Hypershield

Mastering Next-Generation Security

Jeroen Wittock - TME

BRKSEC-2265
Tuesday February 11 4:30PM

Webex App

Questions?

Use the Webex app to chat with the speaker after the session

How

1. Find this session in the Cisco Events 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, at least, February 28, 2025.

Securing the enterprise is increasingly challenging

Expanding attack surface

- Explosive workload growth
- Inconsistent enforcement
- Environments keep changing

Patching is hard

- High vulnerability rate
- Mitigation is too slow
- Ensure app is available

Change is risky, expensive

- Firmware updates delayed
- Policy changes are behind
- Delayed security posture

Agenda

- Introduction
- Architecture
  - eBPF
  - AI & Graph Engine
  - Digital Twin
  - API
- Use Case: Autonomous Segmentation
- Use Case: Distributed Exploit Protection

Hypershield Architecture

Cloud Protection Suite

Hybrid Mesh Firewall

Cloud Management (Security Cloud Control)

Major trust boundaries		Everywhere		
L7 Threat Protection	AI Model Protection*	Segmentation	Distributed Exploit Protection	
Firewall Threat Defense	Multicloud Defense	Secure Access (FWaaS)*	Hypershield (Smart Switch)	Hypershield (Agent)	Secure Workload

* AI Defense and Secure Access are add-ons to Cloud Protection Suite

Cisco Hypershield

Security Cloud Control

Autonomous Segmentation

Distributed Exploit Protection

Zone L4 Firewalling

Future services

Platform

Fine-grained security controls

Self-qualifying updates

Workload Enforcement

Linux

Windows soon

Network Enforcement

VM

Smart Switch

Network Visibility

Select existing switches Coming Soon

Server Enforcement

Coming Soon

Public Cloud | Private Cloud

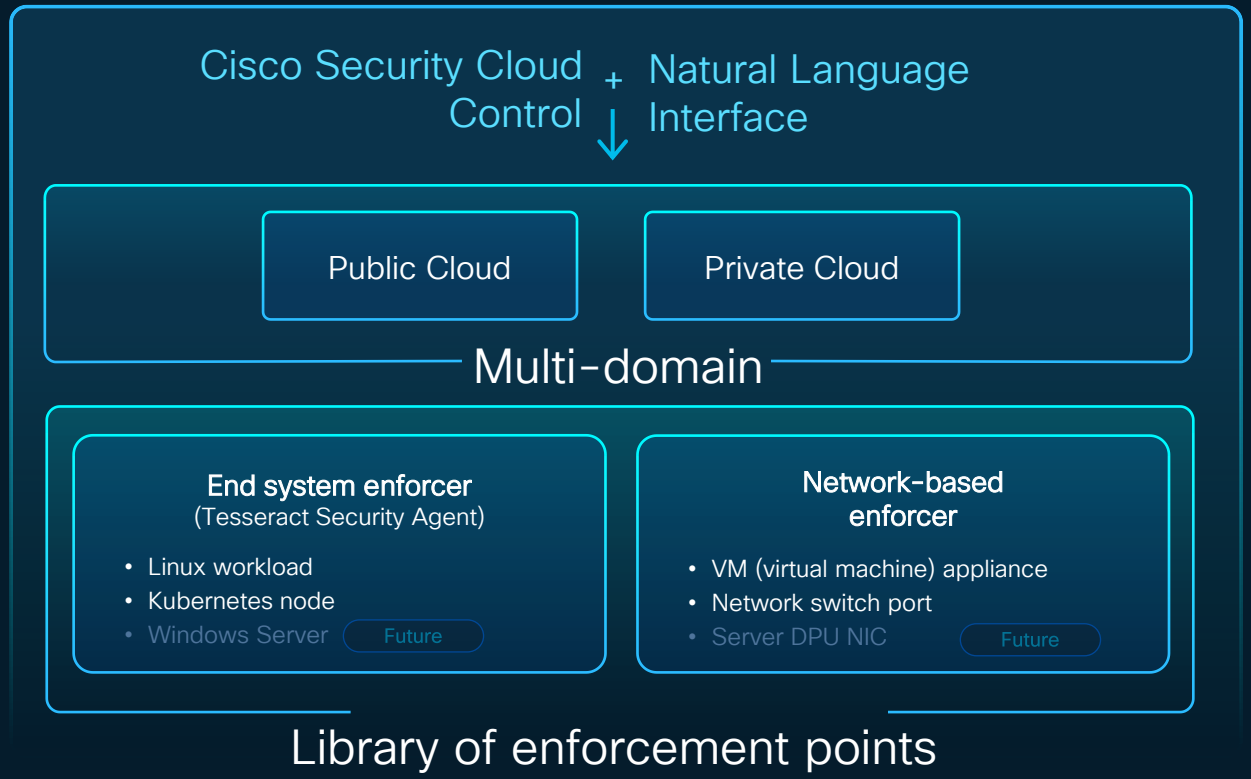
Manage globally, enforce locally

Includes

- Unified management
- Single global policy
- Intelligent placement of shields
- Integrations with cloud/app/infra metadata

Environments

- Kubernetes
- Cloud – Private/Public
- On-prem



Tesseract Security Agent

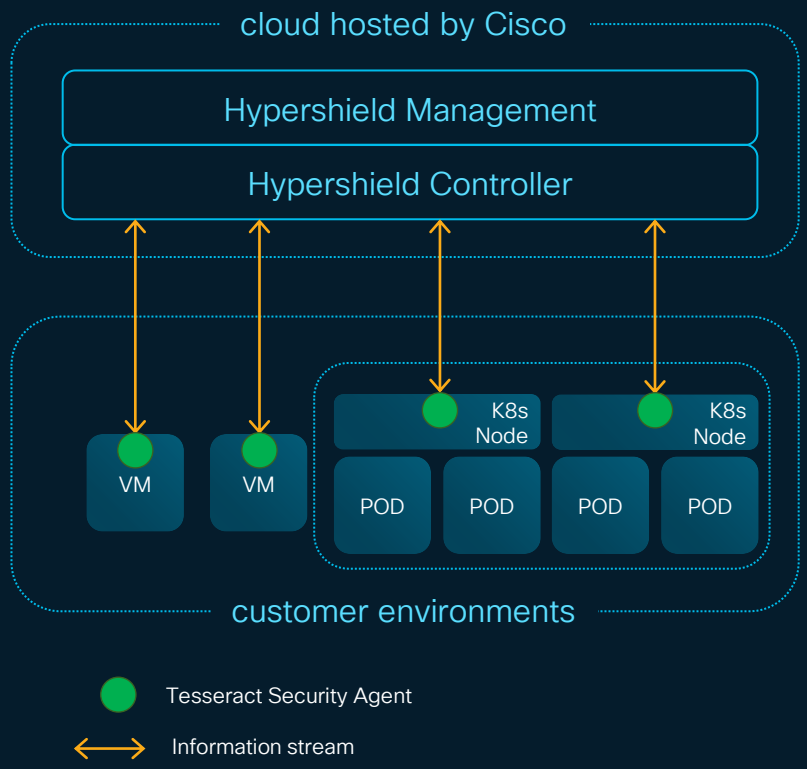
Deployed on customer workloads

On Kubernetes and Linux machines

Managed via Cisco cloud

Distributed analytics, visibility, and control

Covers network, file, execution, and privilege escalations



Cisco N9300 Smart switch

A platform to enable stateful services

Network

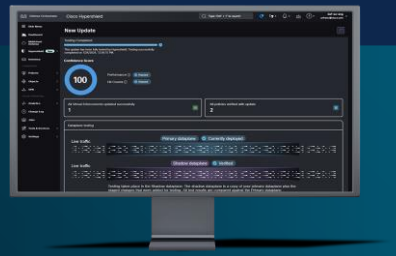
N9300 Series Smart Switches

Converge stateful services and network

- 800G stateful services throughput and scale
- 24-port 100G
- 4.8T Silicon One + 4 AMD DPU
- 1 RU

Security

Cisco Hypershield

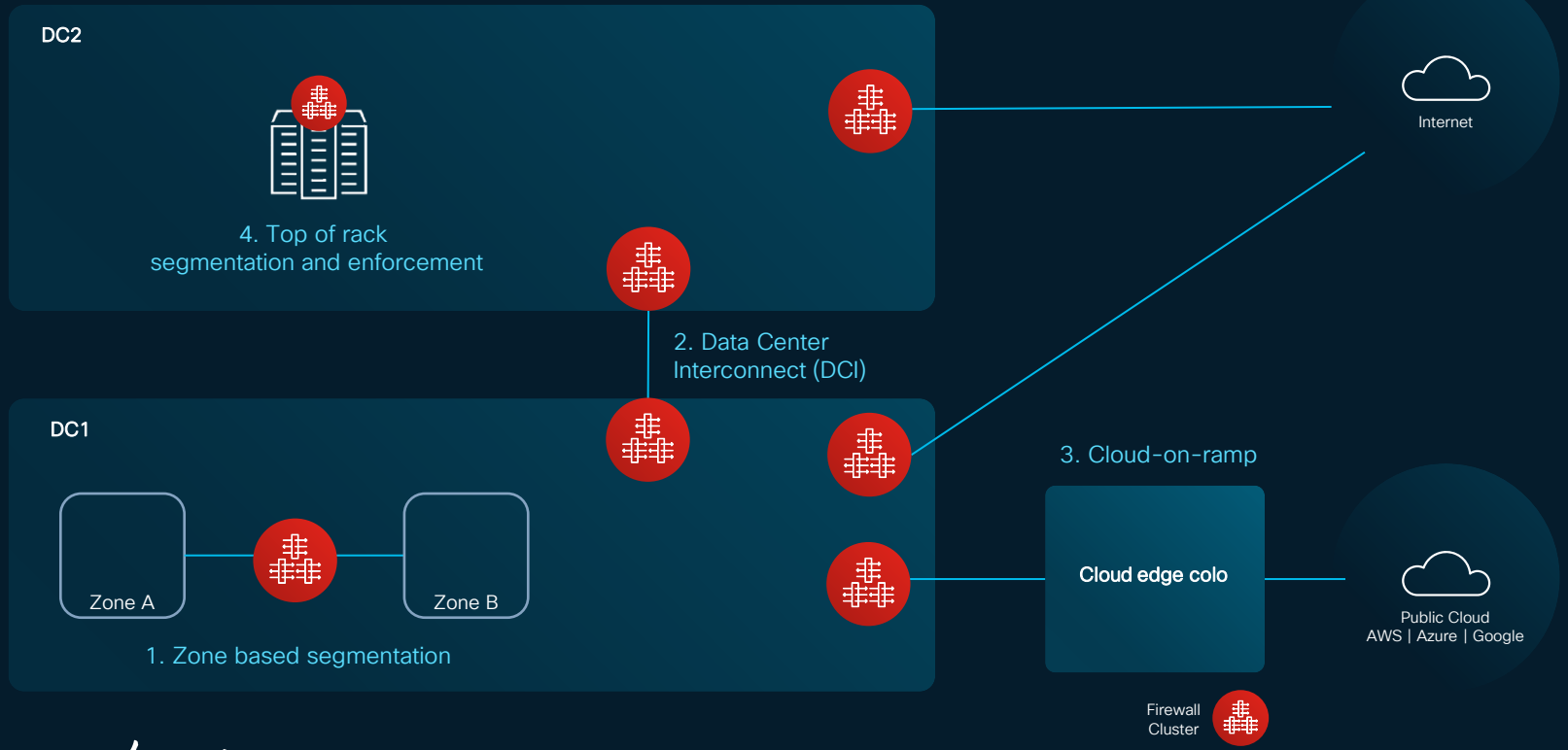


Integrated security (license add-on)

- Intelligent security policy placement
- Self-qualifying policy updates
- Policy unified with workload/network enforcement, public and private clouds

