

# 一篇文章带你搭建CTF平台-基于CTFd的H1ve

原创

Qwzf 于 2020-04-01 00:43:09 发布 4185 收藏 36

分类专栏: [CTF](#) 文章标签: [CTF](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/qq\\_43625917/article/details/105229456](https://blog.csdn.net/qq_43625917/article/details/105229456)

版权



[CTF 专栏收录该内容](#)

30 篇文章 6 订阅

订阅专栏

## 前言

前段时间发现了一个界面精美的CTF平台-H1ve, 发现它是在CTFd的基础上进行美化的。然后, 我们团队的lemon搭建了一下, 由于某些原因不能用了(服务器到期)。于是我参考其搭建笔记又搭了一遍, 发现了几个新的问题。继续总结一下搭建过程及所可能遇到的问题吧。。。

## 0x00 搭建环境

Centos 7 64位(其他版本暂未测试)

Docker

python2.7

## 0x01 搭建流程

### 1、服务器

准备一个云服务器, 阿里云、腾讯云和华为云均可。镜像选择Centos 7.x 64位的, 只要是Centos 7 64位就行, 具体版本无所谓。

### 2、安装docker

#### 1.安装所需软件包

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

#### 2.设置yum源

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

设置yum源主要是提高yum下载速度

#### 3.安装docker

```
sudo yum install docker-ce
```

如果安装失败, 可采用下面的脚本脚本安装

```
curl -fsSL https://get.docker.com | bash -s docker --mirror Aliyun
```

## 4.启动并加入开机启动

```
sudo systemctl start docker #或sudo service docker start
sudo systemctl enable docker
```

## 5.验证是否安装成功

```
docker version
```

```
[root@iZ2zecj761e18gcn2t5w8Z ~]# docker version
Client: Docker Engine - Community
 Version:           19.03.8
 API version:       1.40
 Go version:        go1.12.17
 Git commit:        afacb8b
 Built:             Wed Mar 11 01:27:04 2020
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.8                                qwzf
```

## 3、安装docker-compose

### 1.安装epel-release

```
yum -y install epel-release
```

### 2.安装python-pip

```
yum -y install python-pip
```

如果出现 未找到匹配的参数: python-pip 错误:没有任何匹配: python-pip

(1) 可能原因:

Centos中有的源是被禁用的, 通过以下命令可以查看启用的源和禁用的源。

```
#查看启用的源列表
yum repolist
#查看禁用的源列表
yum repolist disabled
```

查看之后, 发现epel在禁用列表里。

(2) 修复方法:

启用的方法:

```
vi /etc/yum.repos.d/epel.repo
```

将第一个和第三个的enabled=0改成enabled=1, 第二个是测试版的可能不稳定所有别修改。

修改后就可以安装 python-pip 了。

也可参考:

[CentOS下使用yum安装python-pip失败的完美解决方法](#)

### 3.升级pip

```
pip install --upgrade pip
```

如果出现 `ModuleNotFoundError: No module named 'pip._internal'`

(1) 可能原因:

与旧版冲突导致了这个报错。

(2) 修复方法:

```
python -m ensurepip
python -m pip install --upgrade pip
```

更多修复方法参考:

[ModuleNotFoundError: No module named 'pip.\\_internal', pip 无法下载软件 解决办法](#)

#### 4. 安装 docker-compose

```
pip install docker-compose
```

如果出现 `ReadTimeoutError: httpConnectionPool(host='pypi.python.org', port=443): Read timed out`

(1) 可能原因: (暂时我不知道)

(2) 修复方法:

```
pip --default-timeout=200 install -U docker-compose
```

如果出现

由于忘了截图了，上边都没有配图。但这个发现我们团队的lemon截图了，于是拿来即用。

(1) 可能原因:

缺少openssl-devel支持

(2) 修复方法:

```
yum install gcc libffi-devel python-devel openssl-devel -y
```

也可参考: [解决"command 'gcc' failed with exit status 1"错误问题](#)

