利用 Docker 搭建 Vulhub 靶场

一、利用 V2rayN 解决 Docker 镜像下载问题（注：配置之前需要自行购买科学上网节点）

1、下载客户端

访问下方链接下载 V2rayN 客户端并完成安装，安装后从桌面打开或在目录中打开 **v2rayN.exe**

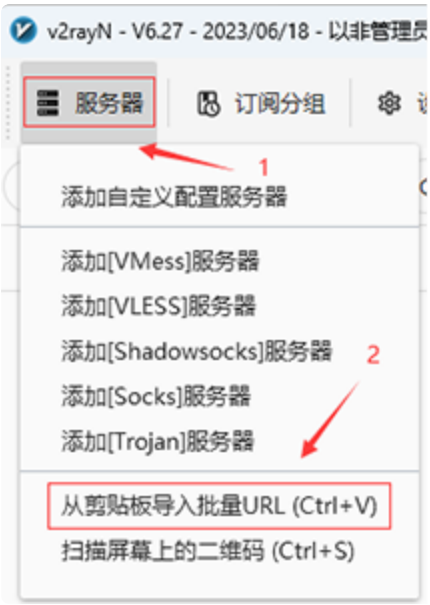
```
链接：https://pan.quark.cn/s/97f48d9395fc
提取码：UaGu
```

2、设置订阅地址

从科学上网供应商处获得 V2ray 订阅后打开 V2rayN 客户端

点击 **服务器 **，然后点击 **从剪贴板导入批量URL（Ctrl+V）**

点击 **订阅分组**，然后点击 **更新全部订阅（通过代理）**可以得到节点信息，剩余流量与过期时间



如果弹出配置错误提示，请重启客户端并重新操作一遍

3、选择节点

在软件主界面右击需要的节点， 点击 **设为活动服务器（Enter）**可自己根据使用情况选择需要的节点

4、开启局域网连接

点击 **设置**，然后点击 **参数设置**

在新出现的页面中点击 **允许来自局域网的连接** 后方的按钮，将该功能打开

5、安装docker

(1) 安装一些必要的依赖

```Bash
sudo yum install -y yum-utils device-mapper-persistent-data lvm2
```

(2) 设置镜像仓库（阿里云仓库）：

```Bash
sudo yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
```

(3) 更新 yum 软件包索引

```bash
sudo yum makecache fast
```

（4）安装 Docker CE 最新版本：

```bash
sudo yum install -y docker-ce
```

（5）启动docker

```bash
sudo systemctl start docker #启动docker
sudo systemctl restart docker #重启docker
sudo systemctl enable docker #加入开机启动docker
```

