






1. 目录遍历
2. PHPINFO
3. 备份文件下载
 - 3.1 网站源码
 - 3.2 bak文件
 - 3.3 vim缓存
 - 3.4 .DS_Store
4. Git泄露
 - 4.1 Log
 - 4.2 Stash
 - 4.3 Index
5. SVN泄露
6. HG泄露

1. 目录遍历



Index of /flag_in_here

Name	Last modified	Size	Description
 Parent Directory		-	
 1/	2020-12-20 10:07	-	
 2/	2020-12-20 10:07	-	
 3/	2020-12-20 10:07	-	
 4/	2020-12-20 10:07	-	

Apache/2.4.38 (Debian) Server at challenge-07e8eb2501d30c5e.sandbox.ctfhub.com Port 10080

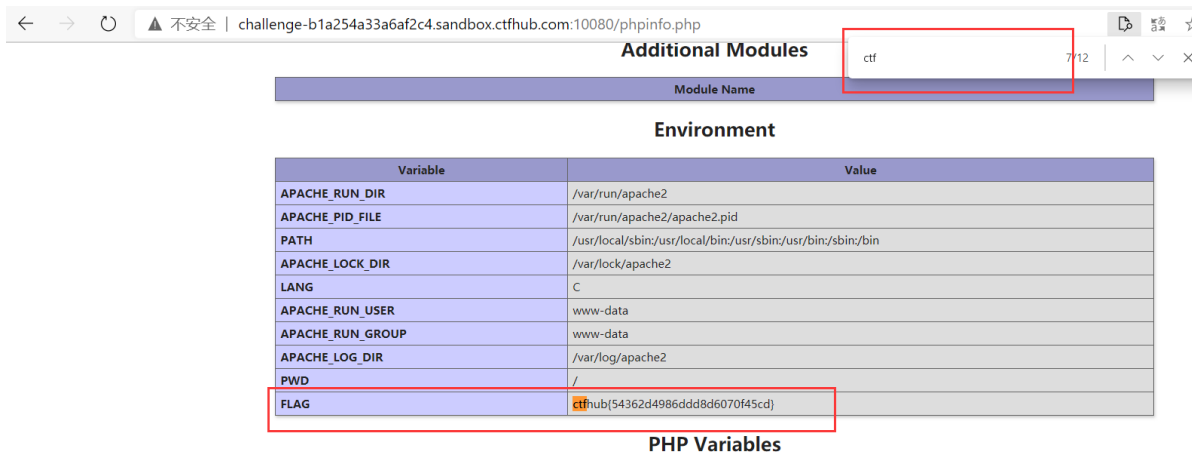
直接多找找就能找到了，我这次是在：

Index of /flag_in_here/3/1

Name	Last modified	Size	Description
 Parent Directory		-	
 flag.txt	2020-12-20 10:07	33	

Apache/2.4.38 (Debian) Server at challenge-07e8eb2501d30c5e.sandbox.ctfhub.com Port 10080

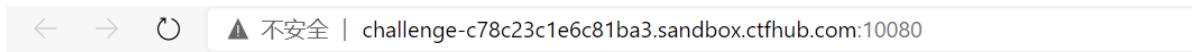
2. PHPINFO



在PHPINFO的页面里面，直接查找flag就可以了。

3. 备份文件下载

3.1 网站源码



备份文件下载 - 网站源码

可能有点用的提示

常见的网站源码备份文件后缀

- tar
- tar.gz
- zip
- rar

常见的网站源码备份文件名

- web
- website
- backup
- back
- www
- wwwroot
- temp

根据网站的提示，用文件名+备份后缀去访问。

懒得一个一个的去输入，直接用burp爆破。

Burp Suite Professional v2.0beta - Temporary Project - licensed to surferxyz

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

1 x 2 x ...

Target Positions Payloads Options

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Cluster bomb

```
GET /$$.$$ HTTP/1.1
Host: challenge-c78c23c1e6c81ba3.sandbox.ctfhub.com:10080
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.88 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
If-None-Match: W/"5e4a9605-260"
If-Modified-Since: Mon, 17 Feb 2020 13:32:53 GMT
Connection: close
```

0 matches Clear

2 payload positions Length: 578

Burp Suite Professional v2.0beta - Temporary Project - licensed to surferxyz

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

1 x 2 x ...

