# 企业应用容器化的攻与防

腾讯蓝军 {neargle}

# About Neargle

**腾讯安全平台部 安全工程师**

**安全开源组织OpenSec成员**

blog.neargle.com

weib.com/neargle

github.com/neargle

# What did we do?

# 蓝军:这叫我怎么打?

## 蓝军(RedTeam)的目的？

悄悄的/什么可以做/什么不能做

# RedTeam:容器网络的乐与哀

1. Rename parameter (e.g. `router-mac -> hahaha`?)
2. Build with another name (e.g. `nmap/masscan -> email`?)
3. Just Run:

```
> {masscan} -e eth1 --{router-mac} XX-XX-XX-XX-XX-XX -p 1-65535 --rate 10000 -iL tencent.force -oX out.xml
Starting masscan 1.0.4 (http://bit.ly/14GZzcT) at 20XX-XX-XX XX:XX:49 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [0000000 ports/host]
...
```

# Detection of Port Scanning


1. Necessity and importance of Detection
2. Worm/Malware Spreads/Scanner/SSRF...
3. Finding and finding local files(log/conf...) is no longer safe o(╥﹏╥)o
4. Cmdline(string/regex base) + Network Event

# RedTeam:容器网络的[乐]与哀

```
nc -llnvp 8888
```

Net NameSpace.1
net:[4026531993]

Net NameSpace.2
net:[4026533024]

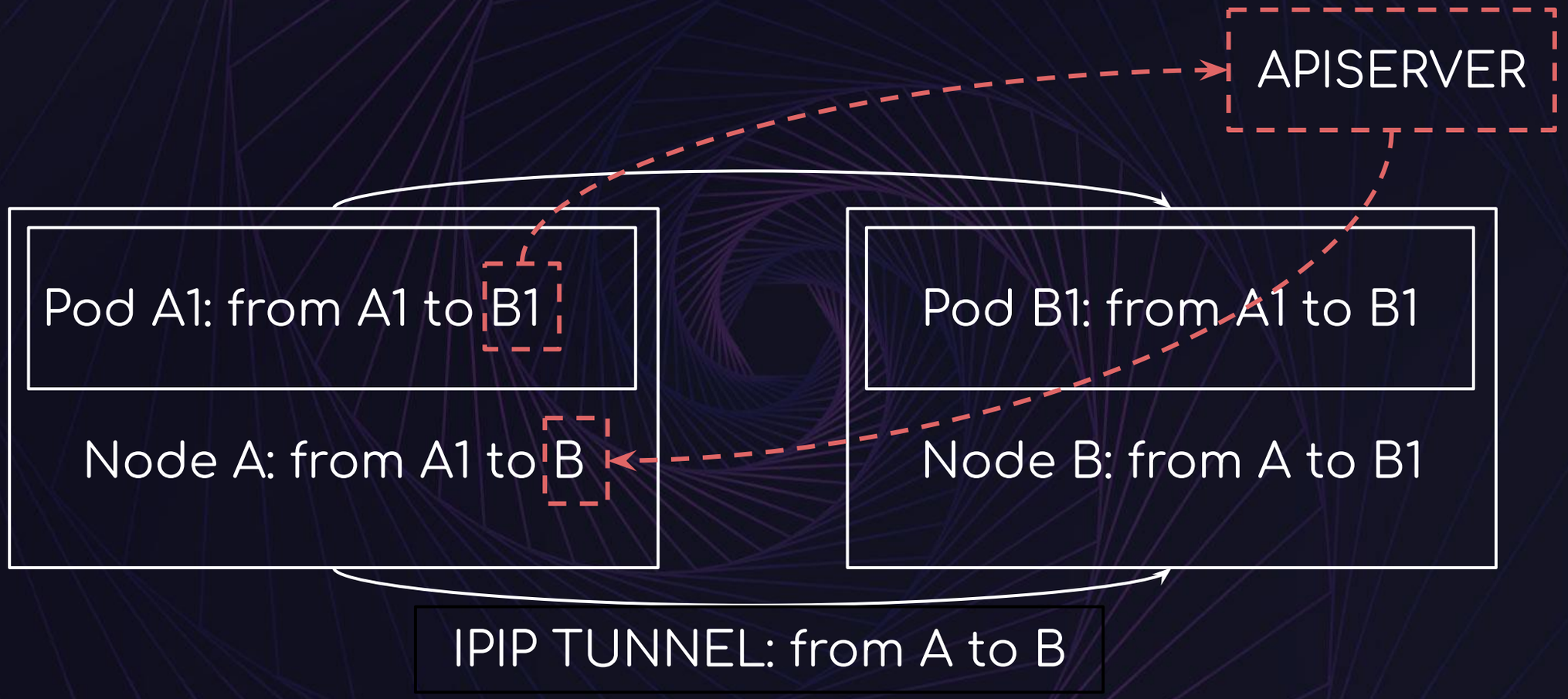
```
curl net1_ip:8888 -v
```

```
ls -l /var/run/netns
total 0
lrwxrwxrwx 1 root root 17 Dec  1 23:54 6723 -> /proc/6875/ns/net
-r--r--r-- 1 root root  0 Nov 12 19:53 ns1
-r--r--r-- 1 root root  0 Nov 12 19:53 ns2

find . -name "net" | grep "ns/net" | grep -v "task" | xargs /bin/ls -l
lrwxrwxrwx 1 root   root  0 Dec  2 00:34 ./11801/ns/net -> net:[4026532732]
lrwxrwxrwx 1 root   root  0 Nov  8 11:13 ./15946/ns/net -> net:[4026532433]
```

