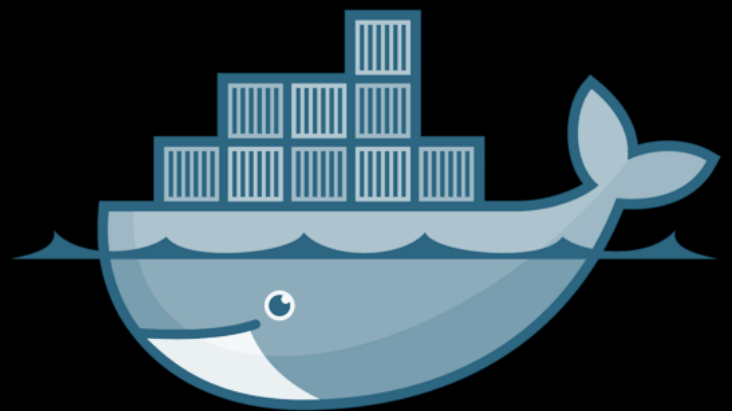


# LAB DE DETECÇÃO

CRIANDO LABORATORIO DE DETECÇÃO EM  
CIBERSEGURANÇA COM DOCKER



docker



Diogo Caldas

# Introdução

## Laboratório de detecção com Docker !

Bem-vindo ao guia essencial para montar seu próprio laboratório de detecção de ameaças usando Docker! Este eBook vai orientá-lo na configuração de um ambiente prático e eficiente para testar e detectar ameaças cibernéticas, utilizando apenas o Docker. Vamos começar com o básico e, em seguida, seguir para exemplos práticos.



# 01

## **O Que é Docker e Por Que Usá-lo?**



# O Que é Docker e Por Que Usá-lo?

Docker é uma plataforma que permite criar, testar e implantar aplicações em ambientes isolados chamados de containers. Ele é leve, rápido e facilita a replicação de ambientes, tornando-o ideal para um laboratório de detecção de ameaças.

## Por que Docker?

- Isolamento: Cada container é isolado, o que impede que uma ameaça afete todo o sistema.
- Portabilidade: Pode ser executado em qualquer lugar, desde que o Docker esteja instalado.
- Eficiência: Consome menos recursos do que máquinas virtuais tradicionais.

02

# **Preparando Seu Ambiente**



# Preparando Seu Ambiente

Antes de começar, você precisa instalar o Docker.  
Veja como fazer isso:

## Instalando Docker

No Windows/Mac:

1. Baixe o Docker Desktop [aqui](#).
2. Execute o instalador e siga as instruções.

No Linux:

```
sudo apt-get update  
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Verifique a instalação:

```
docker --version
```

03

# **Criando Seu Primeiro Container**



# Criando Seu Primeiro Container

- Abra o terminal.
- Execute o comando:

```
docker run --name meu-nginx -d -p 8080:80 nginx
```

Isso baixa a imagem do Nginx e a executa no modo daemon (-d), vinculando a porta 8080 do host à porta 80 do container.

- Acesse <http://localhost:8080> no seu navegador. Você verá a página padrão do Nginx.

## Comando Explicado:

- `docker run`: Inicia um novo container.
- `--name meu-nginx`: Nomeia o container como "meu-nginx".
- `-d`: Executa o container em segundo plano.
- `-p 8080:80`: Mapeia a porta 8080 do host para a porta 80 do container.
- `nginx`: Usa a imagem do Nginx.



# 04

## **Configurando um Ambiente de Detecção de Ameaças**



# Configurando um Ambiente de Detecção de Ameaças


Vamos configurar um ambiente simples de detecção usando o Suricata, uma ferramenta de detecção de intrusões (IDS).

## Exemplo: Container Suricata

### 1. Criar um Dockerfile:

- Crie uma pasta chamada `suricata-lab` e dentro dela um arquivo chamado `Dockerfile`.
- Adicione o seguinte conteúdo:

Dockerfile

 Copiar código


```
FROM ubuntu:20.04
RUN apt-get update && apt-get install -y suricata
CMD ["suricata", "-c", "/etc/suricata/suricata.yaml", "-i", "eth0"]
```

Isso cria uma imagem base Ubuntu e instala o Suricata.

### 2. Construir a Imagem:

- No terminal, navegue até a pasta `suricata-lab` e execute:

bash

 Copiar código


```
docker build -t meu-suricata .
```

Isso cria a imagem chamada `meu-suricata`.

### 3. Executar o Container:

- Execute o comando:

bash

 Copiar código

```
docker run --name suricata-container --net=host -it meu-suricata
```

Isso inicia o Suricata em um container com a rede do host.

# Configurando um Ambiente de Detecção de Ameaças

Comando Explicado:

`docker build -t meu-suricata .`: Constrói a imagem usando o Dockerfile na pasta atual e a nomeia meu-suricata.

`docker run --name suricata-container --net=host -it meu-suricata`: Executa o container com a rede do host e no modo interativo.

05

# Testando Seu Laboratório




# Testando Seu Laboratório

Agora, vamos gerar tráfego de rede e verificar se o Suricata consegue detectar algo.

1. No terminal, dentro do container Suricata, instale `curl`:


bash

 Copiar código

```
apt-get update && apt-get install -y curl
```

2. Execute um comando para gerar tráfego:

bash


 Copiar código

```
curl http://testmyids.com
```

Isso deve gerar um alerta que o Suricata detectará.

3. Verifique os logs do Suricata:


bash

 Copiar código

```
cat /var/log/suricata/fast.log
```

Você deve ver uma linha semelhante a:

CSS

 Copiar código

```
[**] [1:2100498:1] GPL ATTACK_RESPONSE id check returned root [**] [Classification: Po
```

## Comando Explicado:

- ``curl http://testmyids.com`` [Sem título] Faz uma solicitação que deve disparar um alerta IDS.
- ``cat /var/log/suricata/fast.log``: Mostra os logs de alertas do Suricata.

06

# **Automatizando e Expandindo Seu Laboratório**



# Automatizando e Expandindo Seu Laboratório


Para automatizar e expandir seu laboratório, você pode usar o Docker Compose para orquestrar múltiplos containers.

## Exemplo: Usando Docker Compose

### 1. Criar um Arquivo `docker-compose.yml`:

- Na pasta `suricata-lab`, crie um arquivo `docker-compose.yml` com o seguinte conteúdo:

yaml


 Copiar código

```
version: '3'
services:
  suricata:
    image: meu-suricata
    network_mode: host
    container_name: suricata-container
  nginx:
    image: nginx
    ports:
      - "8080:80"
```

### 2. Executar com Docker Compose:

- No terminal, execute:

bash

 Copiar código

```
docker-compose up
```

Isso iniciará ambos os containers, Nginx e Suricata.

### Comando Explicado:

- `docker-compose up`: Inicia todos os serviços definidos no `docker-compose.yml`.

# Conclusão

Você agora tem um laboratório de detecção de ameaças funcional usando Docker! Com essa configuração, você pode simular diferentes cenários de ataque, verificar a eficácia de suas regras IDS e expandir conforme necessário. Experimente adicionar mais ferramentas ou scripts para aumentar a complexidade e utilidade do seu laboratório. Boa sorte!

Este eBook servirá como sua base para explorar e implementar soluções mais complexas e robustas em seu ambiente de cibersegurança. Continue aprendendo e adaptando as ferramentas às suas necessidades específicas.