Target Positions Payloads Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 7
Payload type: Simple list Request count: 4

Payload Options [Simple list]

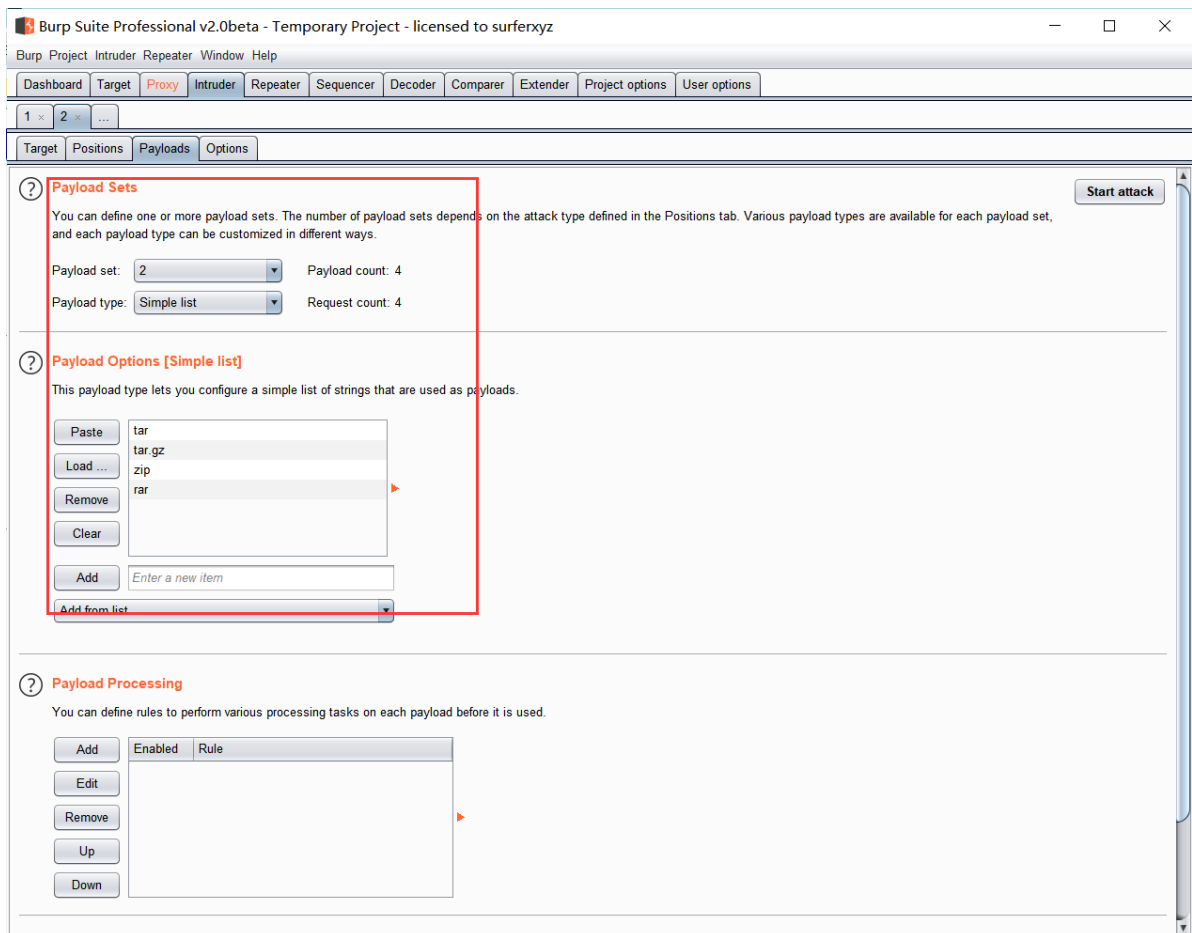
This payload type lets you configure a simple list of strings that are used as payloads.

Paste web
Load ... website
Remove backup
Clear back
www
temp
Add Enter a new item
Add from list ...

