

## Buscando informações usando OSINT

- ▼ Proprietário - Whois
- ▼ Portas - Shodan
- ▼ Geolocalização - Shodan
- ▼ Subdomínios – SharingmyIP
- ▼ Informações - Google Hacking
- ▼ Enumerar Serviços - Censys/Shodan
- ▼ Enumerar CMS - WhatCMS
- ▼ Checar IP - SharingMyIP

## Introdução ao Docker

- ▼ O que é o Docker
- ▼ Vantagens
- ▼ Porque eu devo usar o Docker
- ▼ Como a estrutura de servidores é projetada
- ▼ Conhecendo a arquitetura e o conceito
- ▼ Alguns comandos básicos do Docker



## Docker para Pentesters

- ▼ Brute force em diretórios - DIRB
- ▼ Analise de portas - Nmap/netcat
- ▼ Enumeração de serviços - FTP/mysql-client/SSL Client
- ▼ Analise de vulnerabilidade - Nessus/OpenVas/NSE
- ▼ Buscando exploits públicos - Pompem/Searchsploit
- ▼ Labs para estudo - DVWA/SSH/Worspress
- ▼ Criando wordlists personalizadas - CeWL
- ▼ Outras interessantes - metagoofil/theharveste/recon-ng

# ./OSINT -tema 'Proprietário - Whois'

O que podemos encontrar com o Whois ?

- ▼ Nome do domínio
- ▼ Quem registrou o domínio
- ▼ Quando foi criado
- ▼ Até quando é válido
- ▼ Servidores DNS
- ▼ País
- ▼ email de contato
- ▼ Nome do responsável
- ▼ CPF/CNPJ

Veja mais: <https://github.com/ABase-BR/OSINT/blob/master/2-Whois.md>

# ./OSINT -tema 'Range de IP - Whois'

Além dos domínios também podemos pesquisar por IPS ,  
dessa forma podemos buscar informações sobre a rede e  
assim encontrar informações como

- ▼ Range de rede
- ▼ CIDR
- ▼ Organização responsável
- ▼ Endereço
- ▼ Cidade
- ▼ País

Veja mais: <https://github.com/ABase-BR/OSINT/blob/master/2-Whois.md>

# ./OSINT -tema 'Portas - Shodan'

Com o Shodan podemos encontrar dispositivos conectados a internet e buscar por informações.

- ▼ Portas
- ▼ Serviços
- ▼ Organização
- ▼ Hostnames

Podemos usar operadores e assim filtrar por portas determinados serviços que estão disponíveis na internet.

Veja mais:

<https://github.com/ABase-BR/OSINT/blob/master/1-Shodan.md#iniciando-os-testes>

# ./OSINT -tema 'Geolocalização - Shodan'

Podemos filtrar por cidade usando operadores , além disso ao inserir um alvo ele vai retornar a possível localização.

- ▼ Localização do servidor
- ▼ Cidade
- ▼ País

Veja mais:

<https://github.com/ABase-BR/OSINT/blob/master/1-Shodan.md#shodan-e-seus-operadores>

# Subdomínios - SharingmyIP

Com o SharingmyIP é possível que encontre informações de um site e podemos encontrar informações como

- ▼ Sites que dividem o mesmo IP
- ▼ Subdomínios
- ▼ Entradas de DNS

Veja mais: <https://github.com/ABase-BR/OSINT/blob/master/4-Sharingmyip.md>



# Informações de site - Google Hacking

O Google faz o uso de operadores , assim faz uma busca para localizar sequências específicas de texto dentro de resultados de pesquisa.

- ▼ Versões específicas vulneráveis
- ▼ Localizar páginas que possuem um determinado texto
- ▼ Páginas de administração
- ▼ Backups de arquivos

E tudo que pode estar sendo indexado pelo Google.

Veja mais: <https://github.com/ABase-BR/OSINT/blob/master/10-Google.md>

# Enumerar Serviços - Censys/Shodan

Censys nos auxilia na busca de dispositivos , redes e infraestruturas. Com ele é possível ver informações como

- ▼ Latitude/Longitude
- ▼ País
- ▼ Portas abertas e protocolos
- ▼ Title do IP usado
- ▼ Nome da rede de sites e IPS.

