rulez: get a “Bill of Behavior” through the supply chain

for anomaly-based runtime security

Dr. Constanze Roedig

SBA Research, fusioncore.ai

Key Researcher

SBA Research

Dr Constanze Roedig

Special Thanks to:

Redpill Linpro Cloud

TU WIEN TECHNISCHE UNIVERSITÄT WIEN

ASC Austrian Scientific Computing

This project is partially funded by EOSC Future INFRAEOSC-03-2020 Grant 101017536, part of ASC and TUW

EUROPEAN OPEN SCIENCE CLOUD

GÉANT

Federal Ministry Innovation, Mobility and Infrastructure Republic of Austria

Federal Ministry Economy, Energy and Tourism Republic of Austria

FFG Promoting Innovation.

vienna business agency

For the City of Vienna

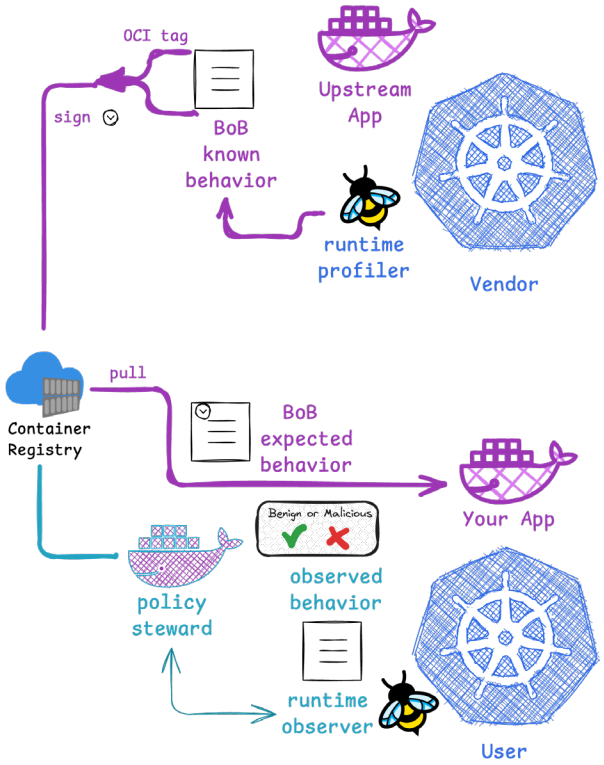
netidee OPEN INNOVATIONS

FWF Austrian Science Fund

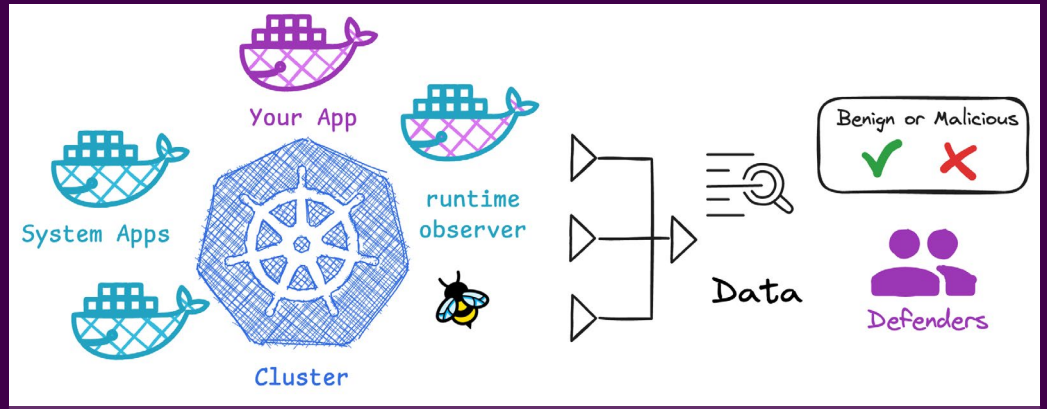
European Commission

k8sstormcenter

(S)BoB as a new standard - SBOM for runtime



- Software Bill of Behavior
  - How do you detect tampering?
  - Who writes runtime rules?
  - Who should write runtime rules?
  - Can we transfer runtimes rules from A -> B?
  - Alerting vs Enforcing
  - Attack Surface Shrinking
  - Integration into the ecosystem
  - What you can do today to benefit & help!

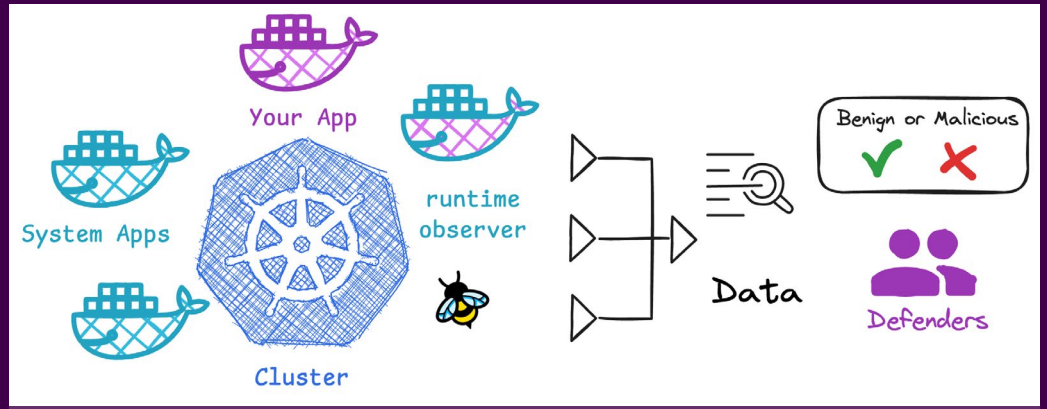


How do you detect tampering?

observe: get some data out

analyze: was this stuff bad?

k8sstormcenter



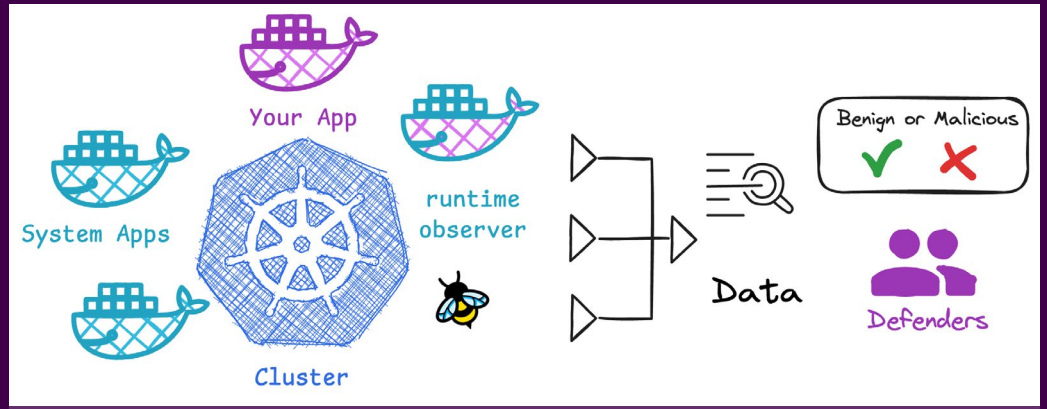
How do you detect tampering?

observe: get some data out

analyze: was this stuff bad?

You use RULES !

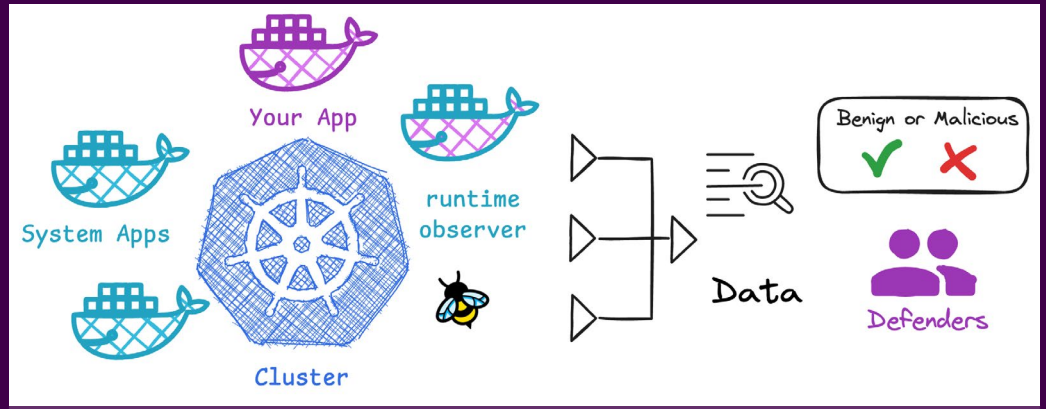
k8sstormcenter



Who writes these rules?

Currently: you the user

k8sstormcenter



Who should write these rules?