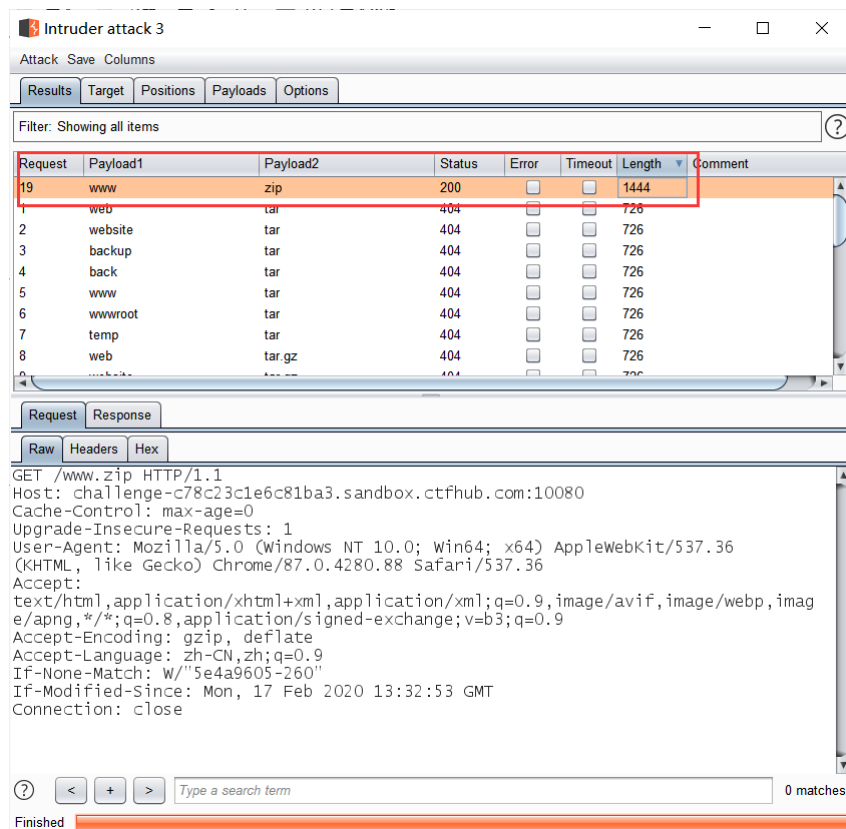
Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

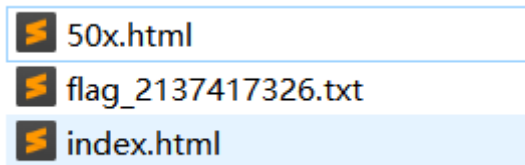
Add Enabled Rule
Edit
Remove
Up
Down



爆破结果：

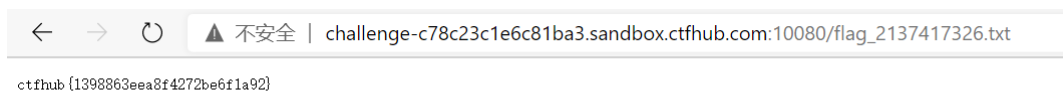


所以网站上应该存在有www.zip，去下载下来，发现三个文件



打开flag文件，发现里面什么都没有。

在网站上进入url/flag_2137417326.txt，发现了flag。



3.2 bak文件



Flag in index.php source code.

网站上说flag在index.php的源码里面，估计是在php里面被注释了。

说是存在bak备份，于是尝试 index.php.bak，果然存在，然后下载下来了备份文件。

打开便看到了flag



3.3 vim缓存

如果意外退出就会保留，文件名为 `.filename.swp`，
第一次产生的交换文件名 `.filename.txt.swp`
再次意外退出后，将会产生名为 `.filename.txt.swo` 的交换文件；
第三次产生的交换文件则为 `.filename.txt.swn`

