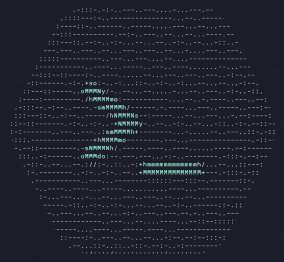
>_cmd

DETECTION IN LINUX CONTAINERS:

MITRE ATT&CK IN CONTAINER ENVIRONMENTS & UPDATES TO CUSTOMER ADOPTION



Martin Bowyer &
Jake King

Agenda_

- Deploying **MITRE ATT&CK Detection policies** in Production.
- Your feedback from the last session - **gaps in Container Observability**
- Applying **MITRE ATT&CK & OSS tooling Kubernetes environments**

Gathering better data for TTP's on Linux systems

- **We need data -** Execution logs and detailed process / filesystem information
- **We need context -** User, Session, parent / child process information, prevalence, enrichment
- **We need Retention -** Over time, and across environments
- **We must have control -** Whitelists aren't super easy & remediation is post-breach

Gathering better data for TTP's on Linux systems

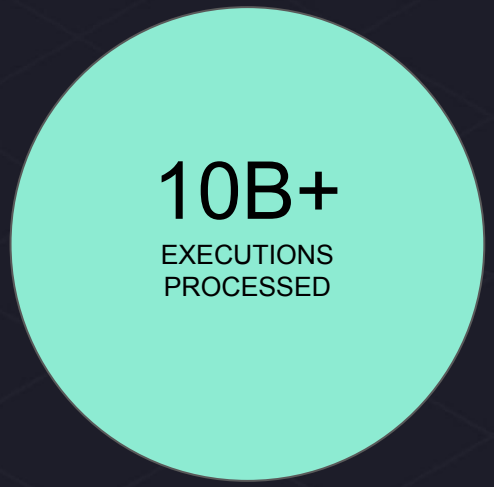
- **We need data -** Execution logs and detailed process / filesystem information
- **We need context -** User, Session, parent / child process information, prevalence, enrichment
- **We need Retention -** Over time, and across environments
- **We must have control -** Whitelists aren't super easy & remediation is post-breach

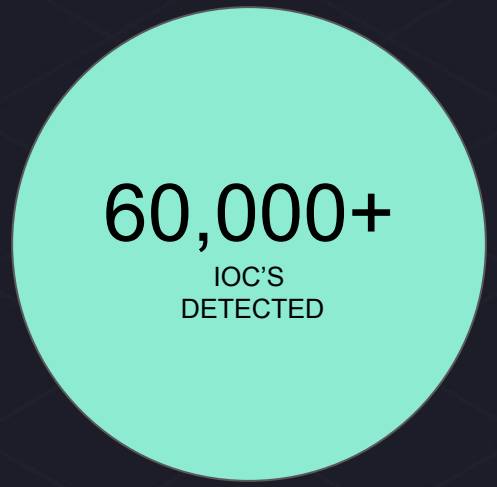
Observation

What tools are used for Linux?

- **OS Logs:** AuditD, Syslog, eBPF
  - Augmented alerting via Auditd Policies
- **FOSS Sensors:** ES-Beats, ProcSpy, ExecSnoop, SSH-Bastion
- **Vendors Sensors:** EDR / HIDS / PAM tools
- **Vendor / FOSS SIEM:** Splunk, Sumo, ELK
- **Defense / Remediation Tools:** Ansible, Chef, Puppet, LimaCharlie