Receive and modify the rules: you the user

Create and maintain the rules: the vendor or supplier

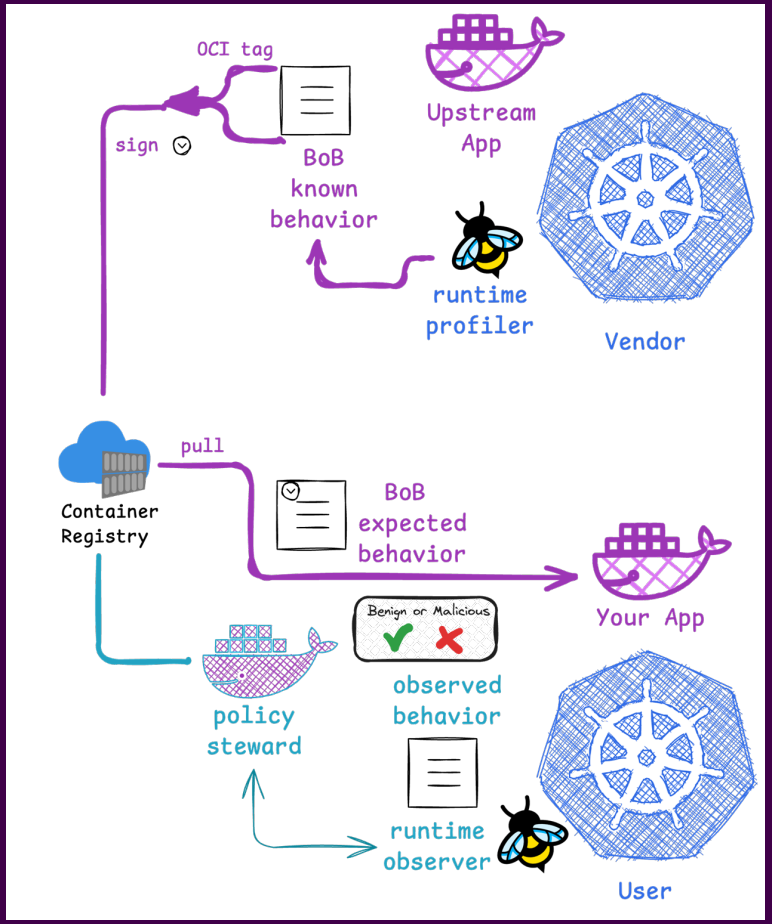
The people who made the SW

k8sstormcenter

Transferability??!

it works on my cluster

BUT: will it work on your cluster

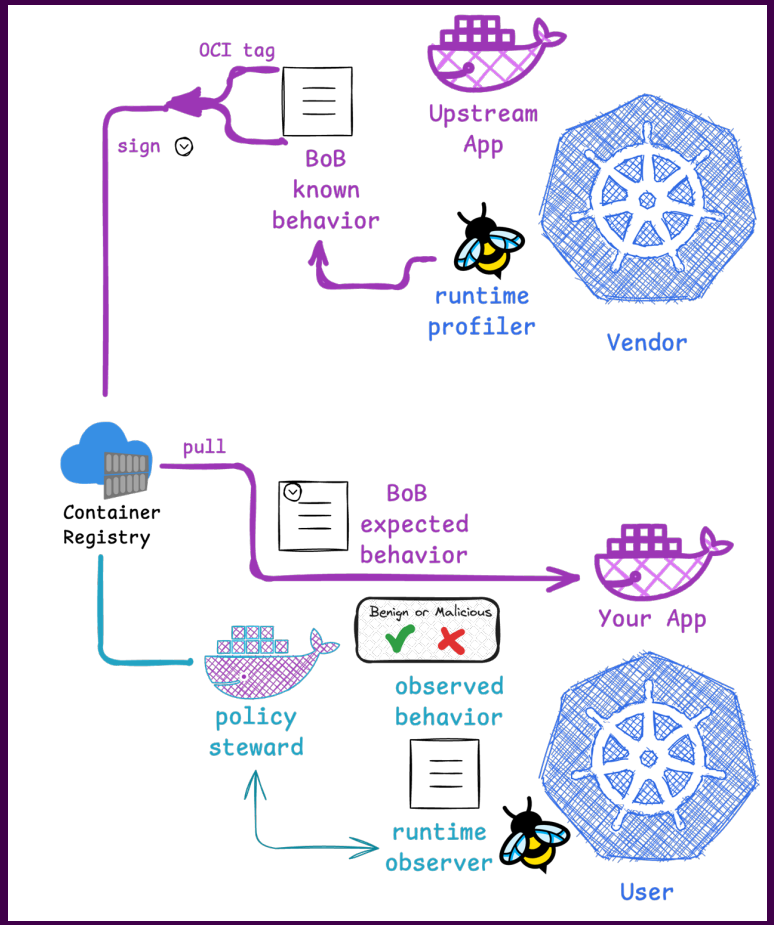


k8sstormcenter

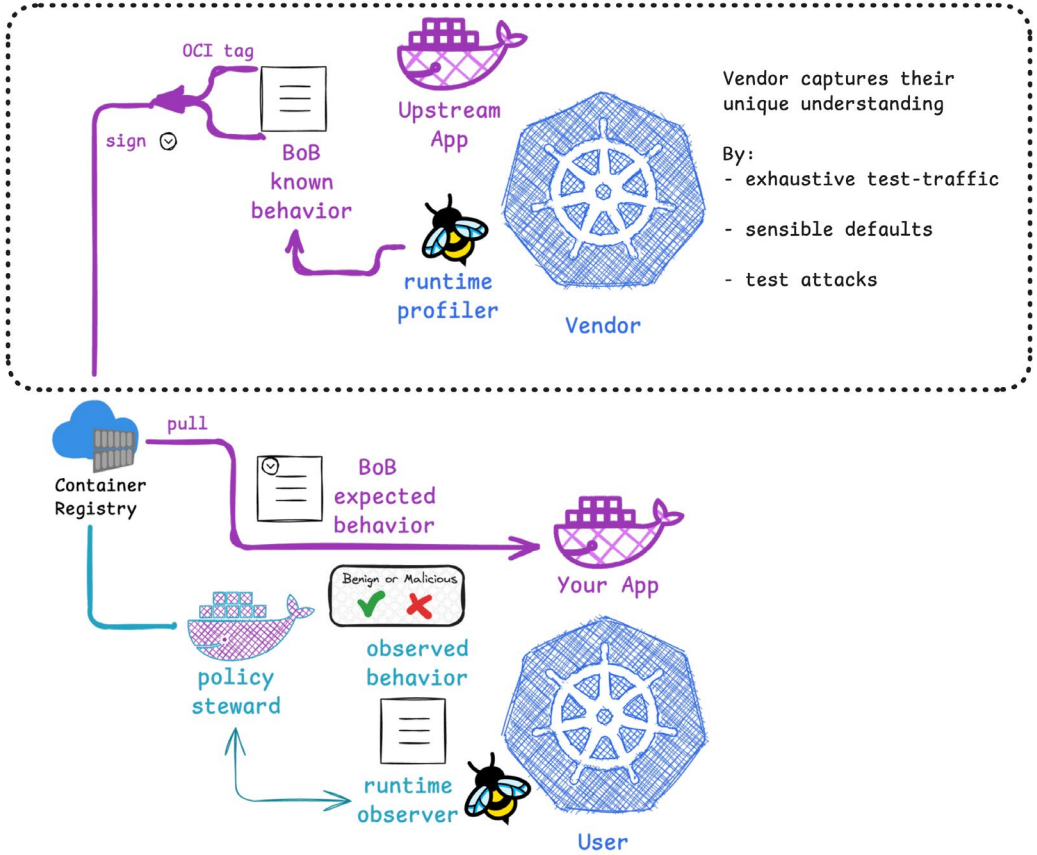
Idea of the year

attach a Bill of Behavior to software

to “supply” benign runtime rules



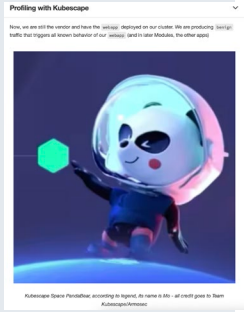
Vendor produces a BoB in spdx spec!