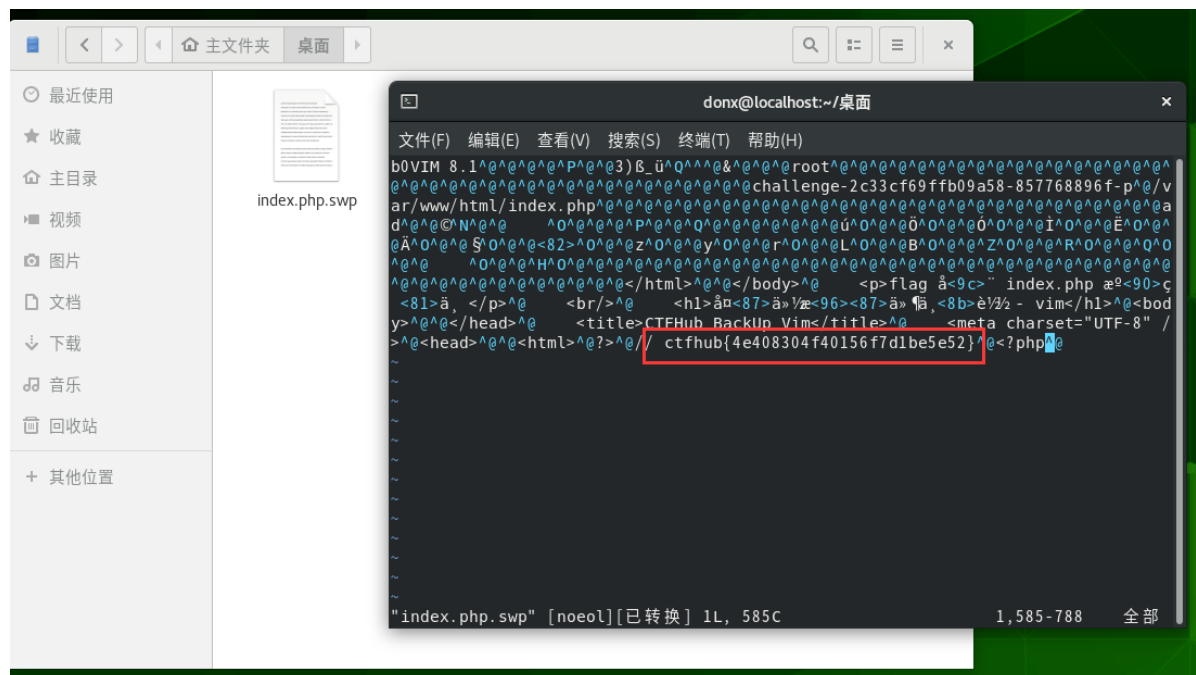


直接就下载下来了，但是用sublime打开发现是二进制文件，于是想到用vim来打开。



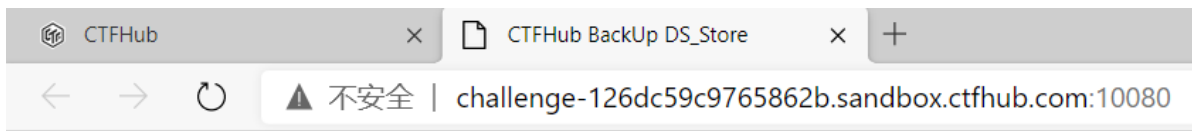
1	6230	5649	4d20	382e	3100	0000	0010	0000
2	3329	df5f	fc11	1e00	2600	0000	726f	6f74
3	0000	0000	0000	0000	0000	0000	0000	0000
4	0000	0000	0000	0000	0000	0000	0000	0000
5	0000	0000	6368	616c	6c65	6e67	652d	3263
6	3333	6366	3639	6666	6230	3961	3538	2d38
7	3537	3736	3838	3936	662d	7000	2f76	6172
8	2f77	7777	2f68	746d	6c2f	696e	6465	782e
9	7068	7000	0000	0000	0000	0000	0000	0000
10	0000	0000	0000	0000	0000	0000	0000	0000
11	6164	0000	a90e	0000	090f	0000	0010	0000
12	1100	0000	0000	0000	fa0f	0000	d60f	0000
13	d30f	0000	cc0f	0000	cb0f	0000	c40f	0000
14	a70f	0000	820f	0000	7a0f	0000	790f	0000
15	720f	0000	4c0f	0000	420f	0000	1a0f	0000
16	120f	0000	110f	0000	090f	0000	080f	0000
17	0000	0000	0000	0000	0000	0000	0000	0000
18	0000	0000	0000	0000	0000	0000	0000	0000
19	0000	0000	0000	0000	003c	2f68	746d	6c3e
20	0000	3c2f	626f	6479	3e00	2020	2020	3c70
21	3e66	6c61	6720	e59c	a820	696e	6465	782e
22	7068	7020	e6ba	90e7	a081	e4b8	ad3c	2f70
23	3e00	2020	2020	3c62	722f	3e00	2020	2020
24	3c68	313e	e5a4	87e4	bbbd	e696	87e4	bbb6
25	e4b8	8be8	bdbd	202d	2076	696d	3c2f	6831
26	3e00	3c62	6f64	793e	0000	3c2f	6865	6164
27	3e00	2020	2020	3c74	6974	6c65	3e43	5446