MITRE ATT&CK for Linux_





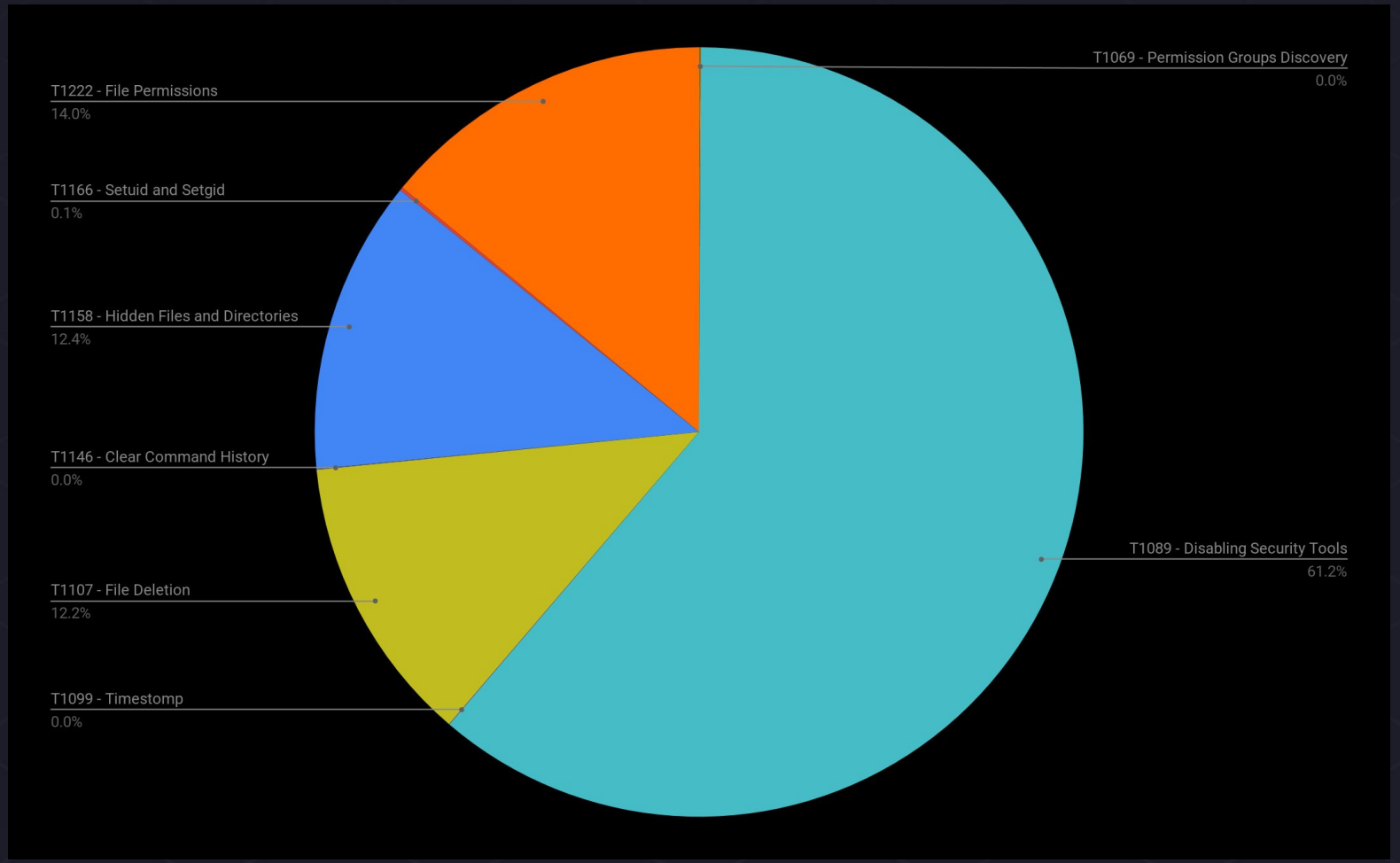
One Caveat_

Media & Streaming