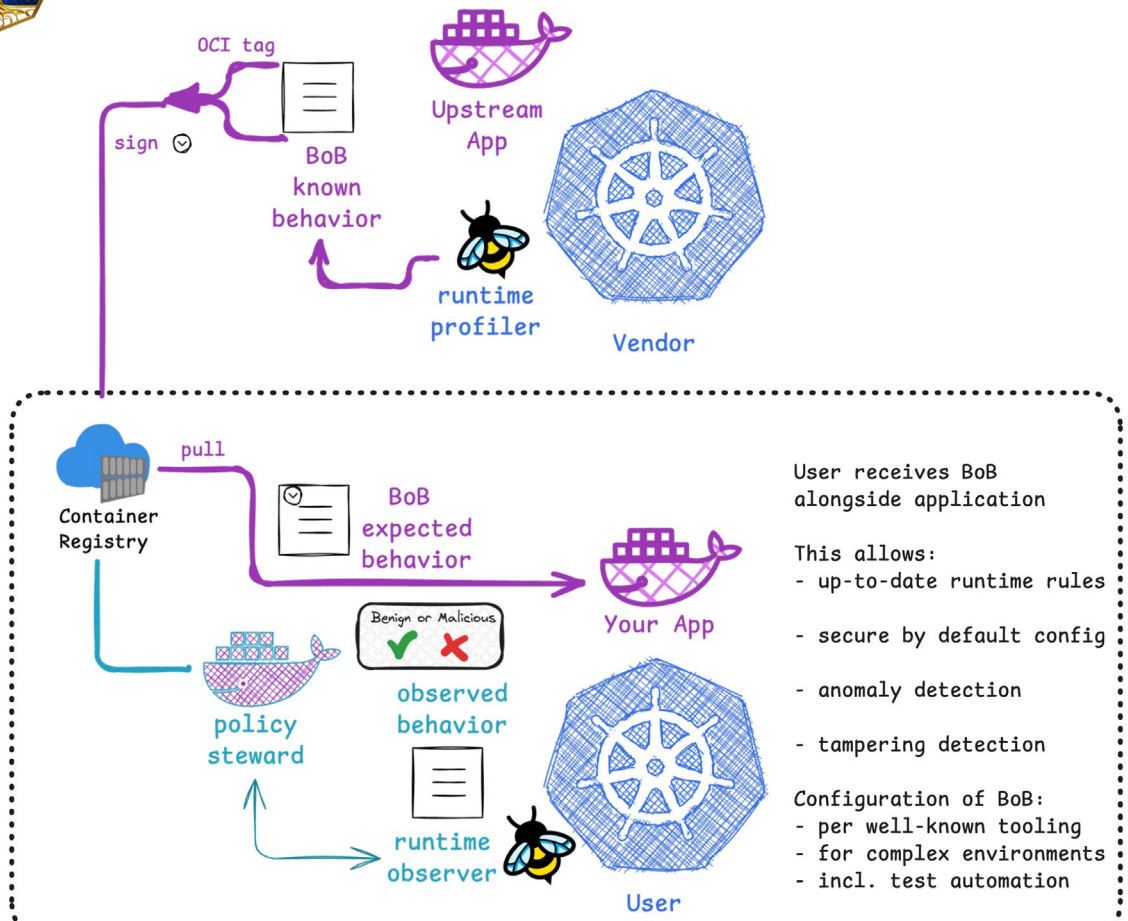
README Apache-2.0 license

```
apiVersion: spdx.softwarecomposition.kubescape.io/v1beta1
kind: ApplicationProfile
metadata:
  name: bob-application123
  namespace: {{ .Release.Namespace }}
spec:
  architectures:
  - amd64
  containers:
  - capabilities:  # KNOWN CAPABILITIES
    - DAC_OVERRIDE
    - SETGID
    - SETUID
    endpoints:     # KNOWN NETWORK
    - direction: inbound
      endpoint: :8080/ping.php
      headers:
        Host: # User accessible Overrides
        - {{ include "mywebapp.fullname" . }}.{{ .Release.Namespace }}.svc.cluster.local:{{ .V
      internal: false
      methods:
      - GET
    execs:       #KNOWN EXEC
    - args:
      - /usr/bin/dirname
      - /var/lock/apache2
      path: /usr/bin/dirname
    - args:
      - /bin/sh
      - -c
      - ping -c 4 172.16.0.2
      path: /bin/sh
    imageID: ghcr.io/k8sstormcenter/webapp@sha256:e323014ec9befb76bc551f8cc3bf158120150e2e277b
    opens:     #KNOWN FILE OPENS
    - flags:
      - O_CLOEXEC
      - O_DIRECTORY
      - O_NONBLOCK
      - O_RDONLY
      path: /etc/apache2/* #globs that generalize UUIDs or well-known FS structures
    - flags:
      - O_CLOEXEC
      - O_RDONLY
      path: /etc/group
    - flags:
      - O_CLOEXEC
      - O_RDONLY
      path: /etc/ld.so.cache
    rulePolicies:  # SPECIFIC EXCEPTION RULES
      R0001:
        processAllowed:
        - ping
        - sh
      R0002: {}
      ...
    syscalls:    # KNOWN SYSCALLS
    - accept4
    - access
    - arch_prctl
    - getegid
```

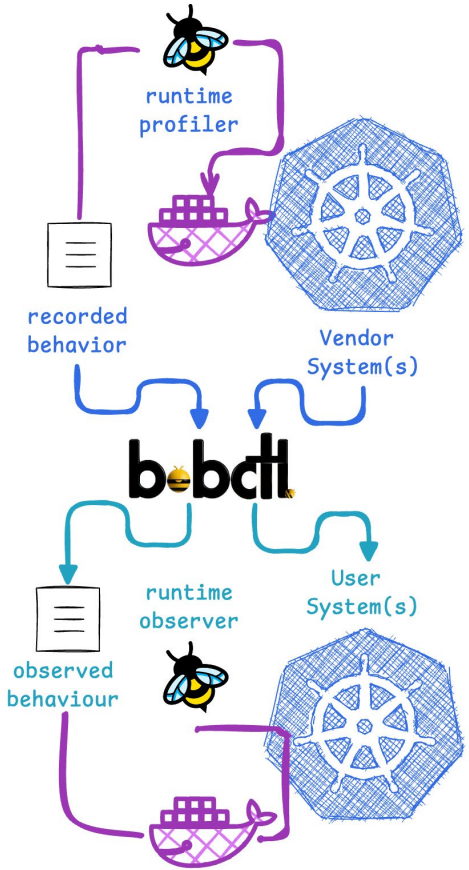
Thanks Matthias!



User receives the BoB with each update



BoBctl translates between systems



In order to allow for vendor neutrality:

Packaging systems:

- Helm

RunTime Engines:

- Kubescape 4.0 (Oct 2025)
- *NeuVector (possible)*
- *AppArmor (planned)*

Kubernetes:

- K8s: GKE, Vanilla, Talos, k3s, k0s
- Kind, Minikube
- *Openshift, SAP … (planned)*

https://github.com/k8sstormcenter/bobctl

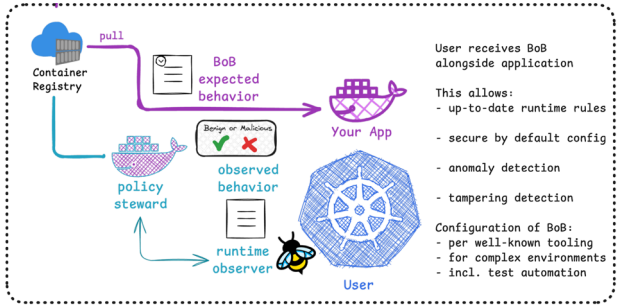
How much does it reduce the attack surface?

```
❯ ./attacksurface.sh
```

Profile	Capabilities	Network	Opens (#)	Execs (#)	Allowed Syscalls
Kubernetes Default (v1.33)	unconfined	CNI	unconfined	unconfined	363
catalogsource	CHOWN,DAC_OVERRIDE,DAC_READ_SEARCH,NET_ADMIN,SETGID,SETPCAP,SETUID,SYS_ADMIN	0	19	2	106
kelvin	DAC_OVERRIDE,DAC_READ_SEARCH,NET_ADMIN	0	97	1	76
pem	BPF,DAC_READ_SEARCH,IPC_LOCK,NET_ADMIN,PERFMON,SYS_ADMIN,SYS_PTRACE,SYSLOG	0	390	1	122
pletcd	NET_ADMIN,NET_RAW,SETGID,SETPCAP,SETUID,SYS_ADMIN	0	39	10	92
plnats	DAC_OVERRIDE,DAC_READ_SEARCH,NET_ADMIN	3	11	1	57
querybroker	DAC_OVERRIDE,DAC_READ_SEARCH,FOWNER,NET_ADMIN	0	20	1	107
viziercloudconnector	DAC_OVERRIDE,DAC_READ_SEARCH,NET_ADMIN	0	18	1	70
viziermeta	NET_ADMIN	0	16	1	65
vizieroperator	NET_ADMIN	1	13	1	104

CNCF Pixie provides real-time observability and debugging (based on eBPF)
pem = Pixie edge module (the agent that handles the tracing)

Enforcing vs Alerting




```
❯ ./attacksurface.sh
Profile                 Capabilities                                                                  Network  Opens (#)  Execs (#)  Allowed Syscalls
-------                 ------------                                                                  -------  ---------  ---------  ----------------
-----
pem                     BPF,DAC_READ_SEARCH,IPC_LOCK,NET_ADMIN,PERFMON,SYS_ADMIN,SYS_PTRACE,SYSLOG    0        390        1          122
-----
```