Veja mais: <https://github.com/ABase-BR/OSINT/blob/master/3-Censys.md>

# Enumerar CMS - WhatCMS

O WhatCMS nos auxilia no reconhecimento CMS , podemos ver até qual a versão e ele possui mais de 422 CMS.

- ▼ Wordpress
- ▼ Joomla
- ▼ Moodle
- ▼ Drupal
- ▼ Presta shop
- ▼ PhpBB

Veja mais: <https://github.com/ABase-BR/OSINT/blob/master/5-WhatCMS.md>

# O que é o Docker?

O docker é uma plataforma aberta criada com o intuito de

- ▼ Facilitar no desenvolvimento
- ▼ Agilizar a implementação
- ▼ E a possibilidade de ter aplicações em ambientes isolados

O docker foi abraçado pela comunidade e graças a

- ▼ Facilidade de gerenciar
- ▼ A agilidade para subir um novo ambiente
- ▼ E a simplicidade de realizar modificações

Veja mais:

<https://github.com/ABase-BR/Docker-intro/tree/master/1-O-que-%C3%A9-o-Docker>

# Vantagens - Principais vantagens

Algumas das vantagens do uso do Docker é ter um ambiente

- ▼ Extremamente leve
- ▼ Isolado e ajudando com projetos legados

Temos a possibilidade de criar diversos containers e serem executados simultaneamente em um mesmo host.

Veja mais: <https://github.com/ABase-BR/Docker-intro/tree/master/2-Vantagens>

# Vantagens - Dependências

Podemos ter um gerenciamento de dependência melhor e assim dando a possibilidade de cada contêiner contem as dependências de cada aplicação.

Depois de um projeto criado podemos facilmente

- ▼ Importar
- ▼ Exportar

Assim podemos subir de forma fácil para um outro ambiente ou até para a produção.

# Vantagens - Diversas versões

- ▼ Podemos ter diversos imagens e containers para cada versão do projeto.
- ▼ Depois de criado podemos ter a portabilidade de testar em qualquer ambiente e facilmente ter uma escalabilidade para produção.
- ▼ A facilidade de testar diversas versões de um mesmo projeto em diversos ambientes com agilidade

# Porque eu devo usar o Docker

Afinal , porque mudar o modo de trabalhar hoje?

- ▼ Atualmente o Docker está sendo considerada um "idioma" , pois ele auxilia na comunicação entre o código e a infraestrutura.
- ▼ A facilidade para encontrar exemplos , projetos opensource e ver como outros desenvolvedores trabalham.
- ▼ Poupar tempo de novos desenvolvedores e da equipe.
- ▼ A agilidade de varias pessoas ter o mesmo ambiente



# Ainda não sei se uso o Docker - Desenvolvedor

- ▼ Só precisa se preocupar em desenvolver uma vez e pode executar em qualquer local.
- ▼ Não é necessário se preocupar com dependências ou pacotes.
- ▼ Só precisa ter o foco no desenvolvimento.
- ▼ Com o Docker é possível ter diversos ambientes para testes.
- ▼ Assim evitamos o clássico no localhost funciona