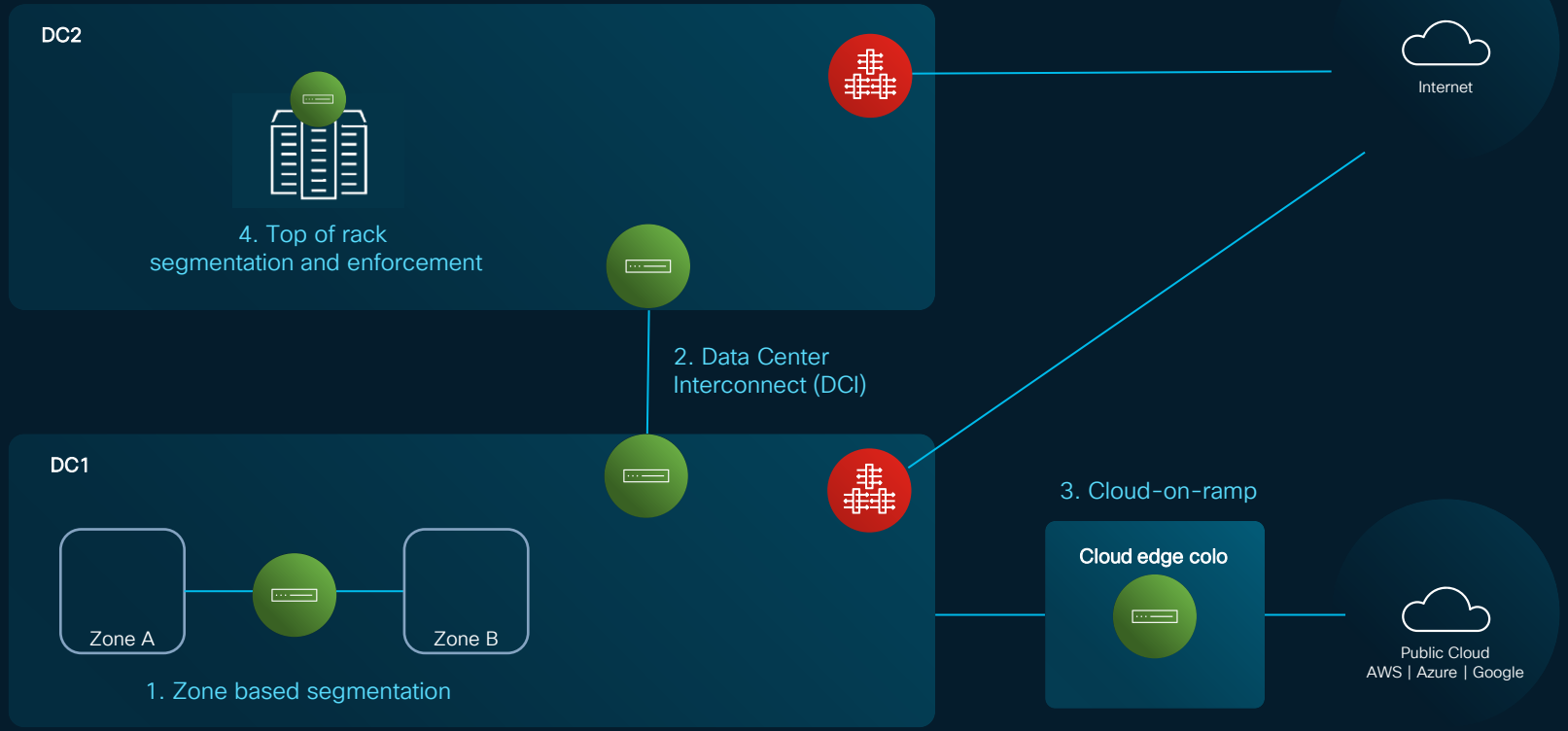
Security infused into the data center fabric

Current implementation is complex and expensive



Security infused into the data center fabric

Use cases with Cisco Smart Switches



Firewall Cluster

Cisco Smart Switch

Hypershield Policy Model

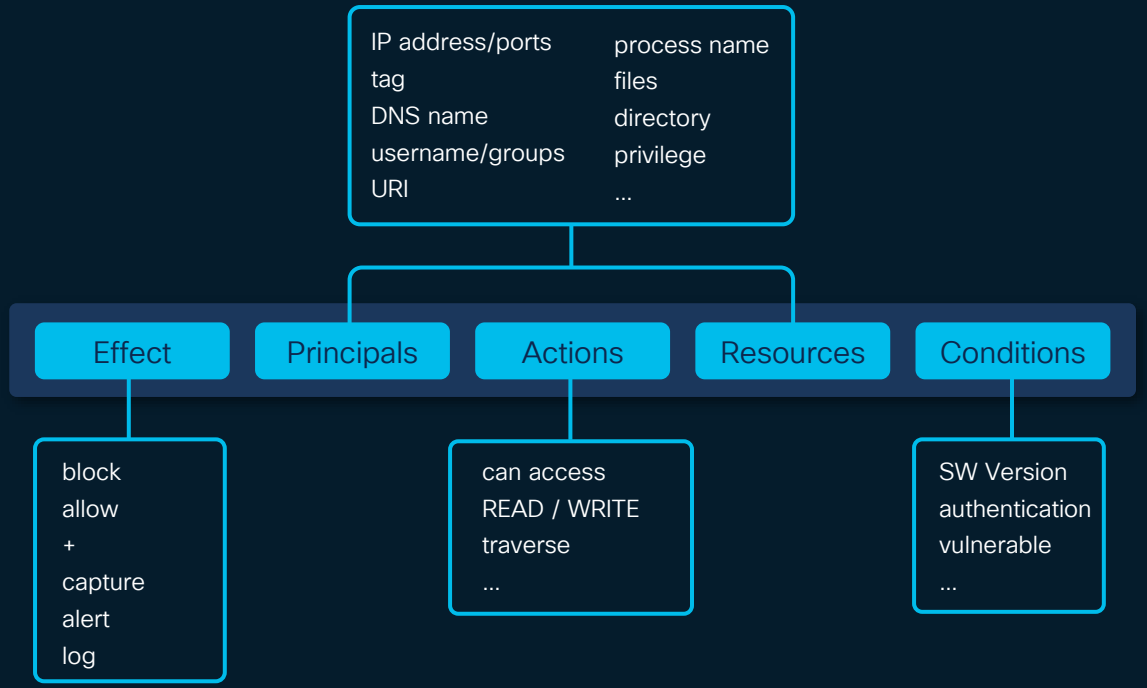
- Declarative Policy Language (PARC) is a Human readable model.
- Automated Reasoning, especially useful in large scale systems, where doing things manually is near impossible.
- AI & ML integration
- Able to define microsegmentation
- Able to define kernel level detail
- Scalable

Why PARC

Policy as Resource Controls
Principal Action Resource Condition

- Inspired by AWS Cedar, but:
- Hypershield adds Exceptions to original Cedar Model
- Decision Process:
  - Default Deny
  - Deny Wins, except for conditionals
  - Unordered Policies
  - Apply Effect
- Policies can Include metadata, but no impact to rules themselves

Single policy covering network and workload



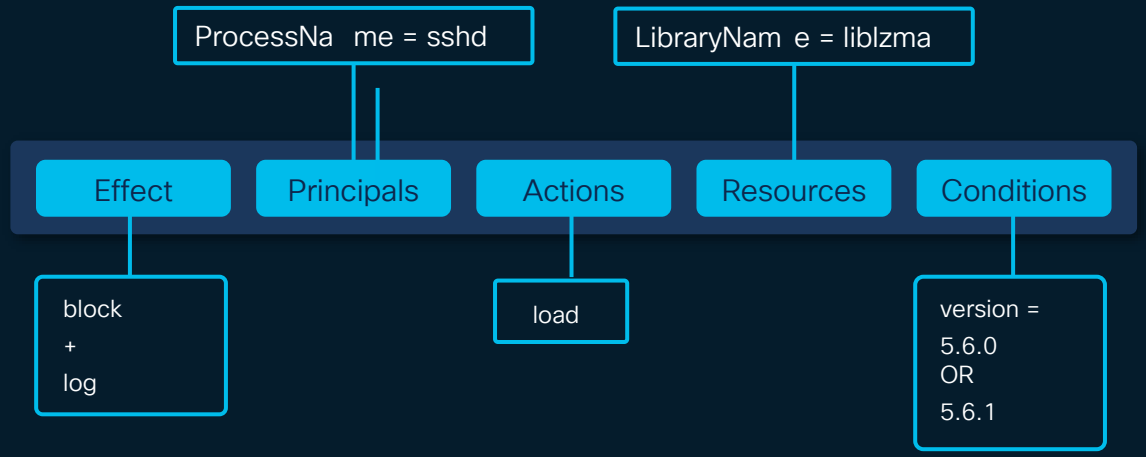
Hypershield Policy Construct

- Intent based
- Order independent
- Rule Auto-compilation
- Intelligent Rule Placement
- Single Global Policy
- Unified management

XZ vulnerability example with eBPF

Natural Language:
Do not allow ssh process to load vulnerable XZ library

Hypershield Global Policy:



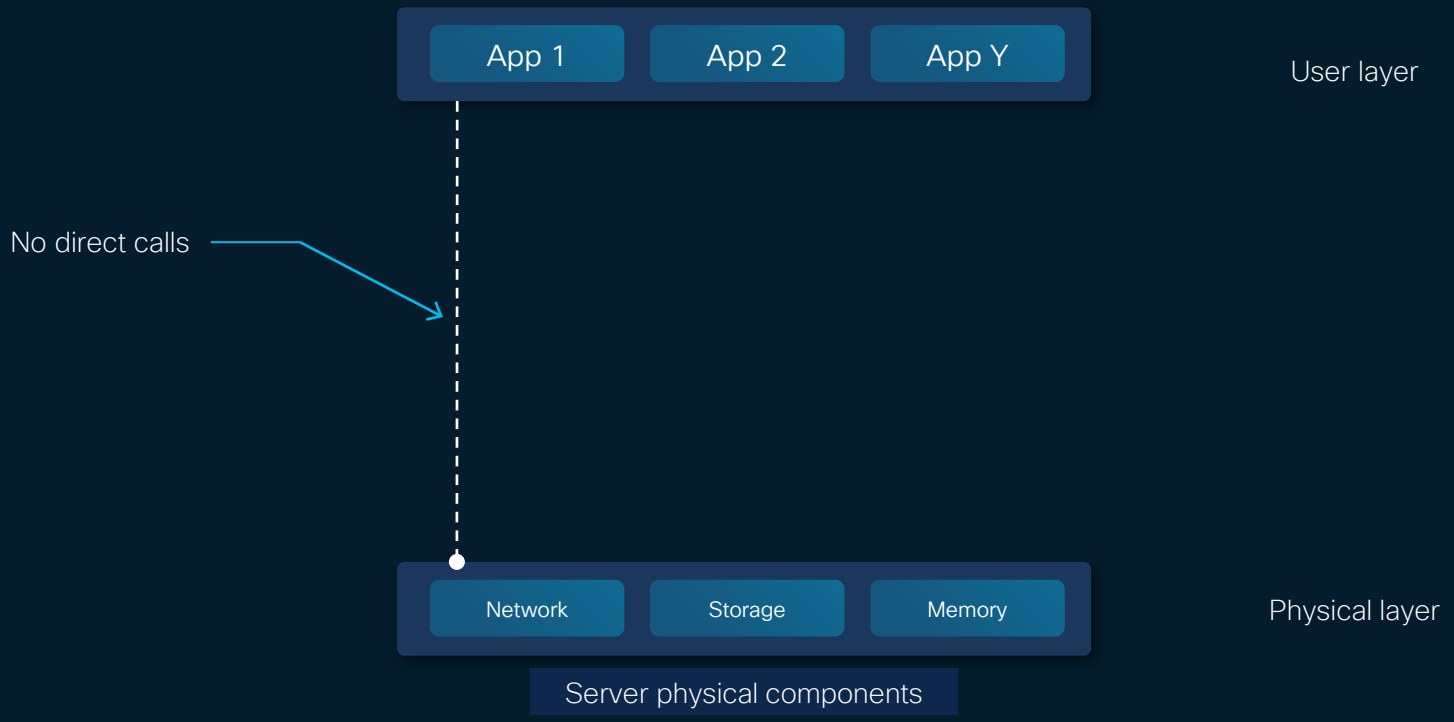
TSA (eBPF policy):

```
selectors:
- matchBinaries:
  - operator: "In"
    values:
    - "/usr/sbin/sshd"
  matchArgs:
  - index: 0
    operator: "Postfix"
    values:
    - "liblzma.so.5.6.0"
    - "liblzma.so.5.6.1"
```