Anomaly Alerting for:

- Network Endpoints, RunTimeRules
- File Opens and SysCalls

RunTime Engines:

- Kubescape 4.0 (Q4 2025)

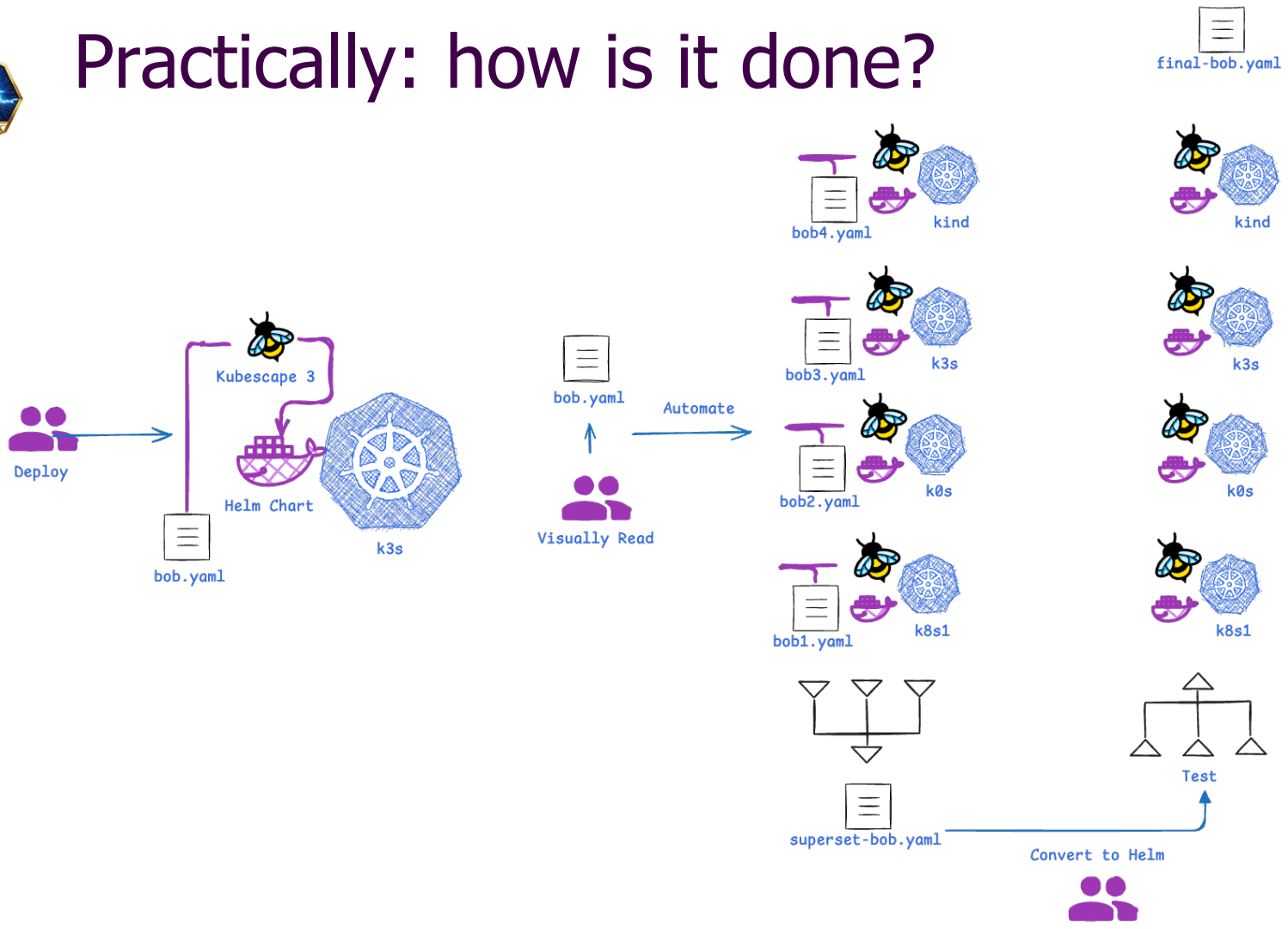
Enforcing for:

- {Capabilities, Executables} tuples

RunTime Engines:

- AppArmor (roadmap)

Practically: how is it done?



Practically: how is it done?

← Create Redis Bill of Behaviour on different Kernels, Kubernetes Versions, Kubernetes Distros and OSs

55 create systematic cicd pipeline with matrix output thats are easy ... #24

Summary

Jobs

- build (ubuntu-22.04, k3s, v1.30....
- build (ubuntu-22.04, k3s, v1.31.0)
- build (ubuntu-22.04, k3s, v1.32....
- build (ubuntu-22.04, k3s, v1.33....
- build (ubuntu-22.04, k3s, v1.33.2)
- build (ubuntu-22.04, kind, v1.30...
- build (ubuntu-22.04, kind, v1.31....
- build (ubuntu-22.04, kind, v1.32...
- build (ubuntu-22.04, kind, v1.33...
- build (ubuntu-22.04, kind, v1.33...
- build (ubuntu-22.04, minikube, ...
- build (ubuntu-22.04, minikube, ...
- build (ubuntu-22.04, minikube, ...
- build (ubuntu-22.04, minikube, ...
- build (ubuntu-22.04, minikube, ...
- build (ubuntu-24.04, k3s, v1.30....
- build (ubuntu-24.04, k3s, v1.31.0)
- build (ubuntu-24.04, k3s, v1.32....
- build (ubuntu-24.04, k3s, v1.33....
- build (ubuntu-24.04, k3s, v1.33....
- build (ubuntu-24.04, kind, v1.30...
- build (ubuntu-24.04, kind, v1.31....
- build (ubuntu-24.04, kind, v1.32...
- build (ubuntu-24.04, kind, v1.33...
- build (ubuntu-24.04, kind, v1.33...
- build (ubuntu-24.04, minikube, ...
- build (ubuntu-24.04, minikube, ...
- build (ubuntu-24.04, minikube, ...
- build (ubuntu-24.04, minikube, ...
- build (ubuntu-24.04, minikube, ...
- package-bobs

Annotations
1 error

package-bobs
succeeded 4 days ago in 3m 10s

- Set up job
- Checkout code
- Download all BoB artifacts
- Create a superset
- Package all BoBs into a tarball
- Upload BoB tarball
- setup kind
- Test superset Bob on kind
- Run helm test
- Upload superset BoB as the new final (and tested) Bill of Behavior
- Post setup kind
- Post Checkout code
- Complete job

summarize summary https://github.com/k8sstormcenter/bobctl/actions/runs/17190024510

Matrix Build Summary

OS	K8s Distro	K8s Version	Kernel Version	Status	Observed Syscalls	Observed Processes	Observed File Access
ubuntu-22.04	k3s	v1.30.0	6.8.0-1031-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-22.04	k3s	v1.31.0	6.8.0-1031-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-22.04	k3s	v1.32.0	6.8.0-1031-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-22.04	k3s	v1.33.0	6.8.0-1031-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-22.04	k3s	v1.33.2	6.8.0-1031-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-22.04	kind	v1.30.0	6.8.0-1031-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-22.04	kind	v1.31.0	6.8.0-1031-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-22.04	kind	v1.32.0	6.8.0-1031-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-22.04	kind	v1.33.0	6.8.0-1031-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-22.04	kind	v1.33.2	6.8.0-1031-azure	❌ failure	`getrusage, pidfd_open, pidfd_send_signal, writev`	N/A	N/A
ubuntu-22.04	minikube	v1.30.0	6.8.0-1031-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-22.04	minikube	v1.31.0	6.8.0-1031-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-22.04	minikube	v1.32.0	6.8.0-1031-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-22.04	minikube	v1.33.0	6.8.0-1031-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-22.04	minikube	v1.33.2	6.8.0-1031-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-24.04	k3s	v1.30.0	6.11.0-1018-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-24.04	k3s	v1.31.0	6.11.0-1018-azure	❌ failure	`alarm, getrusage, times, vfork, writev`	N/A	N/A
ubuntu-24.04	k3s	v1.32.0	6.11.0-1018-azure	❌ failure	`alarm, getrusage, times, vfork, writev`	N/A	N/A
ubuntu-24.04	k3s	v1.33.0	6.11.0-1018-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-24.04	k3s	v1.33.2	6.11.0-1018-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-24.04	kind	v1.30.0	6.11.0-1018-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-24.04	kind	v1.31.0	6.11.0-1018-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-24.04	kind	v1.32.0	6.11.0-1018-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-24.04	kind	v1.33.0	6.11.0-1018-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-24.04	kind	v1.33.2	6.11.0-1018-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-24.04	minikube	v1.30.0	6.11.0-1018-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-24.04	minikube	v1.31.0	6.11.0-1018-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-24.04	minikube	v1.32.0	6.11.0-1018-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-24.04	minikube	v1.33.0	6.11.0-1018-azure	✅ Success	`getrusage, writev`	N/A	N/A
ubuntu-24.04	minikube	v1.33.2	6.11.0-1018-azure	✅ Success	`getrusage, writev`	N/A	N/A

What is a (S)BoB?