# Ainda não sei se uso o Docker - Syadmin

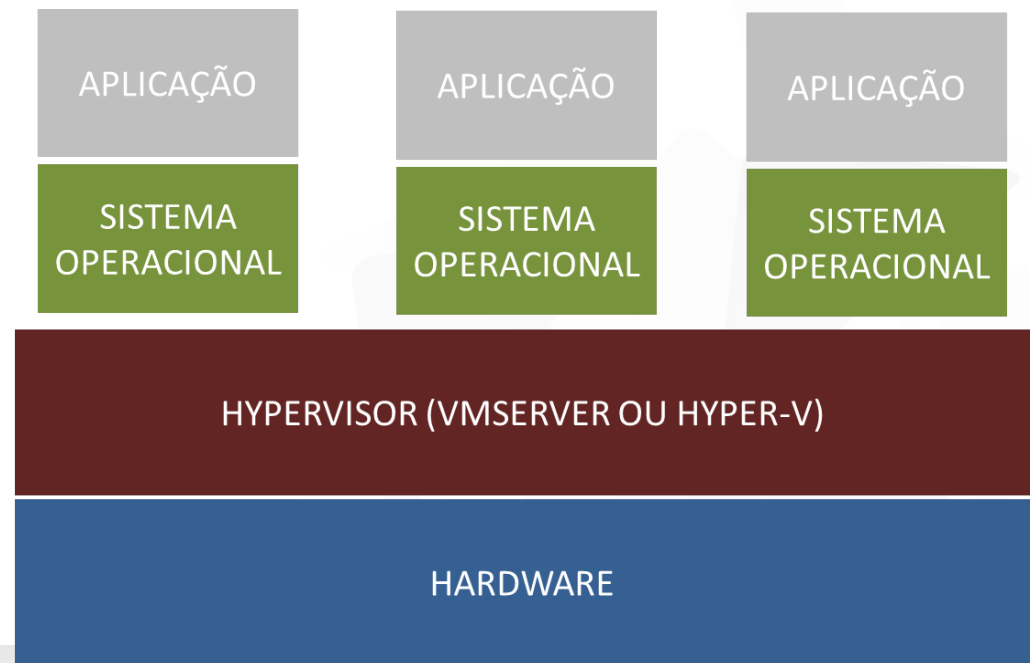
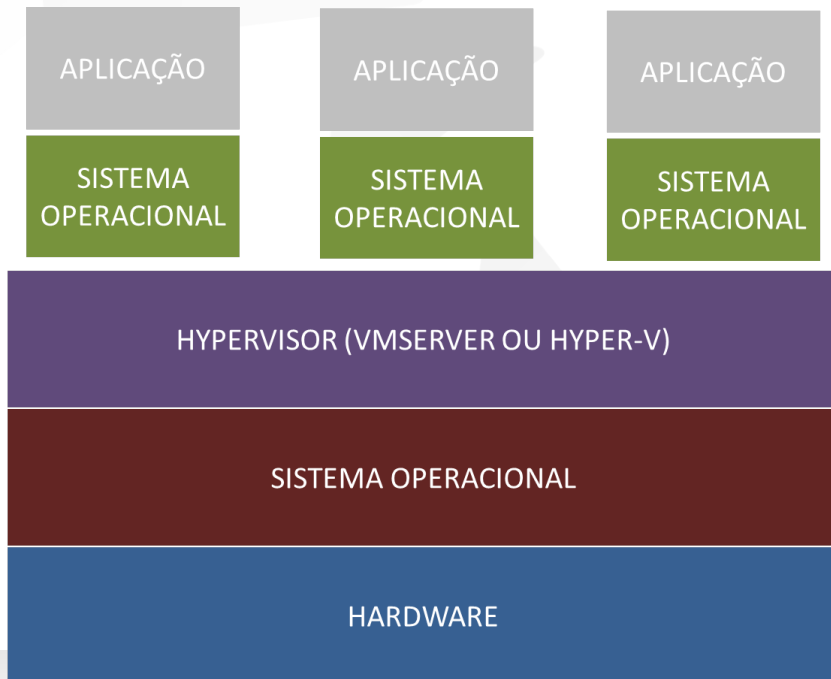
- ▼ O responsável pela infraestrutura configura uma vez e executa em qualquer lugar
- ▼ Podemos eliminar incertezas na entrega das aplicações ou serviços
- ▼ Com o Docker temos um ciclo de trabalho mais eficiente e ágil
- ▼ É possível ter uma infraestrutura escalável facilmente

Veja mais:

<https://github.com/ABase-BR/Docker-intro/tree/master/3-Portue-eu-devo-usar-o-Docker>

# Sem o uso da plataforma de containers - Infra

- Uma infraestrutura sem o uso de containers deixa todo o projeto mais pesado e lento.
- Em toda maquina virtual tem um sistema operacional instalado , deixando o projeto muito mais lento , pesado e até aumentando o custo



# Como funciona com o uso de containers - Infra

Já com o uso de containers podemos ver que só é necessário obter os binários e as bibliotecas.