再把改了之后的文件在CentOS8里面用vim打开。



终于找到了flag。

3.4 .DS_Store

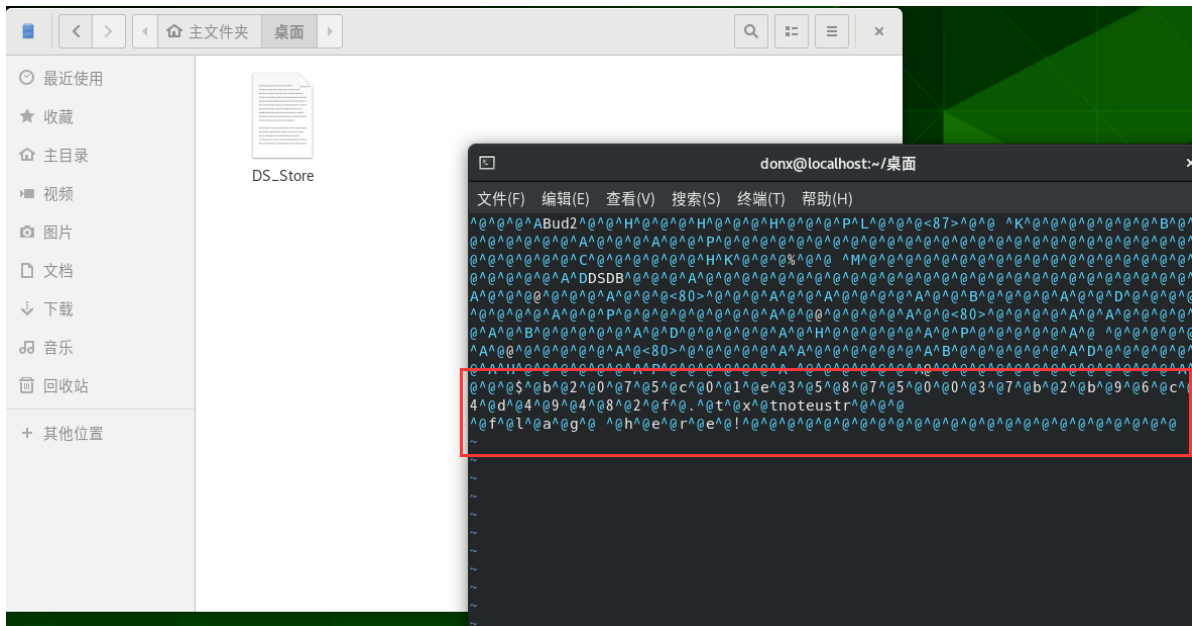


备份文件下载 - DS_Store

试着寻找 flag

直接下载DS_Store。

发现还是二进制文件，删掉里面大多数的0000之后，用vim打开。

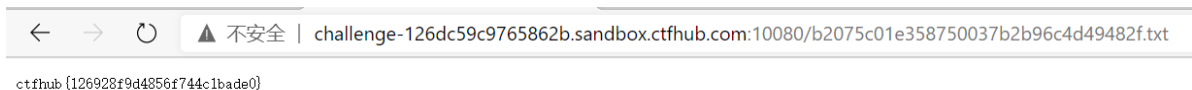


隐隐约约还是可以看见flag here，于是将中间所有的空字符删掉之后，大概可以看到一个文件名



于是去网站上试试。

直接就拿到了flag。



4. Git泄露

4.1 Log

说是将.git文件夹直接部署到了线上环境。直接访问.git，发现被拒绝访问，说明文件是确实存在的。

所以尝试用某种方法访问或者下载这个文件夹。这里尝试使用GitHack工具。（使用python2）

GitHack

.git 泄漏利用工具，可还原历史版本

依赖

不需要安装其它 Python 库，只需要有 git 命令

- git
 - ubuntu/debian: `$ apt-get install git`
 - redhat/centos: `$ yum install git`
 - windows [git-for-windows](#)下载