We introduce the Software “Bill of Behavior” (BoB): a vendor-supplied profile detailing known benign runtime behaviors for software, designed to be distributed directly within OCI artifacts. Generated using eBPF, a BoB codifies expected syscalls, file access patterns, network communications, and capabilities. This empowers powerful, signature-less anomaly detection, allowing end-users to infer malicious activity or tampering in third-party software without the current burden of authoring and maintaining complex, custom security rules.

It solves the problem of distributing **secure-by-default** and **up-to-date** security rules for detection and defense.

Module 1: Record the runtime-profile of known behaviour
Taking the role of a software-vendor, we profile a sample-app and produce its BoB

Lesson: Vendor produces Bill of Behaviour for their application/product
Prerequisites and deployment of "sample app" for which we create a Bill of Behaviour

Module 2: Customer-Usage leverage the Bill of Behaviour for verification and detection

Step 2: Deploy the Application

Using a well-known `demo` ** app, we deploy a ping utility called `webapp` that has:

- **a) Desired functionality:** it pings things.
- **b) Undesired functionality:** it is vulnerable to injection (runtime is compromised)
  - *This is to mimic a CVE in your app.*
- **c) Tampering with the artefact:** In module 2, we will additionally tamper with the
  - *This is to mimic a SupplyChain corruption between vendor and you.*

```
> cd ~/bobctl
```

Step 3: Generate Traffic of Benign Behaviour

Benign (*adjective*) bi-ˈnīn

- **Benignity** (*noun*) bi-ˈnig-nə-tē
- **Benignly** (*adverb*) bi-ˈnīn-lē

Definitions/SYNONYMS:

1. Of a mild type or character that does not threaten health or life. *HARMLESS*.
2. Of a gentle disposition: *GRACIOUS*.
3. Showing kindness and gentleness. *FAVORABLE*, *WHOLESOME*.

User receives the BoB

Live demo

Course lesson by Constanze Roedig

Congrats! ✅

App 1: Deploy Webapp with its BoB

Here, most of the logic is hidden in a `Makefile`, this is the easiest method for the easiest application.

```
> make storage # auxiliary step
> make helm-install
```

```
runtimerulealertbinding.kubescape.io/all-rules-all-pods configured
sleep 5
kubectl rollout restart -n honey ds node-agent
daemonset.apps/node-agent restarted
kubectl wait --for=condition=ready pod -l app=kubevuln  -n honey --timeou
pod/kubevuln-77897b796c-qfdtf condition met
kubectl wait --for=condition=ready pod -l app=node-agent  -n honey --time
pod/node-agent-67b4g condition met
pod/node-agent-8g5nz condition met
pod/node-agent-vwpvx condition met
Installing webapp with BoB configuration ...
helm pull oci://ghcr.io/k8sstormcenter/mywebapp
Pulled: ghcr.io/k8sstormcenter/mywebapp:0.1.0
Digest: sha256:1f35668f2db47ec3e6c34ad597df3decfe3046d5e01963b46f2a73dcf0
Release "webapp" has been upgraded. Happy Helming!
NAME: webapp
LAST DEPLOYED: Fri Jul 18 15:29:37 2025
NAMESPACE: webapp
STATUS: deployed
REVISION: 2
NOTES:
Your webapp-mywebapp application is now deployed.
```

Congrats! ✅

IDE | Explorer | dev-machine | cplane-01 | node-01 | node-02 | dev-machine (2)

```
laborant@dev-machine:~$ cd ~
git clone https://github.com/k8sstormcenter/bobctl.git
cd bobctl
Cloning into 'bobctl'...
remote: Enumerating objects: 867, done.
remote: Counting objects: 100% (106/106), done.
remote: Compressing objects: 100% (55/55), done.
remote: Total 867 (delta 67), reused 61 (delta 47), pack-reused 761 (from 1)
Receiving objects: 100% (867/867), 343.51 KiB | 8.38 MiB/s, done.
Resolving deltas: 100% (450/450), done.
laborant@dev-machine:bobctl$ make storage # auxiliary step
make helm-install
kubectl apply -f https://openebs.github.io/charts/openebs-operator-lite.yaml
namespace/openebs created
serviceaccount/openebs-maya-operator created
clusterrole.rbac.authorization.k8s.io/openebs-maya-operator created
clusterrolebinding.rbac.authorization.k8s.io/openebs-maya-operator created
customresourcedefinition.apiextensions.k8s.io/blockdevices.openebs.io created
customresourcedefinition.apiextensions.k8s.io/blockdeviceclaims.openebs.io created
configmap/openebs-ndm-config created
daemonset.apps/openebs-ndm created
deployment.apps/openebs-ndm-operator created
deployment.apps/openebs-ndm-cluster-exporter created
service/openebs-ndm-cluster-exporter-service created
daemonset.apps/openebs-ndm-node-exporter created
service/openebs-ndm-node-exporter-service created
deployment.apps/openebs-localpv-provisioner created
kubectl apply -f https://openebs.github.io/charts/openebs-lite-sc.yaml
storageclass.storage.k8s.io/openebs-hostpath created
storageclass.storage.k8s.io/openebs-device created
kubectl apply -f storage/sc.yaml
storageclass.storage.k8s.io/local-hostpath created
kubectl patch storageclass local-hostpath -p '{"metadata": {"annotations":{"storage
class.kubernetes.io/is-default-class":"true"}}}'
storageclass.storage.k8s.io/local-hostpath patched
/usr/local/bin/helm repo add kubescape https://kubescape.github.io/helm-charts/
"kubescape" has been added to your repositories
/usr/local/bin/helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "kubescape" chart repository
Update Complete. ⎈Happy Helming!⎈
/usr/local/bin/helm upgrade --install kubescape kubescape/kubescape-operator --vers
ion 1.28.0 -n honey --create-namespace --values kubescape/values.yaml
Release "kubescape" does not exist. Installing it now.
NAME: kubescape
LAST DEPLOYED: Sun Jul 20 05:40:50 2025
NAMESPACE: honey
STATUS: deployed
REVISION: 1
```

```
0:00Z",
          "upperLayer": false,
          "cwd": "/var/www/html",
          "path": "/bin/cat"
        }
      }
    },
    "containerID": "0b748fe5277a36f3
af625f35cd8f390f7dd900dfc853a72c861c
ae99dc"
  },
  "event": {
    "runtime": {
      "runtimeName": "containerd",
      "containerId": "0b748fe5277a36
b4af625f35cd8f390f7dd900dfc853a72c86
e4ae99dc",
      "containerName": "mywebapp-app
      "containerImageName": "ghcr.io
sstormcenter/webapp@sha256:e323014ec
fb76bc551f8cc3bf158120150e2e277bae1
c2da6c56c0a2b",
      "containerImageDigest": "sha25
622cf306b94e8a6e7cfd718f048015e0336
0f19228d8beee23a0ccc57bb"
    },
    "k8s": {
      "namespace": "webapp",
      "podName": "webapp-mywebapp-67
968bb-4t8cg",
      "podLabels": {
        "app.kubernetes.io/instance"
webapp",
        "app.kubernetes.io/name": "m
bapp",
        "pod-template-hash": "679659
b"
      },
      "containerName": "mywebapp-app
      "owner": {}
    },
    "timestamp": 1752990479151726967
    "type": "normal"
  },
  "level": "error",
  "message": "Unexpected process lau
ed: /bin/cat",
  "msg": "Unexpected process launche
  "time": "2025-07-20T05:47:59Z"
}
```

User receives the BoB
Live demo

Course lesson by Constanze Roedig

Congrats! ✅

2 Verify BoB (included in Webapp) via supplied test

Now, as user/customer, verify the app is working as intended by the vendor:

In a new tab `new terminal`, open the logs (for any anomalies)

```
> kubectl logs -n honey -l app=node-agent -f
```

Please, switch back to the original `dev-machine` tab, and proceed to test

```
> helm test webapp -n webapp
```

```
# make helm-test
NAME: webapp
LAST DEPLOYED: Fri Jul 18 15:29:37 2025
NAMESPACE: webapp
STATUS: deployed
REVISION: 2
TEST SUITE:     webapp-mywebapp-test-connection
Last Started:   Fri Jul 18 16:28:03 2025
Last Completed: Fri Jul 18 16:28:11 2025
Phase:          Succeeded
NOTES:
Your webapp-mywebapp application is now deployed.
```

The only logs, you should see are that the test-container was monitored

```
{"level":"info","ts":"2025-07-18T16:28:08Z","msg":"ApplicationProfileMana
{"level":"info","ts":"2025-07-18T16:28:37Z","msg":"ApplicationProfileMana
```