- ▼ O Docker usa um modelo de isolamento utilizado é o virtualização a nível de sistema operacional



Veja mais:

<https://github.com/ABase-BR/Docker-intro/tree/master/4-Como-a-infraestrutura-de-servidor-e-projetada>

# Conhecendo a arquitetura - Imagens

- ▼ As imagens podemos entender como formas de bolo ou template usados para criar containers
- ▼ As imagens não containers , mais dão base consistente para que aja um container.
- ▼ Temos imagens oficiais ou criadas pela comunidades.
- ▼ Podemos armazenar elas localmente usando o Docker Registry ou publicamente no Dockerhub

# Conhecendo a arquitetura - Containers

- ▼ Já os containers podemos entender como bolos prontos , não podemos criar um container sem uma imagem previamente.
- ▼ Os containers contem o necessario para executar uma aplicação e são baseados nas images.
- ▼ Os mantem o isolamento da aplicação e de recursos.
- ▼ Os containers são voláteis e depois de desligado todo o seu conteúdo é perdido.

Veja mais:

<https://github.com/ABase-BR/Docker-intro/tree/master/5-Conhecendo-a-arquitetura>

# Conhecendo o conceito – Docker Engine

O docker engine é um daemon , ele é nos auxilia na

- ▼ Construção
- ▼ No envio
- ▼ E na execução dos nossos containers.

# Conhecendo o conceito - Docker Client

- ▼ Já o Docker client é responsável por receber as entradas do usuário e as enviar para a engine.
- ▼ Podemos ter um client e o engine na mesma máquina. Porém eles podem ser executados em hosts diferentes.



# Conhecendo o conceito - Docker Registry

- ▼ O Docker registry é responsável por armazenar nossas imagens , ele pode ser usado de forma local ou criar o nosso próprio servidor de imagens.
- ▼ Esse servidor pode ser privado ou publico , um exemplo de servidor publico é o Dockerhub onde podemos encontrar diversas imagens e até subir a nossa propria imagem.

# Conhecendo o conceito - Dockerfile

- ▼ O dockerfile é um arquivo que auxiliar na criação de uma imagem.
- ▼ É usado instruções e elas são aplicadas em uma determinada imagem para que outra imagem seja criada baseada nas modificações.
- ▼ Depois de criada podemos subir nossa imagem para o dockerhub.