使用前需确保 git 在 环境变量中

使用方法

```
1 python GitHack.py http://www.example.com/.git/
```

还原后的文件在 `dist/` 目录下

工作流程

1. 尝试获取 `packs` 克隆
2. 尝试目录遍历克隆
3. 尝试从缓存文件(index)、commit记录中恢复

相关链接

- [BugScan](#)
- [GitHack - lijiejie](#)

使用之后。。。发现clone成功。

GitHack (0.0.5)

因为clone下来了，直接在clone下来的文件夹里查看log

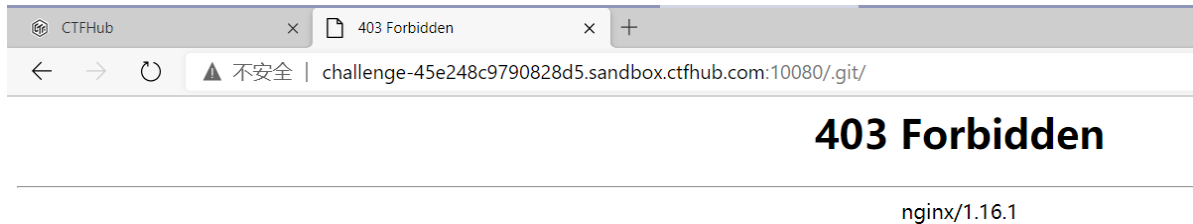
发现在最近的一次提交中移除了flag，直接比较两个版本就好。

也可以直接强行回到上一个版本: `git reset --hard 3884215`

```
1 ctfhub{a861b6f1266b5e0a45df6a33}
```

4.2 Stash

进去发现还是403:



还是用刚刚那个工具先clone下来再说。

```
D:\GitHack-master>python2 GitHack.py http://challenge-45e248c9790828d5.sandbox.ctfhub.com:10080/.git/
```

然后对着log比较了几次都没发现flag。

想了想题目的要求是stash，所以猜测可能是把flag存到了stash里面。

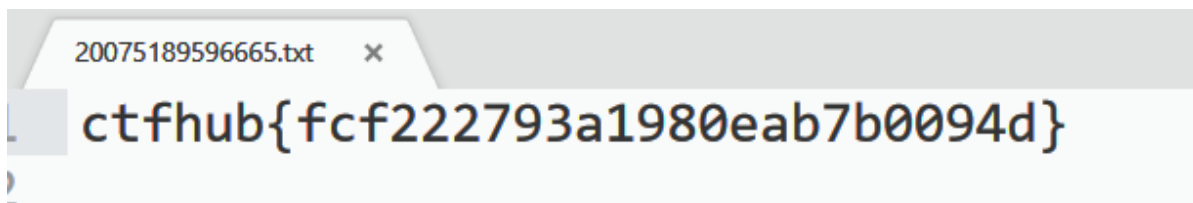
这里是[git stash的用法](#)。

使用 `git stash list` 命令之后发现有一个存档。

```
GitHack-master/dist/challenge-45e248c9790828d5.sandbox.ctfhub.com_10080 (m
aster)
$ git stash list
stash@{0}: WIP on master: f846aa6 add flag
```