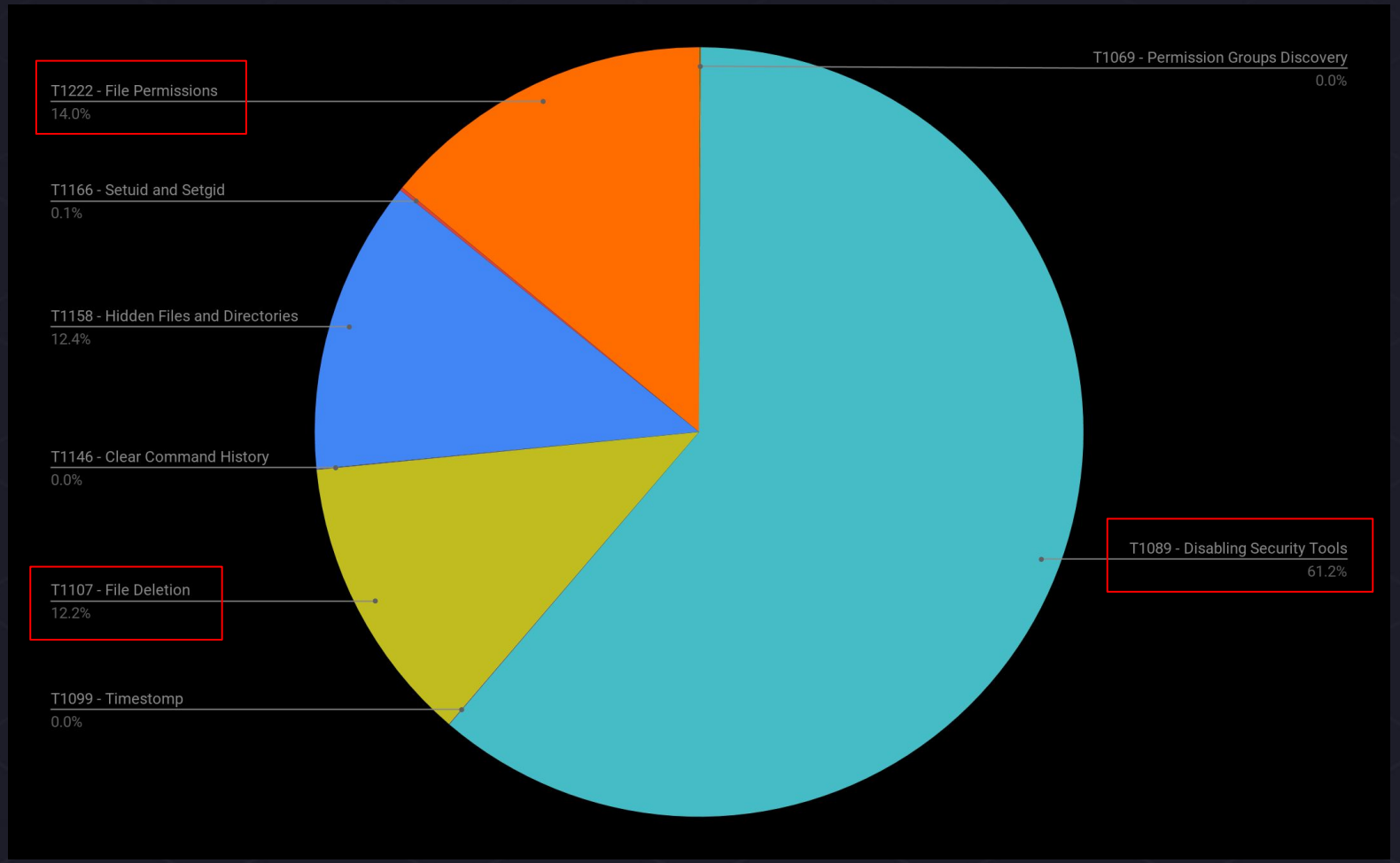
Banking

Ecommerce

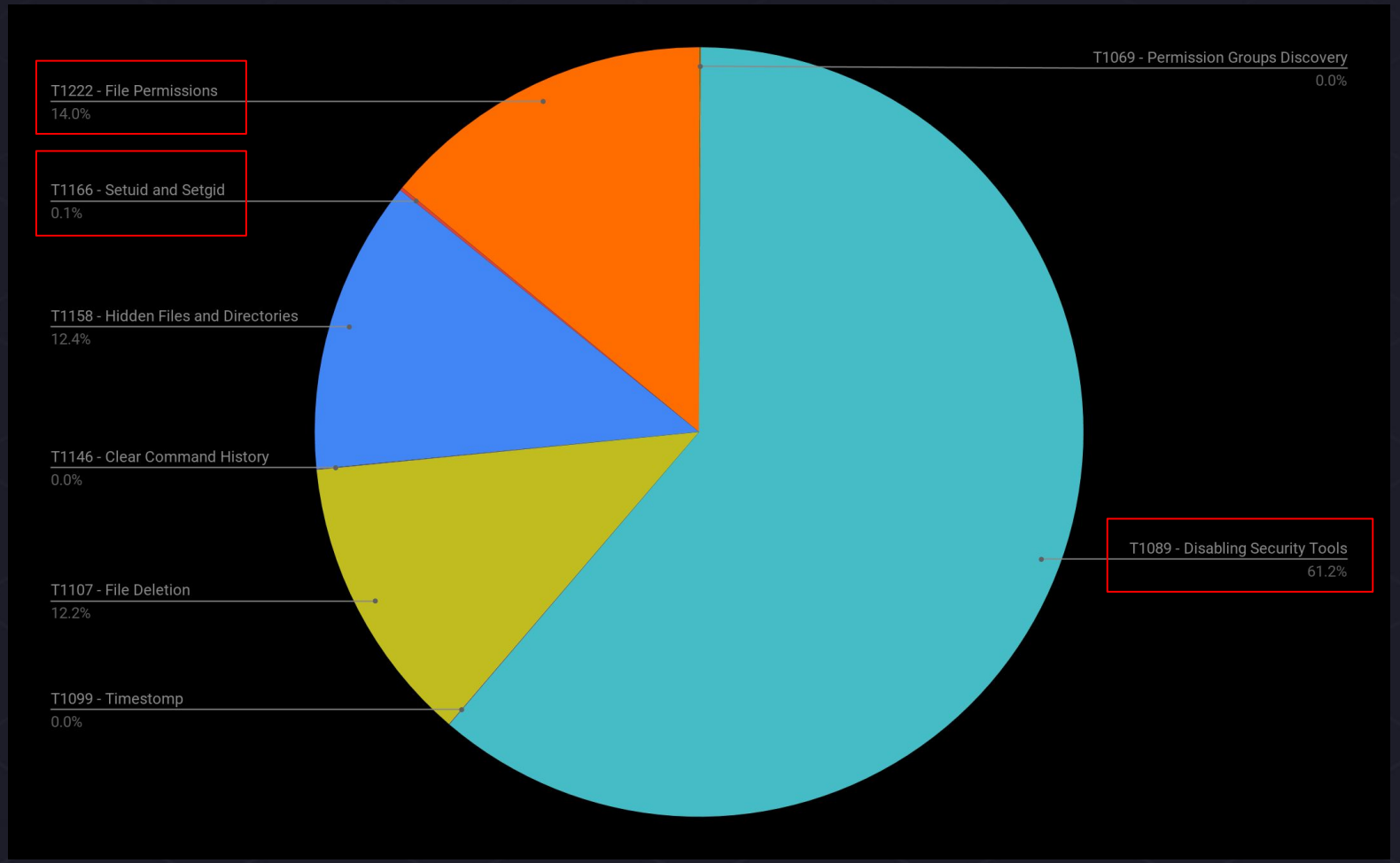