然后重新使用pip安装docker-compose

```
pip install docker-compose
```

#### 5. 验证是否安装成功

```
docker-compose version
```

如果出现 `/usr/lib/python2.7/site-packages/requests/__init__.py:91: RequestsDependencyWarning: urllib3 (1.25.8) or chardet (2.2.1) doesn't match a supported version!`

(1) 可能原因:

python库中urllib3 (1.25.8) or chardet (2.2.1) 的版本不兼容

(2) 修复方法:

```
pip uninstall urllib3
pip uninstall chardet
pip install requests
```

如果出现 `from backports.shutil_get_terminal_size import get_terminal_size ImportError: No module named shutil_get_terminal_size`

(1) 可能原因:

- 1) 未安装ipython或安装有问题(我遇到的是这个原因)
- 2) (第二个原因暂时不清楚)

(2) 修复方法:

- 1) 针对原因 1), 可以用pip安装ipython:

```
pip install ipython
```

如果安装之后, 还有这个错误, 可使用下面这个命令:

```
pip install --force ipython
```

2) 针对原因 2), 报错信息显示在 `/usr/local/lib/python2.7/dist-packages/IPython/utils/terminal.py` 的 `from backports.shutil_get_terminal_size import get_terminal_size as _get_terminal_size` 出错。

`sudo vi vi /usr/lib/python2.7/site-packages/IPython/utils/terminal.py` 打开此脚本

将出错的代码段修改如下:

```
import struct
import sys
import warnings
try:
    from shutil import get_terminal_size as _get_terminal_size
except ImportError:
    from shutil_backports import get_terminal_size as _get_terminal_size

from . import py3compat
```

然后再次输入进行测试

```
docker-compose version
```

```
[root@iZ2zecj761e18gcnp2t5w8Z ~]# docker-compose version
docker-compose version 1.25.4, build unknown
docker-py version: 4.2.0
CPython version: 2.7.5
OpenSSL version: OpenSSL 1.0.2k-fips 26 Jan 2017
```

注意: 如果docker-compose版本过低, 则无法启动H1ve-CTFd

### 3、安装与启动H1ve

开源平台地址

```
https://github.com/D0g3-Lab/H1ve
```



看着挺精美的，那就开始安装吧！

## 1.搭建前准备

### (1) 安装git

一般情况下Centos上可能没安装 `git`，从github上克隆项目需要用到这个。(当然也可以用本地自己的电脑下载好，然后上传到服务器。这样的话，就不需要安装git)

安装git命令：

```
git clone http://github.com/D0g3-Lab/Hive.git
```

### (2) 配置docker源为国内源

这个主要是因为当拉取镜像时，由于远端仓库的服务器是在国外，国内用户拉取镜像的速度特别慢，或很可能都访问不了，产生报错。

下面有我发现的两个好用的国内docker源：

1) 阿里云镜像加速器

```
https://(yourid).mirror.aliyuncs.com
```

2) ustc(中国科学技术大学)

```
https://docker.mirrors.ustc.edu.cn
```

下面是以阿里云镜像加速器为例，配置docker国内源：

阿里云的docker镜像加速器需要注册账号，每个人都有自己唯一的地址。



```
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<-'EOF'
{
  "registry-mirrors": ["https://[yourid].mirror.aliyuncs.com"]
}
EOF
sudo systemctl daemon-reload
sudo systemctl restart docker
```

可以采用里边给定的配置docker镜像加速器的方法，也可以：

- (1) 使用vi编辑器写入/etc/docker/daemon.json

```
vi /etc/docker/daemon.json
```

添加下面内容：

```
{
  "registry-mirrors": ["https://(yourid).mirror.aliyuncs.com"]
}
```