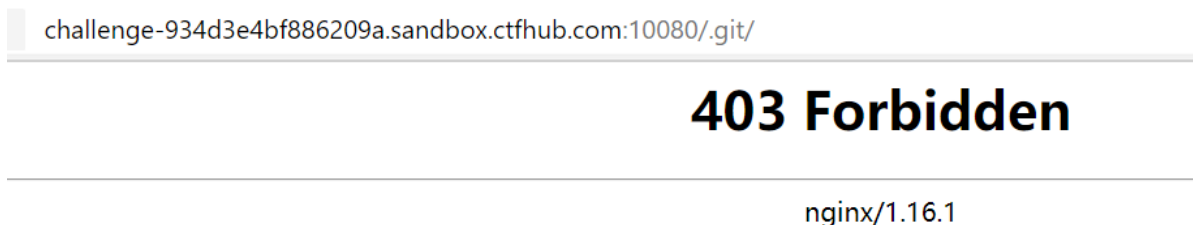
直接弹出存档。 `git stash pop`

发现这个flag被弹出来了。直接打开文件，里面就是flag。

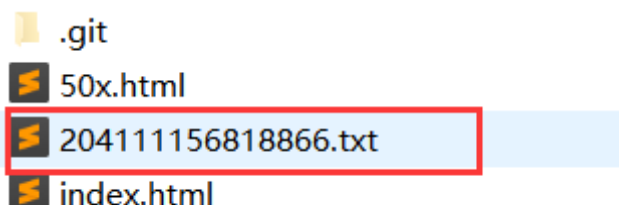


4.3 Index

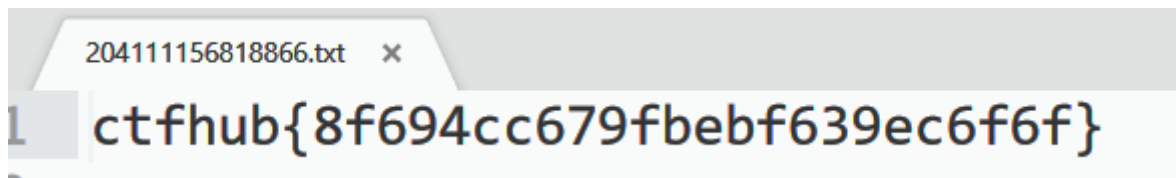
老规矩，先clone下来。



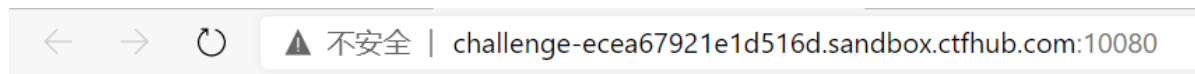
唔，怎么clone下来就有flag了...去查查这道题怎么回事。



查了查好像也差不多，那就直接交吧。



5. SVN泄露



信息泄露 - Subversion

Flag 在服务端旧版本的源代码中

题目说flag在旧版本的源码里面，题目说的是用SVN进行版本控制，去找找svn文件。

▲ 不安全 | challenge-ecea67921e1d516d.sandbox.ctfhub.com:10080/.svn/

403 Forbidden

nginx/1.16.1

状态码 403 说明文件是存在的但是禁止访问。去找找有什么方法拿到文件。

这里使用 `dvcs-ripper` 工具中的 `rip-svn.pl` 进行clone。

```
[root@localhost dvcs-ripper-master]# ./rip-svn.pl -v -u http://challenge-ecea67921e1d516d.sandbox.ctfhub.com:10080/.svn/
[i] Found new SVN client storage format!
REP INFO => 1:file:///opt/svn/ctfhub:e43e7ef8-82fb-4194-9673-81c29de69c33
[i] Trying to revert the tree, if you get error, upgrade your SVN client!
已恢复"index.html"
```

发现多了一个文件和一个文件夹。

221	12月	20	20:09	index.html
18027	8月	18	00:38	LICENSE
5597	8月	18	00:38	README.md
6401	8月	18	00:38	rip-bzr.pl
4717	8月	18	00:38	rip-cvs.pl
15114	8月	18	00:38	rip-git.pl
6102	8月	18	00:38	rip-hg.pl
6157	8月	18	00:38	rip-svn.pl
113	12月	20	20:09	.svn

