实验 1：利用入侵检测系统检测 ICMP 恶意扫描

环境配置

1. 安装容器 Docker

容器：容器是软件的可执行单元，它采用通用方式封装了应用程序代码及其库和依赖项，因此可以随时随地运行容器（无论是在桌面、传统 IT 还是云端）。

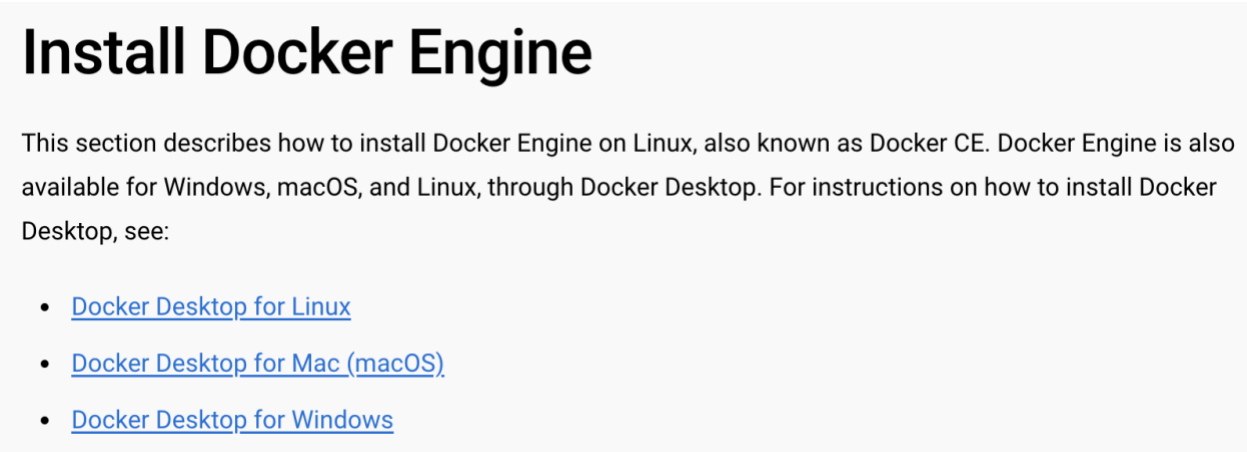
为此，容器利用一种操作系统 (OS) 虚拟化的形式，进而可以利用操作系统内核的功能（例如 Linux 命名空间和 cgroups、Windows 孤岛和作业对象）来隔离进程，并控制进程可以访问的 CPU、内存和磁盘的数量。

容器小巧轻便、速度快且可移植；与虚拟机不同，容器不需要在每个实例中都包含访客操作系统，只需利用主机操作系统的功能和资源。

容器在几十年前就已出现在 FreeBSD Jails 和 AIX Workload Partitions 等版本中，但大多数现代开发人员都记得容器时代是始于 2013 年，因为 Docker 诞生于 2013 年。