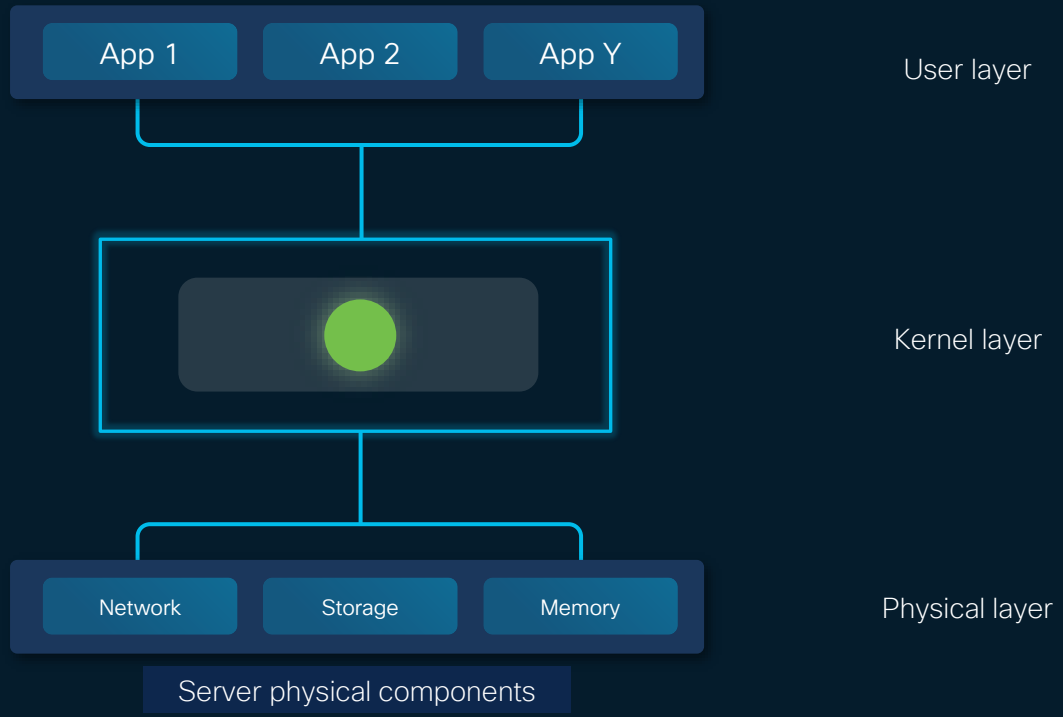
Where are we:

- Introduction
- Architecture
  - eBPF
  - AI & Graph Engine
  - Digital Twin
  - API
- Use Case: Autonomous Segmentation
- Use Case: Distributed Exploit Protection

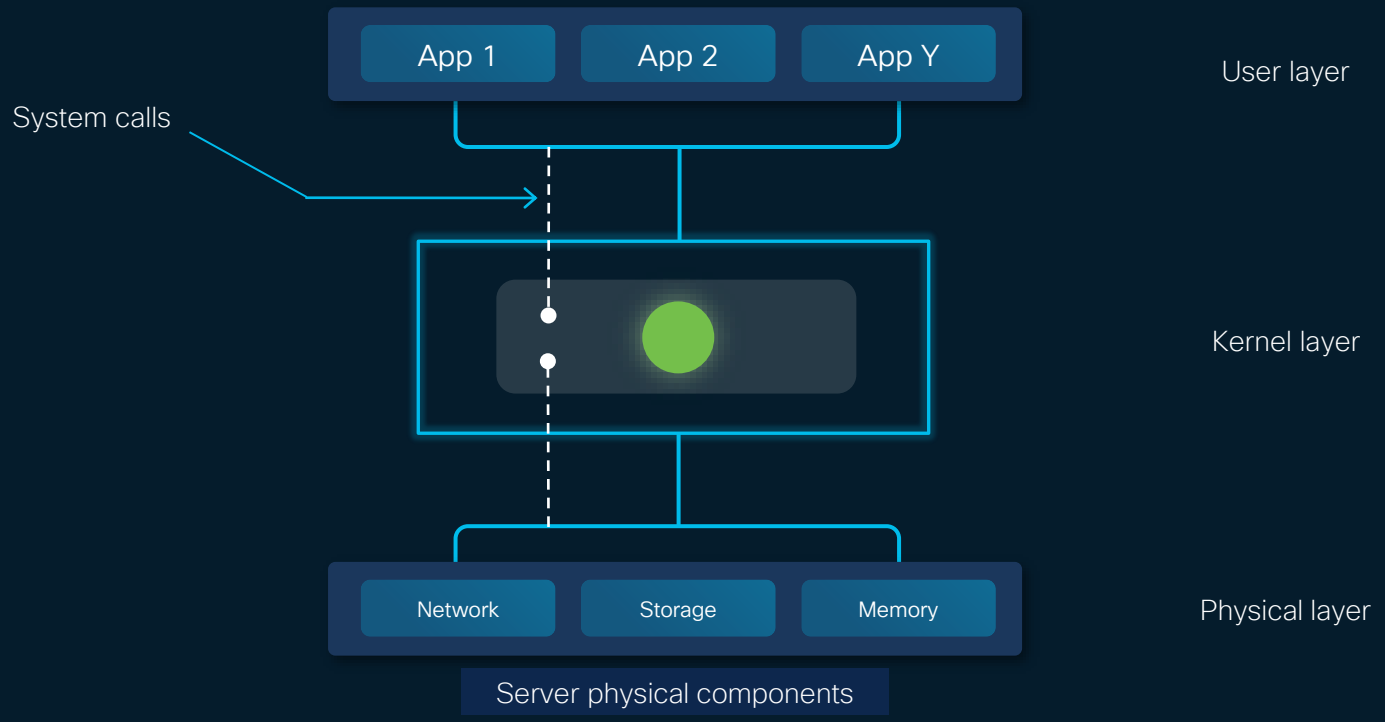
How do operating systems work?



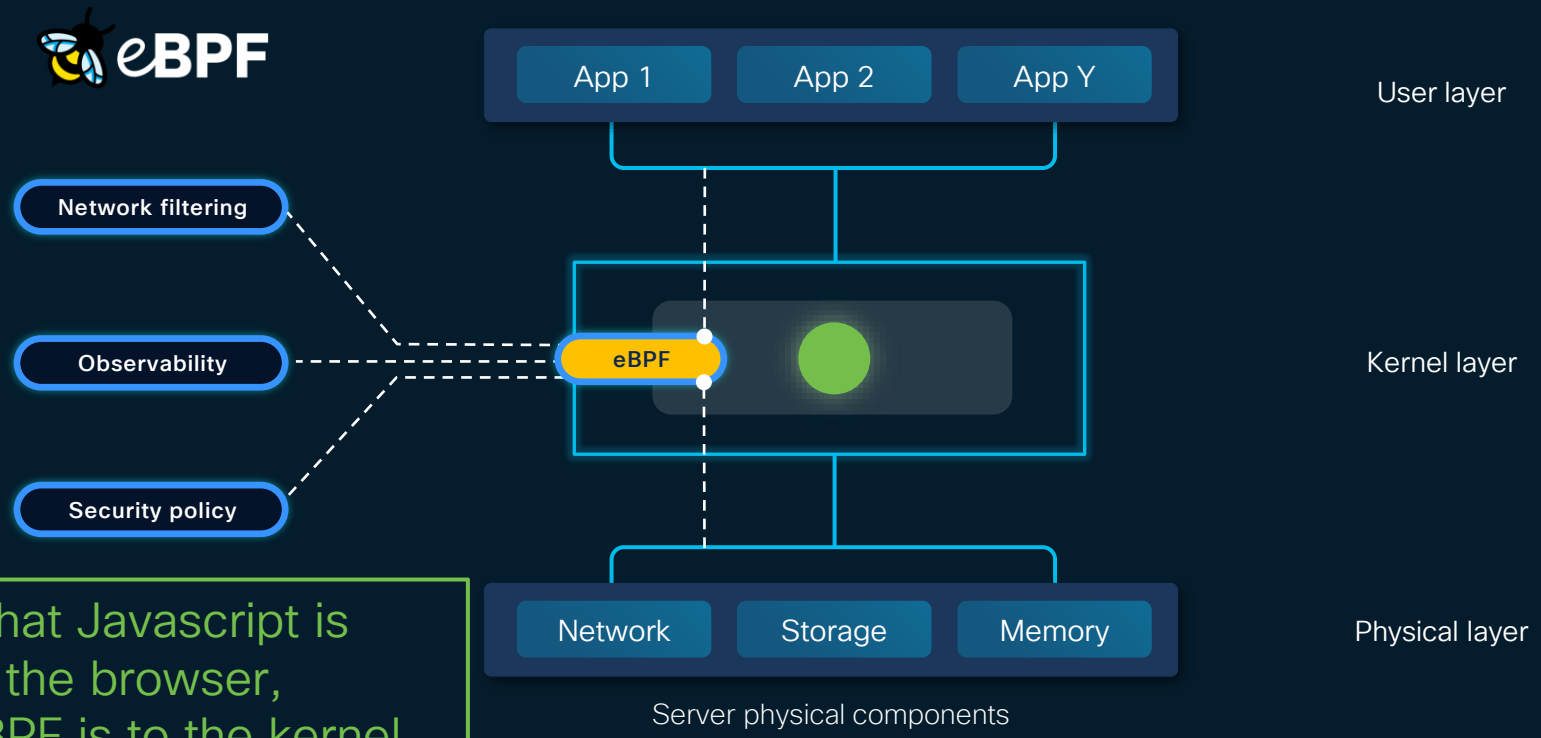
How do operating systems work?