IDE | Explorer | dev-machine | cplane-01 | node- | dev-machine (2)

```
pp.kubernetes.io/name=mywebapp,app.kubernetes.io/instance=web
app" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace webapp port-forward $POD_NAME 8080:80
rm -rf mywebapp-0.1.0.tgz
HASH=$(kubectl get rs -n webapp -o jsonpath='{.items[0].metad
ata.labels.pod-template-hash}')
The template has is
helm upgrade --install webapp oci://ghcr.io/k8sstormcenter/my
webapp --version 0.1.0  --namespace webapp --set bob.create=t
rue --set bob.ignore=false --set bob.templateHash=$(kubectl g
et rs -n webapp -o jsonpath='{.items[0].metadata.labels.pod-t
emplate-hash}')
Pulled: ghcr.io/k8sstormcenter/mywebapp:0.1.0
Digest: sha256:606bc61786ec40972cc7578a60d50b77ee895b661745a5
7ecfde4787b3964484
Release "webapp" has been upgraded. Happy Helming!
NAME: webapp
LAST DEPLOYED: Sun Jul 20 05:41:49 2025
NAMESPACE: webapp
STATUS: deployed
REVISION: 2
NOTES:
Your webapp-mywebapp application is now deployed.
To access your application, you can use port-forwarding:

  export POD_NAME=$(kubectl get pods --namespace webapp -l "a
pp.kubernetes.io/name=mywebapp,app.kubernetes.io/instance=web
app" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace webapp port-forward $POD_NAME 8080:80
kubectl wait --for=condition=ready pod -l app.kubernetes.io/n
ame=mywebapp -n webapp
pod/webapp-mywebapp-67965968bb-4t8cg condition met
laborant@dev-machine:bobctl$ helm test webapp -n webapp
NAME: webapp
LAST DEPLOYED: Sun Jul 20 05:41:49 2025
NAMESPACE: webapp
STATUS: deployed
REVISION: 2
TEST SUITE:     webapp-mywebapp-test-connection
Last Started:   Sun Jul 20 05:45:40 2025
Last Completed: Sun Jul 20 05:45:48 2025
Phase:          Succeeded
NOTES:
Your webapp-mywebapp application is now deployed.
To access your application, you can use port-forwarding:

  export POD_NAME=$(kubectl get pods --namespace webapp -l "a
pp.kubernetes.io/name=mywebapp,app.kubernetes.io/instance=web
app" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace webapp port-forward $POD_NAME 8080:80
```

```
  "ts": "2025-07-20T05:41:25Z",
  "msg": "started ptrace tracing"
}
{
  "level": "info",
  "ts": "2025-07-20T05:41:25Z",
  "msg": "started io_uring tracing"
}
{
  "level": "info",
  "ts": "2025-07-20T05:41:25Z",
  "msg": "started http tracing"
}
{
  "level": "info",
  "ts": "2025-07-20T05:41:25Z",
  "msg": "main container handler started"
}
{
  "level": "info",
  "ts": "2025-07-20T05:41:30Z",
  "msg": "RBCache - ruleBinding added/modified",
  "name": "/all-rules-all-pods"
}
{
  "level": "info",
  "ts": "2025-07-20T05:42:01Z",
  "msg": "ApplicationProfileManager - start monitor on co
ner",
  "preRunning": false,
  "container index": 0,
  "container ID": "0b748fe5277a36f38b4af625f35cd8f390f7dd
fc853a72c861dde4ae99dc",
  "k8s workload": "webapp/webapp-mywebapp-67965968bb-4t8c
748fe5277a36f38b4af625f35cd8f390f7dd900dfc853a72c861dde4a
c"
}
{
  "level": "error",
  "ts": "2025-07-20T05:42:53Z",
  "msg": "SbomManager - failed to save SBOM",
  "error": "Operation cannot be fulfilled on sbomsyfts.sp
oftwarecomposition.kubescape.io \"ghcr.io-k8sstormcenter-
pp-sha256-e323014ec9befb76bc551f8cc3bf158120150e2e277bae1
c2da6c56c0a2b-6c0a2b\": the object has been modified; ple
apply your changes to the latest version and try again",
  "namespace": "webapp",
  "pod": "webapp-mywebapp-67965968bb-4t8cg",
  "container": "mywebapp-app",
  "sbomName": "ghcr.io-k8sstormcenter-webapp-sha256-e3230
```

User receives the BoB
Live demo

3 Abuse the Webapp manually

We keep the terminal with the logs open, and proceed to manually exploit our app.

> This app is intentionally vulnerable. Do not use in production under any circumstances.

```
> make fwd
```

```
> curl localhost:8080/ping.php?ip=172.16.0.2\;ls
```

In the logs tab, you should see the `anomaly alerts`

```
{"BaseRuntimeMetadata":{"alertName":"Unexpected process launched","argume
{"BaseRuntimeMetadata":{"alertName":"Unexpected file access","arguments":
```

▶ Inspect the BoB profile (aka ApplicationProfile)

4 Abuse the Webapp via an automated negative test

We can also create known undesired behavior and test that it is detecteable in a predictable and consistent way:

In the next unit, we ll supply such an `attack-test` as part of the `BoB`, here we do it via `Makefile`. In the `original tab`, please run

```
> make attack
```

In the `logs tab`, you may observe a very predictable set of anomalies

IDE | Explorer | dev-machine | cplane-01 | node- | dev-machine (2)

```
namespace webapp -l "app.kubernetes.io/name=mywebapp,app.kube
rnetes.io/instance=webapp" -o jsonpath="{.items[0].metadata.n
ame}") 8080:80 &
laborant@dev-machine:bobctl$ Forwarding from 127.0.0.1:8080 -
> 80
Forwarding from [::1]:8080 -> 80
^C
laborant@dev-machine:bobctl$ curl localhost:8080/ping.php?ip=
172.16.0.2\;ls
Handling connection for 8080
<pre><strong>Ping results for 172.16.0.2;ls:</strong><br>PING
 172.16.0.2 (172.16.0.2) 56(84) bytes of data.<br><span style
='color: #4caf50;'>64 bytes from 172.16.0.2: icmp_seq=1 ttl=6
4 time=0.106 ms</span><br><span style='color: #4caf50;'>64 by
tes from 172.16.0.2: icmp_seq=2 ttl=64 time=0.127 ms</span><b
r><span style='color: #4caf50;'>64 bytes from 172.16.0.2: icm
p_seq=3 ttl=64 time=0.128 ms</span><br><span style='color: #4
caf50;'>64 bytes from 172.16.0.2: icmp_seq=4 ttl=64 time=0.10
4 ms</span><br><br>--- 172.16.0.2 ping statistics ---<br>4 pa
ckets transmitted, 4 received, 0% packet loss, time 3063ms<br
>rtt min/avg/max/mdev = 0.104/0.116/0.128/0.011 ms<br>index.h
tml<br>ping.php<br></pre><strong>Return status:</strong> 0lab
orant@dev-machine:bobctl$ makmake attack
curl 127.0.0.1:8080/ping.php?ip=1.1.1.1\;ls
Handling connection for 8080
<pre><strong>Ping results for 1.1.1.1;ls:</strong><br>PING 1.
1.1.1 (1.1.1.1) 56(84) bytes of data.<br><span style='color:
#4caf50;'>64 bytes from 1.1.1.1: icmp_seq=1 ttl=55 time=5.80
ms</span><br><span style='color: #4caf50;'>64 bytes from 1.1.
1.1: icmp_seq=2 ttl=55 time=5.83 ms</span><br><span style='co
lor: #4caf50;'>64 bytes from 1.1.1.1: icmp_seq=3 ttl=55 time=
5.79 ms</span><br><span style='color: #4caf50;'>64 bytes from
 1.1.1.1: icmp_seq=4 ttl=55 time=5.81 ms</span><br><br>--- 1.
1.1.1 ping statistics ---<br>4 packets transmitted, 4 receive
d, 0% packet loss, time 3005ms<br>rtt min/avg/max/mdev = 5.78
9/5.805/5.829/0.014 ms<br>index.html<br>ping.php<br></pre><st
rong>Return status:</strong> 0curl  127.0.0.1:8080/ping.php?i
p=1.1.1.1%3Bcat%20/proc/self/mounts
Handling connection for 8080
<pre><strong>Ping results for 1.1.1.1;cat /proc/self/mounts:<
/strong><br>PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.<br><
span style='color: #4caf50;'>64 bytes from 1.1.1.1: icmp_seq=
1 ttl=55 time=5.79 ms</span><br><span style='color: #4caf50;'
>64 bytes from 1.1.1.1: icmp_seq=2 ttl=55 time=5.83 ms</span>
<br><span style='color: #4caf50;'>64 bytes from 1.1.1.1: icmp
_seq=3 ttl=55 time=5.80 ms</span><br><span style='color: #4ca
f50;'>64 bytes from 1.1.1.1: icmp_seq=4 ttl=55 time=5.81 ms</
span><br><br>--- 1.1.1.1 ping statistics ---<br>4 packets tra
nsmitted, 4 received, 0% packet loss, time 3005ms<br>rtt min/
avg/max/mdev = 5.789/5.809/5.831/0.015 ms<br>overlay / overla
y rw,relatime,lowerdir=/var/lib/rancher/k3s/agent/containerd/
```


```
    "podName": "webapp-mywebapp-67965968bb-4t8cg",
    "podNamespace": "webapp",
    "podLabels": {
      "app.kubernetes.io/instance": "webapp",
      "app.kubernetes.io/name": "mywebapp",
      "pod-template-hash": "67965968bb"
    },
    "workloadName": "webapp-mywebapp",
    "workloadNamespace": "webapp",
    "workloadKind": "Deployment"
  },
  "RuntimeProcessDetails": {
    "processTree": {
      "pid": 7511,
      "cmdline": "/bin/sh -c ping -c 4 1.1.1.1;cat index.htm
",
      "comm": "sh",
      "ppid": 6630,
      "pcomm": "apache2",
      "hardlink": "/bin/dash",
      "uid": 33,
      "gid": 33,
      "startTime": "0001-01-01T00:00:00Z",
      "upperLayer": false,
      "cwd": "/var/www/html",
      "path": "/bin/dash",
      "childrenMap": {
        "cat 7516": {
          "pid": 7516,
          "cmdline": "/bin/cat index.html",
          "comm": "cat",
          "ppid": 7511,
          "pcomm": "sh",
          "hardlink": "/bin/cat",
          "uid": 33,
          "gid": 33,
          "startTime": "0001-01-01T00:00:00Z",
          "upperLayer": false,
          "cwd": "/var/www/html",
          "path": "/bin/cat"
        }
      }
    },
    "containerID": "0b748fe5277a36f38b4af625f35cd8f390f7dd90
dfc853a72c861dde4ae99dc"
  },
  "event": {
    "runtime": {
      "runtimeName": "containerd",
      "containerId": "0b748fe5277a36f38b4af625f35cd8f390f7dd
00dfc853a72c861dde4ae99dc",
```