Some Results from Production_



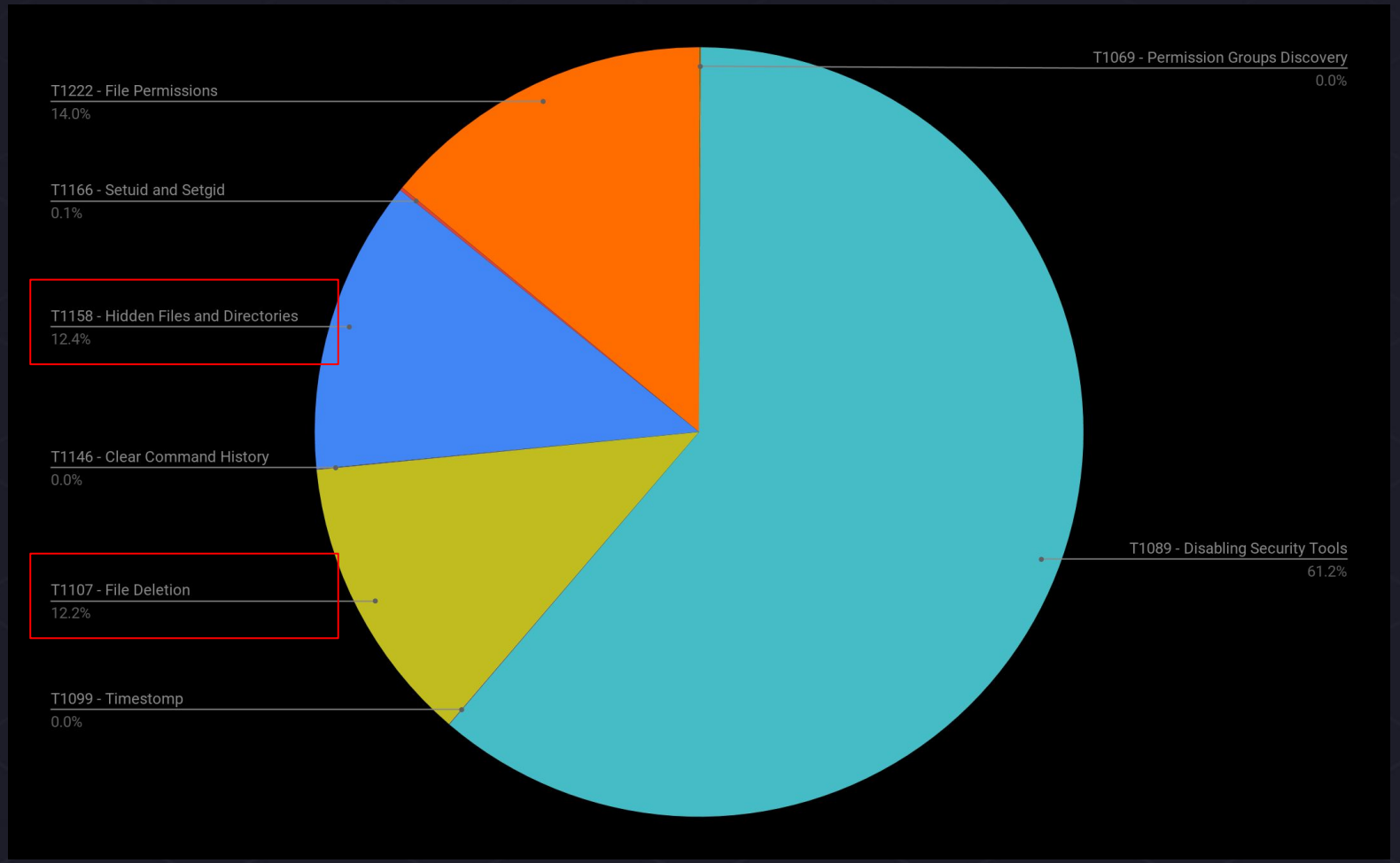
Some Results from Production_



Some Results from Production_



Some Results from Production_



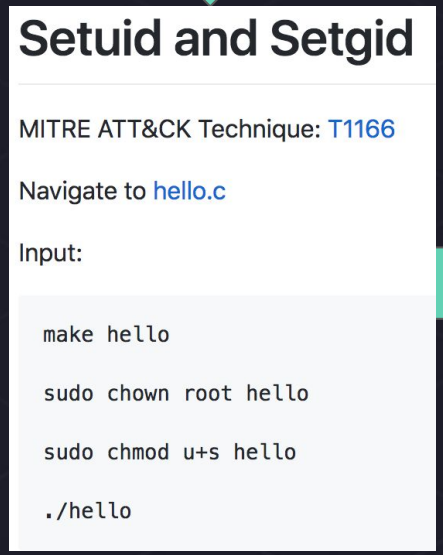
Not...Sightings?