下载 Docker 平台：
https://docs.docker.com/engine/install/

选择对应 OS：



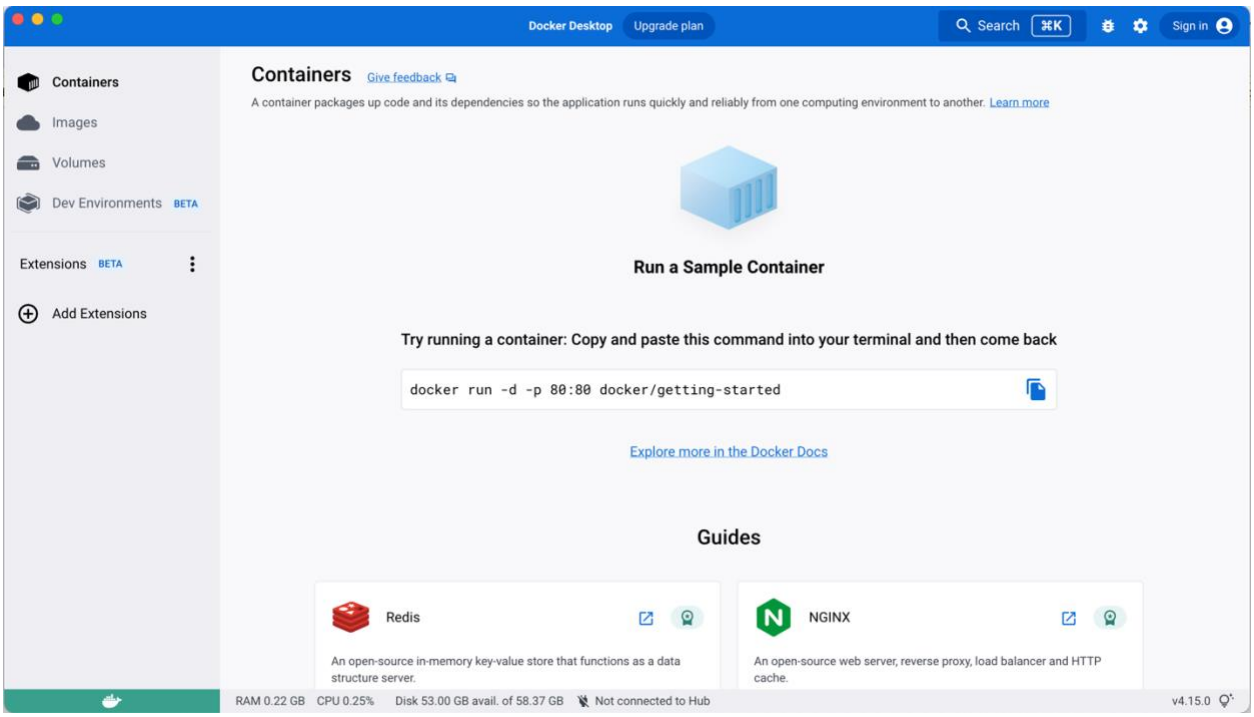
以 windows 为例：
https://docs.docker.com/desktop/install/windows-install/

Install Docker Desktop on Windows

Install interactively | Install from the command line

1. Double-click **Docker Desktop Installer.exe** to run the installer.
2. When prompted, ensure the **Use WSL 2 instead of Hyper-V** option on the Configuration page is selected or not depending on your choice of backend.
   If your system only supports one of the two options, you will not be able to select which backend to use.
3. Follow the instructions on the installation wizard to authorize the installer and proceed with the install.
4. When the installation is successful, select **Close** to complete the installation process.
5. If your admin account is different to your user account, you must add the user to the **docker-users** group. Run **Computer Management** as an **administrator** and navigate to **Local Users and Groups** > **Groups** > **docker-users**. Right-click to add the user to the group. Sign out and sign back in for the changes to take effect.

安装完成界面：



2. 安装入侵检测防御系统 Snort

入侵检测系统 Snort 介绍:

在 1998 年，Martin Roesch 用 C 语言开发了开放源代码(Open Source)的入侵检测系统 Snort。Snort 已发展成为一个具有多平台(Multi-Platform)、实时(Real-Time)流量分析、网络 IP 数据包(Pocket)记录等特性的强大的网络入侵检测/防御系统(Network Intrusion Detection/Prevention System)，即 NIDS/NIPS。Snort 符合通用公共许可(GPL——GNU General Public License)，在网上可以通过免费下载获得 Snort，并且只需要几分钟就可以安装并开始使用。

Snort 有三种工作模式：嗅探器、数据包记录器、网络入侵检测系统。嗅探器模式仅仅是从网络上读取数据包并作为连续不断的流显示在终端上。数据包记录器模式把数据包记录到硬盘上。网络入侵检测模式是最复杂的，而且是可配置的。我们可以让 snort 分析网络数据流以匹配用户定义的一些规则，并根据检测结果采取一定的动作。

Snort 能够对网络上的数据包进行抓包分析，但区别于其它嗅探器的是，它能根据所定义的规则进行响应及处理。Snort 通过对获取的数据包，进行各规则的分析后，根据规则链，可采取 Activation（报警并启动另外一个动态规则链）、Dynamic（由其它的规则包调用）、Alert（报警），Pass（忽略），Log（不报警但记录网络流量）五种响应的机制。

Snort 有数据包嗅探，数据包分析，数据包检测，响应处理等多种功能，每个模块实现不同的功能，各模块都是用插件的方式和 Snort 相结合，功能扩展方便。例如，预处理插件的功能就是在规则匹配误用检测之前运行，完成 TCP 碎片重组，http 解码，telnet 解码等功能，处理插件完成检查协议各字段，关闭连接，攻击响应等功能，输出插件将得理后的各种情况以日志或警告的方式输出。