1. Process/thread
2. Veth Pair not in host#ifconfig
3. libpcap cry
4. unnecessary

# RedTeam:容器网络的[乐]与哀



# RedTeam:容器网络的[乐]与[哀]

1. Calico(Istio)/Flannel/Canal/Weave
2. Data packet may be encrypted
3. Never run security agents on every container
   (single-process-per-container/OPS???/ReDev)
4. Kube-proxy is similar but more simple

# RedTeam:容器网络的[乐]与[哀]

# 容器进程

## Container Process

```
\_ containerd-shim -namespace moby -workdir /var/lib/containerd/io.containerd.runtime.v1.linux/moby/0x0x0x
|   \_ /bin/sh -c while true; do sleep 10000; done
|   |   \_ sleep 10000
|   \_ bash
|       \_ python -m SimpleHTTPServer 8080
```

1. A normal process in linux
2. MNT/PID/NET/IPC/USER/UTS Namespace & Cgroup
3. Image/LogPath&Log/ContainerName/NetworkMode/ Privileged/Memory/CpuShares/Mounts/NetworkSettings

# 容器文件: Web目录文件监控

```
> /proc/14185/exe -V
configure arguments: --prefix=/etc/nginx --sbin-path=/usr/sbin/nginx
--conf-path=/etc/nginx/nginx.conf --error-log-path=/var/log/nginx/error.log

> ls -l /etc/nginx/nginx.conf
ls: cannot access '/etc/nginx/nginx.conf': No such file or directory
```

```
> md5sum /proc/14185/root/usr/share/nginx/html/neargle.php
> md5sum /var/lib/docker/overlay2/{LayerID}/diff/usr/share/nginx/html/neargle.php
> docker exec -it {ContainerID} md5sum /usr/share/nginx/html/neargle.php
```

d41d8cd98f00b204e9800998ecf8427e

# 容器文件: 文件监控

| | |
|---|---|
| /data/docker/aufs | AUFS |
| /data/docker/btrfs | Btrfs |
| /data/docker/devicemapper | Device mapper |
| /data/docker/overlay2 | OverlayFS |
| /data/docker/zfs | ZFS |
| /data/docker/vfs | VFS |

```
/data/docker/aufs/mnt/{LayerID}/usr/local/app/nginx/
    return path.Join(devices.root, "devicemapper")
```

# Container 2 PC

Container

```
> kubectl cp <namespace>/<pod>:/anyfile /tmp/safe
```

```go
func (o *CopyOptions) copyFromPod(src, dest fileSpec) error {
    ....
    options := &exec.ExecOptions{
        StreamOptions: exec.StreamOptions{
            ...
            Namespace: src.PodNamespace,
            PodName:   src.PodName,
        },
        Command:  []string{"tar", "cf", "-", src.File},
        Executor: &exec.DefaultRemoteExecutor{},
    }
    go func() {
        defer outStream.Close()
        err := o.execute(options)
        cmdutil.CheckErr(err)
    }()
    ...
    prefix = stripPathShortcuts(prefix)
    return o.untarAll(src, reader, dest.File, prefix)
}
```

# 其他

```
...
   "auths": {
        "hubbak.neargle.com": {
            "auth":
"bmVhcmdsZTpkZW5naHVpZXJ5b3Vob25nYmFv
",
            "email": ""
...
```

1. ~/.docker/config.json & ~/.kube/config
2. imageRepository (Self-built)
3. unix:///var/run/docker.sock & WebConsole & Docker in Docker

# THANKS