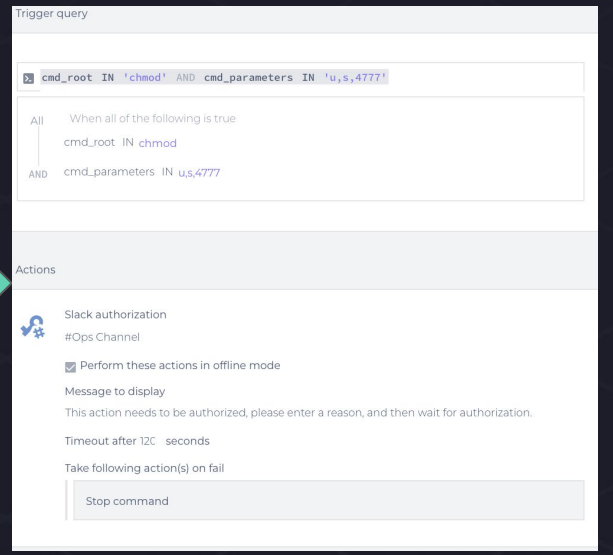
How do we create Policies?

Developing **policies via Atomic Red Team** - Still working well, but we want to translate / share in SIGMA.

Work to be done to polish rules into effectively for Containers / Kube

40+ Policies created, **but can be noisy without custom profiles / Prioritization**





Feedback from our last Session_

*“What about visibility into **Kubernetes**?”*

*“Visibility into **containers**?”*

A gap in Observability: Containers

- OSS Offerings are limited, and hard to deploy for Containers without the right knowledge
- Not as low-touch as you'd assume - Constantly touched / interacted with
- Common abuses
  - Insider Threats
  - API key compromise (Kubectl)
  - Service/image Compromise*

* Check out the talk from Hack.lu on Containing containers.