直接一通查找：

```
[root@localhost .svn]# ls -la
总用量 128
drwxr-xr-x. 5 root root    113 12月 20 20:09 .
drwxrwxr-x. 4 root root    204 12月 20 20:09 ..
-rw-r--r--. 1 root root     3 12月 20 20:09 entries
-rw-r--r--. 1 root root     3 12月 20 20:09 format
drwxr-xr-x. 4 root root     26 12月 20 20:09 pristine
drwxr-xr-x. 2 root root     6 12月 20 20:09 text-base
drwxr-xr-x. 2 root root     6 12月 20 20:09 tmp
-rw-r--r--. 1 root root 122880 12月 20 20:09 wc.db
-rw-r--r--. 1 root root     0 12月 20 20:09 wc.db-journal
[root@localhost .svn]# cd pristine/
[root@localhost pristine]# ls -la
总用量 0
drwxr-xr-x. 4 root root   26 12月 20 20:09 .
drwxr-xr-x. 5 root root  113 12月 20 20:09 ..
drwxr-xr-x. 2 root root   63 12月 20 20:09 4d
drwxr-xr-x. 2 root root   63 12月 20 20:09 bf
[root@localhost pristine]# cd 4d
[root@localhost 4d]# ls -la
总用量 4
drwxr-xr-x. 2 root root   63 12月 20 20:09 .
drwxr-xr-x. 4 root root   26 12月 20 20:09 ..
-rw-r--r--. 1 root root   33 12月 20 20:09 4d697e4deb3cd472c8c80fd6984a07e9bfcab
[root@localhost 4d]# cat 4d697e4deb3cd472c8c80fd6984a07e9bfcab6f4.svn-base
ctfhub{28439e1c06efcbba169c6692}
[root@localhost 4d]#
```

最终找到了flag。

6. HG泄露

←
→
↺
⚠ 不安全 | challenge-abe92eb262179135.sandbox.ctfhub.com:10080

信息泄露 - Mercurial

Flag 在服务端旧版本的源代码中, 不太好使的情况下, 试着手工解决。

尝试找.hg文件夹:

challenge-abe92eb262179135.sandbox.ctfhub.com:10080/.hg/

403 Forbidden

nginx/1.16.1

依旧是403, 尝试用使用 `dvcs-ripper` 工具中的 `rip-hg.pl` 进行clone。

```
1 ./rip-hg.pl -v -u http://challenge-
  abe92eb262179135.sandbox.ctfhub.com:10080/.hg/
```

发现clone下来了一个.hg文件。里面有这些东西。

```
[root@localhost dvcs-ripper-master]# cd .hg
[root@localhost .hg]# ls -la
总用量 28
drwxr-xr-x. 3 root root 153 12月 20 20:26 .
drwxrwxr-x. 4 root root 185 12月 20 20:26 ..
-rw-r--r--. 1 root root  57 12月 20 20:26 00changelog.i
-rw-r--r--. 1 root root 128 12月 20 20:26 dirstate
-rw-r--r--. 1 root root   8 12月 20 20:26 last-message.txt
-rw-r--r--. 1 root root  59 12月 20 20:26 requires
drwxr-xr-x. 3 root root  86 12月 20 20:26 store
-rw-r--r--. 1 root root   7 12月 20 20:26 undo.branch
-rw-r--r--. 1 root root   9 12月 20 20:26 undo.desc
-rw-r--r--. 1 root root 128 12月 20 20:26 undo.dirstate
```

查看last-message.txt，说是加了一个flag，这就去找找在哪。

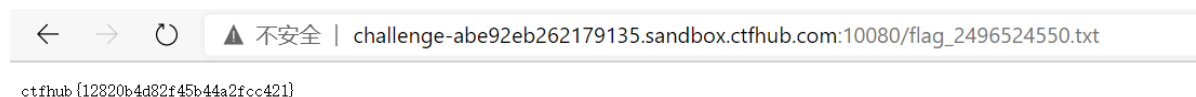
```
[root@localhost .hg]# cat last-message.txt
add flag [root@localhost .hg]#
```

访问store文件夹里的fncache文件，发现了flag的位置。但是进入data之后却发现没有这个文件，估计是被删了。

```
[root@localhost store]# cat fncache
data/index.html.i
data/50x.html.i
data/flag_2496524550.txt.i
```

没办法，去看看网页上这个文件还有没有。

还好，网页上这个文件存在。



直接拿到flag。