User receives the BoB
Live demo

▶ Inspect the BoB profile (aka ApplicationProfile)

4 Abuse the Webapp via an automated negative test

We can also create known undesired behavior and test that it is detecteable in a predictable and consistent way:

In the next unit, we ll supply such an `attack-test` as part of the `BoB`, here we do it via `Makefile`. In the `original tab`, please run

```
> make attack
```

In the `logs tab`, you may observe a very predictable set of anomalies


```
{"BaseRuntimeMetadata":{"alertName":"Unexpected process launched","argume
{"BaseRuntimeMetadata":{"alertName":"Unexpected file access","arguments":
{"BaseRuntimeMetadata":{"alertName":"Unexpected process launched","argume
{"BaseRuntimeMetadata":{"alertName":"Unexpected process launched","argume
{"BaseRuntimeMetadata":{"alertName":"Unexpected file access","arguments":
{"BaseRuntimeMetadata":{"alertName":"Unexpected system call","arguments":
{"BaseRuntimeMetadata":{"alertName":"Unexpected process launched","argume
{"BaseRuntimeMetadata":{"alertName":"Unexpected domain request","argument
{"BaseRuntimeMetadata":{"alertName":"Unexpected system call","arguments":
{"BaseRuntimeMetadata":{"alertName":"Unexpected system call","arguments":
{"BaseRuntimeMetadata":{"alertName":"Unexpected system call","arguments":
{"BaseRuntimeMetadata":{"alertName":"Unexpected process launched","argume
{"BaseRuntimeMetadata":{"alertName":"Unexpected file access","arguments":
{"BaseRuntimeMetadata":{"alertName":"Unexpected Service Account Token Acc
```


5 Mimic a supply chain attack

WIP: make this automated and pretty

We also have a tampered version of `webapp` and there you'll notice

IDE | Explorer | dev-machine | cplane-01 | node- | dev-machine (2)

```
orant@dev-machine:bobctl$ make attack
curl 127.0.0.1:8080/ping.php?ip=1.1.1.1\;ls
Handling connection for 8080
<pre><strong>Ping results for 1.1.1.1;ls:</strong><br>PING 1.
1.1.1 (1.1.1.1) 56(84) bytes of data.<br><span style='color:
#4caf50;'>64 bytes from 1.1.1.1: icmp_seq=1 ttl=55 time=5.80
ms</span><br><span style='color: #4caf50;'>64 bytes from 1.1.
1.1: icmp_seq=2 ttl=55 time=5.83 ms</span><br><span style='co
lor: #4caf50;'>64 bytes from 1.1.1.1: icmp_seq=3 ttl=55 time=
5.79 ms</span><br><span style='color: #4caf50;'>64 bytes from
 1.1.1.1: icmp_seq=4 ttl=55 time=5.81 ms</span><br><br>--- 1.
1.1.1 ping statistics ---<br>4 packets transmitted, 4 receive
d, 0% packet loss, time 3005ms<br>rtt min/avg/max/mdev = 5.78
9/5.805/5.829/0.014 ms<br>index.html<br>ping.php<br></pre><st
rong>Return status:</strong> 0curl  127.0.0.1:8080/ping.php?i
p=1.1.1.1%3Bcat%20/proc/self/mounts
Handling connection for 8080
<pre><strong>Ping results for 1.1.1.1;cat /proc/self/mounts:<
/strong><br>PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.<br><
span style='color: #4caf50;'>64 bytes from 1.1.1.1: icmp_seq=
1 ttl=55 time=5.79 ms</span><br><span style='color: #4caf50;'
>64 bytes from 1.1.1.1: icmp_seq=2 ttl=55 time=5.83 ms</span>
<br><span style='color: #4caf50;'>64 bytes from 1.1.1.1: icmp
_seq=3 ttl=55 time=5.80 ms</span><br><span style='color: #4ca
f50;'>64 bytes from 1.1.1.1: icmp_seq=4 ttl=55 time=5.81 ms</
span><br><br>--- 1.1.1.1 ping statistics ---<br>4 packets tra
nsmitted, 4 received, 0% packet loss, time 3005ms<br>rtt min/
avg/max/mdev = 5.789/5.809/5.831/0.015 ms<br>overlay / overla
y rw,relatime,lowerdir=/var/lib/rancher/k3s/agent/containerd/
io.containerd.snapshotter.v1.overlayfs/snapshots/101/fs:/var/
lib/rancher/k3s/agent/containerd/io.containerd.snapshotter.v1
.overlayfs/snapshots/100/fs:/var/lib/rancher/k3s/agent/contai
nerd/io.containerd.snapshotter.v1.overlayfs/snapshots/99/fs:/
var/lib/rancher/k3s/agent/containerd/io.containerd.snapshotte
r.v1.overlayfs/snapshots/98/fs:/var/lib/rancher/k3s/agent/con
tainerd/io.containerd.snapshotter.v1.overlayfs/snapshots/97/f
s:/var/lib/rancher/k3s/agent/containerd/io.containerd.snapsho
tter.v1.overlayfs/snapshots/96/fs:/var/lib/rancher/k3s/agent/
containerd/io.containerd.snapshotter.v1.overlayfs/snapshots/9
5/fs:/var/lib/rancher/k3s/agent/containerd/io.containerd.snap
shotter.v1.overlayfs/snapshots/94/fs:/var/lib/rancher/k3s/age
nt/containerd/io.containerd.snapshotter.v1.overlayfs/snapshot
s/93/fs:/var/lib/rancher/k3s/agent/containerd/io.containerd.s
napshotter.v1.overlayfs/snapshots/92/fs:/var/lib/rancher/k3s/
agent/containerd/io.containerd.snapshotter.v1.overlayfs/snaps
hots/91/fs:/var/lib/rancher/k3s/agent/containerd/io.containe
r.snapshotter.v1.overlayfs/snapshots/90/fs:/var/lib/rancher/k
3s/agent/containerd/io.containerd.snapshotter.v1.overlayfs/sn
apshots/89/fs:/var/lib/rancher/k3s/agent/containerd/io.contai
nerd.snapshotter.v1.overlayfs/snapshots/88/fs:/var/lib/ranche
r/k3s/agent/containerd/io.containerd.snapshotter.v1.overlayfs
```


```
      "containerId": "0b748fe5277a36f38b4af625f35cd8f390f7dd5
00dfc853a72c861dde4ae99dc",
      "containerName": "mywebapp-app",
      "containerImageName": "ghcr.io/k8sstormcenter/webapp@sh
a256:e323014ec9befb76bc551f8cc3bf158120150e2e277bae11844c2da
c56c0a2b",
      "containerImageDigest": "sha256:c622cf306b94e8a6e7cfd71
8f048015e033614170f19228d8beee23a0ccc57bb"
    },
    "k8s": {
      "namespace": "webapp",
      "podName": "webapp-mywebapp-67965968bb-4t8cg",
      "podLabels": {
        "app.kubernetes.io/instance": "webapp",
        "app.kubernetes.io/name": "mywebapp",
        "pod-template-hash": "67965968bb"
      },
      "containerName": "mywebapp-app",
      "owner": {}
    },
    "timestamp": 1752990476105800465,
    "type": "normal"
  },
  "level": "error",
  "message": "Unexpected domain communication: github.com. fr
om: mywebapp-app",
  "msg": "Unexpected domain request",
  "time": "2025-07-20T05:47:56Z"
}
{
  "BaseRuntimeMetadata": {
    "alertName": "Unexpected system call",
    "arguments": {
      "syscall": "getpeername"
    },
    "infectedPID": 6602,
    "md5Hash": "4e79f11b07df8f72e945e0e3b3587177",
    "sha1Hash": "b361a04dcb3086d0ecf960d3acaa776c62f03a55",
    "severity": 1,
    "size": "730 kB",
    "timestamp": "2025-07-20T05:47:57.688648007Z",
    "trace": {},
    "uniqueID": "69bbdde26311ca1a112c3449cc03d209",
    "profileMetadata": {
      "status": "completed",
      "completion": "complete",
      "name": "replicaset-webapp-mywebapp-67965968bb",
      "failOnProfile": true,
      "type": 0
    }
  },
```