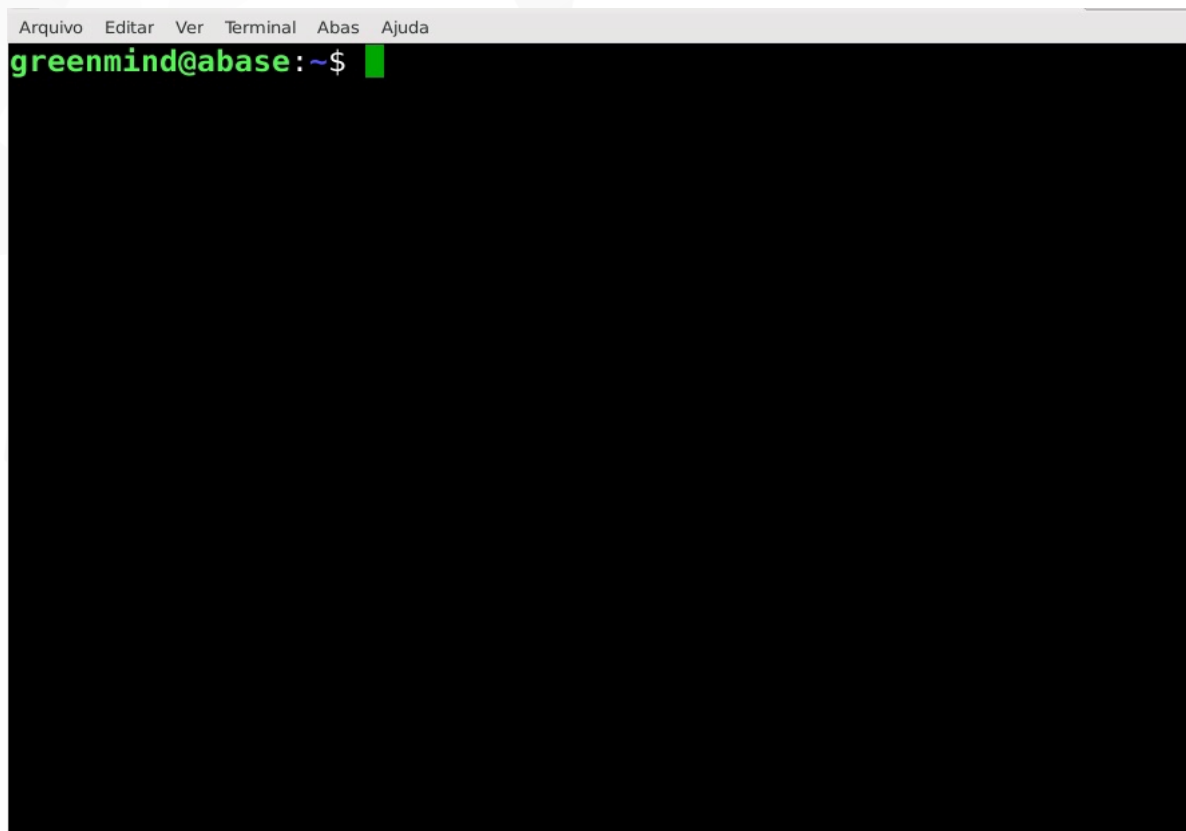
# Conhecendo o conceito – Docker compose

- ▼ O docker compose é uma ferramenta para nos auxiliar na criação de múltiplos containers Docker
- ▼ Assim como no Dockerfile podemos configurar todos os parâmetros necessários para executar um container a partir de um arquivo que é o docker-compose.yml
- ▼ Nesse arquivo de execução podemos selecionar definir determinados serviços , podemos setar portas abertas , variáveis de ambiente , volumes , configurar redes e muitas possibilidade que não conseguimos apenas com o Dockerfile.

Veja mais: <https://github.com/ABase-BR/Docker-intro/tree/master/6-Entendo-o-conceito>

# Comandos basicos - help

Podemos obter mais informações sobre o docker e o docker-compose usando a opção

A screenshot of a terminal window. The title bar at the top contains the menu items: 'Arquivo', 'Editar', 'Ver', 'Terminal', 'Abas', and 'Ajuda'. The terminal content shows a green prompt 'greenmind@abase:~\$' followed by a green cursor block. The rest of the terminal area is black and empty.

```
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
greenmind@abase:~$
```

# Comandos basicos - pull

O comando **pull** é responsável por obter uma imagem, essa imagem pode ser do DockerHub ou de outro servidor de imagens.

Podemos querer escolher uma determinada tag, senão ele vai baixar a versão latest.

# Comandos basicos - images

O comando **images** é resposavel por listar as imagens que temos em nossa maquina.

Dessa forma podemos ver qual imagem usar e até qual TAG.

# Comandos basicos - ps

Podemos listar os container que estão em funcionamento usando o **ps** e ele nos auxilia para saber quais containers estão em funcionamento.

Vamos ter informações como por exemplo

- ▼ CONTAINER ID
- ▼ IMAGE
- ▼ COMMAND
- ▼ CREATED
- ▼ STATUS
- ▼ PORTS
- ▼ NAMES

# Comandos basicos - run

Agora que já sabemos como obter uma imagem e como listar elas vamos executar um container.

- ▼ Podemos executar ele tambem em background
- ▼ Podemos executar ele interagindo diretamente com ele e quando terminarmos ele ser excluido.



# Comandos basicos - Volumes

Vamos realizar mapeamento da seguinte forma.

- ▼ Primeiro precisamos especificar qual origem do no host e segundo onde devemos montar dentro do container.