How do operating systems work?



eBPF – Foundation of Hypershield



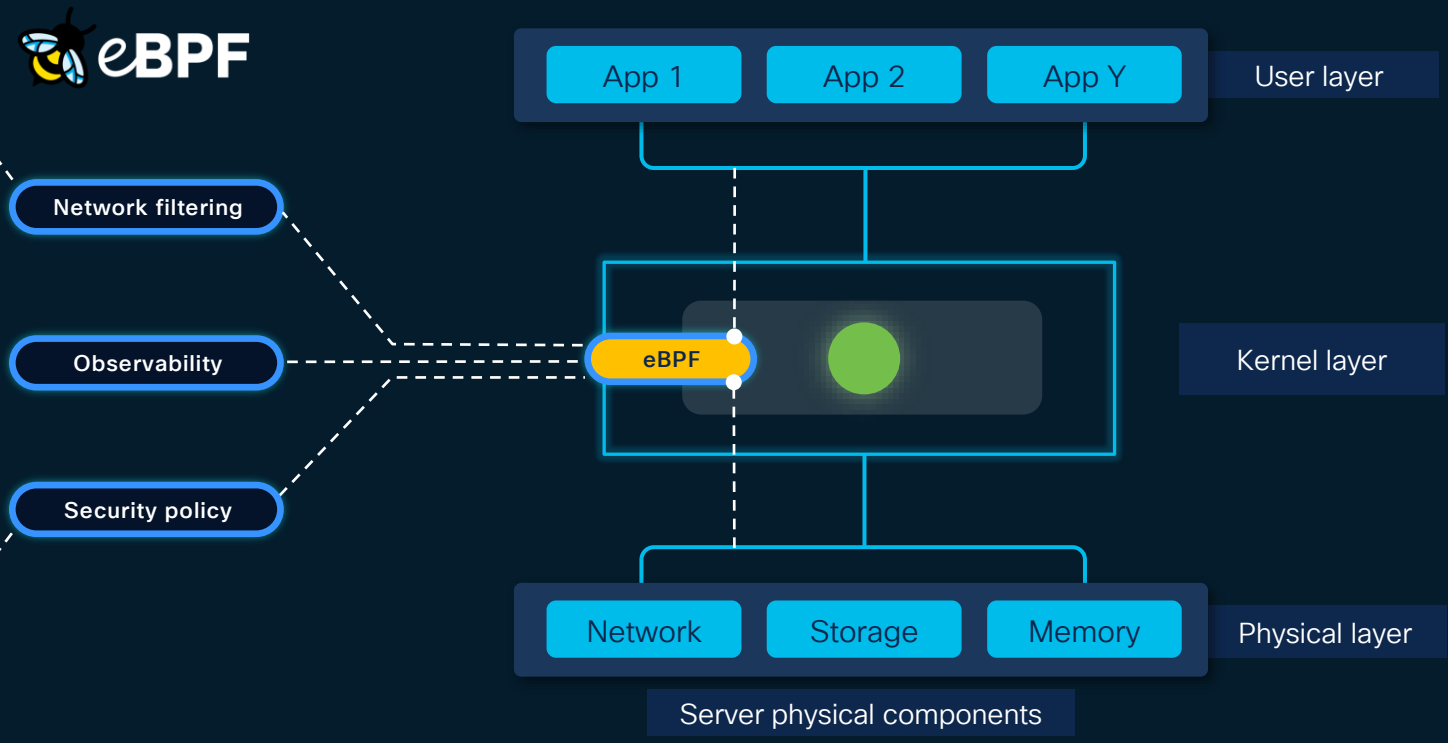
What Javascript is to the browser, eBPF is to the kernel.

eBPF – Foundation of Hypershield

- Kubernetes networking
- Load balancing
- Kubernetes services
- Identity-based security
- L7 policies

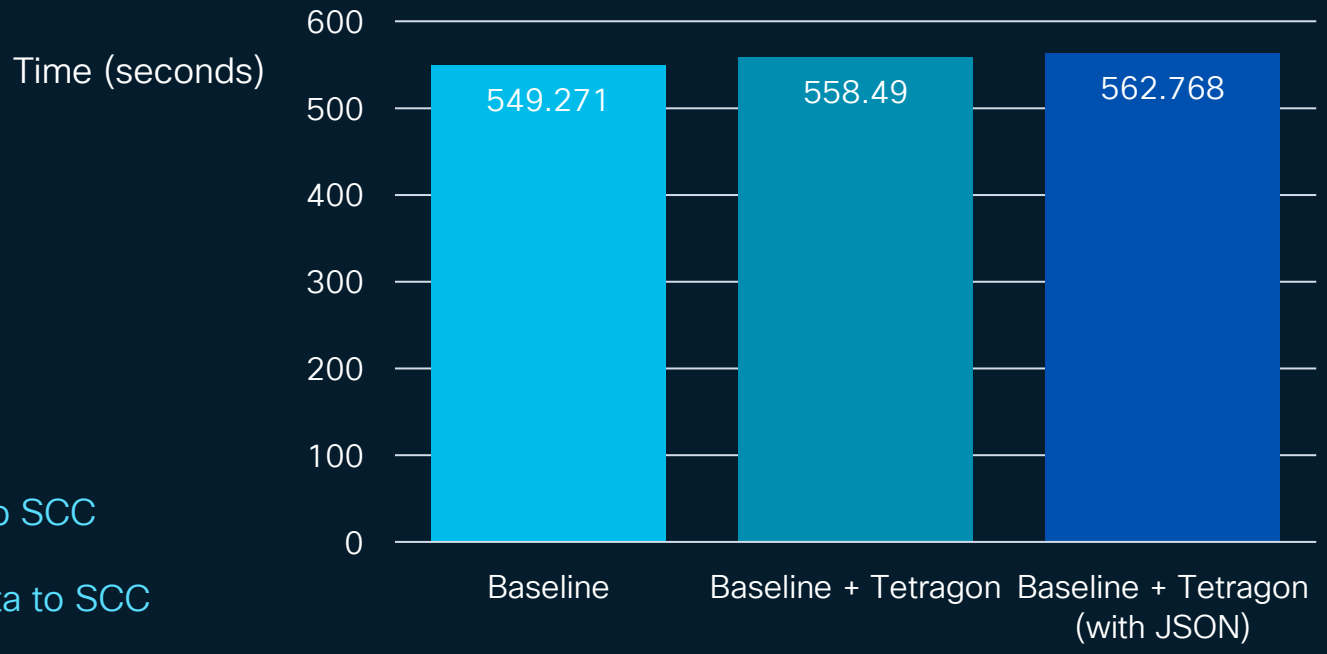
- Dependencies map (service and flows)
- Monitoring and alerting
- App monitoring

- Monitor process execution
- Runtime security policies
- Real time enforcement



eBPF Performance

Build 6.1.13 Kernel
(~1.5M Total events):
Time Elapsed - Lower is Better

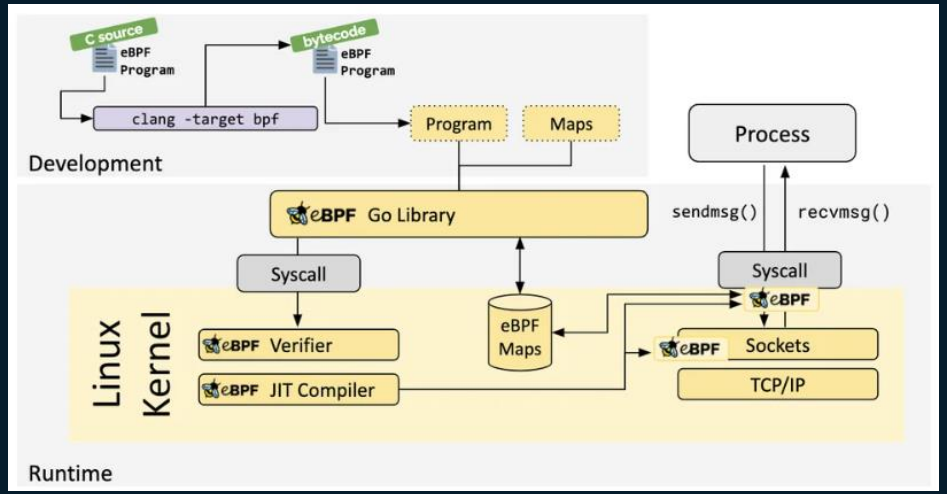


Remember, agent only:
1. maintains connection to SCC
2. maintains local graph
3. sends and receives data to SCC

The actual data plane work is done by eBPF

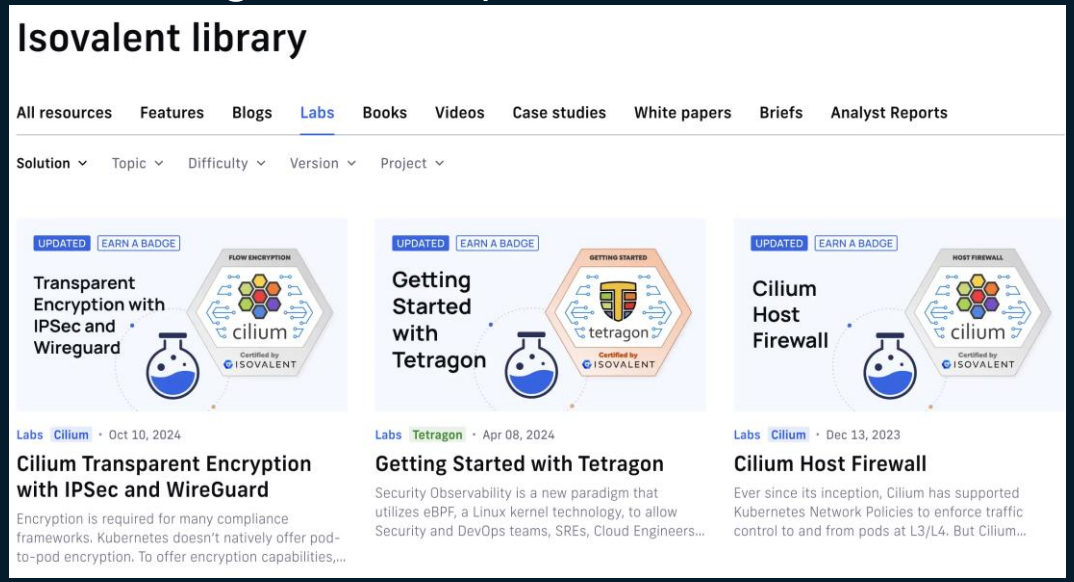
eBPF: Safety

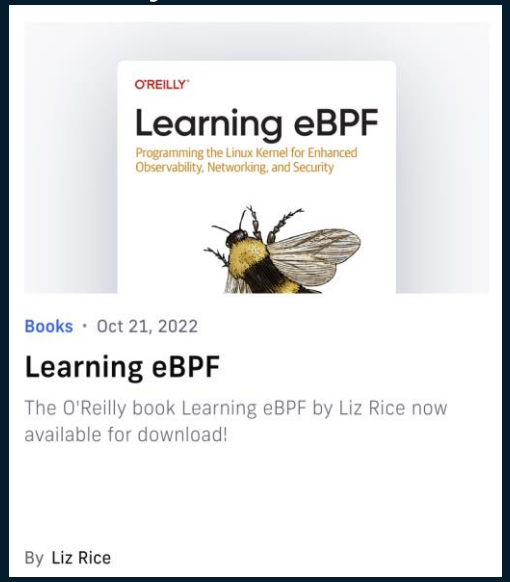
- eBPF runs in a restricted execution environment
- Before eBPF script is compiled:
  - verifier ensures things such as memory safety
  - hardening process
    (program execution protection,
    mitigation against spectre,
    constant bliinding, ...)
- eBPF can only access pre-approved kernel functions and datastructures



eBPF Learning Resources

- Books: https://isovalent.com/resource-library/books/
- Learning Labs: https://isovalent.com/resource-library/labs



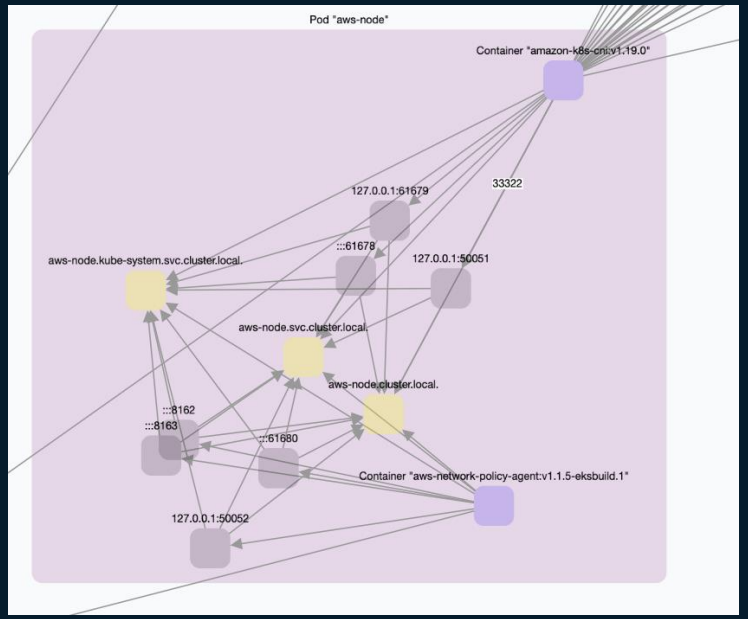


Where are we:

- Introduction
- Architecture
  - eBPF
  - AI & Graph Engine
  - Digital Twin
  - API
- Use Case: Autonomous Segmentation
- Use Case: Distributed Exploit Protection

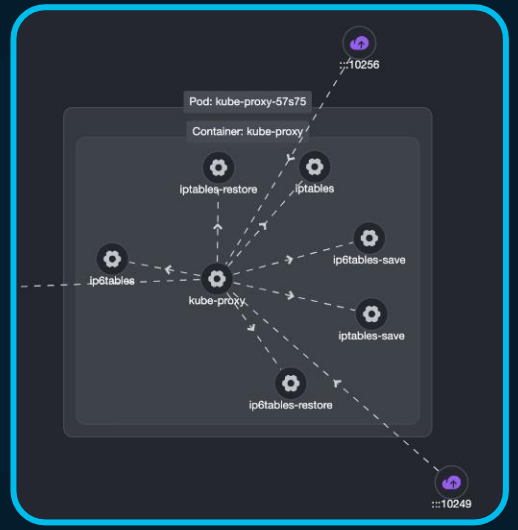
Why Use Graphs in Cybersecurity?