保存退出

- (2) 重启docker

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

## 2.开始搭建H1ve

进到下载好的项目H1ve根目录，执行下面的命令：

```
docker-compose -f single.yml up
```

默认的端口是8000(可在Dockerfile文件中进行修改，尽量不要改)，访问一下服务器的IP地址，就会发现搭建成功了。

如果运行过程中出现 **响应码500错误**，重启环境即可(暂时没发现好的方法解决)

在项目根目录下，重启命令：

```
docker-compose -f single.yml down
docker-compose -f single.yml up
```

## 0x02 修改首页内容

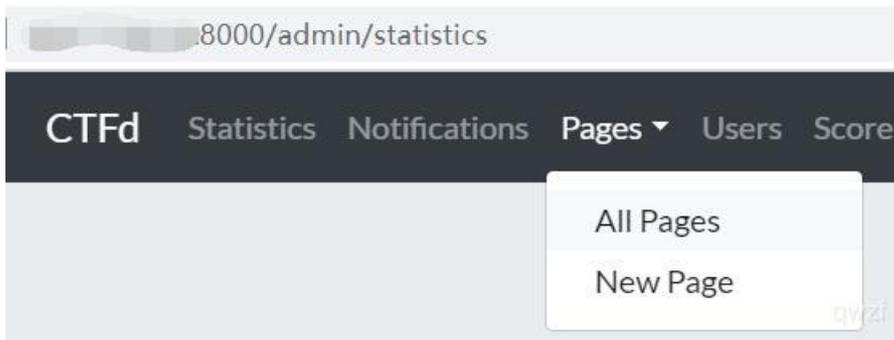
### 1.修改首页信息

即修改下面箭头所示信息：

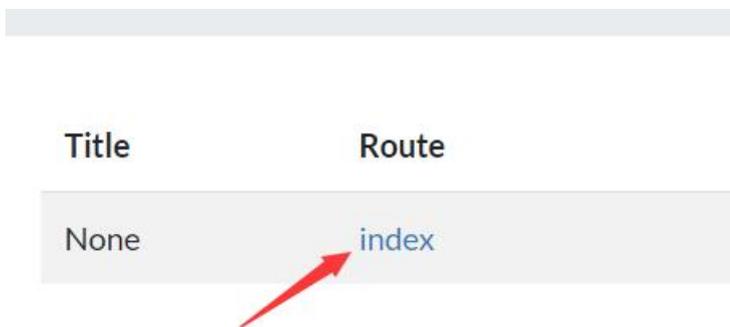




修改方法如下:

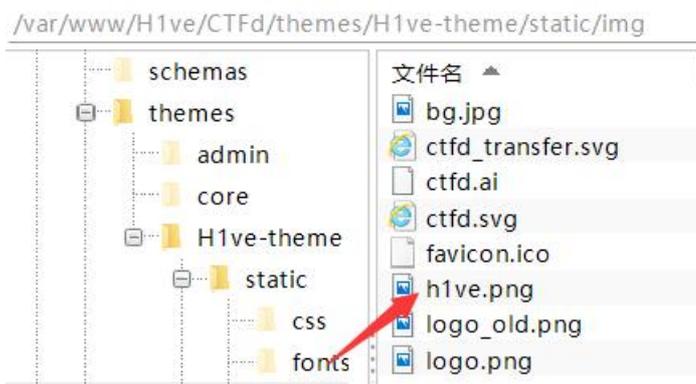


然后修改下面这个文件即可:



## 2.修改首页 logo

如果要换页面首页的logo的话, 在服务器里找到下面这个文件替换即可:



当然我只是测试了下，不知道替换后会不会侵权，能不替换还是不要替换了。也可以替换一些其他的地方信息，但不晓得会不会侵权。我只知道如果修改后用于商业用途是一定会侵权的，不用作商业用途应该不会侵权。

## 0x03 后记

搭建过程中，解决报错还是挺有趣的，不仅培养了自己的解决报错的能力，而且还可以让自己产生一些思考，真的挺不错的。继续努力吧，做一个奋进的精神小火。。。