Veja mais:

<https://github.com/ABase-BR/Docker-intro/tree/master/7-Conhecendo-comandos-basicos-do-Docker#volumes>

# Comandos basicos - Portas

- ▼ Podemos realizar o mapeamento de portas da seguinte forma.
- ▼ Primeiro precisamos saber qual porta será mapeada no host e qual deve receber essa conexão dentro do container.

Veja mais:

<https://github.com/ABase-BR/Docker-intro/tree/master/7-Conhecendo-comandos-basicos-do-Docker#passando-uma-porta-ao-container>

# Comandos basicos - exec

Depois de criar um container tambem podemos ter acesso a eles , para isso vamos usar o exec.

Vamos supor que temos um container com o nome **web\_server** e quero ter acesso a ele.

Veja mais:

<https://github.com/ABase-BR/Docker-intro/tree/master/7-Conhecendo-comandos-basicos-do-Docker#conhecendo-o-exec>

# Comando basicos – criando container

Depois que aprendemos a inserir nome em um container e subir ele para funcionar em background podemos realizar comandos como por exemplo

- ▼ stop
- ▼ start

Dessa forma podemos iniciar e parar container com maior facilidade.

Veja mais:

<https://github.com/ABase-BR/Docker-intro/tree/master/7-Conhecendo-comandos-basicos-do-Docker>

# Para fazer depois – Criar imagem

A ideia é você criar a sua propria imagem usando o Docker para agilizar algum processo ou teste do seu dia a dia.

- ▼ Depois de criado listar a sua imagem
- ▼ Subir para o Dockerhub
- ▼ Compartilhar com os amigos

Veja mais:

<https://github.com/ABase-BR/Docker-intro/tree/master/8-Criando-a-propria-imagem>

<https://github.com/ABase-BR/Docker-intro/tree/master/9-Listando-Imagens-e-Containers>

<https://github.com/ABase-BR/Docker-intro/tree/master/10-Subindo-imagem-ao-Dockerhub>

# Para fazer depois - Redes e Armazenamento

Podemos também configurar redes em nossos containers , configurar armazenamento para manter arquivo que seriam apagados caso ele seja apagado.

Veja mais:

<https://github.com/ABase-BR/Docker-intro/tree/master/11-Trabalhando-com-armazename>  
nto

<https://github.com/ABase-BR/Docker-intro/tree/master/12-Trabalhando-com-redes-no-Docker>

# Bonus – Gerenciando containers com Portainer

Agora que já aprendemos o basico podemos conhecer um projeto chamado Portainer.

O portainer nos auxilia na administração do docker e ainda

- ▼ Gerenciar container
- ▼ Auxilia na criação de imagens
- ▼ Redes
- ▼ Volumes

E tudo isso com duas linhas de comando.

Veja mais: <https://www.portainer.io/installation/>

# Brute force em diretórios - DIRB

O Dirb é um projeto que tem como objetivo realizar o reconhecimento de diretórios usando brute force.

- ▼ Podemos usar a wordlist padrão.
- ▼ É possível passar a nossa propria wordlist personalizada.

Saiba mais: <https://github.com/ABase-BR/Docker-pentesters/tree/master/dirb>



# Analise de portas - Nmap/netcat

NMAP é uma ferramenta muito conhecida , ela nos auxilia em diversas tarefas como por exemplo

- ▼ Scan de portas
- ▼ Scanner
- ▼ Ataques a formularios
- ▼ E uma infinidade de soluções.

Alem do NMAP temos o NSE e tem um conjunto de scripts maior a cada nova versão auxiliando no reconhecimento.