- Data Reduction
- Automated Policy Creation
- Real-Time Attack Detection
- Pattern Matching with Talos
- Multi-Host Detection
- Efficient Data Sharing
- Hierarchical Graphs to enable joining



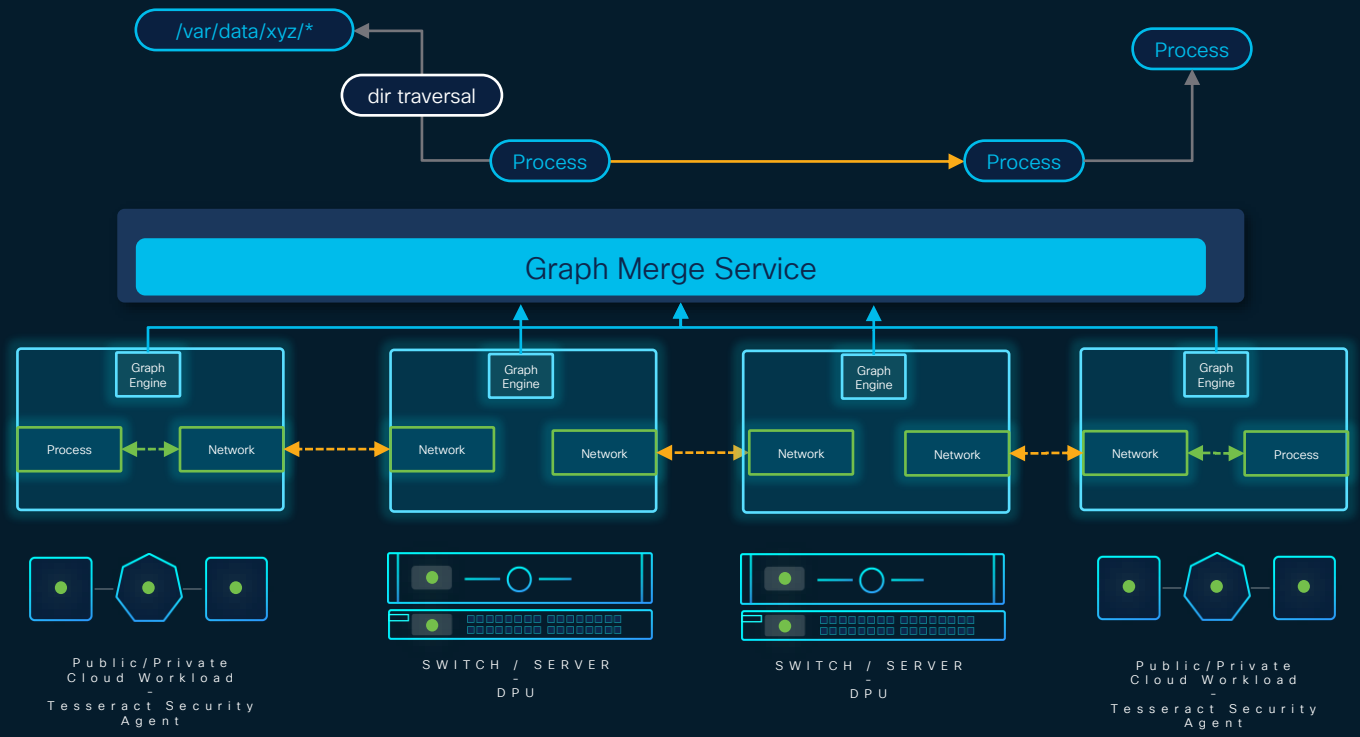
Visibility Today

UI only shows a fraction of the actual graph contents



191b0f35b1f9f1d8c51e7f1f163c76886ab76d9ac0a63d0945adc70d", "@type": "@id"}]}, "csco:kube_label
-storage"}, {"app.kubernetes.io/created-by": "rook-ceph-operator"}, {"app.kubernetes.io/compon
naged-by": "rook-ceph-operator"}, {"app.kubernetes.io/instance": "ocs-storagecluster-cephfiles
}, {"odf-resource-profile": "balanced"}, {"ceph_daemon_id": "ocs-storagecluster-cephfilesystem
tem": "ocs-storagecluster-cephfilesystem"}, {"mds": "ocs-storagecluster-cephfilesystem-a"}, {"
{"app.kubernetes.io/part-of": "ocs-storagecluster-cephfilesystem"}], "csco:kube_workload": "ro
namespace": "openshift-storage", "csco:kube_workload_kind": "Deployment"}, {"@id": "https://sb

Distributed Graph Engine

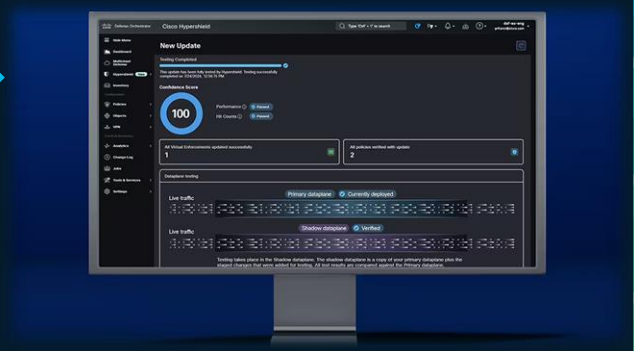


AI in Hypershield

Kenna

- CVE related info
- exploit code
- advisories
- …

Ground Truths



It is not just an LLM!
It is not a connection to OpenAI or ChatGPT!

Cisco Responsible ML/AI

Guidance & Oversight

Controls

Incident Management

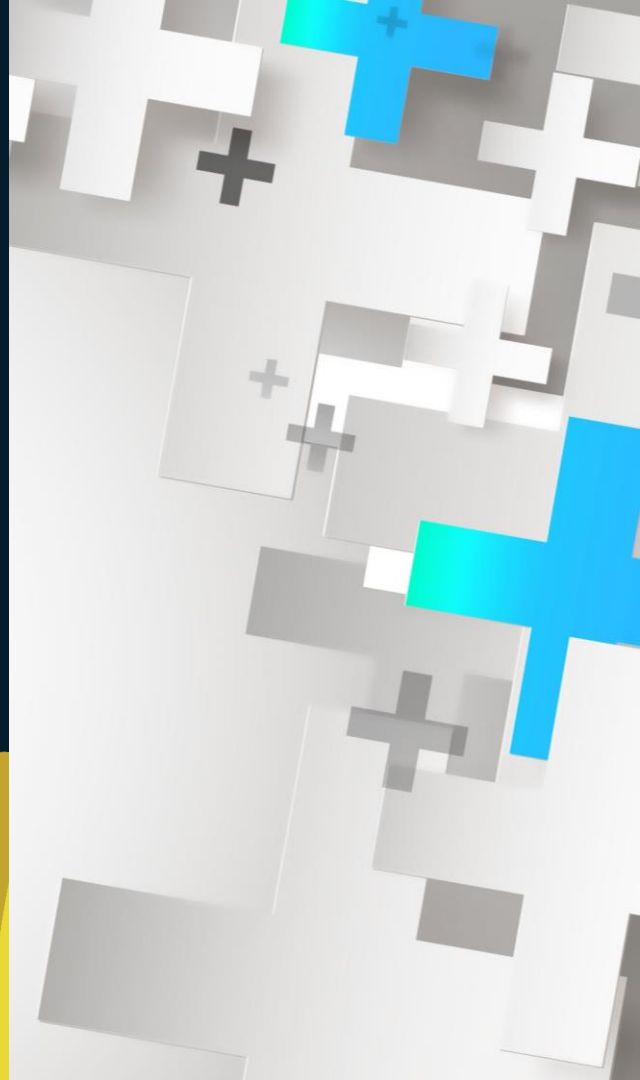
Industry Leadership

External Engagement

https://www.cisco.com/c/dam/en_us/about/doing_business/trust-center/docs/cisco-responsible-artificial-intelligence-framework.pdf

Where are we:

- Introduction
- Architecture
  - eBPF
  - AI & Graph Engine
  - Digital Twin
  - API
- Use Case: Autonomous Segmentation
- Use Case: Distributed Exploit Protection



Policy Updates

Hypershield Policy Proposal

Reason:
- CVE shield
- Behavioral Anomaly
- Segmentation update

Policy Details:
- PARC policy

Challenges:
- will this enforce the desired outcome?
- what will be the impact to rest of the environment
- will this introduce unexpected side effects?

~~Hypershield is bug free, we will be fine~~

~~Hypershield has AI, we will be fine~~

~~We will do all the testing and validation for every policy proposal~~

Digital Twin
- Automatic Test in Production environment
- No impact during testing
- Detailed test report

Changes can be validated on your live, production environment

Network-based enforcer's dual data plane: Earning your trust

Primary Data Plane

VERSION 2.0

VERSION 2.1

Shadow Data Plane

Self qualifying software updates

Primary Data Plane

DEPLOYED POLICY

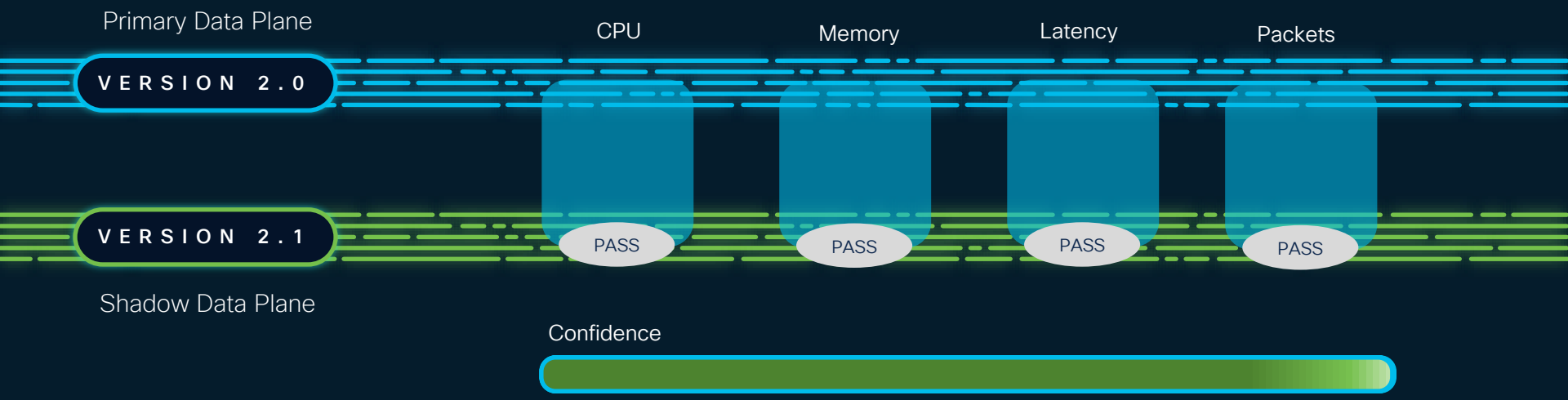
POLICY GROUP A

Shadow Data Plane

Policy verification, exploit protection test

Self Qualifying Updates

Earns your trust:
- ❑ Edit / Update
- ❑ Test Reports
- ❑ Schedule Deployment
- ❑ Deploy



Self Qualifying Updates: Test Report

Cisco Hypershield

Type 'Ctrl' + '/' to search

daf-ea-eng
jwittock@cisco.com

New Update

Testing Completed

This update has been fully tested by Hypershield. Testing successfully completed on 24/07/2024, 21:56:15.