SBoB makes behavior prescriptive

No longer descriptive

Ask your primary care engineer
in case of unexpected side-channels

Tracing has been possible since the dawn of time: but

Instead of everyone recording profiles (potentially incl attacker behavior)

Explicit positives! Makes a DB look like a DB (and a debug pkg like a debug pkg)

Explicit negatives! If negative tests are supplied, users can verify their incident response

some room: to make it transfer well and allow good UX

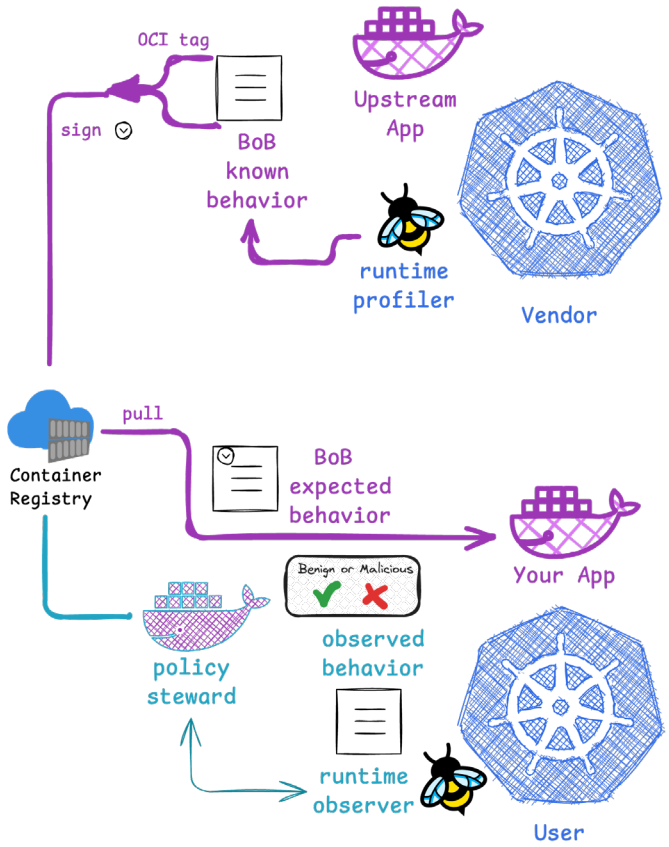
WANTED:
CNCF projects
with their test-suite

needs YOU !

k8sstormcenter

Roadmap

Edge cases, caveats



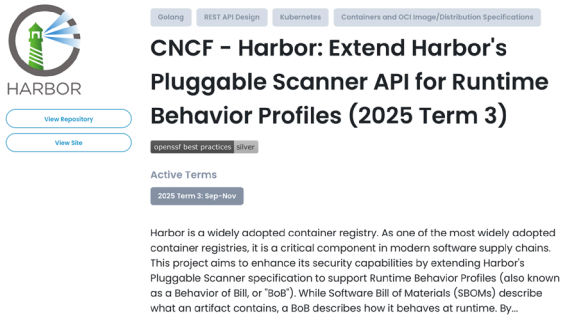
Research-Collaborations welcome!

Research funding requested for

- Additivity (e.g. various Sidecars like istio) ☑
- Large Parameter Studies ✅ TU WIEN TECHNISCHE UNIVERSITÄT WIEN
- UX Evaluation and Implementation
- Optimal Granularity 👨‍🎓

  (Performance vs Alert Fatigue)
- Integration into cloudnative ecosystem
- Maintenance

Ecosystem Integration Harbor



- SBOB integration into Harbor
- Attaching SBOB information alongside the OCI artefact
- Extending Harbor interface to 3rd party scanners
    - UI and reporting
    - Interfaces and API
- SBOM, SBOB, etc to represent different aspect of an application (see Open Component Model OCM)
- Implementation as part of the LFX mentorship program

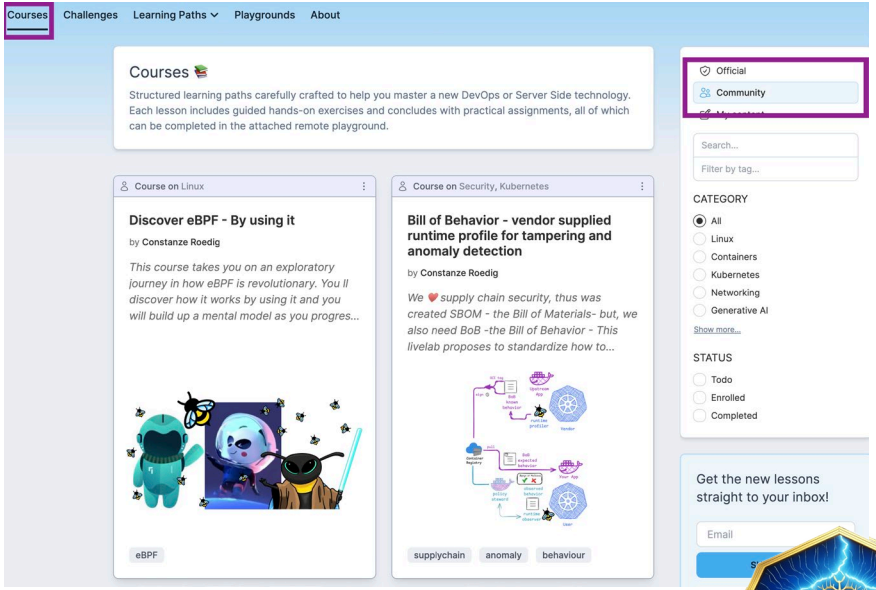
How you can benefit today
Don’t believe it … try out the lab

CNCF SW with test-suite WANTED

Bill of Behavior - vendor supplied runtime profile for tampering and anomaly detection

supplychain anomaly behaviour oci ebpf

We ❤️ supply chain security, thus was created SBOM - the Bill of Materials- but, we also need BoB -the Bill of Behavior - This livelab proposes to standardize how to record a benign behavior profile, extract, sign and publish it, for users to consume, ingest, verify it and detect attacks and tampering. For software-vendors, a BoB creates trust and transparency, For users, it makes anomaly detection realistically achievable.



https://labs.iximiuz.com/courses/bill-of-behaviour-c070da3a

Summary

Ask your primary care engineer
in case of unexpected effects

Container Beipackzettel

BoB: Software Bill of Behavior

What? generalized runtime profile encoding benign syscalls, paths, network, capas, execs

Who? Vendors attach to software at “the factory”

Why? Allows users to inherit maintained, up-to-date rules for anomaly detection

so that: Users stop play catch-up

CLOUD NATIVE DENMARK
October 7-8, 2025 • Aarhus

CNDA
Cloud Native Days Austria

ITSECURITY SUMMIT

Thank you !!

SOFIA | BULGARIA
KCD
GROWING CLOUD NATIVE TOGETHER

Redpill Linpro
Cloud

iximiuz Labs

SBoB would not have been possible without: Matthias, Ben, Thomas, Norman, Stefan, Peter, Markus, David, Pepijn, Mario, Ivan, Markus, Stephie, Andi, Ernst, Michi, Reini and Alois

https://cloudnativecompass.dev

https://www.letsboot.ch

https://kubescape.io

https://github.com/k8sstormcenter/bobctl

⭐ Star it if you want it ⭐