CDR Attack Coverage

- Ransomware
- Data Exfiltration
- Container Escaping
- Crypto-mining
- Data Destruction
- Unknown



Complete Your Session Evaluations

Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to **win 1 of 5 full conference passes** to Cisco Live 2025.

Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.

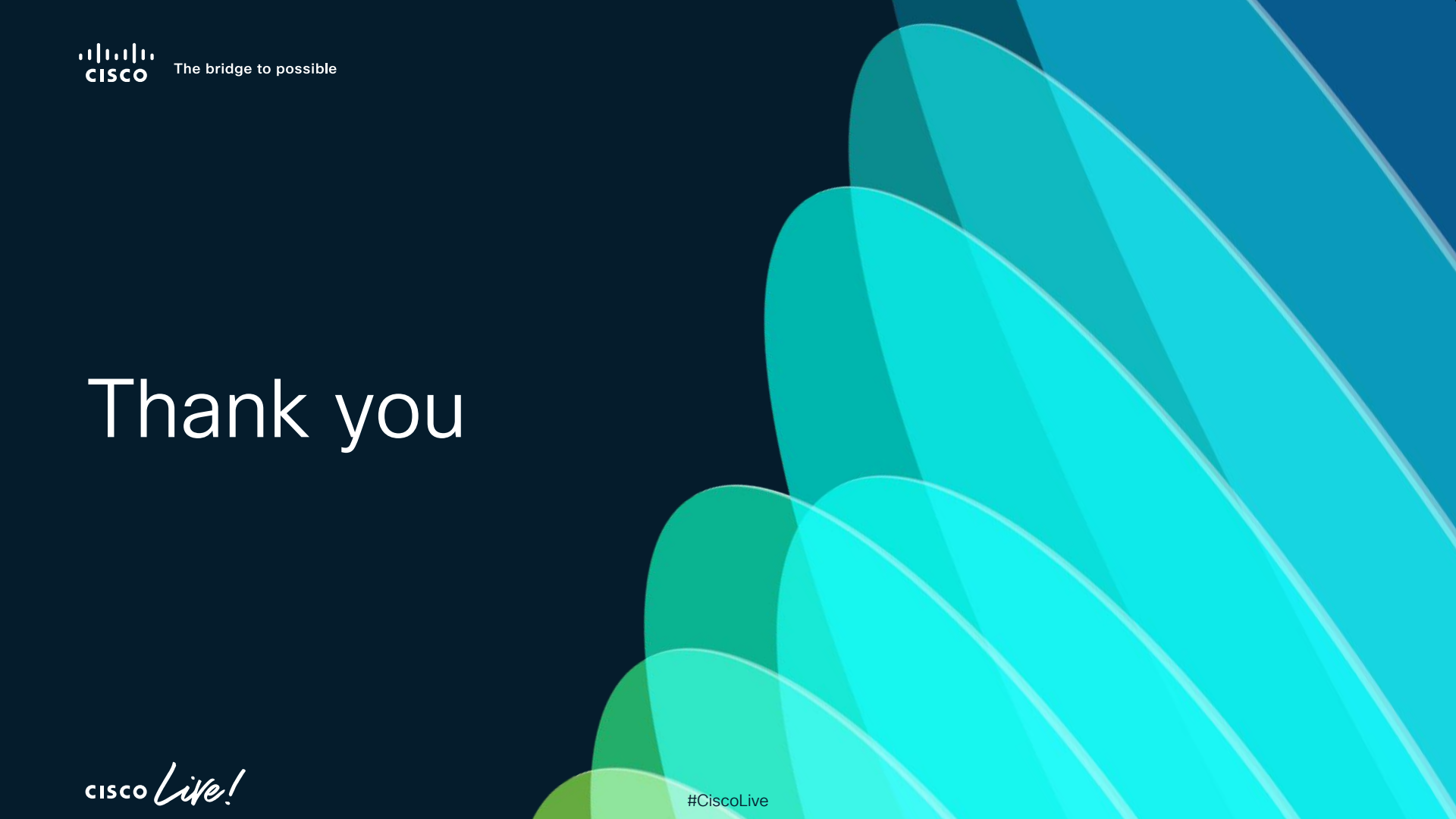
Level up and earn **exclusive prizes!**

Complete your surveys in the **Cisco Live mobile app.**

Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Contact me at: Insert preferred comms method


The bridge to possible
Thank you
cisco Live!
#CiscoLive