Docker-snort 安装链接：
https://github.com/John-Lin/docker-snort/

安装与运行流程：

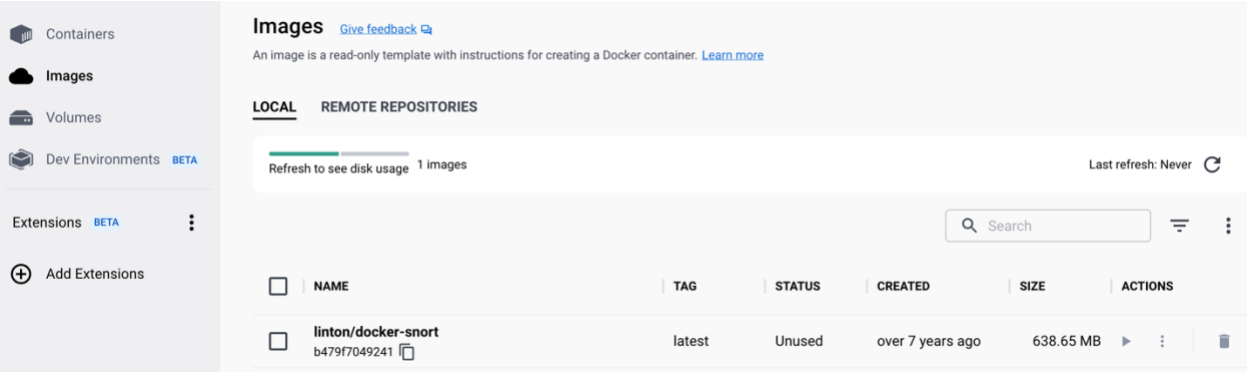
You may need to run as `sudo` Attach the snort in container to have full access to the network

```
$ docker run -it --rm --net=host linton/docker-snort /bin/bash
```

Or you may need to add --cap-add=NET_ADMIN or --privileged (unsafe)

```
$ docker run -it --rm --net=host --cap-add=NET_ADMIN linton/docker-snort /bin/bash
```

在 Docker 界面验证安装完成：



实验 1：Snort 入侵检测系统的恶意扫描检测

检测 ICMP 数据包

1. 运行 Snort-Docker

```
docker run -it --rm --net=host linton/docker-snort /bin/bash
```

2. 修改 Snort 检测规则

Snort 规则介绍:

###标题字段（必须）

action：snort 发现匹配规则时要执行的动作，包括（alert、log、pass、activate、dynamic）

Protocol：流量使用的协议（ip、tcp、udp、icmp 等）

Source IP：攻击者/访问者的 IP

Source port：攻击者/访问者的 端口

方向运算符：" ->"、"<>" 表示发送方和接收方网络流量方向

Destination IP：目标 IP，一般是收 IDS 保护的主机 IP

Destination Port：目标端口，一般是接受流量的主机端口

###选项字段（在括号中，可选。参数和值之间以冒号分隔，参数和参数之间以分号分隔）

###选项字段有四大类（常规、有效载荷、非有效载荷、检测后触发器）

###如下是一般规则

.msg：在报警和日志中打印的消息,

.logto：把日志记录到一个用户指定的文件，而不是输出到标准的输出文件，如：logto:'<filename>';

.ttl：测试 IP 包头的 TTL 域的值 .tos：测试 IP 包头的 TOS 域的值

.id：测试 IP 分组标志符(fragment ID)域是否是一个特定的值

.ipoption：查看 IP 选项（IP option)域

.fragbits：测试 IP 包头的分片位(fragmentation bit)

.dsize：测试数据包包数据段的大小

.flags：测试 TCP 标志(flag)是否是某个值

.seq：测试 TCP 包的序列号是否是某个值

.ack：测试 TCP 包的确认(acknowledgement)域是否为某个值

.itype：测试 ICMP 数据包的类型(type)域

Ping 检测规则:

For testing it's work. Add this rule in the file at `/etc/snort/rules/local.rules`

```
alert icmp any any -> any any (msg:"Pinging...";sid:1000004;)
```