Confidence Score

80

Performance ⓘ Passed

Hit Counts ⓘ Passed

All Network-based Enforcers updated successfully
1

All policies verified with update
0

Dual dataplane testing

Live traffic — Primary dataplane — Currently deployed

Live traffic — Shadow dataplane — Verified

Testing takes place in the Shadow dataplane. The shadow dataplane is a copy of your primary dataplane plus the staged changes that were added for testing. All test results are compared against the Primary dataplane.

Self Qualifying Updates: Test Report

Performance

Review

We test if the update passed or failed according to our metric. This ensures the update does not have a negative impact on the system.

7
Expected differences

CPU

Metric	Difference	Threshold for success	Result	Reason
CPU min	-0.59%	within 5%	Passed	Difference as expected
CPU max	0%	within 5%	Passed	*No difference*
CPU avg	0.05%	within 5%	Passed	Difference as expected

Memory

Metric	Difference	Threshold for success	Result	Reason
Memory min	-0.01%	within 5%	Passed	Difference as expected
Memory max	0.15%	within 5%	Passed	Difference as expected
Memory avg	0.13%	within 5%	Passed	Difference as expected
Memory growth	0%	within 5%	Passed	*No difference*

Latency

Metric	Difference	Threshold for success	Result	Reason
Latency min	0%	within 5%	Passed	*No difference*

Self Qualifying Updates: Test Report

Flow

Metric	Difference	Threshold for success	Result	Reason
Flow Age max	0%	within 5%	Passed	*No difference*
Flow Age avg	0%	within 5%	Passed	*No difference*

Hit Counts

Review

Hit counts are tracked to show any discrepancies between the number of times a policy was exercised between the primary dataplane and this update currently on the shadow dataplane.

Policy	Number of Hits Primary dataplane	Number of Hits Shadow dataplane	Difference	Review
Permit ICMP APP	0	n/a	n/a	Needs review
Test Policy	0	0	0	Needs review
Test Policy	0	0	0	Needs review
Test MS	0	0	0	Needs review
Test Policy	0	0	0	Needs review
Test	0	0	0	Needs review
sakhter-test-1	n/a	0	n/a	Needs review

Staged changes

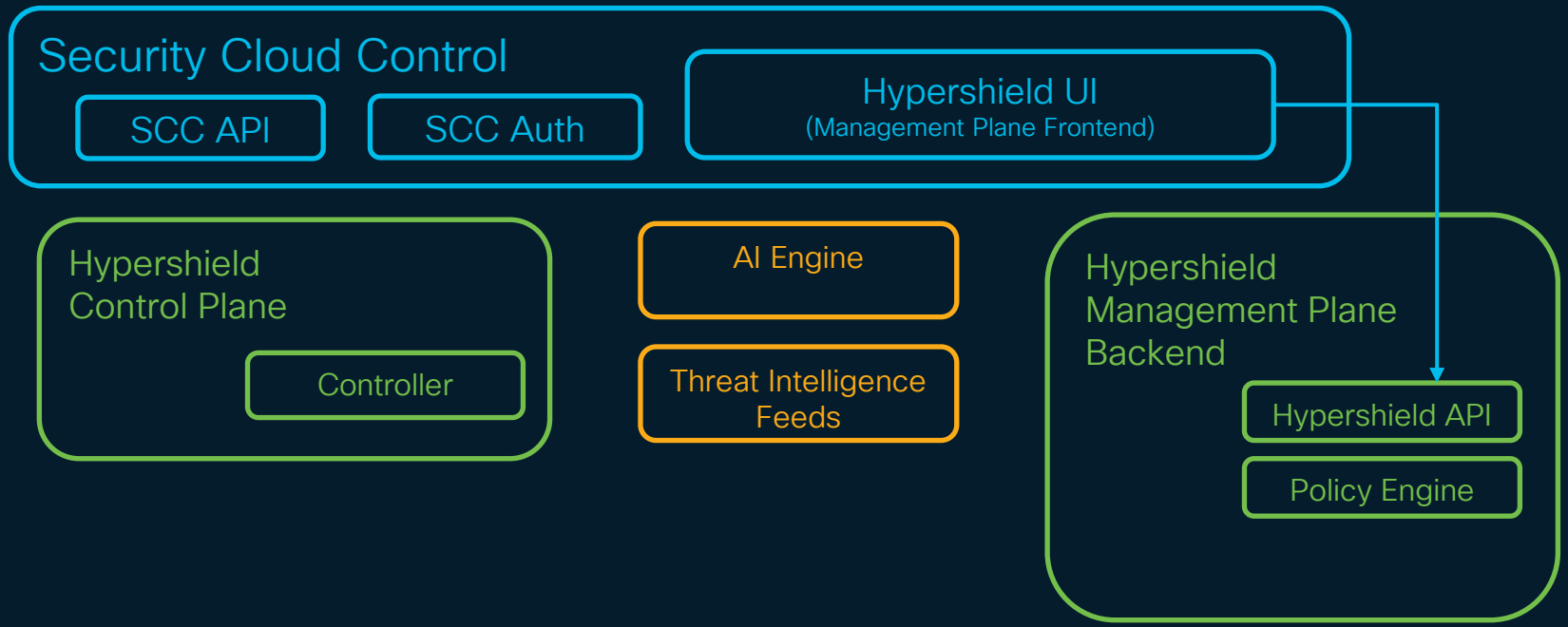
1 policy added

1 policies deleted

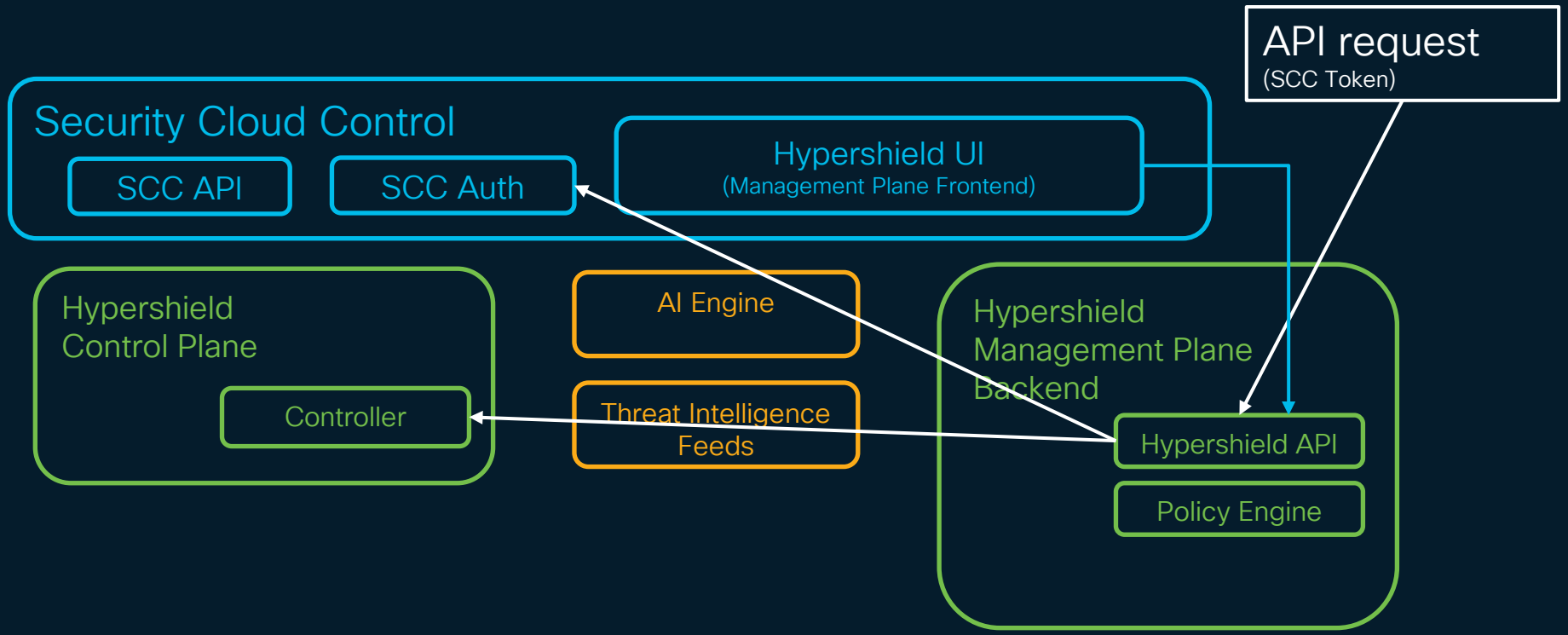
Where are we:

- Introduction
- Architecture
  - eBPF
  - AI & Graph Engine
  - Digital Twin
  - API
- Use Case: Autonomous Segmentation
- Use Case: Distributed Exploit Protection

Hypershield Architecture



Hypershield API



API Documentation (Swagger)

Swagger

Cisco Hypershield API 1.0.0 OAS 3.1

Notes:

- Any objects that take a pagination parameter returns a paginated response of { data: T[], total: number, options: PaginationOptions } (not shown).

Servers

/api

Authorize

AI

POST /v1/ai

FwAgent

GET /v1/fw-agent

POST /v1/fw-agent

GET /v1/fw-agent/{id}

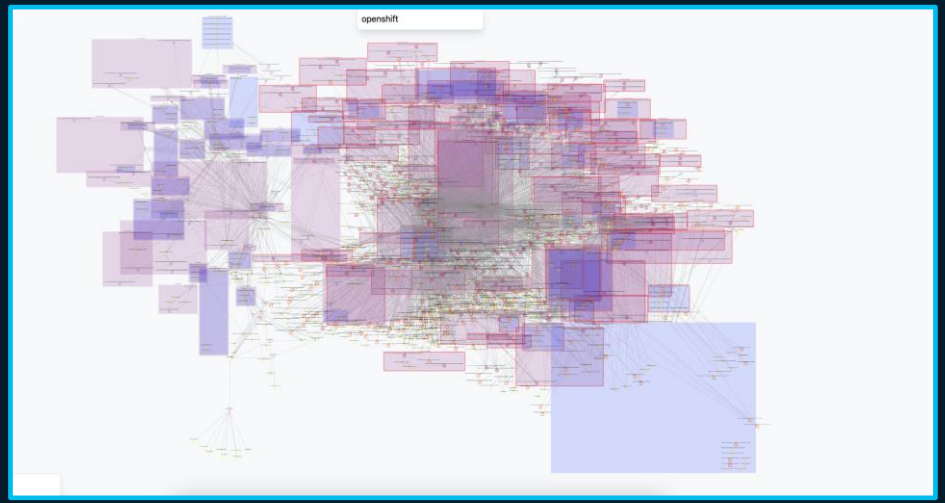
POST /v1/fw-agent/{id}

DELETE /v1/fw-agent/{id}

Object

GET /v1/object

Visibility through the API




```python
import requests
import json

url = "https://mp-api.prod.hypershield.engineering/api/v1/"
operation_id = "workload-overview"
api_token = ""

with open('tenant.config', 'r') as f:
    config = json.load(f)
    api_token = config['token']

payload = {}
headers = {"Authorization" : "Bearer " + api_token}

response = requests.request('GET', url+operation_id, headers=headers, data=json.dumps(payload))

print(json.dumps(response.json(), indent=2))
```


```json
{
  "id": "123b2153-3ab9-4321-9a2f-a7a426963e97",
  "tenantId": "6a32347a-a9db-430f-8dc7-e210c1274252",
  "data": {
    "@id": "http://sbg.cisco.com/graph/merged/hosts/pods/conts/net/e663
    "@graph": [
      {
        "@id": "https://sbg.cisco.com/host/9a0de0b68b602a3ddc9f8835ca8
        "@type": "csco:Host",
        "csco:debug": "mid:0,hid:ip-10-0-1-22.us-east-2.compute.interna
        "csco:host_name": "ip-10-0-1-22.us-east-2.compute.internal",
        "csco:machine_id": "0"
      },
      {
        "@id": "https://sbg.cisco.com/flow/2c4ef73abdd2c04891de6627ebd
        "@type": "csco:NetEndpoint",
        "csco:prot": "IPPROTO_TCP",
        "csco:debug": "net:IPPROTO_TCP-10.0.1.121:6676",
        "csco:endpoint": "10.0.1.121:6676",
        "csco:resolved_from": {
          "@list": [
            {
              "@id": "https://sbg.cisco.com/dns/027810d9dba7dc21b56170
              "@type": "@id"
            },
            {
              "@id": "https://sbg.cisco.com/dns/1b106cd399d086e197cbc0
              "@type": "@id"
            },
            {
              "@id": "https://sbg.cisco.com/dns/6d29d3cf425ff4372e31328
              "@type": "@id"
            }
          ]
        }
```