Veja mais: <https://github.com/ABase-BR/Docker-pentesters/tree/master/nmap>

# Enumeração - FTP/mysql-client/SSL Client

Não temos a necessidade de ter instalado diversos clientes em nossa maquina host como por exemplo

- ▼ Mysql-client
- ▼ Openssl
- ▼ FTP-client

Saiba mais:

<https://github.com/ABase-BR/Docker-pentesters/tree/master/openssl>

<https://github.com/ABase-BR/Docker-pentesters/tree/master/ftp-client>

<https://github.com/ABase-BR/Docker-pentesters/tree/master/mysql-client>

# Analise de vulnerabilidade - Nessus/OpenVas

- ▼ Nessus é um scanner de vulnerabilidades mantido pela Tenable, tem diversas versões pagas e uma para a comunidade
- ▼ Openvas é um framework que nos auxilia na análise de vulnerabilidade e é aberto para a comunidade
- ▼ NSE (Nmap Script Engine) usa um conjunto de scripts junto com o Nmap para auxiliar no reconhecimento, testes e ainda temos a possibilidade de modificar e inserir nossos scripts.

Veja mais: <https://github.com/ABase-BR/Docker-pentesters/tree/master/nessus>

<https://github.com/ABase-BR/Docker-pentesters/tree/master/openvas>

<https://github.com/ABase-BR/Docker-pentesters/tree/master/nmap>

# Buscando exploits públicos - Pompem

O Pompem auxilia na busca de exploits publicos

- ▼ PacketStorm security
- ▼ CXSecurity
- ▼ ZeroDay
- ▼ Vulners
- ▼ National Vulnerability Database
- ▼ WPScan Vulnerability Database

Veja mais: <https://github.com/ABase-BR/Docker-pentesters/tree/master/pompem>

# Criando wordlists personalizadas - CeWL

O CeWL é um projeto que tem como meta auxiliar na criação de wordlists personalizadas , só precisamos passar a URL que queremos testar e podemos encontrar o projeto no github.

Veja mais: <https://github.com/ABase-BR/Docker-pentesters/tree/master/cewl>

# Labs para estudo - DVWA/SSH/Wordpress

Além de usar diversas ferramentas para testes , também é possível a criação de laboratórios para estudo como por exemplo

## ▼ DVWA

Veja mais: <https://hub.docker.com/r/vulnerables/web-dvwa>

<https://github.com/opsxcq/docker-vulnerable-dvwa>

## Outras - metagoofil/theharvester/recon-ng

- ▼ Metagoofil é um projeto para extrair metadados usando documentos públicos (PDF, docs, XLS, PPT) em sites alvo.
- ▼ Theharvester é uma ferramenta que auxilia na busca de informações públicas (email, nomes , subdomínios , IPS , servidores DNS , banco de dados e em muitas outras fontes)
- ▼ O recon-ng é uma ferramenta desenvolvida em Python , com funcionalidades como , whois , localização geográfica , validação de emails , força bruta DNS , adsense e etc

Veja mais:

<https://github.com/ABase-BR/Docker-pentesters/tree/master/metagoofil>

<https://github.com/ABase-BR/Docker-pentesters/tree/master/theHarvester>

<https://github.com/ABase-BR/Docker-pentesters/tree/master/recon-ng>

Perguntas?







## Referencias e agradecimentos ...

- ▼ <https://github.com/gomex/docker-para-desenvolvedores/>
- ▼ <https://www.linuxtips.com.br/>
- ▼ E toda a comunidade foda que está a nossa volta



All text and image content in this document is licensed under the [Creative Commons Attribution-Share Alike 3.0 License](https://creativecommons.org/licenses/by-sa/3.0/) (unless otherwise specified). "LibreOffice" and "The Document Foundation" are registered trademarks. Their respective logos and icons are subject to international copyright laws. The use of these therefore is subject to the [trademark policy](#).



Obrigado ...

- ▼ <https://github.com/greenmind-sec/>
- ▼ <https://github.com/ABase-BR/Docker-intro>
- ▼ <https://www.juliusec.com/>
- ▼ <https://wiki.juliusec.com/>
- ▼ <https://ezdevs.com.br/>



All text and image content in this document is licensed under the [Creative Commons Attribution-Share Alike 3.0 License](#) (unless otherwise specified). "LibreOffice" and "The Document Foundation" are registered trademarks. Their respective logos and icons are subject to international copyright laws. The use of these therefore is subject to the [trademark policy](#).