3. 在 Snort-Docker 的 Terminal 运行 Snort

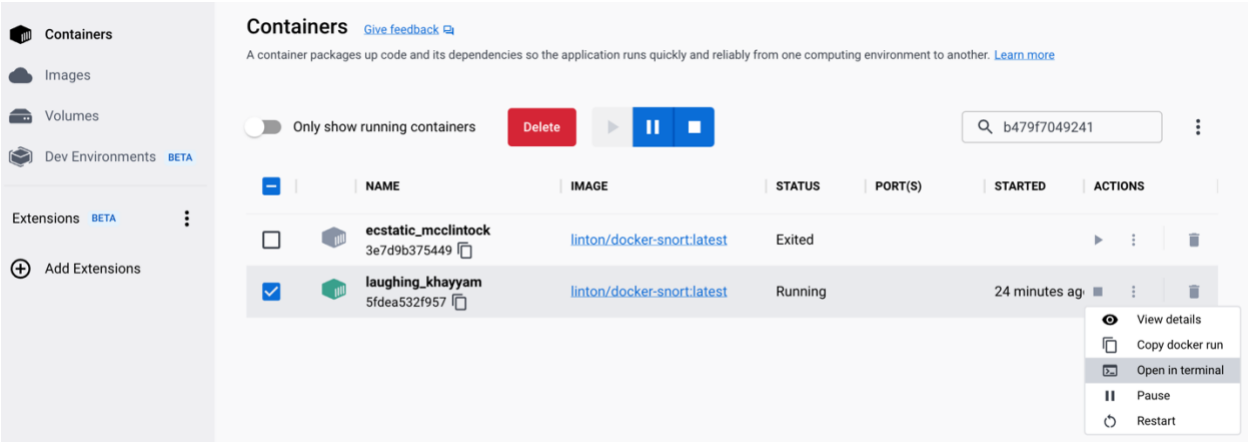
```
snort -i eth0 -c /etc/snort/etc/snort.conf -A console
```

结果如下：

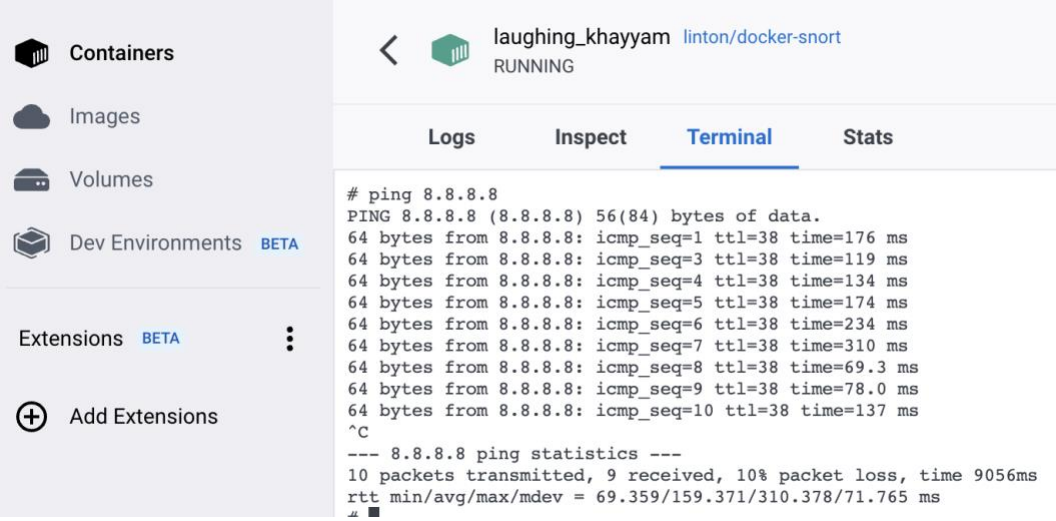
```
root@docker-desktop:/opt# snort -i eth0 -c /etc/snort/etc/snort.conf -A console
Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/etc/snort.conf"
```