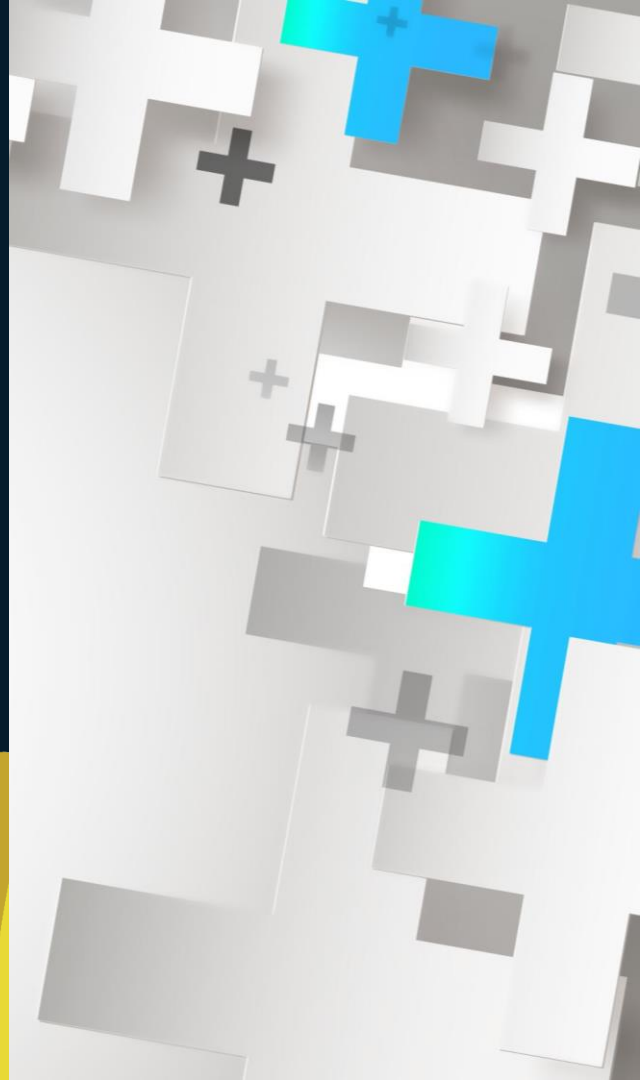
Shields using the API

```python
import requests
import json

url = "https://mp-api.prod.hypershield.engineering/api/v1/"
operation_id = "policy/tetragon"
api_token = ""

with open('tenant.config', 'r') as f:
    config = json.load(f)
    api_token = config['token']

payload = {}
headers = {"Authorization" : "Bearer " + api_token}

response = requests.request('GET', url+operation_id, headers=headers, data=json.dumps(payload))

print(json.dumps(response.json(), indent=2))
```

```
JWITTOCK-M-M4X7:example_scripts jwittock$ python3 get_shields.py
{
  "data": [
    {
      "id": "3a0563a0-1d4b-43ec-a89b-151d2a26966e",
      "tenantId": "d206f38f-6c4f-4e1c-b3b4-e42e5d33c671",
      "policyGroupId": null,
      "name": "CVE-HS-TEST",
      "description": "Test vulnerable package for testing.",
      "createdBy": "system",
      "updatedBy": "system",
      "policyType": "TETRAGON",
      "compensatingControlId": "12bbe803-935c-4a11-a044-38cc5edf83de",
      "autodeploy": false,
      "cedar": {
        "effect": "forbid",
        "principal": {
          "op": "==",
          "entity": {
            "type": "package",
            "id": "hs-test"
          }
        },
        "action": "security_bprm_check",
        "resource": {
          "op": "==",
          "entity": {
            "type": "linux_binprm"
          }
        },
        "conditions": [
          {
            "type": "when",
            "left": "principal.version",
            "op": "<=",
            "right": "1.0.0"
          }
```

Where are we:

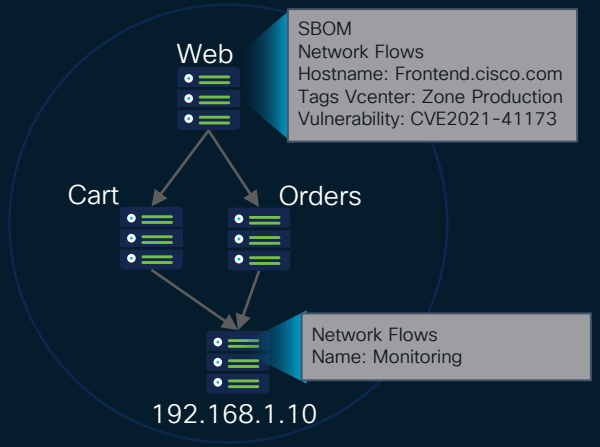
- Introduction
- Architecture
  - eBPF
  - AI & Graph Engine
  - Digital Twin
  - API
- Use Case: Autonomous Segmentation
- Use Case: Distributed Exploit Protection

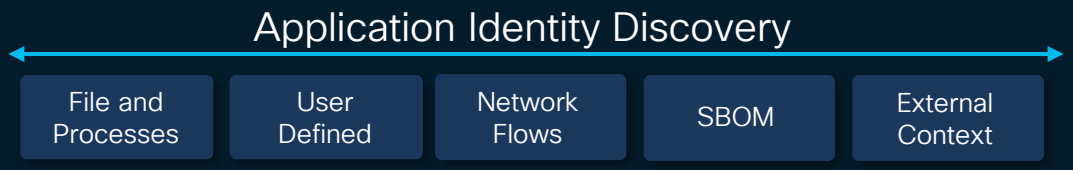


Application Identity Discovery

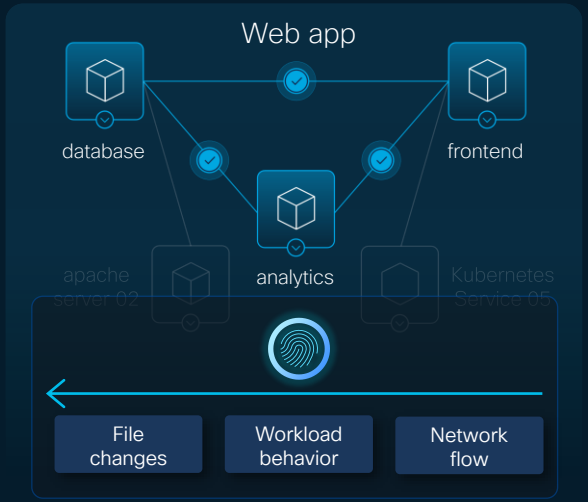
Discovering the Application Runtime Identities!

- eBPF based, realtime, factual
- Auto-Discovery of Application Runtime Inventory
  - Network Endpoints and Workloads
  - Network Flows and Processes
  - Filesystem and others
- Enrichment
  - SBOM
  - Vulnerabilities
  - User defined
  - External Systems

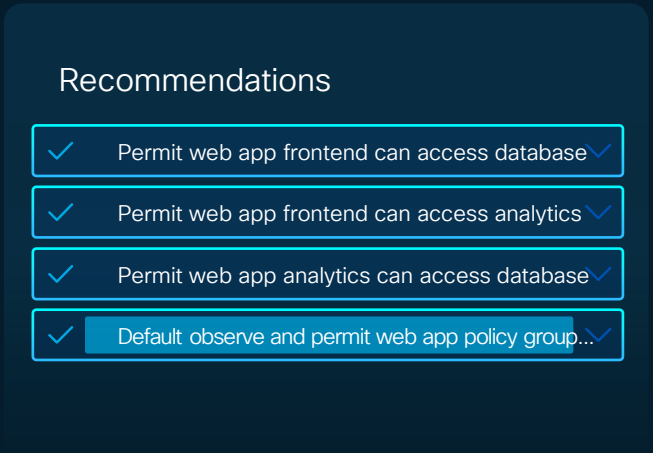




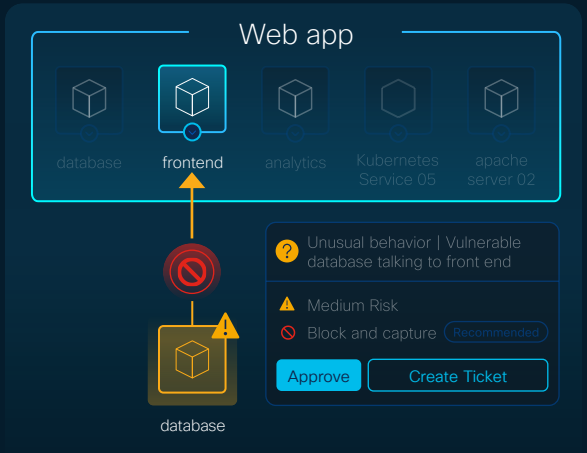
Segmentation that is effective and keeps up with changing applications



Complete understanding of changing app behavior from network to workload to pre-prod



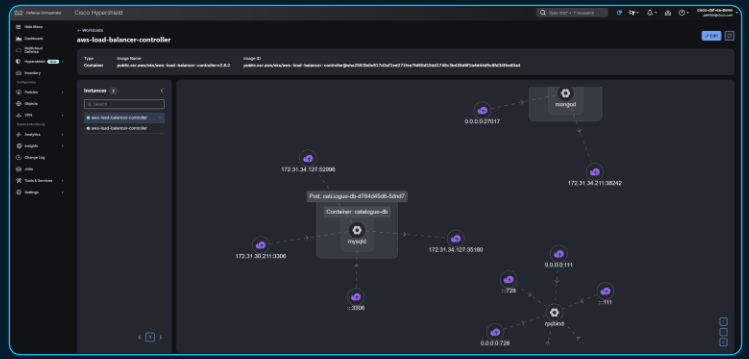
Flexible segmentation rules that help avoid app fragility

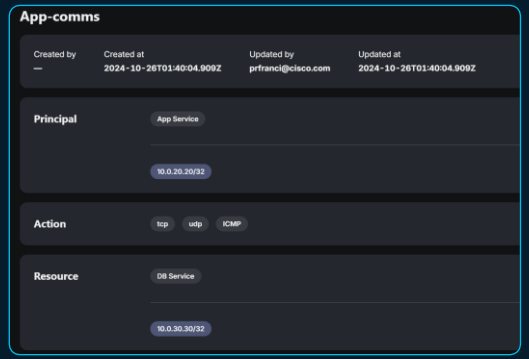


Policies updated to stricter rules in response to suspicious events

Note: Images are not an exact product UI representation

Autonomous Segmentation





Simplified policy management
Singular policy across multiple enforcement points on public and private clouds

Application Fingerprinting
Autonomous discovery, tagging, and grouping of workloads

Deployment confidence
Know how policies and software updates would perform in real time

Autonomous Segmentation

Baselining & Scanning Modes

New workloads automatically go into baselining mode.
In scanning mode, Hypershield scans for deviations & Anomalies.