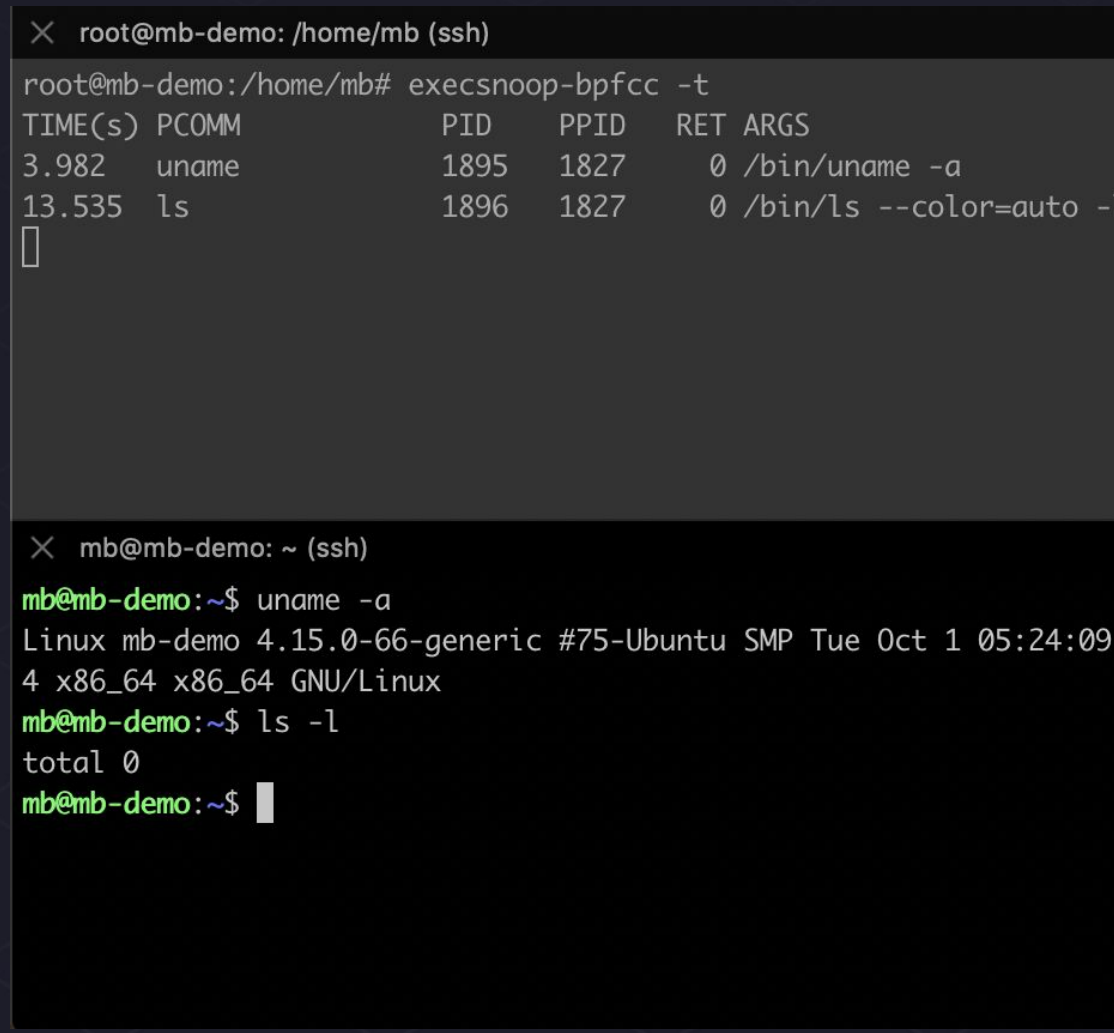
Observability with eBPF

What is eBPF?

- Bytecode interpreter in the Linux kernel
- Traditionally used for packet filtering
- Extended to other kernel objects
  - kprobes, uprobes, tracepoints

BPF Compiler Collection

- **execsnoop, opensnoop,** many more!
- **github.com/iovisor/bcc**



Monitoring a Kubernetes Cluster

- Package eBPF programs in a container image
- Deploy container image to cluster as a Daemonset
  - Single instance per cluster node
  - Automatically scales with the cluster
- Map data collected to ATT&CK leveraging either open signatures and/or custom signatures

```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  labels:
    k8s-app: cmd-bcc
    kubernetes.io/cluster-service: "true"
  name: cmd-bcc
spec:
  selector:
    matchLabels:
      k8s-app: cmd-bcc
      kubernetes.io/cluster-service: "true"
  template:
    metadata:
      annotations:
        scheduler.alpha.kubernetes.io/critical-pod: ""
        scheduler.alpha.kubernetes.io/tolerations: '[{"key":"dedicated
      labels:
        k8s-app: cmd-bcc
        kubernetes.io/cluster-service: "true"
    spec:
      containers:
      - name: cmd-bcc
        image: cmdinc/bcc:latest
        command: ["bash"]
        args: ["/scripts/bcc_setup.sh"]
```

Demo

Applying MITRE ATT&CK to eBPF Data_

- Standard hardening / minimization will defend against numerous attacks, but not all
- eBPF data can be used to capture syscalls being made with **Atomic Red Team** to build effective preventative rules
- Policies can then be created to whitelist syscalls at the point of a binary execution with **SecComp / SELinux / AuditD**
- Kernel 5.0 brings some of these features to user space for awesome future projects!

>_cmd

THANKYOU!

Martin Bowyer &
Jake King
cmd.com