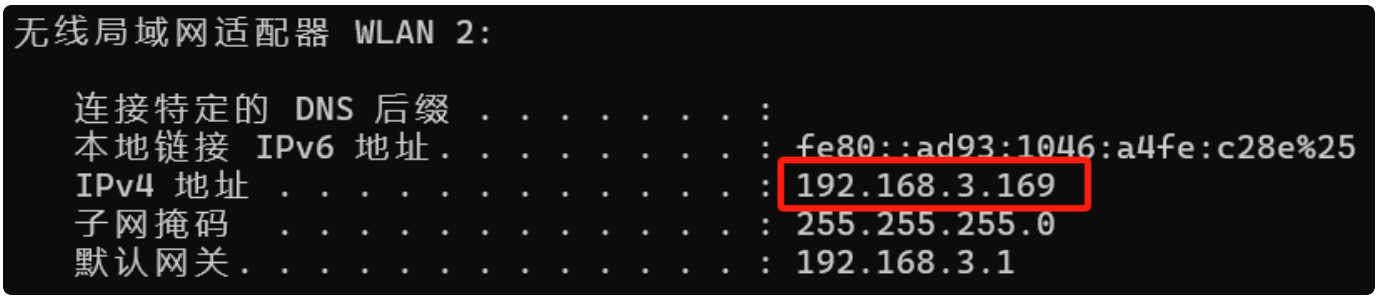
6、设置docker

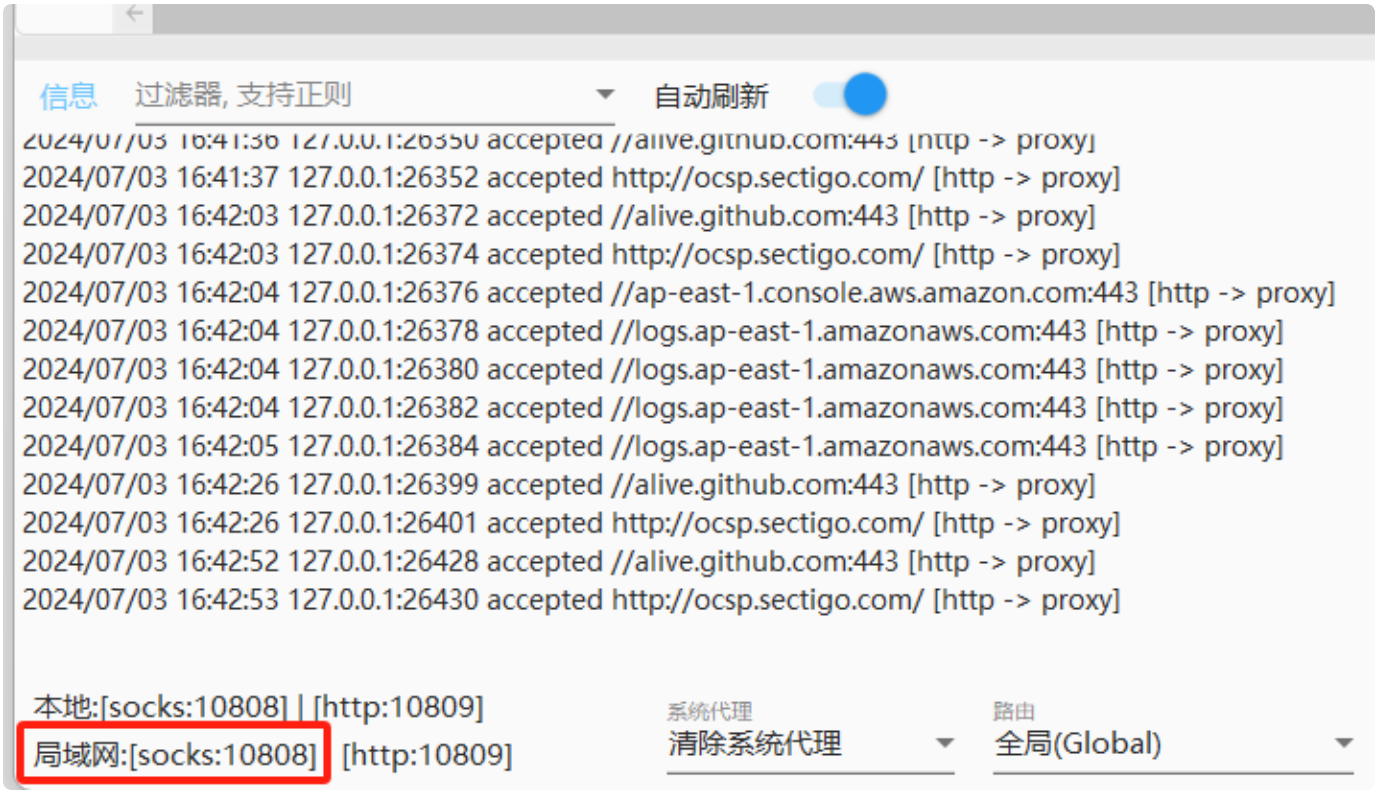
（1）执行下列命令创建配置目录和文件

```bash
sudo mkdir -p /etc/systemd/system/docker.service.d
sudo touch /etc/systemd/system/docker.service.d/proxy.conf
```

```
[root@localhost ~]# sudo mkdir -p /etc/systemd/system/docker.service.d
[root@localhost ~]# sudo touch /etc/systemd/system/docker.service.d/proxy.conf
[root@localhost ~]#
```

（2）查看并记录安装了 V2rayN 软件的电脑的 IP（如果虚拟机是 nat 模式，那么这里要查看并记录的 IP 为VMware Network Adapter VMnet8 网卡的 IP），以及 V2rayN 中的局域网连接时的端口号





（3）将下列内容添加到刚才创建的 **proxy.conf** 文件中，**代理 IP** 和 **代理端口号** 填刚才记录的 IP 和端口即可。

```
[Service]
Environment="HTTP_PROXY=socks5://代理IP:代理端口号"
Environment="HTTPS_PROXY=socks5://代理IP:代理端口号"
Environment="NO_PROXY=localhost,127.0.0.1,.example.com"
```

```
[Service]
Environment="HTTP_PROXY=socks5://192.168.3.169:10808"
Environment="HTTPS_PROXY=socks5://192.168.3.169:10808"
Environment="NO_PROXY=localhost,127.0.0.1,.example.com"
```