4. 在 Docker UI 界面运行 Ping 命令

进入 Snort 容器 Terminal：



在 Terminal 中运行 Ping （可用 Ctrl + C 停止）：



Snort 检测结果：

```
Commencing packet processing (pid=18)
^[[B12/25-08:12:06.704603  [**] [1:1000004:0] Pinging... [**] [Priority: 0] {ICMP} 192.168.65.3 -> 8.8.8.8
12/25-08:12:07.705716  [**] [1:1000004:0] Pinging... [**] [Priority: 0] {ICMP} 192.168.65.3 -> 8.8.8.8
12/25-08:12:08.730523  [**] [1:1000004:0] Pinging... [**] [Priority: 0] {ICMP} 192.168.65.3 -> 8.8.8.8
12/25-08:12:09.735000  [**] [1:1000004:0] Pinging... [**] [Priority: 0] {ICMP} 192.168.65.3 -> 8.8.8.8
12/25-08:12:10.737854  [**] [1:1000004:0] Pinging... [**] [Priority: 0] {ICMP} 192.168.65.3 -> 8.8.8.8
12/25-08:12:11.743928  [**] [1:1000004:0] Pinging... [**] [Priority: 0] {ICMP} 192.168.65.3 -> 8.8.8.8
12/25-08:12:12.747286  [**] [1:1000004:0] Pinging... [**] [Priority: 0] {ICMP} 192.168.65.3 -> 8.8.8.8
12/25-08:12:13.750468  [**] [1:1000004:0] Pinging... [**] [Priority: 0] {ICMP} 192.168.65.3 -> 8.8.8.8
12/25-08:12:14.754128  [**] [1:1000004:0] Pinging... [**] [Priority: 0] {ICMP} 192.168.65.3 -> 8.8.8.8
12/25-08:12:15.760416  [**] [1:1000004:0] Pinging... [**] [Priority: 0] {ICMP} 192.168.65.3 -> 8.8.8.8
```

检测 NMAP 扫描

安装 NMAP：

在 docker 容器 Terminal 运行 sudo apt-get update

```
root@docker-desktop:/opt# sudo apt-get update
Ign http://archive.ubuntu.com trusty InRelease
Get:1 http://archive.ubuntu.com trusty-updates InRelease [56.4 kB]
Get:2 http://archive.ubuntu.com trusty-security InRelease [56.4 kB]
Get:3 http://archive.ubuntu.com trusty Release.gpg [933 B]
Get:4 http://archive.ubuntu.com trusty-updates/main Sources [809 kB]
Get:5 http://archive.ubuntu.com trusty-updates/restricted Sources [9993 B]
Get:6 http://archive.ubuntu.com trusty-updates/universe Sources [435 kB]
Get:7 http://archive.ubuntu.com trusty-updates/main amd64 Packages [1460 kB]
Get:8 http://archive.ubuntu.com trusty-updates/restricted amd64 Packages [28.7 kB]
Get:9 http://archive.ubuntu.com trusty-updates/universe amd64 Packages [884 kB]
Get:10 http://archive.ubuntu.com trusty-security/main Sources [314 kB]
```

在 docker 容器 Terminal 运行 sudo apt-get install nmap

```
root@docker-desktop:/opt# sudo apt-get install nmap
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libblas3 liblinear-tools liblinear1 liblua5.2-0
Suggested packages:
  libsvm-tools liblinear-dev
The following NEW packages will be installed:
  libblas3 liblinear-tools liblinear1 liblua5.2-0 nmap
0 upgraded, 5 newly installed, 0 to remove and 122 not upgraded.
Need to get 4238 kB of archives.
After this operation, 18.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

修改 Snort 规则：

vim /etc/snort/rules/local.rules

加入以下 Snort 规则：

alert icmp any any -> 8.8.8.8 any (msg:"NMAP Ping Sweep Scan";dsize:0;sid:1000004;)

运行 snort：

snort -i eth0 -c /etc/snort/etc/snort.conf -A console

在 Docker UI 界面进行 Nmap Scan：
nmap -sP 8.8.8.8 -disable-arp-ping

```
# nmap -sP 8.8.8.8 -disable-arp-ping

Starting Nmap 6.40 ( http://nmap.org ) at 2023-12-27 07:31 UTC
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.018s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds
```

在 Docker UI 的 Log 界面显示成功检测结果：

```
2023-12-27 15:28:47            Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
2023-12-27 15:28:47            Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
2023-12-27 15:28:47            Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
2023-12-27 15:28:47            Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
2023-12-27 15:28:47            Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
2023-12-27 15:28:47            Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
2023-12-27 15:28:47            Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
2023-12-27 15:28:47 Commencing packet processing (pid=179)
2023-12-27 15:31:35 12/27-07:31:34.206534  [**] [1:1000004:0] NMAP Ping Sweep Scan [**] [Priority: 0] {ICMP} 192.168.65.3 -> 8.8.8.8
```