Hosts | Pods | Containers | User-defined

Search

Autonomous Segmentation	Name	Kubernetes Namespace
Baselining	aws-load-balancer-controller-5ddf449c7c-d8xt2	kube-system
Baselining	cart-78dbff49b-rdgg2	robot-shop
Updating baseline	carts-666b98fdc4-pph8k	sock-shop
Baselining	catalogue-7b4b777975-77p2f	robot-shop
Scanning	catalogue-db-d764d45d6-5dnd7	sock-shop
Baselining	dispatch-7d4ff989d7-96kmg	robot-shop

Anomalies

Search

Anomalous instance

catalogue-db

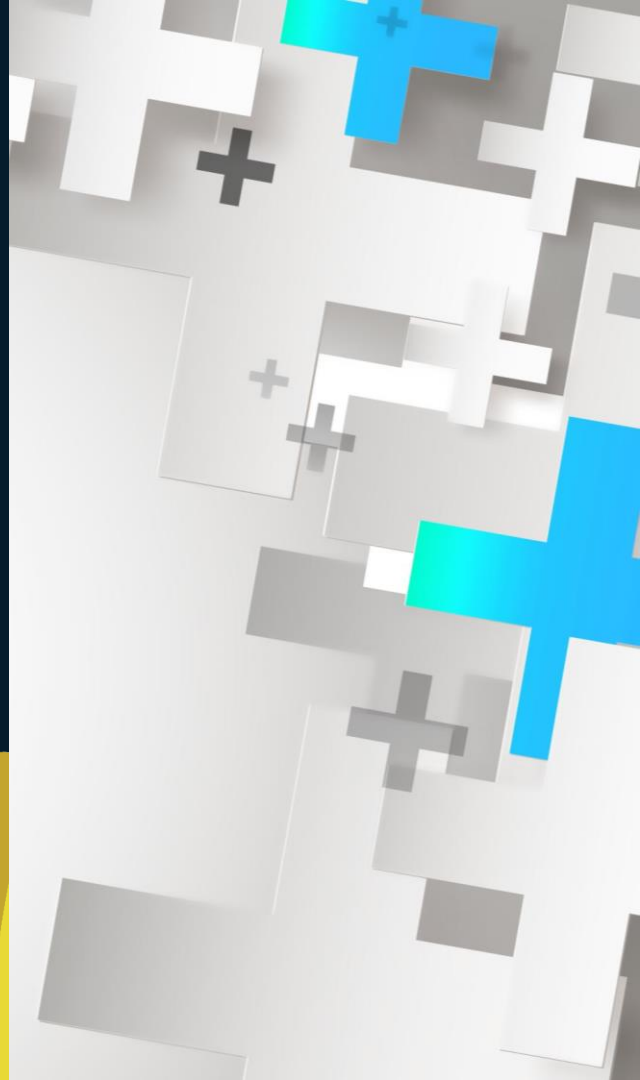
Anomaly
New process, gosu
New process, mysql
New process, mysqld
New process, awk
New process, sleep
New process, mysqld
New process, id
New process, mysql_tzinfo_to_sql
New process, chown
New process, mkdir

Anomaly Detection

Detect deviations from baseline and act upon them.

Where are we:

- Introduction
- Architecture
  - eBPF
  - AI & Graph Engine
  - Digital Twin
  - API
- Use Case: Autonomous Segmentation
- Use Case: Distributed Exploit Protection



End-to-end vulnerability management, accelerated

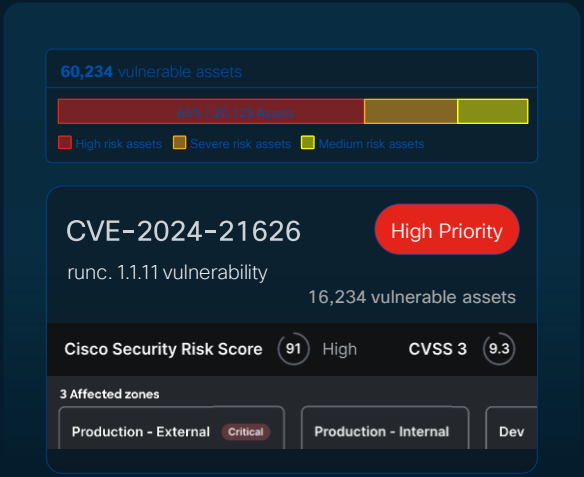
Weeks of expert analysis compressed to minutes

Automated inspection, pre-built integrations, AI recommendations, tests, and deployment

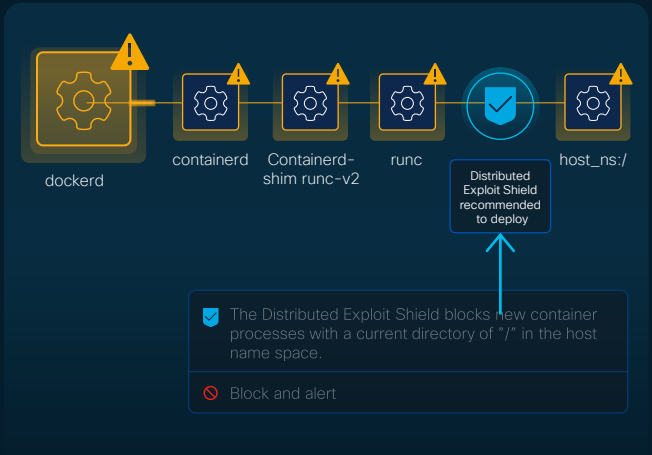
Multiple compensating controls evaluated; environment best fit recommended

Give the app team time to patch while working at speed of the adversary

Close the exploit gap against growing vulnerabilities with automated workflows



Complete view of the vulnerabilities, prioritized by severity and critical business flows



Surgical mitigating control in the path of the process that keeps application running

Tested against live production traffic to earn trust and increase confidence

Note: Images are not an exact product UI representation

Distributed Exploit Protection

Public Global Data Sources

- Kenna
- Public CVE info
- Talos

Hypershield AI → Shields for each CVE

Private Local Data Sources

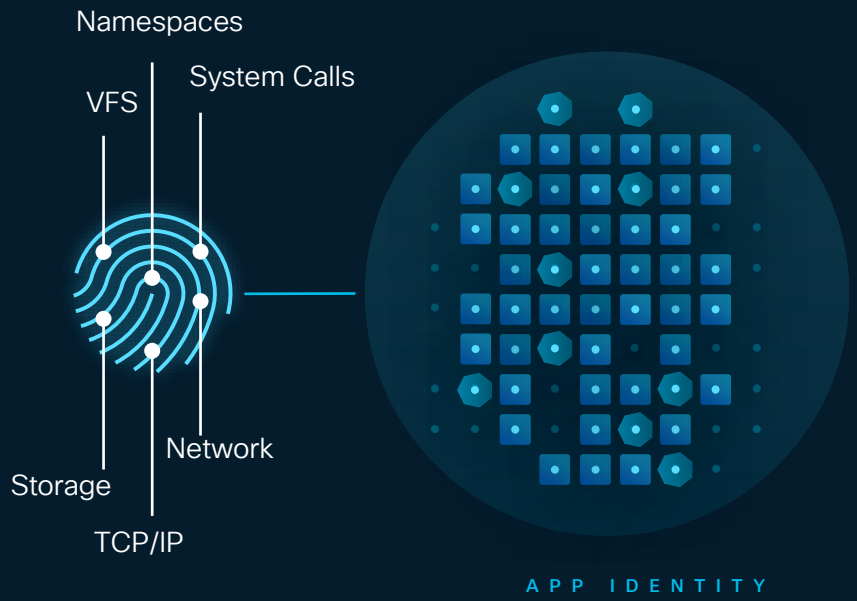
In-Depth Visibility
- what
- where
- which version
- which libraries

Local Graph → Global Graph → Detect CVE's

Local Graph → Global Graph

Distributed Exploit Protection

Unknown vulnerability protection



VALIDATED

dir traversal → file read

- /var/www
- httpd
- /etc/apache2/httpd_conf

SUSPICIOUS

- CWE-78 — OS command injection
- CWE-200 — Unauthorized access to sensitive information

MALICIOUS

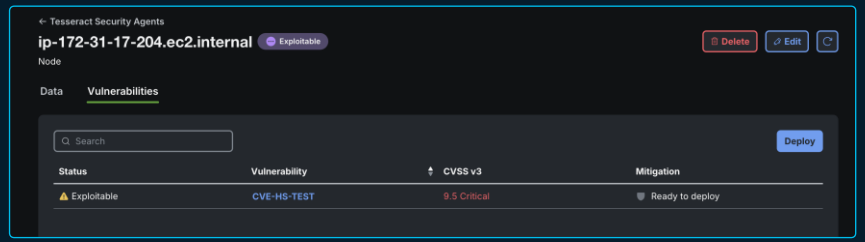
Web shell payload → Saved in PHP dir → Execute web shell

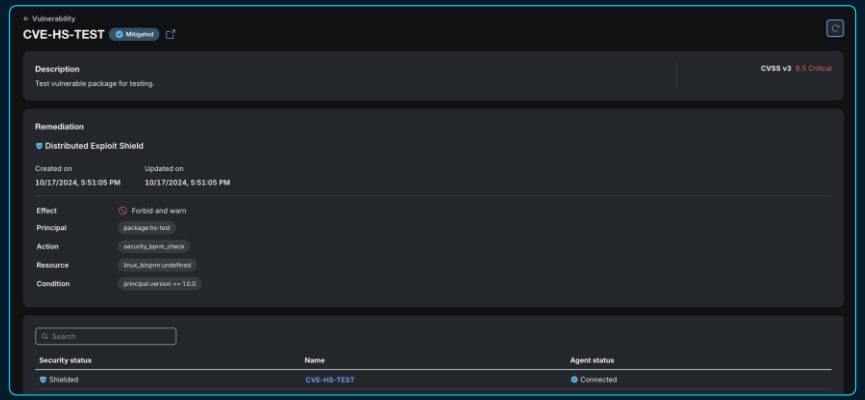
Application-specific behavior analysis | Common weakness enumeration and analysis

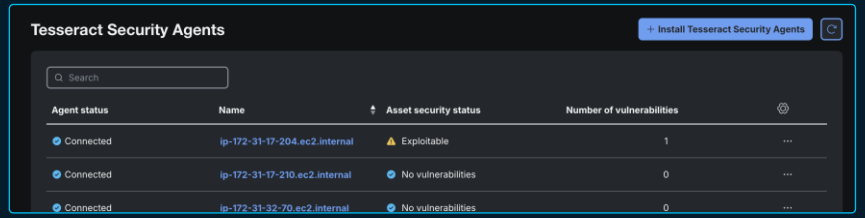
Distributed Exploit Protection Progress – Nov 2024

Vulnerability detection
Identify known software vulnerabilities (CVEs)

Mitigation shields
Protect applications by automatically implementing precise mitigating security controls







Thank you!

Join our Security Research Community

Participating in design research gives you a place to share your thoughts and experiences to influence the future of Cisco Security Products.

Webex App

Questions?

Use the Webex app to chat with the speaker after the session

How

1. Find this session in the Cisco Events 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, at least, February 28, 2025.

Fill Out Your Session Surveys

Participants who fill out a minimum of 4 session surveys and the overall event survey will get a unique Cisco Live t-shirt.

(from 11:30 on Thursday, while supplies last)

All surveys can be taken in the Cisco Events mobile app or by logging in to the Session Catalog and clicking the ‘Participant Dashboard’

Content Catalog

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 ciscolive.com/on-demand. Sessions from this event will be available from March 3.

Thank you

cisco Live!

GO BEYOND