（4）执行下列命令重载 systemd 并重启 docker

```
sudo systemctl daemon-reload
sudo systemctl restart docker
```

（5）此时就可以愉快的使用 docker 了

二、Docker 的使用

练习使用 docker 搭建 sqli-labs 靶场

（1）在终端输入下列命令查找 sqli-labs 镜像

```bash
docker search sqli-labs
```

```
[root@localhost ~]# docker search sqli-labs
NAME                    DESCRIPTION                                     STARS     OFFICIAL
c0ny1/sqli-labs         sqli-labs是一个sql注入的练习靶机，项目地址为…   10
monstertsl/sqli-labs    sqli-labs靶场镜像                               0
745184472/sqli-labs     sqli-labs是一个注入sql地址的练习靶机，项目为…   0
acgpiano/sqli-labs      sql injection labs                              39
tavenli/sqli-labs       靶机 sqli-labs                                  0
cuer/sqli-labs          sql injection labs ，sqli-labs项目地址为：ht…   0
vulfocus/sqli-labs                                                      0
howhacker/sqli-labs     sqli-labs靶场                                   0
hulb/sqli-labs          sql 注入练习靶场                                0
```

（2）使用下列命令拉取镜像

```bash
docker pull 镜像名称
```

```
[root@localhost ~]# docker pull acgpiano/sqli-labs
Using default tag: latest
latest: Pulling from acgpiano/sqli-labs
10e38e0bc63a: Extracting [==========================================>        ]  57.38MB/67.2MB
0ae7230b55bc: Download complete
fd1884d29eba: Download complete
4f4fb700ef54: Download complete
2a1b74a434c3: Downloading [==========>                                        ]  17.68MB/84.66MB
fb846398c5b7: Download complete
9b56a3aae7bc: Download complete
1dca99172123: Waiting
1a57c2088e59: Waiting
b3f593c73141: Waiting
d6ab91bda113: Waiting
d18c99b32885: Waiting
b2e4d0e62d16: Waiting
91b5c99fef87: Waiting
bf0fd25b73be: Waiting
b2824e2cd9b8: Waiting
97179df0aa33: Waiting
```

（3）拉取完成后使用下列命令启动容器

```bash
docker run -dit --name sqli-labs -p 8088:80 --rm acgpiano/sqli-labs

参数说明:
run: 表示在Docker容器中运行一个容器
-d: 后台运行容器，并返回容器ID
-i: 以交互模式运行容器，通常与 -t 同时使用
-t: 为容器重新分配一个伪输入终端，通常与 -i 同时使用
-p: 指定端口映射，格式为: 主机(宿主)端口:容器端口
--name: 为容器指定一个名称;
--rm: 当容器退出时自动删除容器。这可以确保不会留下无用的容器占用资源
```

```
[root@localhost ~]# docker run -dit --name sqli-labs -p 8088:80 --rm acgpiano/sqli-labs
07d8e785819158e59ce90af14efe83f069f353a5c7da8ad7cb2b494b70e198c2
```

（4）在浏览器访问 http://IP地址:8088 即可访问到启动的容器

（5）容器相关命令

```bash
docker ps -s #查看当前正在运行的容器
docker ps -a #查看已经创建的容器
docker start test #启动容器名为 test 的容器
docker stop test #停止容器名为 test 的容器
docker rm test #删除容器名为 test 的容器
docker rename old_name new_name #重命名一个容器
docker exec -it [容器名/id] bash #进入到docker容器
  -it 交互式终端
  bash 运行shell程序
exit #退出容器
docker rmi 镜像id #删除镜像
```

三、搭建vulhub

Vulhub是一个面向大众的开源漏洞靶场，无需docker知识，简单执行两条命令即可运行一个完整的漏洞靶场镜像。旨在让漏洞复现变得更加简单，让安全研究者更加专注于漏洞原理本身。

1、 下载vulhub 的源码

（1）切换到根目录

```bash
cd /
```

（2）下载靶场源码文件

```bash
wget https://github.com/vulhub/vulhub/archive/master.zip
```

（3）解压源码文件

```bash
unzip master.zip
```

2、安装 dockers compose

（1）切换到 docker-compose 的安装目录

```Bash
cd  /usr/local/bin
```

（2）安装docker-compose

```Bash
curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose && chmod +x /usr/local/bin/docker-compose
```

（3）查看docker-compose的版本

```Bash
docker-compose -version
```

（4）启动漏洞环境时进入到相应的漏洞目录，然后执行下方命令即可

```Bash
docker-compose up -d
```